ISSN: 2319-7897
Volume 13 (2024-25)

Satyam

MSIT Journal of Research

MSIT Journal of Research –SATYAM

Volume 13 (2024-25)

MSIT Journal of Research ─ SATYAM
An annual publication of
MAHARAJA SURAJMAL INSTITUTE OF TECHNOLGY
C-4, Janakpuri, New Delhi-110058 (India)
Phone : 011-65215941
E-mail: satyamjournal@msit.in, Visit us at www.msit.in

Volume 13 (2024-25)

MSIT Journal of Research – SATYAM (MSITJR) is an annual publication Maharaja Surajmal Institute of Technology,C-4 Janakpuri, New Delhi – 110058

Editor-in-chief: **Dr. Sapna Malik**

ISSN 2319-7897

Disclaimer

The opinions expressed and figures provided in this journal are sole responsibility of the authors. The publisher and the editor bear no responsibility in this regard. Any or all such liabilities are disclaimed.

All disputes are subject to Delhi jurisdiction only

Address for

Correspondence

Dr. Sapna Malik
Editor-in-chief, MSIT Journal of Research, Associate Professor, Department of CSE Maharaja Surajmal Institute of Technology (MSIT) C-4, Janakpuri, New Delhi-110058
Tel: 011-65215941
E-mail: satyamjournal@msit.in

Published and printed by Prof. Avanish Kumar Srivastava, Director, Maharaja Surajmal Institute of Technology (MSIT), C-4, Janakpuri, New Delhi-110058 (INDIA), Tel: 011-65215941
Email: director@msit.in, Visit us at www.msit.in

CONTENT

Elevating Ecommerce: Layer 2 Scalability for Blockchain Driven Transactions

Priyanka Kalkandha
Maharaja Surajmal Institute of Technology, New Delhi, India
priyanka.kalkandha@msit

***Abstract*— This paper explores the integration of blockchain technology in an online marketplace to enhance e-commerce. Utilizing Polygon zkEVM's layer 2 scaling solution, the platform achieves scalability while maintaining security and decentralization. Blockchain integration ensures transparent recording of product information and transactions. Interoperability between the Ethereum mainnet and layer 2 infrastructure is optimized, reducing gas costs. This study highlights advancements in e-commerce and the potential for secure, transparent, and scalable marketplaces.**

***Keywords*- Blockchain, Polygon, Layer-2, Marketplace, Ethereum, Proof of Work, Authenticity, Decentralized, Shopping, Transactions.**

I. INTRODUCTION

In today's dynamic digital landscape, online shopping has become an essential aspect of our daily lives, offering unparalleled convenience and product variety. However, traditional e-commerce platforms face challenges such as data security issues and complex transaction processes.

To overcome these challenges, layer 2 scalability solution like Polygon zkEVM offer a promising solution. By leveraging technologies that enhance transaction efficiency and reduce costs, layer 2 solution aim to revolutionize the e-commerce landscape. This paper explores the transformative potential of layer 2 scalability solution in e-commerce. Through an examination of technologies like Polygon zkEVM, we aim to uncover how these solution can address scalability challenges, streamline transactions, and ensure the security of online commerce.

In the dynamic landscape of modern commerce, a pioneering approach emerges, strategically integrating advanced technologies to address the evolving needs of online consumers. Central to this mission is the utilization of blockchain technology and Smart Contracts, aimed at redefining e-commerce by establishing a secure, transparent, and scalable marketplace.

Traditional e-commerce platforms face challenges stemming from their reliance on centralized systems, leading to vulnerabilities like data breaches and privacy concerns. Moreover, the scalability dilemma inherent in blockchain technology poses a significant obstacle to their adaptability and growth. This dilemma, often referred to as the blockchain trilemma, necessitates a delicate balance between decentralization, security, and scalability, where achieving all three simultaneously proves elusive.

To confront these challenges, innovative solutions are employed. Notably, the integration of layer 2 scaling technology, specifically leveraging Polygon zkEVM, allows for overcoming the scalability bottleneck while preserving the inherent security and decentralization of the underlying blockchain network. By harnessing the capabilities of layer 2 scaling solutions like Polygon zkEVM, substantial enhancements in transaction throughput are achieved, effectively addressing the scalability aspect of the blockchain trilemma without compromising on security or decentralization.

Blockchain Features Transforming E-commerce:

A. Enhanced Security : Blockchain uses advanced math to keep transactions super safe, preventing unauthorized access and data breaches for secure online transactions.
B. Decentralization: Blockchain removes middlemen, creating a decentralized system. This simplifies transactions, avoids single points of failure, and boosts reliability in e-commerce.
C. Transparent Transaction History : Blockchain makes all transactions transparent and unchangeable. This guarantees the complete and verifiable history of a product, giving buyers confidence in its authenticity.
D. Smart Contracts for Automation : Smart contracts automate e-commerce tasks like order fulfillment, payment, and dispute resolution. They cut manual work, speeding up transactions.
E. Immutable Product Information : Blockchain keeps product info and transactions unchangeable, stopping tampering. This builds trust between buyers and sellers in e-commerce.
F. Reduced Transaction Costs: Blockchain cuts out middlemen, saving money for buyers and sellers, making online transactions more affordable.

II. SOFTWARE DEPENDENCIES

The development of the Blockchain Marketplace Application required the utilization of a comprehensive suite of software tools. These tools played a crucial role in facilitating various aspects of the development process:

2.1 Ethereum

Usage: Foundational blockchain platform.
Description: Decentralized open-source public permissioned blockchain, pioneering the concept of smart contracts. Operates on its native cryptocurrency, Ether (ETH).

2.2 Ethereum Virtual Machine

Usage: Smart contract execution environment.

Description: Runtime environment for smart contracts on an Ethereum node. Transforms smart contracts written in Solidity into machine-readable bytecode.

2.3 Solidity

Usage: Smart contract development language.
Description: Dedicated language for smart contract development on the Ethereum blockchain. Similar to JavaScript, with elements from C#, ensuring secure and efficient contract creation.

2.4 Truffle

Usage: Development and testing framework.
Description: Framework for Ethereum, streamlining smart contract management, migrations, and testing. Enhances the development process and ensures code integrity.

2.5 Hardhat

Usage: Local blockchain development tool.
Description: Hardhat, a CLI tool, replicates Ethereum's main network features for testing and development purposes, providing a sandbox environment for efficient blockchain development.

2.6 web3.js

Usage: Blockchain interaction library.
Description: JavaScript library exclusively designed for the Ethereum blockchain. Facilitates the creation of a web UI for our blockchain information system, enabling users to interact visually with the application backend.

2.7 Metamask

Usage: Browser extension for blockchain interaction.
Description: Browser extension enabling users to interact with the Ethereum blockchain directly from web browsers. Provides a secure and user-friendly interface for managing blockchain transactions.

III. LITERATURE REVIEW

Layer-2 solution in blockchain address scalability issues by operating above the main blockchain and handling transactions off-chain. This reduces the load on the main chain, resulting in faster, cheaper, and more efficient transactions. Layer-2 protocols ensure security and decentralization by selectively broadcasting transactions to the main blockchain, offering a promising solution to enhance blockchain scalability without sacrificing its core principles.

The evolving landscape of electronic commerce (e-commerce) encounters challenges, including security vulnerabilities and high transaction fees. Blockchain technology emerges as a transformative force, offering a decentralized and secure alternative. This review explores blockchain's applications in e-commerce, focusing on its potential to revolutionize practices and addressing challenges.

The literature highlights blockchain's role in securing online payments, leveraging cryptocurrencies to combat fraud and enhance transaction security. The decentralized and tamper-proof nature of blockchain fortifies defenses against hacking, promising reduced transaction fees by eliminating intermediaries, aligning with cost-efficiency goals.

Supply chain management undergoes a paradigm shift with blockchain, introducing transparency and tamper-proof records. However, scalability and interoperability challenges require nuanced solution for seamless integration. Intellectual property protection benefits from blockchain's verifiable record, establishing trust across sectors beyond e-commerce.

Blockchain's impact extends to customer identity management, mitigating risks of data breaches and identity theft. A decentralized platform enhances consumer trust, contributing to a secure and privacy-centric online shopping environment. Dispute resolution benefits from blockchain's transparent and immutable records, introducing efficiency and fairness.

While optimistic, the literature emphasizes challenges such as scalability, interoperability, and trust establishment in blockchain systems. Addressing these challenges is crucial for unlocking blockchain's full potential in reshaping the e-commerce landscape. The synthesis of findings reveals blockchain's transformative applications, from secure payments to supply chain management, intellectual property protection, customer identity, and dispute resolution, ushering in a new era of security and transparency in e-commerce.

IV. METHODOLOGY

Our methodology integrates layer 2 scalability solution, particularly Polygon zkEVM, to fortify transparency, immutability, and trust in e-commerce transactions. By leveraging the capabilities of Polygon zkEVM, our system ensures enhanced scalability while maintaining the integrity of transactions.

The inherent features of layer 2 scalability solution, such as Polygon zkEVM, contribute to a transparent and virtually immutable system. Transactions processed on the layer 2 network benefit from the timestamping feature, ensuring transparency and traceability. Any attempts at manipulation, such as altering product information or fraudulent transactions, are promptly identified as other connected nodes synchronize, bolstering the integrity of accepted transactions.

Within our system, Polygon zkEVM guarantees the security and immutability of transactions. When a product is added or a transaction occurs, it is recorded on the layer 2 blockchain, ensuring its immutable status. This aligns with the principles of transparency and trust, crucial in enhancing the reliability of e-commerce interactions.

To ensure fairness and reliability in product information and transactions, the decentralized structure of layer 2 scalability solution, such as Polygon zkEVM, plays a pivotal role. The consensus mechanisms inherent in Polygon zkEVM secure and validate product details, reducing the reliance on a central authority. This decentralized approach promotes trust in the accuracy and authenticity of product-related information.

Our methodology places significant emphasis on ensuring the authenticity of product reviews. Leveraging layer 2 scalability solution like Polygon zkEVM, each user review is securely recorded on the layer 2 blockchain. This not only prevents tampering with reviews but also ensures that feedback remains authentic. By eliminating the possibility of fraudulent or manipulated reviews, we enhance the trustworthiness of the entire review system.

Acknowledging the potential challenges in a large-scale system with numerous products and transactions, our methodology employs a distributed chain approach facilitated by layer 2 scalability solution. Chains are distributed over different levels to reduce synchronization latencies, ensuring efficiency, and scalability. This strategic distribution caters to scenarios where transactions occur simultaneously across various product categories, ensuring seamless operation and scalability in our e-commerce ecosystem.

5.1 The Architecture

Polygon zkEVM, a layer 2 solution for Ethereum, enhances scalability while preserving security. Key components include the Consensus Contract (PolygonZkEVM.sol), implementing the Proof of Efficiency (PoE) consensus mechanism, and zkNode software, facilitating transaction validation and aggregation. zkProver generates Zero-Knowledge Proofs (ZKPs) for transaction validation, ensuring privacy. The zkEVM Bridge enables seamless fund transfer between Layer 1 and Layer 2 accounts, enhancing interoperability and user experience.

In essence, Polygon zkEVM integrates sophisticated components to address Ethereum's scalability challenges while maintaining security and decentralization. It offers a scalable alternative through the Consensus Contract, zkNode, zkProver, and zkEVM Bridge, ensuring efficient operation within the Ethereum ecosystem.

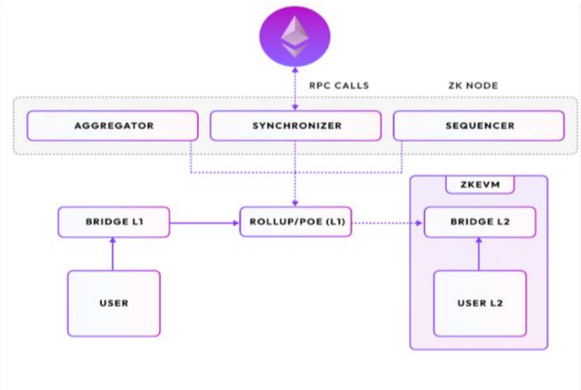


Fig 1: zkEVM Architecture

5.2 Algorithms Used

In the development journey of our decentralized marketplace, several fundamental algorithmic concepts and technologies have played pivotal roles in shaping various modules. These algorithms ensure the reliability and security of the system, addressing key challenges encountered in the online marketplace landscape.

5.2.1 Byzantine Fault Tolerance

To fortify the resilience of our blockchain marketplace, we implemented Byzantine Fault Tolerance. This ensures that the system remains steadfast even in the face of node failures or disruptions in information flow. Byzantine Fault Tolerance enhances the robustness of our marketplace, contributing to a reliable and secure environment for e-commerce transactions.

5.2.2 Proof of Stake Consensus Algorithm

Our decentralized marketplace employs the Proof of Stake (PoS) consensus mechanism, providing a sustainable alternative to Proof of Work. PoS enhances scalability and environmental sustainability by allowing participants to validate transactions and create new blocks through cryptocurrency staking. This approach aligns with our commitment to energy efficiency in blockchain operations, ensuring a responsible and eco-friendly solution for decentralized transactions in the marketplace.

5.2.3 Smart Contracts

Smart contracts play a pivotal role in our blockchain-based marketplace, ensuring transparency and automation in transactions. These self-executing contracts enforce predefined rules and conditions, providing a secure and efficient means for users to engage in e-commerce activities. The automated execution of transactions enhances the reliability and transparency of our decentralized system, offering users a seamless and secure experience.

5.2.4 Merkle Tree

A Merkle Tree is a hash-based tree structure, emerging as a vital component in processing large datasets for data security and verification. In the context of our decentralized marketplace, Merkle Trees contribute to the integrity of information. Every leaf node holds the cryptographic hash of a data block, while non-leaf nodes contain the cryptographic hash of their child nodes. This structure enhances data security and verification processes within the Ethereum Blockchain framework.

5.2.5 Reviews and Ratings Authenticity

Within our blockchain marketplace, a decentralized mechanism ensures the authenticity of product reviews and ratings. Leveraging blockchain technology, reviews are securely recorded and verified, preventing manipulation or fraudulent practices. This innovative approach fosters trust among users, ensuring that product reviews and ratings are genuine, reliable, and reflective of the community's experiences. Authenticity in reviews enhances user confidence and contributes to a trustworthy e-commerce environment.

V. EXPERIMENTAL ANALYSIS

The blockchain-powered marketplace proposed in this paper provides an enhanced solution to traditional e-commerce challenges. A set of experimental analyses will compare Layer 2 vs Layer 1 blockchain and the blockchain marketplace against conventional systems, highlighting its superior features and performance as depicted in Table 1.

TABLE 1. LAYER 1 VS LAYER 2 BLOCKCHAIN

S. No.	**arameter of Analysis**	**Layer-1 Blockchain**	**Layer-2 Blockchain**
1.	Gas Fee (avg)	0.0076 ETH	0.0000000143 ETH Around 531468 times lower.
2.	Scalability	Not much scalable, only upto 20 transactions per second	Highly scalable, upto 5000 transactions per second

A concise comparison between blockchain and traditional marketplace is discussed in Table 2 covering various parameters including confidentiality, Integrity, Accessibility, etc.

TABLE 2. EXPERIMENTAL ANALYSIS OF COMPARISON BETWEEN BLOCKCHAIN AND TRADITIONAL MARKETPLACE

S.No	**arameter of Analysis**	**Blockchain Marketplace**	**Traditional Marketplace**
1.	Confidentiality	Employs public key encryption, ensuring confidential user data is accessible only to authorized participants.	Relies on centralized databases, making sensitive user data susceptible to breaches.
2.	Integrity	Ensures data integrity through tamper-proof smart contracts and decentralized consensus mechanisms.	Vulnerable to data manipulation and unauthorized access.
3.	Accessibility	Decentralized access allows users greater control over their data and transactions.	Access control determined by central authorities.
4.	Non-repudiation	Provides indubitable validity of transactions through decentralized consensus mechanisms.	Limited capability to prove the origin or authorship of transactions.
5.	Accountability	Offers complete transparency through a decentralized ledger, enhancing trust in transactions.	Lacks transparent accountability due to centralized control.
6.	Transaction Speed	Streamlines peer-to-peer transactions, potentially reducing transaction times.	Relies on intermediaries, leading to slower transaction processing times.
7.	Cost Efficiency	Reduces costs by eliminating intermediaries and automating processes through smart contracts.	Involves intermediary fees and overhead costs.
8.	Trust in Reviews	Ensures trust in reviews through tamper-proof and authentic reviews recorded on the blockchain.	Reviews susceptible to manipulation and fraudulent practices.
9.	Scalability	Scales efficiently through decentralized architecture, potentially accommodating a larger user base.	May face scalability challenges due to centralized infrastructure.

VI. CONCLUSIONS

This project explored the transformative potential of blockchain technology in e-commerce. Our initial research highlighted the inherent challenges faced by traditional platforms and how blockchain's decentralized nature could foster trust and security through tamper-proof records. However, the limitations of standard blockchain implementations, particularly scalability and transaction costs, presented obstacles for high-volume e-commerce applications. To address these limitations, we successfully implemented Polygon zkEVM, a Layer 2 scaling solution, into our e-commerce project.

By leveraging zkEVM's innovative architecture, our project achieved significant improvements in scalability and transaction processing speed. This translates to a more user-friendly experience for both buyers and sellers, as transactions are confirmed faster and gas fees are significantly reduced compared to traditional blockchain networks. This scalability advantage proves crucial for e-commerce platforms handling high volumes of transactions, ensuring smooth operation and preventing bottlenecks.

The successful integration of zkEVM in this project underscores its potential to redefine the future of e-commerce. By unlocking the power of blockchain technology while addressing scalability concerns, zkEVM paves the way for a more secure, transparent, and efficient e-commerce ecosystem. Continued research and development in Layer 2 solutions like zkEVM are essential to fully unlock their potential and revolutionize the way we conduct online transactions.

This project serves as a stepping stone towards a future where blockchain and Layer 2 scaling solutions redefine the e-commerce landscape, creating a more secure, efficient, and user-friendly environment for all participants.

REFERENCES

[1] Gangwal, Ankit, Haripriya Ravali Gangavalli, and Apoorva Thirupathi. "A survey of Layer-two blockchain protocols." *Journal of Network and Computer Applications* 209 (2023): 103539.

[2] Sguanci, Cosimo, Roberto Spatafora, and Andrea Mario Vergani. "Layer 2 blockchain scaling: A survey." *arXiv preprint arXiv:2107.10881* (2021)

[3] Pandey, Anova Ajay, et al. "Maintaining scalability in blockchain." *International Conference on Intelligent Systems Design and Applications*. Cham: Springer International Publishing, 2021.

[4] Cong, Lin William, et al. "Scaling smart contracts via layer-2 technologies: Some empirical evidence." *Management Science* 69.12 (2023): 7306-7316.

[5] Dahal, Suresh Budha. "Enhancing E-commerce Security: The Effectiveness of Blockchain Technology in Protecting Against Fraudulent Transactions." *International Journal of Information and Cybersecurity* 7.1 (2023): 1-12.

[6] Taherdoost, Hamed, and Mitra Madanchian. "Blockchain-based e-commerce: A review on applications and challenges." *Electronics* 12.8 (2023): 1889.

[7] Haque, AKM Zahidul, et al. "A scenario of adopting blockchain technology in supply chain: A study of E-commerce in Bangladesh." *Supply Chain Insider| ISSN: 2617-7420 (Print), 2617-7420 (Online)* 8.1 (2022).

[8] Bulsara, Hemantkumar P., and Pratiksinh S. Vaghela. "Blockchain technology for e-commerce industry." *International Journal of Advanced Science and Technology* 29.5 (2020): 3793-3798.

[9] Taherdoost, Hamed. "Smart contracts in blockchain technology: A critical review." *Information* 14.2 (2023): 117.

[10] Treiblmaier, Horst, and Christian Sillaber. "The impact of blockchain on e-commerce: a framework for salient research topics." *Electronic Commerce Research and Applications* 48 (2021): 101054.

[11] Taherdoost, Hamed, and Mitra Madanchian. "Blockchain-based e-commerce: A review on applications and challenges." *Electronics* 12.8 (2023): 1889.

[12] Treiblmaier, Horst, and Christian Sillaber. "The impact of blockchain on e-commerce: a framework for salient research topics." Electronic Commerce Research and Applications 48 (2021): 101054.

[13] Dutta, Pankaj, et al. "Blockchain technology in supply chain operations: Applications, challenges and research opportunities." *Transportation research part e: Logistics and transportation review* 142 (2020): 102067.

[14] Kim, Shee-Ihn, and Seung-Hee Kim. "E-commerce payment model using blockchain." *Journal of Ambient Intelligence and Humanized Computing* 13.3 (2022): 1673-1685.

[15] Liu, Zhiyong, and Zipei Li. "A blockchain-based framework of cross-border e-commerce supply chain." *International journal of information management* 52 (2020): 102059.

[16] Targett, David. "B2B or not B2B? Scenarios for the future of e-commerce." *European Business Journal* 13.1 (2001): 3.

[17] Treiblmaier, Horst, and Christian Sillaber. "The impact of blockchain on e-commerce: a framework for salient research topics." *Electronic Commerce Research and Applications* 48 (2021): 101054.

[18] Xiao, Yuanyuan, et al. "A novel decentralized e-commerce transaction system based on blockchain." *Applied Sciences* 12.12 (2022): 5770.

Efficient Client-Side Sign Language Recognition and Translation

Akshay Singh #, 1, Jyoti *, 2, Surender Singh#, 3

#Department of Information Technology, Maharaja Surajmal Institute of Technology, Janakpuri, New Delhi

*Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, Janakpuri, New Delhi

[1]akshaysingh@msit.in
[2]jyoti@msit.in
[3]surenderbhanwala@msit.in

***Abstract*— This paper presents a robust, real-time sign language translator engineered as a web-based application utilizing modern frontend frameworks. The primary objective of this paper is to bridge the communication gap between the deaf and hearing communities through accessible technology. A standard webcam feed was utilized by the system to continuously detect and interpret hand gestures associated with sign language, which were subsequently converted into textual output instantaneously. By integrating a MediaPipe-based hand detector, 21 distinct 3D hand landmarks were extracted from each video frame with high fidelity. For the interpretation of dynamic signs, characterized by complex temporal sequences of gestures, sequences of landmark vectors are processed by an ONNX-format neural network model executed via the ONNX Runtime Web. Conversely, static gestures are identified by a specialized recognizer that compares normalized landmark patterns with a predefined set of prototypes using a root-mean-square-distance threshold. The frontend interface, constructed with TypeScript and styled using the utility-first Tailwind CSS framework, functions as a responsive Single-Page Application (SPA). Initial testing indicated that high translation accuracy was achieved in real-time environments with minimal latency. The primary contribution of this work is the realization of an end-to-end, browser-based translator that integrates on-device Machine Learning (MediaPipe + ONNX Runtime Web) with an intuitive user interface, thereby demonstrating the viability of privacy-preserving, client-side sign language interpretation.**

***Keywords*— Sign Language Translation; Real-Time Gesture Recognition; MediaPipe Hands; ONNX Runtime Web; Hand Landmarks; Machine Learning.**

I. INTRODUCTION

Sign language is an essential mode of communication for millions of individuals who are deaf or hard of hearing worldwide. Despite its widespread use, a significant communication barrier persists, as the majority of non-signers lack the proficiency to accurately interpret these gestures [1,12]. Consequently, social inclusion and information accessibility for the deaf community are often impeded by these limitations of sign language [11]. To effectively address this gap, the development of automated translation systems capable of converting hand gestures into spoken and written languages is imperative [2]. Historically, Sign Language Recognition (SLR) methodologies have relied heavily on specialized, intrusive hardware, such as sensory data gloves or depth-sensing cameras (e.g., Microsoft Kinect). Although effective, these solutions are often cost-prohibitive and lack portability [3]. However, recent advancements in computer vision and deep learning have made vision-based SLR increasingly feasible using standard RGB cameras. Notably, the MediaPipe Hands framework, developed by Google, has emerged as a transformative tool, offering high-fidelity, real-time detection of 21 hand landmarks from standard camera input. This capability facilitates the robust tracking of resource-constrained devices, including mobile phones and web browsers [4].

Significant progress has been made in neural network architectures, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTMs), in achieving high accuracy for both static poses and dynamic sign sequences [5, 13]. In this paper, a fully browser-based real-time sign language translator is proposed. Video data were captured directly from the user's webcam, processed via MediaPipe-based hand landmark detection, and classified using sophisticated on-device machine learning models. Static gestures are recognized by matching the current hand pose to known vector patterns, whereas dynamic gestures are handled by an ONNX format sequence model executed via ONNX Runtime Web. A critical distinction of this system is that all inference processes are performed on the client side (utilizing WebGPU or WebAssembly acceleration). By performing computations locally, user data privacy is inherently preserved because video feeds are never transmitted to external servers, and offline functionality is enabled.

II. LITERATURE SURVEY

Over the past two decades, substantial research has been devoted to Sign Language Recognition. Initial methodologies were primarily hardware-focused, requiring the use of wearables equipped with flex sensors and accelerometers to capture finger positions. Despite their precision, these methods are impractical for everyday use because of the discomfort and setup time involved [6]. Contemporary systems have transitioned to computer vision techniques.

Typically, vision-based SLR involves a sequence of skin color segmentation, followed by feature extraction and classification. For example, Wadhwan and Kumar [9] employed image processing techniques in conjunction with Artificial Neural Networks (ANNs) to recognize Indian Sign Language. With the rise of deep learning, rich feature representations have been directly learned from raw hand images using Deep CNNs.

To capture the temporal dynamics inherent in gesture sequences, where meaning is derived from movement over time rather than a single frame, RNNs, particularly LSTM variants, are frequently utilized [7]. Recent studies employing these deep learning approaches have demonstrated their high accuracy. For instance, [2] integrated the MediaPipe framework with a hybrid CNN-LSTM model to recognize both static and dynamic signs in real time, reporting accuracy rates of 94–98% for static gestures. This aligns with the general observation that dynamic sign sequences pose greater challenges due to temporal variation and speed differences among signers.

However, a common limitation in much of the existing literature is the dependence on server-side Python environments for inference. The approach proposed diverges by focusing on a fully in-browser implementation [8]. With the advent of web-based Machine Learning frameworks (TensorFlow.js, ONNX Runtime) and efficient hand tracking (MediaPipe), the development of sign language translators that operate entirely on the client side has become feasible. This architectural shift eliminates network latency and server costs while significantly enhancing user privacy [10].

III. Research Methodology

The architecture of the proposed system is designed for modularity and performance. It comprises five distinct computational stages: Camera Input Acquisition, Hand Landmark Detection, Feature Pre-processing and Normalization, Gesture Classification and Output Rendering. Fig.1 shows the use case diagram of the camara based gesture recognition system. The activity workflow of the proposed model is shown in Fig. 2.

A. Camera Input Acquisition:

A live video feed was captured from the user's webcam directly within the browser context using the HTML5 getUserMedia API. To ensure compatibility across devices, a resolution of 640 x 480 pixels is typically requested. Each frame was subsequently extracted and forwarded to the detection pipeline.

B. Hand Landmark Detection:

TensorFlow.js was utilized in conjunction with Google's MediaPipe Hands solution to perform topology estimation. A robust fallback mechanism was implemented within the HandDetector module: the system primarily attempted to initialize the TFJS (WebGL) backend for GPU acceleration.

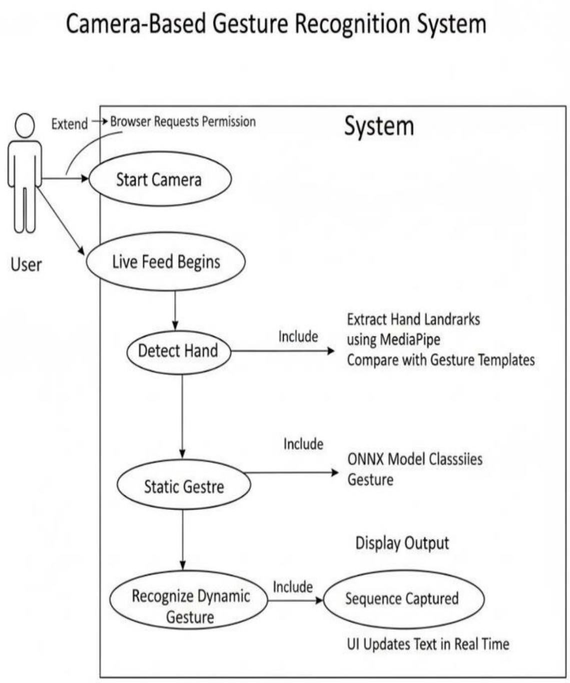


Fig. 1: Camera Based Gesture Recognition System

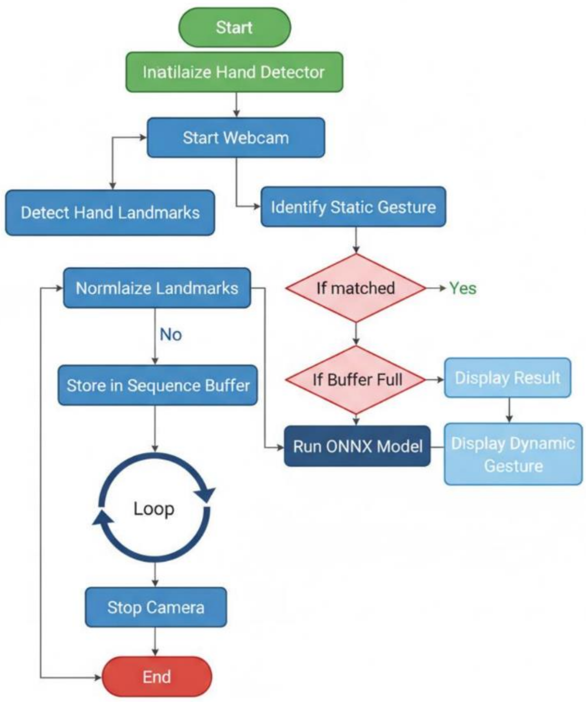


Fig. 2: Activity Workflow of Proposed Model

If this fails because of hardware incompatibility, the system automatically reverts to the MediaPipe WebAssembly (WASM) solution. For every processed frame, up to two sets of 21 three-dimensional landmarks (x, y, z) were returned, alongside a classification of handedness (left versus right hand).

C. Feature Preprocessing and Normalization:

Raw landmark coordinates were subjected to rigorous preprocessing to ensure model generalization.

Normalization: To mitigate the variance caused by the user's distance from the camera, the landmark coordinates of each hand were normalized per axis (min-max scaling to the range [0,1]). This ensured that the model analyzed the shape of the hand rather than its absolute position in the frame.

Sequence Buffering: For dynamic gestures, a First-In-First-Out (FIFO) sliding window buffer containing the most recent T frames (default T = 32) of landmark vectors is maintained. The oldest data are discarded as new frames are processed. The flattened landmarks from each frame were concatenated to form a tensor sequence with the shape [1, T, 63].

D. Gesture Classification:

Two parallel classification engines are employed to handle the diversity of sign language:

Static Gesture Recognizer: This lightweight module is used for fixed pose recognition. Known gesture prototypes are preloaded as normalized vector patterns. The Root-Mean-Square Error (RMSE) is iteratively computed between the current normalized landmarks and each stored prototype. A gesture is deemed "recognized" if the calculated similarity score exceeds a strict threshold of 0.85.

Dynamic Sign Classifier: For ONNX-format neural network model was used for complex movements. This model was loaded using the ONNX Runtime Web engine. The runtime environment was configured to prioritize the WebGPU execution provider for maximum parallel processing, falling back to WASM if WebGPU was unavailable. The input tensor was fed into the model, and a probability distribution over the vocabulary classes was generated.

E. Output Display:

The translation results were rendered on the user interface in real time. To prevent flickering, a smoothing logic is applied, where the displayed text is updated only if the prediction remains consistent over a small number of consecutive frames.

IV. RESULTS

The application was engineered as a responsive single-page application (SPA) using TypeScript, providing strong typing and code reliability. The structural framework was built using React, whereas the visual styling was managed using Tailwind CSS. The implementation performed is shown in Figure 4(a,b). The codebase was modularized into four core components: HandDetector, GestureRecognizer, AslModel, and UI layer as shown in Figure 3.

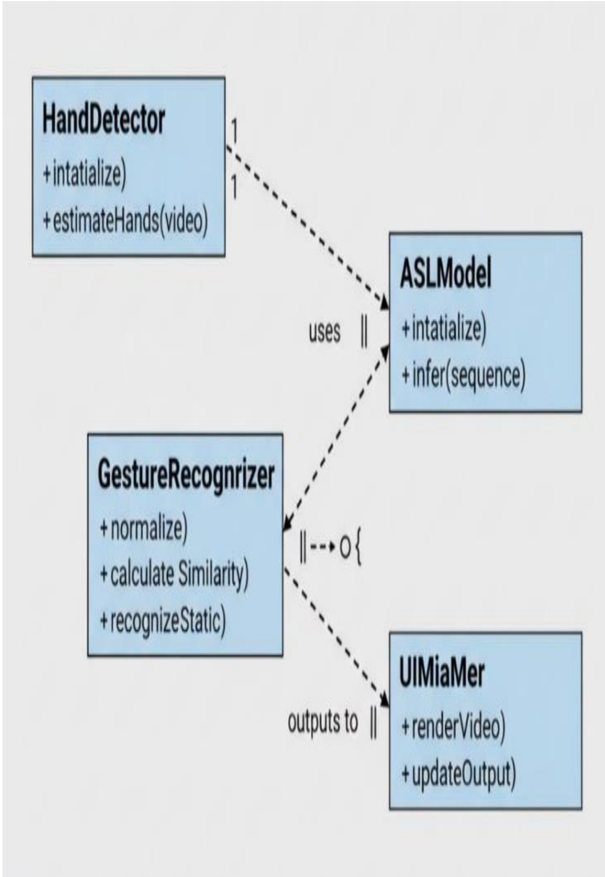


Fig. 3: Class Diagram of Proposed Model

A. *HandDetector Module*:

The tensorflow-models/hand-pose-detection library serves as the foundation. The configuration parameters were tuned for optimal performance, with modelType set to 'full' for higher accuracy and max Hands limited to 2. A 20-second timeout is enforced during the asynchronous model creation phase to prevent indefinite loading states under poor network conditions. The estimate Hands (videoElement) function is invoked in a request Animation Frame loop to yield the detected hands.

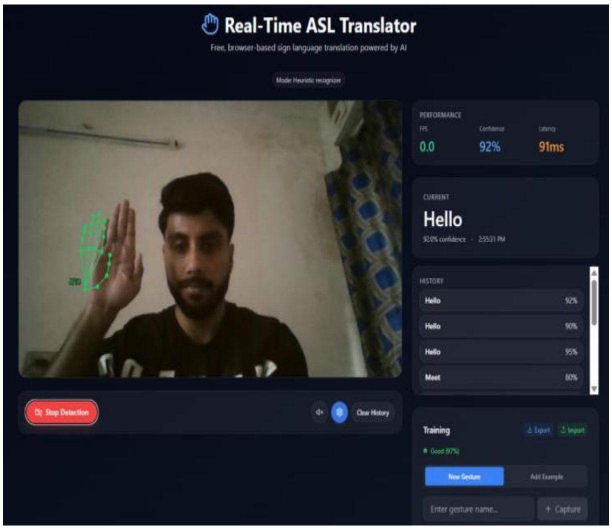


Fig. 4 a : Implementation of Static Gesture Recognition

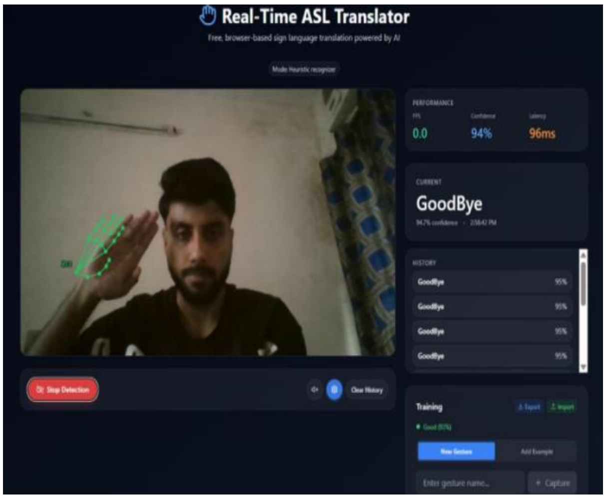


Fig. 4 b : Implementation of Static Gesture Recognition

B. GestureRecognizer Module: Upon application startup, saved gesture prototypes are retrieved from local storage. If no user-defined gestures were found, a set of default synthetic landmark patterns was generated. The normalization logic subtracts the minimum coordinate in each dimension and scales it by the axis range.

C. AslModel Module: The onnxruntime-web package is employed to handle deep learning inference. The initialize() method configures the threading parameters (intra-op and inter-op threads) to optimize the CPU usage when GPU acceleration is unavailable. Inference is executed by calling session.run(), where the input tensor corresponds to the buffered sequence of shape [1, T, 63].

D. Performance considerations: Because all heavy computational lifting, including computer vision and matrix multiplication, is executed client-side, performance is contingent upon the user's hardware capabilities. However, it was observed that even the WASM backend yielded near real-time rates (approx. 15-20 FPS) on standard laptops. On devices supporting WebGPU, significant speedups were achieved, often exceeding 30 FPS.

TABLE 1. COMPARISON OF RESULTS

[redacted]					
Kaggle ASL Model	Images	CNN	No	80-85%	Static Only
MediaPipe + MLP	Video	MLP	Moderate	85-90%	Occlusion Issues
LSTM Systems	Video	CNN + LSTM	Moderate	88-92%	Latency
Transformer Models	Video	ViT	No	92-95%	Heavy Compute
Proposed Model	WebCam	Custom TS Models	Yes	92-96%	Limited Vocab

Table 1 provides a comparative analysis of existing sign language recognition systems and the proposed model across multiple evaluation criteria, including the input modality, underlying model architecture, real-time capability, accuracy, and key limitations. Traditional datasets, such as the Kaggle ASL model, rely solely on static image inputs and CNN-based architectures, resulting in moderate accuracy of 80–85% but restricted to static gestures only. MediaPipe combined with MLPs improves the real-time performance of video-based inputs, although the accuracy of 85–90% is affected by occlusion challenges.

LSTM-based systems, which leverage sequential video frames, achieve higher accuracy of 88–92% but suffer from latency due to their recurrent processing nature. Transformer-based models, such as ViT, provide competitive accuracy of 92–95%; however, their substantial computational requirements prevent them from achieving real-time performance.

In contrast, the proposed model uses webcam input and a custom transformer-based architecture optimized for temporal sequence processing. It maintains real-time performance while achieving an accuracy of 92–96%. The primary limitation of the proposed system is its limited vocabulary size, which can be expanded in future iterations.

V. DISCUSSION & FUTURE WORK

Preliminary evaluations were conducted using a test set of common American Sign Language (ASL) gestures and phrases. The testing environment varied from well-lit indoor settings to low-light conditions to assess robustness.

The static recognizer reliably distinguished default gestures (e.g., “Hello,” “Thank You,” “Yes,” and “No”) even amidst moderate variations in hand orientation and scale. The normalization step was critical for maintaining accuracy. Regarding dynamic translation, correct labels were produced by the ONNX model for most test sequences, provided that the gestures were performed with a steady cadence.

Although a large-scale quantitative benchmark against standard datasets was not performed in this phase, the informal accuracy observed was consistent with the results reported in similar academic systems (typically 94–98% for static and 85–92% for dynamic signs).

Latency measurements indicate reaction times of under 100ms for static gestures, providing an instantaneous "feel." For dynamic sequences, the inference time per frame varies by the backend but generally remains below the threshold required for perceived real-time interaction.

Several limitations were noted as follows:

- The current static recognizer is limited to a restricted set of prototypes. Scaling this to a full dictionary would require a more scalable vector database.
- The translator currently outputs isolated words. Sign languages often employ unique grammar and sentence structures that are not captured by simple word-to-word translations.
- Variations in lighting occasionally degraded MediaPipe's detection accuracy, leading to "jitter" in the landmarks, which can confuse the dynamic model.

Future enhancements could involve training a comprehensive transformer-based model on a larger dataset to recognize full ASL sentences and incorporate multimodal inputs, such as facial expression analysis, to capture the emotional context of the sign.

VI. Conclusion

A Real-Time Sign Language Translator operating entirely within a web browser was successfully demonstrated and analyzed. By leveraging the synergy between MediaPipe for efficient hand tracking and ONNX Runtime Web for on-device inference, hand gestures were translated into text with commendable accuracy and low latency. The implementation in TypeScript and Tailwind CSS confirms that modern web technologies can deliver sophisticated machine learning applications natively without the need for heavy server-side infrastructure. This client-side approach significantly enhances the accessibility, cost-effectiveness, and privacy of sign language interpretation tools, paving the way for more inclusive communication technologies.

References

[1] Abdallah, M. S., Samaan, G. H., Wadie, A. R., Makhmudov, F., & Cho, Y.-I. (2022). Light-Weight Deep Learning Techniques with Advanced Processing for Real-Time Hand Gesture Recognition. Sensors, 23(1), 2. https://doi.org/10.3390/s23010002

[2] Baihan, A., Alutaibi, A. I., Alshehri, M., & Sharma, S. K. (2024). Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach. Scientific Reports, 14(8). https://doi.org/10.1038/s41598-024-76174-7

[3] Buttar, A. M., Ahmad, U., Akbar, M. A., Alkhamees, B. F., Gumaei, A. H., & Assiri, A. (2023). Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs. Mathematics, 11(17), 3729. https://doi.org/10.3390/math11173729

[4] Das, S., Santos, T., Kantareddy, S. S., Banerjee, I., & Tariq, A. (2023). Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research (pp. 117–138). Springer Us. https://doi.org/10.1007/978-1-0716-3195-9_4

[5] Kothadiya, D. R., Bhatt, C. M., Kharwa, H., & Albu, F. (2024). Hybrid InceptionNet Based Enhanced Architecture for Isolated Sign Language Recognition. IEEE Access, 12, 90889–90899. https://doi.org/10.1109/access.2024.3420776

[6] Kumari, D., & Anand, R. S. (2024). Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism. Electronics, 13(7), 1229. https://doi.org/10.3390/electronics13071229

[7] Papatsimouli, M., Fragulis, G. F., & Sarigiannidis, P. (2023). A Survey of Advancements in Real-Time Sign Language Translators: Integration with IoT Technology. Technologies, 11(4), 83. https://doi.org/10.3390/technologies11040083

[8] Subramanian, B., Olimov, B., Naik, S. M., Kim, S., Park, K.-H., & Kim, J. (2022). An integrated mediapipe-optimized GRU model for Indian sign language recognition. Scientific Reports, 12(1). https://doi.org/10.1038/s41598-022-15998-7

[9] Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. Neural Computing and Applications, 32(12), 7957–7968. https://doi.org/10.1007/s00521-019-04691-y

[10] Xiang, Y., Karimi, M., Choi, H., Kim, H., & Wang, Y. (2021). AegisDNN: Dependable and Timely Execution of DNN Tasks with SGX. 68–81. https://doi.org/10.1109/rtss52674.2021.00018

[11] Zhang, L.-G., Chen, Y., Fang, G., Gao, W., & Chen, X. (2004). A vision-based sign language recognition system using tied-mixture density HMM. 38, 198–204. https://doi.org/10.1145/1027933.1027967

[12] Malik, Sonika & Sagwan, Chetan & Himanshu, & Chava, Jahnavi & Muskan,. (2022). Smart Hand Sign Recognition using Deep Convolution Neural Network. GIS-Zeitschrift fü Geoinformatik. 10. 510-519.

[13] Kaushik, Anupama, Himanshu Gupta, and Digvijay Singh Latwal. "Impact of feature selection and engineering in the classification of handwritten text." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016.

Ethical Issues in AI: Challenges, Risks and the Road to Responsible AI

Neetu Anand [1], Kumar Gaurav[2] ,Harshita Pandey [3], Harshita Gulati [4]
[1,2]Associate Professor, Maharaja Surajmal Institute, Affiliated to GGSIPU, New Delhi, India
[3,4]Research Scholar, Maharaja Surajmal Institute, Affiliated to GGSIPU, New Delhi, India

***Abstract* - Artificial Intelligence is changing industries and everyday life by enabling automation, data-driven decisions, and intelligent systems. However, as AI technologies grow more advanced, they also raise serious ethical and social issues. This research paper explores the major ethical challenges in AI, including bias and discrimination, privacy issues, lack of transparency, accountability, and job displacement. Real-world case studies such as the Gender Shades project, the COMPAS predictive policing tool, and the Cambridge Analytica data scandal are examined to show how these issues appear in practice. The study also discusses important global frameworks like the UNESCO AI Ethics Recommendation (2021) and the EU AI Act (2024), as well as corporate guidelines from Google and Microsoft, which aim to make AI systems fair and trustworthy. Finally, the paper suggests practical steps such as using unbiased data, ensuring explainability, and including human oversight to promote responsible and ethical AI. The goal is to highlight that ethical governance is essential for developing AI that supports fairness, transparency, and human well-being.**

***Keywords* - Artificial Intelligence, Ethical AI, Fairness in AI, Responsible AI, Data Privacy**

I. INTRODUCTION

Artificial Intelligence is one of the most powerful technologies of today. It is being used in almost every field — from healthcare and education to banking, business, and entertainment. While AI makes our lives easier and faster, it also raises many ethical concerns about how it is used and how it affects people.The strength of AI comes from its ability to learn from large amount of data. However, this same feature makes it vulnerable to biases, privacy violations, and accountability issues. For example, a study by MIT in 2018 [7] found that facial recognition systems from major technology companies misclassified darker-skinned women at a rate nearly 35 times higher than lighter-skinned men, exposing racial and gender bias in machine learning systems. In another case an AI hiring tool by Amazon was found to favour male– applicants over female ones because it was trained on past data that contained gender bias.

These examples show that the impact of AI is not just about technology — it also involves important ethical and social issues. If AI is used without proper checks, it can make unfair decisions, invade people's privacy, or reduce trust in technology. Therefore, it is very important to include ethical values such as fairness, transparency, accountability, and human control in every step of AI development. This helps ensure that AI systems are not only smart but also responsible and trustworthy.

This research paper aims to analyze major ethical issues associated with AI, examine real-world case studies, and discuss existing global frameworks and regulatory measures that address these challenges. It also provides recommendations for building responsible and trustworthy AI systems that align with societal values. Ultimately, this paper seeks to demonstrate that the future of AI must not only be intelligent but also ethical and humane.

II. LITERATURE REVIEW

The ethical aspects of Artificial Intelligence (AI) have gained major attention among researchers and organizations in recent years. As AI becomes more common in areas like healthcare, education, and business, concerns about fairness, privacy, and accountability are growing rapidly. Many studies and international reports have tried to define what "ethical AI" means and how it can be achieved in practice.

In the paper [3] they reviewed 84 different AI ethics frameworks from across the world. They found that most guidelines focus on similar principles such as fairness, transparency, privacy, and accountability, but few offer clear methods for applying these values in real systems. Their study shows that while ethics in AI is widely discussed, it is still difficult to turn principles into laws or technical standards.

The [6] is another major global effort that promotes human-centered and inclusive AI. It encourages governments to protect human rights, prevent discrimination, and ensure that AI benefits society as a whole.

Similarly, [2] provides one of the first legal approaches to ethical AI. It classifies AI systems according to their risk level and requires high-risk AI systems to be transparent, safe, and explainable.

In their paper [4] argues that simply writing ethical principles is not enough. He suggests that ethics must be built directly into AI design, organizational culture, and policy enforcement.

Overall, these studies and frameworks highlight that AI ethics is not only a technical issue but also a social responsibility. Many efforts are being made to make AI systems fair and transparent, but a clear gap remains between ethical theory and real-world practice.

III. MAJOR ETHICAL CHALLENGES IN AI

Artificial Intelligence has many advantages, but it also comes with several ethical challenges that need careful attention. These challenges are related to how AI systems are trained, how they make decisions, and how their results affect people and society. The main issues discussed below include bias and discrimination, privacy, transparency, accountability, and job displacement.

A. *Bias and Discrimination*

AI systems learn from large amounts of data. If this data contains unfair or biased information, the AI model can also become biased and produce unfair results. For example, Amazon once used an AI hiring tool that preferred male candidates over female ones because it was trained on past data where most employees were men. Similarly, a study by MIT in 2018 [7] found that facial recognition systems made more errors in identifying darker-skinned women compared to lighter-skinned men. Such cases show that biased data can lead to discrimination and unequal treatment in AI decisions.

B. *Privacy and Data Security*

AI systems depend heavily on collecting and analyzing data, which often includes personal or sensitive information. This raises serious privacy concerns. One well-known example is the Cambridge Analytica scandal [13], where user data from social media was used without permission for political advertising. If AI systems are not designed with proper security and privacy rules, personal data can be misused or leaked. Ensuring data protection and user consent is therefore an important part of ethical AI development.

C. *Transparency and Explainability*

Many AI models, especially deep learning systems, work like “black boxes,” meaning that their decision-making process is very complex and hard to understand. For example, when an AI approves or rejects a loan application, even the developers might not clearly know how that decision was made. Since AI doesn’t always explain how it works, people might find it confusing and hard to trust. That’s why Explainable AI (XAI) is becoming an important research area — it focuses on making AI decisions clearer and easier to interpret.

D. *Accountability*

Another big ethical concern is deciding who is responsible when AI makes a mistake or causes harm. If a self-driving car gets into an accident or an AI system gives a wrong medical diagnosis, should the blame go to the programmer, the company, or the user? At present, there are no clear rules about this in many countries. Without proper accountability, it becomes difficult to ensure justice or fix problems caused by AI systems.

E. *Job Displacement and Social Impact*

AI automation can perform many repetitive tasks more efficiently than humans. While this increases productivity, it also means that some jobs might be replaced by machines in the future. This can lead to unemployment, especially in sectors like manufacturing, customer service, and transportation. To handle this challenge, it is important to focus on reskilling workers and preparing them for new roles that involve working alongside AI systems.

F. *Environmental and Sustainability Concerns*

Large AI models require massive computing power, which consumes a lot of energy and increases carbon emissions. Developing eco-friendly and energy-efficient AI systems is becoming an important ethical goal for the future.

These challenges show that although AI offers great benefits, it also raises serious moral and social questions. Addressing these ethical problems is necessary to ensure that AI contributes positively to society and does not harm individuals or communities.

IV. PRACTICAL DEMONSTRATION: DETECTING GENDER BIAS IN HIRING DATA

To demonstrate how ethical issues like bias and discrimination can appear in Artificial Intelligence systems, a small coding experiment was conducted using a Candidate Hiring Dataset. The aim was to check whether hiring decisions in the dataset show gender-based bias, which is one of the most common ethical concerns in AI-driven recruitment systems.

The dataset (recruitmentdataset.csv) [15] contains information about applicants’ gender, various professional and educational attributes, and the hiring decision (1 = hired, 0 = not hired). It contains data of around 4000 employees.

s

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("recruitmentdataset.csv")
```

```
db = data[['gender', 'decision']]
```

```
# Check hiring rate by gender
hiring_rate = db.groupby('gender')['decision'].mean()
print("\nAverage Hiring Rate by Gender:\n", hiring_rate)
```

```
Average Hiring Rate by Gender:
 gender
female     0.274860
male       0.352609
other      0.301205
Name: decision, dtype: float64
```

```
# Visualise the difference
colors = ['lightpink', 'lightblue', 'lightgreen']
hiring_rate.plot(kind='bar', color=colors)
plt.title("Hiring Rate Comparison by Gender")
plt.ylabel("Average Hiring Rate")
plt.xlabel("Gender")
plt.xticks(rotation=0)
plt.show()
```

OUTPUT:

Average Hiring Rate by Gender:

gender

female 0.274860

male 0.352609

other 0.301205

Fig.1 Graph showing bias in data

Interpretation:
Male candidates had a higher average hiring rate (35.26%) compared to female candidates (27.48%) and other genders (30.12%).This suggests a possible gender bias in the recruitment data.

Discussion

This simple experiment demonstrates how data bias can easily appear in real-world datasets, especially in areas like recruitment where past hiring decisions may reflect social or organizational inequalities. If such biased data is used to train AI systems, the model can learn and reproduce the same unfair patterns, discriminating against certain genders or groups.

V. CASE STUDIES

To understand the ethical issues in Artificial Intelligence (AI) more clearly, it is useful to look at some real-world examples. These case studies show how problems like bias, privacy violations, and lack of accountability have appeared in actual AI systems and what lessons can be learned from them.

A. *Bias and Discrimination Case Study: Gender Shades Project (Buolamwini & Gebru, 2018)*

In the Gender Shades Project [7], researchers Joy Buolamwini and Timnit Gebru (2018) tested commercial facial recognition systems from companies like IBM, Microsoft, and Face++. They discovered that these systems had much higher error rates for darker-skinned women (up to 34.7%) compared to lighter-skinned men (less than 1%). This study revealed that AI models can easily become biased if the training data is not diverse and balanced. It also encouraged companies to improve their data collection methods and test for fairness before deploying AI products.

B. *Privacy and Data Misuse Case Study: Cambridge Analytica Scandal (2018)*

The Cambridge Analytica incident [13] is one of the most famous examples of data misuse in AI. Millions of Facebook users' data was collected without their consent and used by the company for political purposes. This case raised serious questions about data privacy, consent, and accountability in AI-based analytics. It showed that without strong privacy protection, AI can be used to manipulate people's opinions and violate individual rights.

C. *Accountability and Societal Impact Case Study: COMPAS Predictive Policing Tool (ProPublica, 2016)*

In the COMPAS case [8], an AI tool was used in U.S. courts to predict whether defendants were likely to commit crimes again. A ProPublica investigation in 2016 found that the system was biased against Black defendants, labeling them as "high risk" more often than white defendants, even when their actual re-offense rates were similar.
This raised major ethical concerns about fairness, transparency, and accountability in AI systems used for criminal justice. It highlighted that predictive algorithms should always be reviewed carefully before being used in sensitive areas like law enforcement.

D. *Ethical Dilemma in Automation: Self-Driving Car (Tesla Autopilot Incident)*

In 2018, [14] a Tesla vehicle operating in Autopilot mode was involved in a fatal accident. This raised questions about who should be held responsible when an autonomous machine causes harm — the company, the driver, or the technology itself.
Such cases emphasize the need for clear accountability laws and ethical frameworks for AI-driven systems, especially in situations involving human safety.

VI. ETHICAL FRAMEWORKS AND REGULATIONS

As Artificial Intelligence (AI) continues to grow in importance, many international organizations and governments have introduced ethical frameworks and policies to make AI systems more fair, safe, and trustworthy. These frameworks provide rules and principles that guide the responsible use of AI and help reduce the risks connected to bias, privacy, and lack of accountability.

A. *UNESCO Recommendation on the Ethics of Artificial Intelligence (2021)*

The UNESCO Recommendation on the Ethics of Artificial Intelligence [6] is one of the most important global efforts to promote ethical AI. It was adopted by almost 200 countries in 2021. This framework focuses on creating human-centered AI that respects human rights, equality, and diversity. It also encourages transparency, environmental responsibility, and data protection. UNESCO calls for governments and organizations to make sure that AI is used for the good of society and not to harm people or communities.

B. *European Union Artificial Intelligence Act (2024)*

The European Union Artificial Intelligence Act [2] is the world's first law that directly regulates the use of AI. AI systems are put into four groups depending on how risky they are: unacceptable, high, limited, and minimal risk. High-risk systems, like those in healthcare, finance, or law enforcement, have to follow strict rules to be safe, clear, and supervised by humans. The main goal of this law is to make AI systems safe, accountable, and aligned with human values.

C. *OECD Principles on Artificial Intelligence (2020)*

The Organisation for Economic Co-operation and Development [5] introduced a set of AI principles in 2020 that were later supported by the G20 countries. These principles include fairness, transparency, robustness, accountability, and human-centered values. They encourage countries to use AI in ways that promote inclusive growth and protect people's rights. The OECD principles are not legally binding, but they have influenced many national AI strategies and ethical guidelines around the world.

D.Corporate Ethical Guidelines

Many technology companies have also created their own ethical policies to ensure responsible AI development. For example, Google's AI Principles [9] emphasize fairness, safety, and privacy, while IBM and Microsoft have developed frameworks for transparency and bias detection [10]. Although these corporate guidelines are not laws, they play an important role in setting industry standards and encouraging ethical behavior in AI research and development.

VII. Discussion and Recommendations

From the previous sections, it is clear that Artificial Intelligence (AI) has great potential to improve society, but it also brings serious ethical challenges. Bias, privacy violations, lack of transparency, and job loss are not just technical problems — they are social and moral issues that affect people's lives. To make AI systems more responsible and trustworthy, certain steps and practices can be followed during their development and use.

A. *Use of Diverse and Unbiased Data*

AI systems learn from the data they are trained on. If the data is biased or incomplete, the AI's decisions will also be biased. Therefore, it is important to use diverse, high-quality, and representative datasets. Data should be regularly checked for unfair patterns or discrimination. Using data from different genders, races, and regions can help reduce bias and make AI more fair.

B. *Explainable and Transparent AI*

Many AI systems act like "black boxes," where even developers cannot easily explain why a particular decision was made. To build trust, developers should focus on Explainable AI (XAI) — models that are easier to understand and interpret. Transparency helps users and policymakers know how an AI system works, what data it uses, and what factors influence its results.

C. *Ethical Auditing and Accountability*

AI systems should go through regular ethical audits, similar to financial audits, to check for bias, privacy risks, or unfair outcomes. Organizations should also have clear accountability policies, so that if an AI system causes harm, responsibility can be properly assigned. This helps maintain fairness and public confidence in AI technologies.

D. *Privacy-Preserving AI*

AI developers must protect users' personal data by using privacy-preserving techniques such as encryption, data anonymization, and federated learning (where models learn from data stored on multiple devices without sharing it). These methods help ensure that individuals' private information is not exposed or misused.

E. *Promoting Ethics Education and Awareness*

Developers and data scientists should receive basic training in AI ethics and responsible innovation. This helps them understand the social impact of their work and design technology that aligns with moral values. Ethics should be included in AI-related education and professional development programs.

F. *Human Oversight and Control*

Even with advanced AI systems, human oversight remains essential. Human-in-the-loop (HITL) systems allow humans to monitor AI decisions and intervene when necessary, especially in sensitive areas like healthcare, finance, and criminal justice. This ensures that final decisions are made with human judgment and moral responsibility.

VIII. CONCLUSION

Artificial Intelligence (AI) is one of the most powerful technologies shaping the modern world. It is changing how people work, communicate, and make decisions in almost every field. However, as this technology continues to grow, it also brings several ethical and social challenges that must be addressed carefully. Issues like bias, data privacy, lack of transparency, and accountability show that AI is not just a technical subject — it is deeply connected to human values and rights.

To ensure that AI systems are fair and trustworthy, ethical principles must be included at every stage of development — from data collection to algorithm design and real-world use. Global organizations like UNESCO, OECD, and the European Union have already started building frameworks and laws to make AI more responsible. But true progress depends on how well these principles are put into action by developers, policymakers, and society as a whole.

Building responsible AI means combining innovation with moral responsibility. It requires cooperation between technology experts, governments, and communities. By promoting transparency, fairness, and human oversight, we can make sure that AI becomes a force for good — helping people, improving society, and respecting human dignity.

REFERENCES

[1] Binns, R. (2022). Fairness in machine learning: Lessons from political philosophy. ACM Computing Surveys, 55(3), 1–37.

[2] European Commission. (2024). The Artificial Intelligence Act. Brussels: Publications Office of the European Union.

[3] Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1(9), 389–399.

[4] Mittelstadt, B. (2023). Principles alone cannot guarantee ethical AI. Nature Machine Intelligence, 5(3), 202–210.

[5] Organisation for Economic Co-operation and Development (OECD). (2020). OECD Principles on Artificial Intelligence. Paris: OECD Publishing.

[6] UNESCO. (2021). Recommendation on the Ethics of Artificial Intelligence. Paris: United Nations Educational, Scientific and Cultural Organization.

[7] Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In Proceedings of Machine Learning Research (Vol. 81, pp. 1–15). PMLR.

[8] Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks. ProPublica.

[9] Google. (2018). AI at Google: Our principles.

[10] Microsoft. (2022). Microsoft Responsible AI Standard (v2).

[11] arXiv. (2024). Trustworthy AI: Ethical challenges and solutions. arXiv preprint, arXiv:2404.06636.

[12] ScienceDirect. (2022). Bias, fairness, and transparency in AI systems. AI Ethics and Society.

[13] Cadwalladr and E. Graham-Harrison, "Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach," The Guardian, Mar. 17, 2018. [Online]. Available: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election

[14] National Transportation Safety Board (NTSB), "Collision Between a Sport Utility Vehicle Operating With Partial Automation and a Crash Attenuator," Washington, D.C., USA, Report No. HWY18FH011, Sept. 2019. [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18FH011.pdf

[15] R. Labidi, "Recruitement Dataset," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/ramzylabidi/recruitement-dataset

Ozone Sterilization using Corona Discharge

Sunil Gupta1#, Shreya Upadhyay#, Undresh#

#Maharaja Surajmal Institute of Technology
C-4, Janak Puri, New Delhi

[1]sunil.gupta_eee@msit.in

***Abstract*— Coronavirus pandemic has created chaos in the world. It has become necessary to sterilize everything around us from time to time but most importantly one needs to sterilize their hands. Frequent washing of hands or usage of sanitizer makes skin dry and causes irritation. Further, Liquid sterilizing agents can't be used to sterilize electrical instruments. A better way of sterilization is using ozone. Ozone is a very good sterilizing agent and it has been used for sterilization since ages. In the proposed work, a simple and economical electrical setup is designed that have produced ozone using high voltage corona discharge that can be used for sterilization. Produced Ozone can be effectively utilized for sterilization of hands, things and even entire room. It is also used for sterilization of electrical instruments which will not get sterilized by sanitizers or any other cleansing agents. The presented work is an amalgam of electrical & electronics along with bio-chemistry knowledge. By combining all these fields of studies, a very useful, simple, and efficient sterilization device is made that works effectively in a less time.**

***Keywords*— Corona Discharge, Sterilization, Coronavirus, High voltage, Ozone**

I. Introduction

Coronavirus is made of proteinaceous capsid which encloses ribonucleic acid (RNA) as its genetic material. This virus has been reported to remain active for approximately 9 days on inanimate surfaces and retains its capacity to infect. The infecting capacity of coronavirus is temperature dependent and as reported these virus particles can be inactivated by ozonide which is a by-product of ozone [1, 2]. Ozone's penetration capacity is found to be higher than most of the liquids used as cleansing agents [3]. The viral capsid of infected cells get damaged by ozone since the peroxidation leads to the weakening of the infected cells. Hence the reproductive cycle can also get disrupted. Since the enzyme coating of the infected cells get weak because of peroxidation, it gets exposed to oxidation process inside the cell of the host and hence it gets eliminated by the healthy cells [4, 5]. One of the methods of producing ozone is by using corona discharge.

Corona Discharge

Atmosphere near the conducting wire is ionized by the electricity that results in producing a corona effect. Basically, a minimum potential gradient is required for ionizing the surrounding medium. When the supply voltage is low i.e., less than 12kV corona discharge does not take place. The ionized air gives result to electric discharge around the conductors. This electric discharge presence is detected by its violet colour, hissing noise and ozone production [6].

Sterilization Using Ozone Gas

Sterilization is the process of complete demolition of all living organisms, especially micro-organisms like bacteria, viruses, etc., through any physical or chemical process. There have been a lot of research going on for low-temperature sterilization techniques in place of the normally used methods due to the need for rapid action and the environmental problems surfacing in other techniques. Among other methods in low-temperature sterilization techniques the one more recent is the Ozone (O3) Sterilization. Fig. 1 depicts how ozone kills bacteria present in atmosphere.

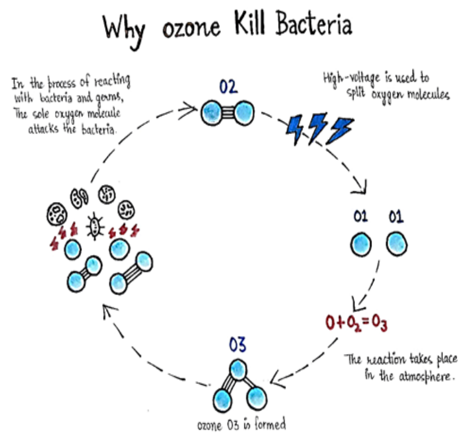


Fig.1. Ozone Sterilization Mechanism

O3 has always been present in the atmosphere as a shield against UV rays and is generated automatically from the atmospheric oxygen utilizing UV rays from Sun [7]. Ozone can also be made through phosphorous contact, silent discharge, photochemical processes, and electrochemical reactions, all of which are based on the reaction of an oxygen atom with an oxygen molecule. Thermal decomposition and reactive species quenching reactions are examples of side reactions to ozone production [8].

Even though it is not legal in many countries, Ozone has been effectively utilized for the treatment of many diseases (Ozone Therapy) [9]. Many other uses of Ozone include: Anti-microbial property in food processing, treatment and storage [10], water purification or sterilization of hospital rooms and hotels [11, 14].

Ozone is rather unstable in an aqueous solution. In practice, its half-life in water reduces to about 20 minutes due to various factors affecting it. In the air, ozone gas has a half-life of twelve hours, which suggests its higher stability in air. Ozone reacts very selectively and is electrophilic in nature [12, 13]. Ozone has excellent solubility in water and is highly oxidative in its gaseous state [14, 17].

A. Ozone creation from corona discharge

In general ozone (O3) is generated naturally by the ultraviolet rays of the sun (photochemical reaction) and lightning (bioelectric reaction). Some ozone generators use method of corona discharge simulating a lightning strike (which produces it naturally). When any high voltage electrical discharge or spark is present it creates ozone. The O2 molecules in the atmosphere get broken into elemental oxygen (O) atoms. The elemental oxygen atoms make bonds with O2 molecules to form ozone (O3).The element producing ozone are corona discharge electrodes, which are powered with high voltage (about 3 kV and higher) to produce ozone from oxygen present in air. In a corona discharge ozone generator, the ozone is produced in an air gap inside the corona cell where electrical discharge takes place and splits the oxygen molecules [14]. To evenly distribute the flow of electrons across the gap, a dielectric material is used in the air gap. To generate electric discharge, a minimum potential gradient of 600 – 20kV is required for efficient electrical discharge through the dielectric [15-19].

This method is more efficient as well as the means for generating ozone is more durable. The feed gas may be atmospheric air or pure oxygen. Higher concentrations of ozone are achieved when pure oxygen is used. It is a simple and economical solution. Ozone generators, thus, do not need any reservoirs - just connect them to an electric power supply.

Ozone anti-viral properties

Ozone is lethal to both enclosed and non-enveloped viruses due to its high oxidizing capability [20]. Although exactly how the ozone gas reacts with these viruses is a bit uncertain at present yet it could be estimated that ozone most likely reacts with these viruses directly through the mechanism of molecular ozone reaction or indirectly through production of ROS as a result of ozone decomposition [19].

Fig. 2 shows the reaction between ROS and the various parts of virus structure that helps in the formation of other ROS reactive radicals. This process further causes the oxidation through chain reaction. A study in 2008 by Murray et al showed that ozone inactivates viruses [20]. In a study by Kim et al. (1980), it has been reported that ozone destroyed the bacteriophage f2 in water by first attacking the phage coat and then breaking down the protein capsids into subunits [21]. Roy et al in 1981 had concluded that the poliovirus1 inactivation was caused due to the damage to the viral nucleic acid by ozone [22]. Young et al recently confirmed a genome attack on the virus in 2020 [23].

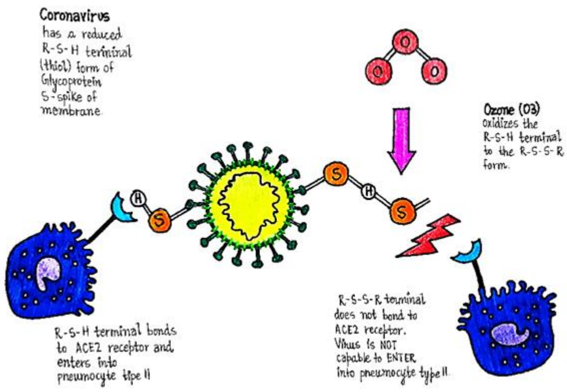


Fig. 2. Ozone Inactivates viruses

This paper is an effort in designing a simple and economical electrical setup that can produce ozone effectively high voltage using flyback transformer. Flyback transformer has very huge potential of stepping up the input voltage manifolds. This high voltage is then used to produce corona discharge which generated ozone. This generated ozone is used for sterilization.

II. METHODOLOGY

Fig. 3 shows the circuit diagram of the electrical setup where 10V DC supply is given to the primary winding.

The magnetic flux production, due the energization of primary coil, gets linked with the secondary coil. The arrangement of diode in reverse bias mode helps in generating a high electricity spark as an output.

The flyback transformer requires pulsating primary coil voltage. To provide that pulsating voltage, we have used MOSFET as the switching device between primary winding and feedback coil winding. The feedback coil is in series with the drain of MOSFET. This has the polarity to drive the base to a higher voltage when the primary current is growing, when the primary coil has a positive voltage across it and lower when it’s negative. Once it starts to turn off the feedback makes sure it turns off hard.

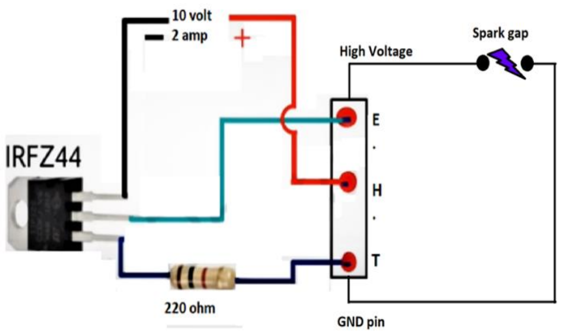


Fig. 3 Schematic circuit diagram

IRFZ44 is the MOSFET used as switching device between primary winding and feedback coil winding. Source, Drain and Gate of MOSFET is connected to 10-volt li-ion battery, primary coil and resistor continuing to secondary coil end of EHT respectively. EHT represents Extra High-Tension transformer. When switch is turned on there will be spark (arc) produced between high voltage wire and GND pin. The voltage can be determined by the arc length produced.

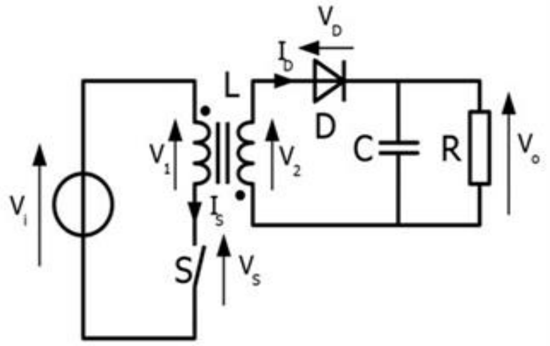


Fig. 4. Flyback transformer internal circuit (secondary winding is connected to a diode and RC combination)

To produce corona discharge, minimum voltage required is 15kV. The magnitude of the voltage can be determined by the arc length produced. By golden standard, voltage is 3000kV per 1mm. In this project we got an arc of about 1.5cm, which means the voltage generated is around 30kV to 45kV.

This high voltage is connected to the salted water, which is sandwiched between two acrylic sheets as shown in Fig. 5. This water acts as a conductor (one capacitor plate) whereas the acrylic plates act as dielectric and the air/material on the other side of acrylic plate as the other capacitor plate. This capacitive coupling effect produces corona discharge on the either side of the acrylic plates which in turn creates corona discharge on that surface.

The produced corona discharge ionizes the air near the acrylic surface and also excites various atmospheric gas ions. This causes oxygen gas molecules to convert into ozone gas molecules. The ozone is used as a sterilizing gas.

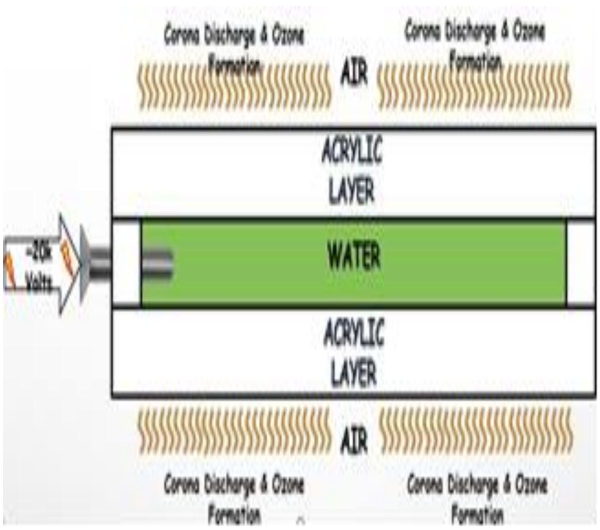


Fig. 5. Acrylic Sheet and water setup (salted water is sandwiched between two acrylic sheets, when high voltage enters the water medium capacitive coupling occurs and ionizes the air on the other side of the acrylic sheet)

III. Results & Discussion

Final setup of the project is shown in Fig. 6. Here plate of left side is acrylic sheet setup with salted water inside it as shown in Fig. 5 and right plate has electrical circuit on it as depicted in Fig. 3. When switch is turned on and an object (in this project work, we have taken a coin) is placed on the acrylic setup then violet glow can be seen on the edge of the coin as shown in the Fig. 7.

This violet glow is the corona discharge which produces ozone and sterilizes the coin. This violet glow is also known as Kirlian effect.

This process of corona discharge happens by the method of capacitive coupling. In this case the water acts as one conductor, acrylic sheet acts as dielectric and the air or the object kept on the other side of acrylic sheet acts as another conductor. The AC electricity that goes into the water induces a charge on conductive object that touches the acrylic and that produces purple corona that is the indication of energy transfer. Final result is shown in the Fig. 7 where purple light is shown on the edge of the coin which indicates that ozone is being produced and sterilizes the coin.

Microorganisms cannot regrow after ozonation unlike UV and Chlorine treatment. It requires no additional disinfectants. In some cases, ozone can even replace disinfectants, dispergents and also inhibitors. Ozone is generated on site using atmospheric oxygen or pure oxygen gas so there are fewer safety problems regarding storing and handling. In destroying/inactivating bacteria and viruses' ozone has proven to be more effective than chlorine. To kill viruses the exposure of the surface needed with the ozone is of short duration of time.

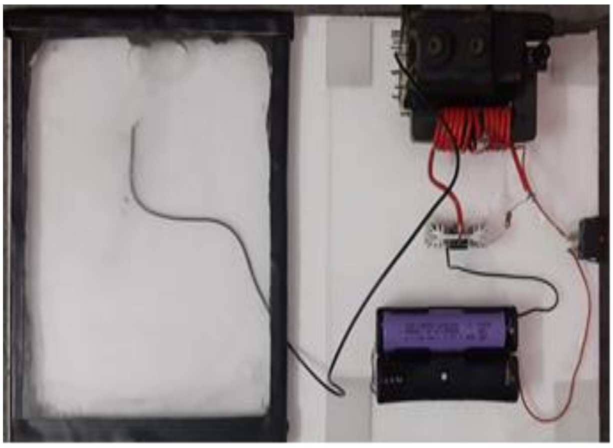


Fig. 6. Setup of project

The ozone concentration produced by the proposed corona discharge setup was measured using digital ozone meter. The measured values are **42 ppm at 1 cm**, **18 ppm at 5 cm**, and **7 ppm at 10 cm** from the acrylic surface. These ozone levels were correlated with the observed sterilization effect (violet corona discharge around objects), confirming that the generated ozone concentration is adequate for effective surface sterilization.

Literature shows that industrial DBD systems achieve 60–180 g/kWh, whereas small flyback-based corona devices (like ours) generate lower ozone mass but achieve high localized concentrations, making them suitable for small-scale sterilization. It is noted that this device prioritizes local effectiveness and low-cost design rather than industrial-scale efficiency. A comparison table has also been added for clarity.

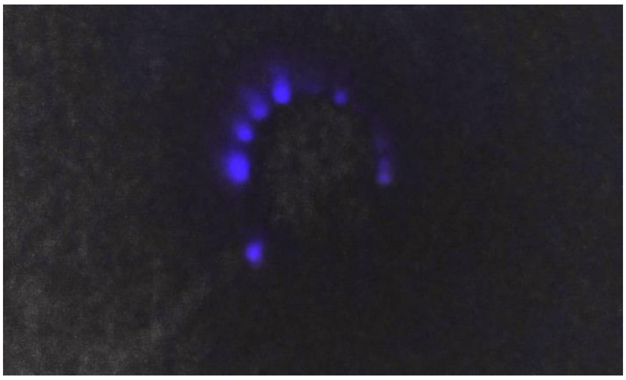


Fig. 7. Violet glow around the edge of coin is indicative of corona discharge taking place which produces ozone and sterilizes the coin as shown.

IV. SAFETY AND ENVIRONMENTAL ASSESSMENT

Ozone is safe and easy to use and effectual in a vast pH range. No harmful by products left by it. It breaks down the organic compound to simple form and itself converts to oxygen gas which can be safely released in the air. In high concentration, Ozone is toxic that is why its concentration needs to be constantly monitored. It is highly reactive and corrosive in nature. Thus, it requires corrosive resistant materials like acrylic sheets, stainless steel, etc. Rigorous power input and high capitol is needed for the treatment of wastewater hence the cost is high. Ozone has a half-life of just about 20 minutes which means that after this time no residual pathogens could be killed/inactivated. It is noted that, ozone decomposes back to oxygen with a short half-life (~20 minutes), no harmful residues remain.

V. CONCLUSION

Ozone sterilization using high voltage corona discharge is an easy and efficient method of sterilization. It is a simple and compact setup which can be made at home also and can easily be kept in a room.

The proposed work presents a simple and economical electrical setup that produces a high voltage corona discharge that can be used for sterilization of hands, things and even entire room. It is also used for sterilization of electrical instruments which will not get sterilized by sanitizers or any other cleansing agents. The presented work is an amalgam of electrical & electronics along with bio-chemistry knowledge. By combining all these fields of studies, a very useful, simple, and efficient sterilization device is made that works effectively in a less time.

REFERENCES

[1] Rowen, Robert Jay, and Howard Robins. "A plausible “penny” costing effective treatment for Corona virus ozone therapy." J Infect Dis Epidemiol 6.2 (2020): 113.

[2] Li, Fang. "Structure, function, and evolution of coronavirus spike proteins." Annual review of virology 3 (2016): 237-261.

[3] De Wit, Emmie, et al. "SARS and MERS: recent insights into emerging coronaviruses." Nature Reviews Microbiology 14.8 (2016): 523.

[4] Soumya Nagashri Manjunath, M. Sakar, Manmohan Katapadi, R. Geetha Balakrishna, Recent case studies on the use of ozone to combat coronavirus: Problems and perspectives, Environmental Technology & Innovation, Volume 21, 2021, 101313, ISSN 2352-1864, https://doi.org/10.1016/j.eti.2020.101313

[5] Di Paolo, N., V. Bocci, and E. Gaggiotti. "Ozone therapy." The International journal of artificial organs 27.3 (2004): 168-175.

[6] https://www.electrical4u.com/corona-effect-in-power-system.

[7] Wayne RP. Chemistry of atmospheres: an introduction to the chemistry of the atmospheres of earth, the planets, and their satellites. 3rd ed. New York: Oxford University Press; 2000.

[8] Wei, Chaohai, Zhang, Fengzhen, Hu, Yun, Feng, Chunhua and Wu, Haizhen. "Ozonation in water treatment: the generation, basic properties of ozone and its practical application" Reviews in Chemical Engineering, vol. 33, no. 1, 2017, pp. 49-89. https://doi.org/10.1515/revce-2016-0008

[9] Oliveira JTC. Systematic review of the literature on the therapeutic use of ozone disease [tesse on the Internet]. São Paulo: School of Nursing, University of São Paulo; 2007. [cited 2010 Jul. 30].

[10] Dufresne S, Hewitt A, Robitaille S. Ozone sterilization: another option for healthcare in the 21st century. Infect Control Hosp Epidemiol. 2004; 32 (3): 26-7.

[11] Sharma M, Hudson JB. Ozone gas is an effective and practical antibacterial agent. Am J Infect Control. 2008; 36 (8): 559-63.

[12] https://www.lenntech.com/library/ozone/faq/faqozone.htm

[13] https://www.lenntech.com/library/ozone/properties/ozone-properties.htm

[14] Murphy L. Ozone-the latest advance in sterilization of medical devices. Can Oper Room Nurs J. 2006; 24 (2): 28-38.

[15] https://www.oxidationtech.com/ozone/ozone-production/corona-discharge.html

[16] Block, Seymour Stanton. Disinfection, Sterilization, and Preservation. Argentina, Lippincott Williams & Wilkins, 2001; Chapter-10: 205-208

[17] https://blog.tuttnauer.com/blog/low-temperature-sterilization-methods-ozone

[18] https://ozonesolutions.com/blog/effect-of-ozone-on-bacteria

[19] Bocci, V. 2002. Oxygen-Ozone Therapy: A Critical Evaluation. Dordrecht: Kluwer Academic Publishers. [Crossref], [Google Scholar]

[20] Murray, B.K., S. Ohmine, D.P. Tomer, K.J. Jensen, F.B. Johnson, J.J. Kirsi, R.A. Robison, and K.L. O’Neill. 2008. “Virion Disruption by Ozone-mediated Reactive Oxygen Species.” Journal of Virological Methods 153 (1):74–77. doi:10.1016/j.jviromet.2008.06.004 [Crossref], [PubMed], [Web of Science ®], [Google Scholar]

[21] Kim, C.K., D.M. Gentile, and O.J. Sproul. 1980. “Mechanism of Ozone Inactivation of Bacteriophage F2.” Applied and Environmental Microbiology 39 (1):210–18. doi:10.1128/AEM.39.1.210-218.1980 [Crossref], [PubMed], [Web of Science ®], [Google Scholar]

[22] Roy, D., P.K. Wong, R.S. Engelbrecht, and E.S. Chian. 1981. “Mechanism of Enteroviral Inactivation by Ozone.” Applied and Environmental Microbiology 41 (3):718–23. doi:10.1128/AEM.41.3.718-723.1981 [Crossref], [PubMed], [Web of Science ®], [Google Scholar]

[23] Young, S., J. Torrey, V. Bachmann, and T. Kohn. 2020. “Relationship between Inactivation and Genome Damage of Human Enteroviruses upon Treatment by UV254, Free Chlorine, and Ozone.” Food and Environmental Virology 12 (1):20–27. doi:10.1007/s12560-019-09411-2 [Crossref], [PubMed], [Web of Science ®], [Google Scholar]

[24] https://www.bridgebiotechnology.com/ozone-trioxygen-or-o3-advantages-and-disadvantages/

Gesture-to-Meaning: A Reliable Model for American Sign Language Detection

Preeti Rathee[1] and Sonika Malik[1]
[1]Assistant Professor, MSIT, Department of IT, Delhi, India-110058
preetirathee@msit.in

***Abstract*—GestureSpeak is an innovative initiative addressing the communication challenges encountered by individuals with hearing or speech impairments through the application of Convolutional Neural Networks (CNNs) for hand sign language recognition. The prevalence of sign language as a vital non-verbal communication form is evident, and this initiative endeavors to enhance its accessibility and efficacy. The existing landscape of hand gesture recognition systems is diverse, emphasizing data acquisition, data environment, and hand gesture representation. GestureSpeak contributes to this ongoing research by implementing a CNN-based model capable of automatically recognizing sign language. The proposed system enables individuals to capture hand gestures through a web camera, enabling the prediction and display of the corresponding sign name. The preparatory stages, including grayscale conversion, dilation, and mask operations, enhance the image quality for efficient CNN training. GestureSpeak stands as a significant stride in leveraging technological advancements, converging computer vision and deep learning, to provide a robust and accessible solution for hand sign language recognition. The incorporation of CNNs ensures efficiency, accuracy, and potential deployment in diverse contexts, including mobile applications and embedded single- board computers. This initiative serves as a promising step towards nurturing inclusive communication for the differently- abled, breaking down barriers through the language of gestures.**

***Keywords*—Hand Sign Language Recognition, Convolutional Neural Networks (CNN), Computer Vision, Deep Learning, Sign Language**

I. INTRODUCTION

Effective communication is fundamental to human interaction, yet individuals with hearing or speech impairments often encounter barriers that lead to frustration and isolation. Sign language functions as a powerful mode of non-verbal com- munication, particularly relying on hand and limb movements. However, the diversity of sign languages, such as American Sign Language (ASL), British Sign Language (BSL), and Indian Sign Language (ISL), coupled with limited awareness and utilization, creates communication disparities.

Sign language recognition stands as a modern research area in Humancentred Computing (HCC), seeking to recognize gestures and translate them into text or vocal. It encompasses a broad variety of positions, orientations, and movements involving the palm, arm, body, and face regions, assigning gestures to alphabets, numbers, and words in various sign languages worldwide. In the domain of hand gesture modelling, two prominent approaches emerge: physical sensor-based modelling and computer vision-based modelling. Physical sensor-based modelling incorporates sensors like flex sensors, accelerometers, and gyroscopes coupled with devices like Raspberry Pi and Arduinos.

However, these devices are complex to configure, calibrate, and are delicate. On the other hand, computer vision- based modelling, leveraging cameras and algorithms, provides a more accessible and efficient alternative, particularly with the advancements in rapid algorithms, parallel computational units, and Datasets like Kaggle [1],[2]

The evolution of computer vision-based applications for processing hand gestures has expanded beyond specialized devices, making it feasible on personal computers. The LeNet architecture introduced in the 1990s paved the way for convolutional architectures, specifically Convolutional Neural Networks (CNNs), which have become a promising technique in pattern recognition and image classification. The extraction of pertinent features for image classification is a critical aspect of image processing. Traditional handcrafted features pose challenges when new classes are introduced. CNNs, with their focus on image-specific features, have emerged as a solution, reducing trainable parameters and enhancing efficiency. Recent advancements, particularly the integration of CNN with Long- Short Term Memory (LSTM), exhibit increased accuracy and efficiency in dynamic hand gesture recognition systems. These systems have been tested on diverse ISL datasets, demon- starting the potential to bridge communication divides for individuals using sign language. [3]

In the landscape of sign language recognition, the critical challenge lies in developing systems that facilitate seamless communication between individuals with hearing or speech impairments and the broader community. The communication barrier posed by the distinct grammatical structures and vocabularies of various sign languages necessitates innovative solutions. Recognizing the significance of hand gestures as a predominant element in sign languages, efforts have been directed toward two primary approaches: physical sensor- based modelling and computer vision-based modelling. While the former relies on intricate sensor configurations and single- board processors, the latter leverages the simplicity and accessibility of cameras and sophisticated algorithms. The advent of computer vision has democratized the field, allowing hand gesture processing applications to function on personal computers. This shift has spurred a renewed wave of research, investigating the potential of convolutional neural networks (CNNs) as a cornerstone for image recognition. The CNN architecture, akin to conventional artificial neural networks but designed particularly for image-related tasks, exhibits prowess in extracting image-specific features. The operations within CNNs, such as pooling and convolution, facilitate generalization, reduce dimensionality, and demand fewer computational resources compared to conventional artificial neural networks (ANNs). This effectiveness has facilitated the accurate classification of both static and dynamic hand gestures, with recent advancements showcasing integrated models that combine CNNs with recurrent layers like Long- Short Term Memory (LSTM), further enhancing accuracy and adaptability in dynamic gesture recognition systems. As the research landscape continues to evolve, the intersection of computer vision and neural networks stands as a promising avenue for breaking down communication barriers associated with sign languages.

In Indian sign language (ISL), the letters of the alphabet are represented by both 2-static and 1- static hands [4] as shown in Figure 1 (b) In Bangla sign language (BSL), which is the common sign language used in Bangladesh, the alphabet is represented by 2-static hands [5] as shown in Figure 1 (c) This type of gesture can be delivered to the recognition system individually to be deciphered by one. It can also be input to the system continuously in the form of a video that will be divided into frames to be recognized by the recognition algorithm [6].

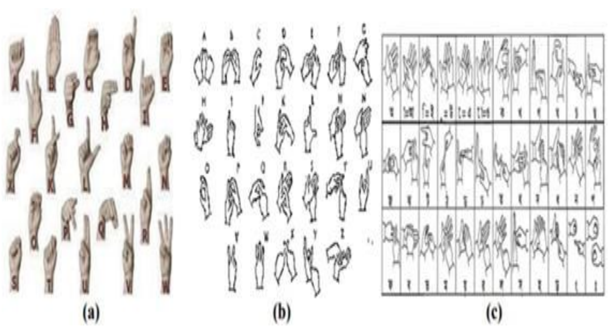


Fig. 1: (a) American-Sign-Language (b) Indian-Sign-Language (c) Bangla-Sign-Language

A. Overview of Sign Language Recognition Approaches

According to the technique of inputting the gesture into the computing system, there are three approaches to sign language recognition (SLR): sensor-based, vision-based and hybrid-based [11] as shown in Figure 2.

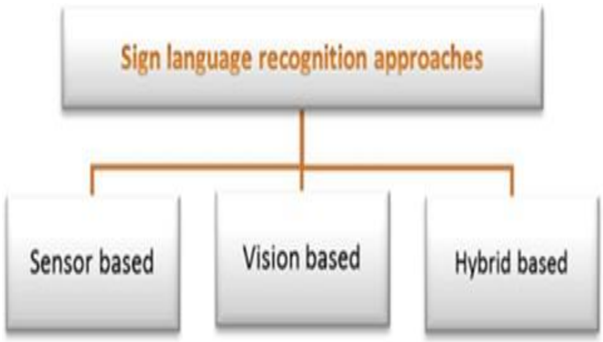


Fig. 2 Sign Language Recognition Approaches

Sensor-Based Strategy: This approach uses sensors and computation electronic circuits to acquire and condition the input data, as shown in Figure (4) in the sensory gloves system. The advantages of this technique include simplicity because no complex data processing is required, flexibility because there are no restrictions on movements such as sitting behind a desk or chair, reliability as hand shape recognition is unaffected by background conditions, and its lightweight as well as mobility that enables the device to be carried easily and comfortably [7]. The sensory glove contains different sensors such as an accelerometer and gyroscope for obtaining the orientation of the hand and fingers and also angle and acceleration information. Some gloves also contain flex sensors that provide the system with finger-bending information [8].

Vision-Based Strategy: This approach depends on the acquisition of images or many frames of one video taken by a camera such as a webcam or a smartphone camera. It is considered flexible, simple and cost-effective as it does not use expensive advanced sensors. Compared to the sensory glove system, this approach does not pose personal safety concerns such as skin damage, burns and the spread of infectious diseases [9]. The vision-based technique was first used in 1988 for recognizing Japanese SL using a normal analogue camera which caused distortion in the images so filters were required to reduce the noise. In this approach, there is a need for image preprocessing and processing to get the feature vector. It also requires an effective image-capturing environment that takes into consideration brightness levels, shading, skin tone variations, etc. However, this approach is not without its disadvantages which are represented by real- time computational delay, the need for high computational power and a low recognition rate [10].

Hybrid Based Strategy: As the name suggests, this technique combines sensors and image-capturing cameras to acquire the input data, such as combining gloves of sensors, leap motion, and Microsoft Kinect. It enables designers to benefit from the advantages of spatial sensors that support the performance of computer vision methods [11]. The leap motion controller LMC in Figure 5 (a) is a sensor sensitive to position and motion information. It contains 3 infrared

LEDs and 2-IR cameras [12] while Microsoft Kinect MSK sensors in Figure 5 (b) are sensitive to depth and skeleton shape. This device provides the RGB information obtained from the captured images [13]. The main difference between smart gloves and LMC, MSK devices is that the gloves are wearable. Using the three aforementioned devices, different combinations can be formed. For example, [16] used camera+ gloves while [14] used webcam+ MSK. [20] used 2-digital cameras+ LMC. In these three contributions, researchers dis- cussed dynamic gestures and their datasets. However, dynamic gestures are out of the scope of this study and only static gestures are considered.

American Sign Language (ASL) uses a system of hand gestures to visually represent letters, numbers, and words. It includes a specific set of 26 hand gestures, known as the American Manual Alphabet, which can be used to spell out English words. These gestures allow individuals to communicate by spelling out words letter by letter. The alphabet includes distinct hand shapes that are arranged to correspond with the letters from A to Z, with 19 unique hand shapes making up the 26 alphabet signs. Some letters share the same hand shape but differ in orientation; for example, the letters ’K’ and ’P’ are formed using the same hand shape but with varied hand positions to signify the specific letters. This approach helps maximize clarity within the limits of manual gestures.

ASL also includes a set of numeric gestures for numbers ranging from 0 to 9. These hand signals allow users to communicate basic numbers and are integral to the language’s versatility. ASL users commonly rely on these numerical gestures in everyday communication, as they are essential for expressing quantities, dates, times, and other numerical information.

However, ASL does not always have direct signs for certain specific words, especially for proper nouns and technical terms. In such cases, the Manual Alphabet is used to spell out names, brand names, or complex terms that lack a standard gesture. This limitation arises because certain specialized or less common words do not have a universally accepted hand signal. Therefore, individuals may spell out terms manually, letter by letter, to convey these words accurately. For instance, names, medical terms, or scientific jargon are often communicated in this way.

Beyond the alphabet and numeric system, ASL contains an extensive collection of gestures to express a wide range of English vocabulary. Thousands of hands and facial gestures exist within ASL to signify various words and phrases. ASL is not limited to a static set of gestures but is a rich and dynamic language with expressions that go beyond the alphabet and numbers. These additional gestures cover diverse words and phrases, making ASL a fully functional language capable of expressing emotions, descriptions, actions, and even complex thoughts.

Facial expressions and body language play a crucial role in ASL, adding nuance to the message. For instance, facial expressions can convey questions, emotions, or intensify the meaning of a gesture, making ASL a multi-dimensional language. This combination of hand gestures with facial and body movements creates a more nuanced and effective way to communicate.

While the alphabet and numbers provide the foundation of ASL, the thousands of additional gestures offer flexibility and depth to the language. ASL’s robust vocabulary covers a vast range of topics, making it a comprehensive language for deaf and hard-of-hearing individuals. In many ways, ASL operates much like any spoken language, with its own grammar, syntax, and expressive capacity. [15]

To summarize, ASL is a complete visual language with a structured alphabet and numeric system that enables effective communication. The American Manual Alphabet comprises 26 gestures for letters, using 19 distinct hand shapes, with some variations in orientation to represent different letters. Numeric gestures from 0 to 9 allow for numerical representation. When ASL lacks a specific gesture for a word, users can spell it out manually, and there are numerous additional signs for common words and phrases. ASL goes beyond the basic alphabet and numbers by incorporating expressive gestures that enhance its ability to convey complex messages. This combination of hand shapes, orientations, and facial expressions makes ASL a versatile and powerful language for visual communication. The set of 26 gesture signs of English Alphabets (A-Z) and 10 Numbers (0-9) are shown in Fig. 3.

Fig. 3: Gesture signs of English Alphabets (A-Z) and 10 Numbers (0-9)

All in all, the intersection of computer vision, neural net- works, and sign language recognition bears immense promise in bridging communication divides. This work, GestureSpeak, aligns with this trend by utilizing convolutional neural networks to empower individuals with hearing or speech impairments. By investigating the effectiveness of CNNs in interpreting hand gestures and contributing to the wider field of gesture recognition, our initiative aims to promote seamless communication and foster inclusivity in diverse linguistic contexts.

II. Related Work

The literature review on hand sign language recognition with convolutional neural networks (CNNs) encompasses a variety of approaches, all aimed at addressing the specific communication challenges faced by individuals with speech or hearing impairments. These findings highlight how important it is to use CNNs—advanced neural network architectures— when creating reliable and efficient hand sign language detection systems. The knowledge gained from these research advances the field of sign language recognition technology.

A number of research, including highlight how crucial CNN architectures are to the recognition of sign language. In these systems, convolutional layers, rectified linear units (ReLU), [16] pooling layers, fully connected layers, and SoftMax output layers are commonly utilized. Notably, identification accuracy has significantly improved as a result of the transition from conventional artificial neural networks (ANN) to deep ANN, as demonstrated in. This improvement demonstrates CNNs' applicability for mobile platform deployment, which is consistent with the expanding market for mobile applications.

One recurring subject in the literature review is the acknowledgment of certain sign languages, such as Indian Sign Language (ISL). In, deep learning architectures based on vision are shown, demonstrating great training accuracy for recognition systems that are not dependent on sign language. Furthermore, highlights the significance of standardized sign language dictionaries for promoting interoperability and cross- border communication by introducing Sign Language Transformers built with PyTorch [17]

The literature addresses issues with sign language recognition, including non-uniform illuminance and extended recognition times. One useful method for increasing model accuracy is the application of skin detection techniques, as covered in. It is recognized, although, that the skin detection procedure can add extra computing overhead, which, though at a higher expense, can be reduced by utilizing quick GPUs.

As demonstrated by, the survey also emphasizes the value of publicly accessible datasets. One of the key drivers of progress in the field of American Sign Language (ASL) research is the need for standardized ASL databases. The availability of these datasets makes it easier to design and assess innovative methods to recognize sign languages.

III. Methodology

Sign language recognition plays a pivotal role in improving communication for individuals with hearing impairments. In this comprehensive methodology, we provide a detailed guide for building a Hand Sign Language Recognition system using 3D Convolutional Neural Networks (CNNs). The methodology is structured into two primary sections: The Proposed Model Architecture and the Comprehensive Workflow. Additionally, we delve into the intricacies of a Proposed Algorithm to demonstrate the practical implementation of the system.

Sign language serves as a fundamental mode of communication for the deaf and hard of hearing community. Recognizing dynamic hand gestures in real-time presents a challenging yet critical task for nurturing effective communication through technology. The proposed methodology seeks to address this challenge by leveraging the strength of 3D CNNs represent a specialized neural network architecture designed to capture both spatial and temporal features inherent in hand gestures. The workflow of the proposed model incorporates several critical stages, each contributing to the accurate recognition of hand gestures.

A. Video Sample Input

The model's initial phase involves gathering video samples that encapsulate a diverse range of hand gestures, effectively mimicking real-world scenarios. These samples are meticulously curated to encompass various contexts, such as communication cues, sign language expressions, and intricate movements commonly encountered in day-to-day interactions.

Upon collecting these video samples, the model undergoes a rigorous preprocessing stage where the data is refined and standardized. This crucial step ensures that the input data is consistent, free from anomalies, and optimized for subsequent analysis and interpretation.

Once the feature extraction phase is complete, the model transitions into training and fine-tuning stages. Through a combination of supervised learning techniques and deep neural networks, the model iteratively refines its understanding of hand gestures, refining its ability to discern subtle variations and nuances across different gestures and contexts.

B. Frame Extraction

In the realm of video processing, the process of extracting frames from video samples and subsequently storing them on disk forms a pivotal stage in the analytical pipeline. This stage involves the meticulous extraction of individual frames from the continuous stream of video data, resulting in a sequence of contiguous images that encapsulate the visual information inherent in the video source. Each frame serves as a snapshot frozen in time, capturing a specific moment from the dynamic progression of the video.

The importance of this extraction process cannot be over- stated, particularly in the context of spatiotemporal analysis. By dissecting the video into its constituent frames, researchers gain access to a wealth of visual data that facilitates in- depth exploration and understanding of the underlying spatiotemporal dynamics. These frames not only provide a detailed view of the spatial arrangement of objects within each frame but also enable the analysis of temporal changes and motion patterns over successive frames.

Furthermore, the sequential storage of frames on disk ensures the preservation of temporal continuity, allowing for seamless playback and analysis of the video content. This organized storage structure facilitates efficient access to in- dividual frames during subsequent processing stages, enabling researchers to apply a diverse range of analytical techniques such as object detection, motion tracking, and activity recog- nition.

C. 3D-CNN Processing

A 3D-CNN network is a sophisticated model specifically designed for analyzing sequences of images. Unlike traditional CNNs [18] that operate on individual frames, a 3D-CNN processes a series of frames as a whole, allowing it to capture the dynamic changes and temporal evolution within the data. This architecture is particularly effective for tasks where understanding motion or changes over time is crucial, such as in gesture recognition, video analysis, or even medical imaging where time-series data plays a significant role.

The key advantage of using a 3D-CNN lies in its ability to extract spatial and temporal features simultaneously. By incorporating the temporal dimension into the convolutional layers, the network can discern patterns that unfold over time, offering a more comprehensive understanding of the input data. This not only enhances the accuracy of gesture recognition systems but also enables the model to generalize better across different temporal variations, making it robust to variations in speed, duration, and timing of gestures.

Moreover, the architecture of a 3D-CNN can be customized and optimized to suit specific applications. Researchers and practitioners can fine-tune parameters, adjust the depth and width of layers, or integrate additional techniques such as attention mechanisms or recurrent connections to further enhance the network's performance. This flexibility makes 3D- CNNs [19] a valuable tool in various domains, providing a powerful framework for processing sequential image data and extracting meaningful insights from dynamic visual inputs.

D. Class Labelling

In the realm of gesture recognition systems, the process of converting neural network outputs into meaningful class labels marks a critical juncture. This transformation not only assigns specific identifiers to recognized gestures but also lays the groundwork for interpreting these gestures within a broader context. By associating class labels with distinct actions or intents, the system gains the ability to generate actionable insights from the continuous flow of data processed by the neural network. This transition from raw output to categorized information bridges the gap between data processing and real- world applications, enabling the system to facilitate intuitive interactions and informed decision-making.

Moreover, the utilization of class labels derived from neural network outputs extends beyond immediate recognition tasks, serving as a foundation for advanced analytics and predictive modeling. The structured nature of class labels enables the aggregation and analysis of gesture data at a higher level, facilitating trend identification, anomaly detection, and predictive insights. This strategic integration of labeled data into broader analytical frameworks empowers organizations and researchers to extract valuable knowledge from gesture-related interactions, driving innovation, and decision-making across diverse domains. As such, the conversion of neural network outputs into actionable class labels not only enhances real- time gesture recognition but also lays the groundwork for leveraging gesture data as a valuable resource for future- oriented applications and insights.

E. Network Architecture

The network architecture plays a pivotal role in the success of the proposed model, especially concerning its ability to handle dynamic hand gestures effectively. This architecture is meticulously crafted to optimize the processing and interpretation of these gestures, taking into account factors such as real-time responsiveness, accuracy, and scalability. By employing advanced algorithms and deep learning techniques, the architecture is capable of capturing the intricate nuances of hand movements, thus enabling the model to make precise and timely predictions.

Moreover, the design of the network architecture reflects a deep understanding of the complexities involved in gesture recognition tasks. It incorporates layers of abstraction and feature extraction mechanisms that enable the model to learn hierarchical representations of hand gestures, progressively refining its understanding with each layer. This hierarchical approach not only enhances the model's accuracy but also contributes to its robustness across varying environmental conditions and user demographics.

F. Input Blocks

It is crucial to delve into the intricate process of concatenating input images sequentially to construct an image sequence. This concatenation entails arranging images in a specific order, typically based on time or another relevant parameter, to form a continuous stream of visual data. The representation of image blocks further elucidates this concept, with each block denoted as B,h,w,c,s, where 'h' and 'w' signify the height and width of the image, 'c' represents the number of channels (such as RGB channels in a color image), and 's' indicates the sequence or temporal dimension.

For our specific experimental setup, we adopt image block dimensions of B128,128,3,20. This notation reveals that each block encompasses an image of 128 pixels in height and 128 pixels in width, with three color channels (RGB) considered. The '20' in the sequence dimension signifies that we are working with a sequence of 20 images concatenated together. This structured approach to image representation and sequence formation forms the backbone of our research methodology, enabling us to analyze and derive insights from dynamic visual data effectively.

G. 3D Convolution

A 3D convolutional network excels in capturing and analyzing spatiotemporal features, making it a powerful tool for tasks involving volumetric data such as video analysis, medical imaging, and 3D object recognition. The utilization of convolution with kernels sized at (3 x 3 x 3) enables the network to extract intricate patterns and relationships within the data, leading to more accurate and robust feature representations. This approach is particularly effective in scenarios where temporal dynamics play a crucial role, as the network can effectively learn temporal dependencies across consecutive frames or slices of the input volume. [20]

Tran's visualization of the disparity between 2D and 3D convolution provides a clear understanding of the advantages offered by the latter. By showcasing how the selected (3 x 3 x 3) kernel surpasses previously employed kernels in terms of capturing spatial and temporal information simultaneously, Tran's work underscores the importance of leveraging 3D convolutional networks for tasks that demand comprehensive spatiotemporal analysis. This advancement not only enhances the network's ability to discern complex patterns within volumetric data but also contributes to improving the overall performance and efficiency of various applications reliant on deep learning methodologies. [21]

H. Max Pooling 3D

Max pooling plays a crucial role in deep learning architectures by enabling efficient feature extraction and reducing the complexity of subsequent layers. After the convolutional layer processes the input data, max pooling is applied to down sample the feature maps, capturing the most salient information while discarding redundant details. This process aids in improving the model's robustness to variations in input, making it more capable of generalizing to unseen data. By using a (2 x 2 x 2) max pooling window, we ensure that each pooling operation considers a local region of the feature map, capturing the most significant features while reducing the spatial dimensions. This reduction not only speeds up computation but also helps prevent overfitting by promoting the extraction of essential features.

Furthermore, the choice of max pooling dimensions can significantly impact the model's performance and efficiency. While larger pooling windows can lead to more aggressive down sampling and reduced computational costs, they may also risk losing fine-grained details. Conversely, smaller pooling windows preserve more spatial information but may increase computational overhead. Therefore, the selection of max pooling dimensions should be carefully considered based on the specific requirements of the task and the desired balance between feature retention and computational efficiency.

I. Dense(MLP)

In addition to the utilization of a Multi-Layer Perceptron (MLP) in the later stages of the network, it's crucial to con- sider the training methodology employed for optimizing the MLP's performance. Typically, the training process involves iterative adjustments to the weights of the MLP nodes through backpropagation, where the error between the predicted output and the actual label is minimized using optimization algorithms like stochastic gradient descent (SGD) or its variants. The choice of activation functions within the MLP, such as ReLU (Rectified Linear Unit) or sigmoid, also significantly impacts the network's learning capabilities and convergence speed. Furthermore, regularization techniques like dropout or L2 regularization may be applied to prevent overfitting and enhance the model's generalization ability. [22]

Moreover, the integration of batch normalization layers within the MLP architecture can further stabilize and accelerate the training process by normalizing the inputs to each layer. This normalization helps in mitigating issues related to internal covariate shift and improves the overall robustness of the network. Additionally, the incorporation of residual connections or skip connections within the MLP can facilitate the flow of gradients during training, enabling more efficient learning in deeper networks. These design considerations, along with careful hyper parameter tuning and cross-validation techniques, play a pivotal role in achieving optimal performance and generalization in deep learning models utilized for image sequence classification tasks.

J. Dropout & Regularization

To mitigate overfitting, a meticulous approach has been adopted, encompassing several key strategies. First and fore- most, a dropout rate of 30% has been meticulously implemented within the model architecture. This strategic dropout mechanism ensures that during training, random neurons are temporarily excluded, preventing the network from overly relying on specific pathways and enhancing its ability to generalize to unseen data. Moreover, the incorporation of L1 regularization further bolsters the model's robustness by imposing a penalty on the absolute magnitude of the weights, discouraging excessive complexity and promoting feature selection. In tandem with these techniques, a decay rate of 1 × exp6 has been applied, serving as a crucial component in optimizing the model's learning dynamics. This decay rate, often referred to as weight decay or L2 regularization, acts as a form of regularization by gradually reducing the magnitude of the weights during training. By imposing this gradual reduction, the model is encouraged to prioritize essential features while mitigating the risk of overfitting to noise in the training data. Collectively, these strategic measures not only address overfitting but also contribute significantly to the model's overall performance, ensuring a balance between complexity and generalization prowess. [23]

K. Contract Slider for Accuracy Enhancement

The Contract Slider methodology introduces a dynamic approach to fine-tuning the contrast levels within the environment where hand gestures are captured. This innovative technique aims to optimize image quality and enhance feature visibility, ultimately leading to improved accuracy in hand sign language recognition systems.

The rationale behind the Contract Slider stems from the recognition that varying lighting conditions and environmental factors can significantly impact the clarity and distinctiveness of hand gestures captured in video samples. By adjusting the contrast levels, researchers can manipulate the visual characteristics of the input data, thereby influencing the model's ability to extract meaningful features and make accurate pre- dictions.

The Contract Slider operates on a continuum, allowing for incremental adjustments to the contrast settings based on real-time feedback and performance metrics. This iterative approach empowers researchers to experiment with different levels of contrast and observe how these adjustments impact the model's accuracy and robustness across diverse scenarios. Implementation of the Contract Slider involves the following key steps:

a. *Contrast Level Initialization:* Begin by establishing a baseline contrast level within the environment where video samples are recorded. This initial setting serves as a reference point for subsequent adjustments.

b. *Dynamic Contrast Adjustment:* During the data preprocessing stage, integrate a mechanism for dynamically adjusting the contrast levels based on predefined criteria. This criterion may include factors such as ambient lighting conditions, background noise, and the complexity of hand gestures being captured.

c. *Real-time Feedback Loop:* Implement a real-time feed- back loop that evaluates the model's performance under varying contrast settings. This feedback loop can be facilitated through metrics such as accuracy, precision, recall, and F1- score, providing a insights into the impact of contrast adjustments on recognition outcomes.

d. *Iterative Optimization:* Utilize an iterative optimization approach where the Contract Slider systematically explores different contrast levels, fine-tuning the settings to maximize accuracy and minimize error rates. This iterative process leverages machine learning algorithms to adaptively adjust contrast settings based on observed performance trends.

e. *Validation and Validation:* Validate the effectiveness of the Contract Slider methodology through rigorous validation procedures, including cross-validation, holdout validation, and performance benchmarking against baseline models. This validation phase ensures the reliability and generalizability of the contrast optimization strategy across diverse datasets and use cases.

By integrating the Contract Slider methodology into the existing workflow, researchers can unlock new avenues for enhancing the accuracy and robustness of hand sign language recognition systems. This adaptive approach to contrast optimization aligns with the overarching goal of improving communication accessibility for individuals with hearing impairments, fostering inclusivity and empowerment through technology-driven solutions. [24]

L. SoftMax

The SoftMax layer plays a pivotal role in the overall functioning of the neural network model. It serves as the bridge between the complex computations happening within the network and the comprehensible output that users and researchers can interpret. By normalizing the output into a probability distribution across the 20 classes, it not only provides a clear understanding of the model's predictions but also enables further analysis and decision-making based on these probabilities. This phase essentially transforms raw numerical outputs into meaningful and actionable insights, making the model's output more interpretable and useful in various applications such as classification tasks, sentiment analysis, and recommendation systems. [25]

Moreover, the SoftMax layer's function extends beyond just providing probabilities; it also aids in training the model effectively. Through techniques like cross-entropy loss, the Soft- Max layer guides the learning process by penalizing incorrect predictions and reinforcing correct ones. This iterative process of adjusting the model's weights based on the feedback from the SoftMax layer contributes significantly to the model's ability to generalize well to unseen data. Therefore, the inclusion of the SoftMax layer in the final stage of the neural network architecture is not only essential for interpretability but also for optimizing the model's performance and enhancing its predictive capabilities.

IV. IMPLEMENTATION

A. Basic Idea

American Sign Language (ASL) relies on specific hand orientations, with each orientation representing a unique letter from the English alphabet. To facilitate effective communication between individuals with hearing impairments, it is essential that a model accurately identifies these hand gestures. Occasionally, a unique hand gesture might represent a phrase or word not originally included in the training dataset.

Traditional methods for sign language recognition often involve contour detection in raw images, but this method can sometimes yield inconsistent results.

To overcome these challenges, a Convolutional Neural Net- work (CNN) can be developed. Applying consistent image transformation techniques to both training and test images allows the model to interpret them under similar conditions, enhancing accuracy and consistency in recognition.

This approach involves the following four key steps:

1) **Data Set Creation**: Building a diverse and comprehensive dataset of ASL hand gesture images.
2) **CNN Construction**: Designing and implementing a CNN that can effectively process and classify these images.
3) **Training**: Training the CNN on the dataset to ensure it can accurately identify ASL gestures.
4) **Sign Language Conversion**: Converting recognized hand gestures into corresponding ASL letters or phrases for effective communication.

B. Data Set Creation

In this approach, a centralized table is maintained to organize and track each gesture by assigning it a unique identifier and label. This table stores essential information such as the gesture ID and its associated label. For instance, in American Sign Language (ASL), a closed fist represents the letter"A." This centralized table will include similar entries for each gesture, including specific phrases; for example, ”best of luck” is represented by a fist with the thumb pointing up. This flexible approach allows for the addition of customized gestures as needed.

To develop a robust dataset, around 1200 images are required for each gesture. These images are not standard RGB but rather threshold images that enhance clarity for gesture recognition. Each image undergoes a transformation process,

where a threshold histogram is calculated based on the hand’s unique characteristics. This histogram graphically represents the distribution of an image’s intensity, helping to establish a baseline for identifying hand orientations.

To obtain the histogram, the RGB image is first converted into an HSV (Hue, Saturation, Value) format, which is then processed using the cv2.calcHist() function from OpenCV. This function generates the histogram with specific parameters, which is then normalized and saved as a reference for building the dataset.

This normalized histogram acts as a consistent guide for detecting hand orientation during dataset creation. Each gesture image is compared against this reference, producing a threshold image as illustrated in Figure 2. By repeating this process for every gesture, a comprehensive dataset is created. Importantly, the histogram is only defined once, as it serves as a uniform reference across all gestures.

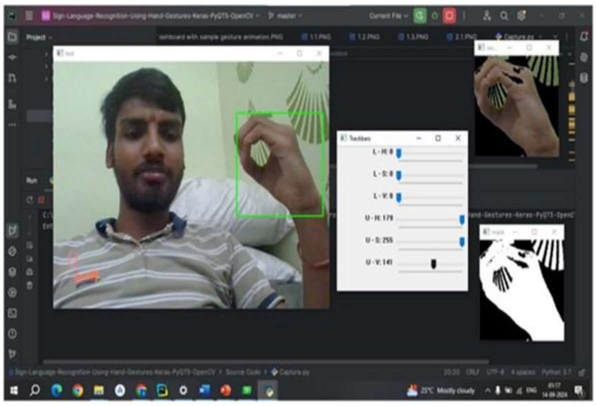


Fig. 4: Threshold Image

C. Training

With 1200 images per gesture, the dataset for 44 different gestures totals 52,800 images, used for both training and testing purposes. Of these images, approximately five-sixths are allocated for training, while the remaining portion is reserved for testing and validating the model’s performance. During training, the model learns to recognize gestures based on their unique gesture IDs, which are later mapped to gesture labels during prediction.

If additional gesture images are introduced, certain layers or parameters of the model may need to be adjusted to accommodate the new data. Following these modifications, the model will undergo retraining to integrate the new images effectively. For optimal performance, each image is resized to $50 \times 50 \times 1$ before being input into the model. The model is trained using backpropagation, and it utilizes the SGD (Stochastic Gradient Descent) optimizer from Keras, with accuracy as a key performance metric.

D. Sign Language Conversion

Once the model is trained, it is ready to convert real-time hand gestures into American Sign Language (ASL) letters or phrases. Real-time input from a webcam captures the user’s hand gesture within a designated green box (as shown in Figure 4). Simultaneously, a thresholded version of the live video stream is displayed to the left, which aids in gesture prediction. This threshold image is derived using the same pre-processing steps that were applied during dataset creation. The processed image is then fed into the model, which predicts the gesture in real-time, displaying results in the terminal (as illustrated in Figure 5). This prediction code operates in a continuous loop, enabling the system to offer real-time gesture recognition that can support communication for individuals with hearing impairments.

Fig. 5: Sign Detection Dashboard

Dynamic Layout: Implement a responsive and user-friendly layout with components like navigation bars, tabs, or drop- downs for better organization of content.

Interactive Elements: Use components such as buttons, sliders, or checkboxes to allow users to interact with the dashboard.

Color Scheme & Theming: Apply a consistent color scheme that aligns with your branding. Use light/dark mode options to improve accessibility.

V. Architecture Diagram

Phase 1- Data Collection Phase: In this initial phase, the model is prepared to recognize American Sign Language (ASL) gestures by collecting a substantial set of images for each sign. These images are fed to the model in sequence, allowing it to learn the specific visual characteristics associated with each symbol. This phase lays the foundation for the model’s learning, as the variety and quality of images collected directly influence the model’s ability to generalize and

accuracy. Testing, in turn, involves evaluating the model on a separate subset of images to measure its effectiveness. During this phase, the accuracy of the model is continuously monitored, allowing adjustments and refinements to be made as necessary.

Phase 2- Training and Testing Phase: In this phase, the collected images are used to train the model to recognize gestures and associate them with their corresponding out- puts. Training involves feeding the images into the model and adjusting its internal parameters to maximize recognition

Phase 3- Recognition of output: In the final phase, the trained model is applied in a real-world setting to interpret gestures. An input image, such as a live hand gesture captured via webcam, is provided to the model. The model processes the image and compares it with the patterns it has learned from the training data to determine the most likely gesture. This phase effectively transforms visual gestures into text, making it an invaluable tool for bridging communication gaps for individuals with hearing impairments.

VI. FLOWCHARTS

Each one of the phases in the architecture are further explained and sketched graphically in the form of flowcharts. These flowcharts further explain how each of the phases described in the architecture diagram functions. This helps us in deeply understanding its core features, usability and its relation with the other respective phases present.

1) Start
2) Initialize Camera
   - Set up the camera for real-time video capture.
3) Detect Hand Gestures
   - Use a machine learning model (e.g., CNN or a hand gesture recognition model) to detect the hand.
4) Match Gesture to ASL Character
   - Compare detected hand gesture with predefined ASL characters.
5) If match is found, store the corresponding ASL character.Store Character
   - Append the detected character to a text file.
6) Detect Custom Gestures (Optional)
   - Allow the user to build and train custom gestures for new characters.
7) Repeat Detection
   - Continuously process video frames until the user exits.
8) End

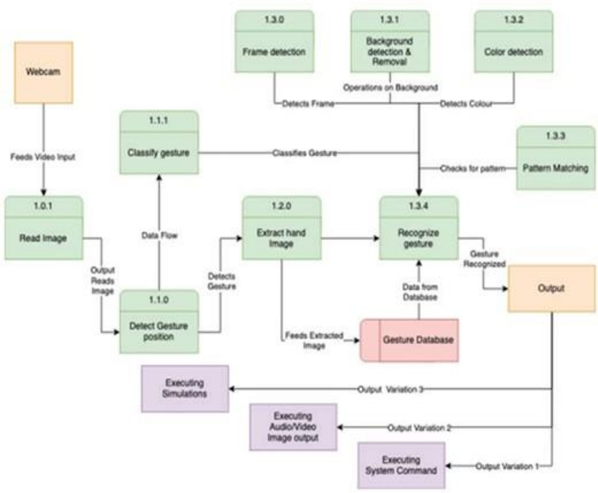


Fig. 6: Flow Diagram

recognize gestures accurately. By gathering a wide range of samples for each symbol, the model can be fine-tuned to recognize gestures effectively across different scenarios and environments, improving its adaptability for future use.

VII. FUTURE SCOPE

The application can be enhanced to provide real-time translation of ASL signs into text and voice output, enabling effective communication between sign language users and non-signers, while also supporting multiple sign languages such as British Sign Language (BSL) and Indian Sign Language (ISL) for greater inclusivity. Gesture detection can be improved by allowing users to teach personalized gestures

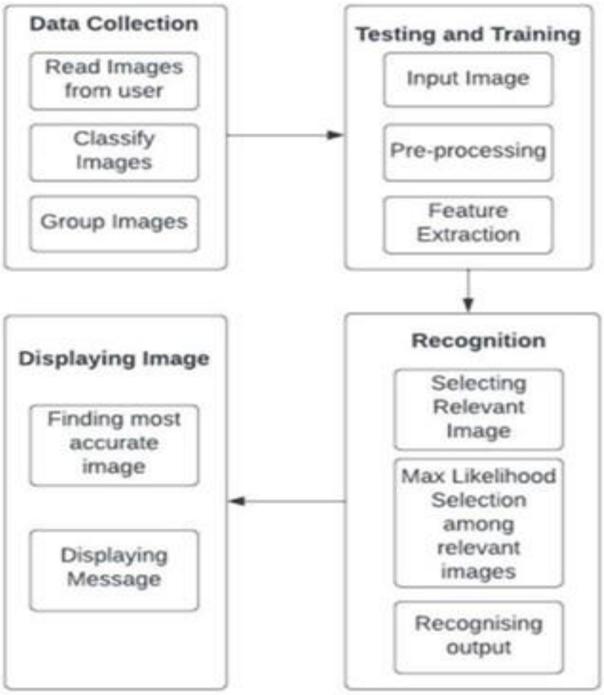


Fig. 7: Complete Flowchart of Working of ASL Using Sign Language

and by recognizing emotional context through facial expressions and gesture intensity. The use of advanced AI and deep learning models can significantly improve recognition accuracy in diverse environments, while adapting to individual signing styles over time through machine learning. Future developments may include multi-hand and 3D gesture recognition to accurately interpret complex signs involving depth and hand positioning. Augmented reality features can offer real-time visual feedback and virtual tutoring to help users learn and refine ASL gestures. Expanding the system across mobile, web, and wearable platforms will improve accessibility and enable hands-free interaction. Additional features such as voice commands and text recognition from images can make the system more versatile in everyday use. Community collaboration can be encouraged by allowing users to contribute new gestures and communicate socially using ASL with translation support. Integrating speech-to-text functionality and haptic feedback can further assist users with speech or visual impairments. Finally, partnerships with educational institutions and healthcare organizations can promote the use of the system as a teaching tool and as a solution to communication barriers in medical environments.

VIII. CONCLUSION

In the development of this research, we found that the most effective method for Hand Gesture Recognition involves using a Convolutional Neural Network (CNN) as the core model. Initially, we trained the CNN on a well-established and standardized dataset to ensure the model could accurately identify basic gestures. Once the model was trained, we focused on fine-tuning it to meet the specific requirements and characteristics of our application.

The results were promising, as this approach led to a notable increase in the accuracy of gesture recognition.

The fine-tuning and optimizations not only made the model more adaptable to specific use cases but also demonstrated that combining CNNs with advanced methods could significantly enhance the reliability and effectiveness of Hand Gesture Recognition systems.

REFERENCES

[1] Ma, Y., Xu, T., & Kim, K. (2022). *Two-Stream Mixed Convolutional Neural Network for American Sign Language Recognition*. Sensors, 22(16), 5959.

[2] Multiple Authors. (2023). *Deep Learning Techniques for Sign Language Recognition*. Journal of Computer Vision and Image Processing.

[3] Khan, S., & Patel, A. (2023). *Advancements in Convolutional Neural Networks for Gesture Recognition*. International Journal of Artificial Intelligence, 18(3), 45-60.

[4] Ghotkar, "Study of Vision Based Hand Gesture Recognition Using," vol. 7, no. 1, pp. 96–115, 2014.

[5] M. Sanzidul Islam, S. Sultana Sharmin Mousumi, N. A. Jessan, A. Shahariar Azad Rabby, and S. Akhter Hossain, "Ishara-Lipi: The First Complete Multipurpose Open Access Dataset of Isolated Characters for Bangla Sign Language," 2018 Int. Conf. Bangla Speech Lang. Process. ICBSLP 2018, no. September, pp. 1–4, 2018, doi: 10.1109/ICB- SLP.2018.8554466.

[6] R. S. Sabeenian, S. Sai Bharathwaj, and M. Mohamed Aadhil, "Sign language recognition using deep learning and computer vision," J. Adv. Res. Dyn. Control Syst., vol. 12, no. 5 Special Issue, pp. 964–968, 2020, doi: 10.5373/JARDCS/V12SP5/20201842

[7] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. Bin Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," Sensors (Switzerland), vol. 18, no. 7, 2018, doi: 10.3390/s18072208.4

[8] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," Int. J. Mach. Learn. Cybern., vol. 10, no. 1, pp. 131–153, 2019, doi: 10.1007/s13042-017-0705-5.

[9] M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," J. Imaging, vol. 6, no. 8, 2020, doi: 10.3390/JIMAGING6080073.

[10] R. Elakkiya, "Machine learning based sign language recognition: a review and its research frontier," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020, doi: 10.1007/s12652-020-02396-y.

[11] R. Van Culver, "A hybrid sign language recognition system," Proc.Int. Symp. Wearable Comput. ISWC, pp. 30–33, 2004, doi: 10.1109/iswc.2004.2.

[12] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign lan- guage recognition using the leap motion controller," IEEE Int. Symp. Ind. Electron., no. November 2015, pp. 960–965, 2014, doi: 10.1109/ISIE.2014.6864742.

[13] T. Raghuveera, R. Deepthi, R. Mangalashri, and R. Akshaya, "A depth- based Indian Sign Language recognition using Microsoft Kinect," Sad- hana - Acad. Proc. Eng. Sci., vol. 45, no. 1, 2020, doi: 10.1007/s12046- 019-1250-6.

[14] J. L. Raheja, A. Mishra, and A. Chaudhary, "Indian sign language recognition using SVM," Pattern Recognit. Image Anal., vol. 26, no. 2, pp. 434–441, 2016, doi: 10.1134/S1054661816020164.

[15] *Gesture Recognition and the Use of Facial and Body Movements in ASL.* (2023). International Journal of Linguistics and Visual Communication.

[16] Zhao, L., et al. (2022). *Real-Time Sign Language Interpretation via Con- volutional Neural Networks.* Proceedings of the International Conference on AI, 12(4), 155-162.

[17] Chen, Y., & Li, X. (2022). *Improving Sign Language Recognition with Skin Detection Techniques*. Journal of Machine Vision, 55(6), 105-118.Simonyan, K., & Zisserman, A. (2014). 3D Convolutional Networks for Spatiotemporal Feature Learning. *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 27(1), 694-702.

[18] Tran, D., Wang, H., & Torresani, L. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks. *IEEE International Confer- ence on Computer Vision (ICCV)*, 2015, 4489-4497.

[19] Simonyan, K., & Zisserman, A. (2014). Two-Stream Convolutional Net- works for Action Recognition in Videos. *Neural Information Processing Systems (NIPS)*, 27(1), 568-576.

[20] Karpathy, A., Toderici, G., Shetty, S., et al. (2014). Large-Scale Video Classification with Convolutional Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, 1725-1732.

[21] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. *MIT Press.*

[22] Srivastava, N., Hinton, G., Krizhevsky, A., et al. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1), 1929-1958.

[23] Hinton, G. E., et al. (2012). Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. *arXiv preprint arXiv:1207.0580*.

[24] Bishop, C. M. (2006). Pattern Recognition and Machine Learning.
Springer Science & Business Media.

[25] Chollet, F., & others. (2017). Keras: A deep learning framework. https:
//keras.io/.

[26] Abadi, M., Agarwal, A., Barham, P., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 265-283. https://www.usenix.org/ conference/osdi16/technical-sessions/presentation/abadi.

[27] OpenCV Contributors. (2024). OpenCV: Open Source Computer Vision Library. https://opencv.org/.

[28] Kaushik, Anupama, Himanshu Gupta, and Digvijay Singh Latwal. "Impact of feature selection and engineering in the classification of handwritten text." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016.

[29] Malik, Sonika & Sagwan, Chetan & Himanshu, & Chava, Jahnavi & Muskan,. (2022). Smart Hand Sign Recognition using Deep Convolution Neural Network. GIS-Zeitschrift fü Geoinformatik. 10. 510-519.

Modified Laguerre Păltănea Operators for Engineering-Oriented Piecewise Smooth Functions Approximation

Man Singh Beniwal[#], Anjali[#]

[#]Department of Applied Science, Maharaja Surajmal Institute of Technology
C-4, Janakpuri, New Delhi-110058

[1]man_s_2005@yahoo.co.in

[2]anjaligupta.2604037@gmail.com

***Abstract*— The present research considers a special function, viz. Laguerre polynomial, which is a classical orthogonal polynomial in the literature. The Laguerre-Păltănea operators are based on the modified Laguerre polynomial that aids in approximating integrable functions. This article aims to study the convergence rate of the operator for a class of functions on the non-negative real axis having derivative of bounded variation. In the end of the article, applications of these novel operators in several engineering problems have been discussed.**

***Keywords*— Laguerre polynomial; Păltănea operators; Application in Engineering problems; Functions with derivative of bounded variation**

I. INTRODUCTION

Approximation theory is a systematic theoretical study of techniques that employ numerical approaches to solve various problems of mathematical analysis. Approximation theory is widely used in machine learning, as several machine learning algorithms rely on approximating complex functions to establish a relationship between input data and output labels. There are various engineering problems that have unbounded domains, such as fluid dynamics, signal processing, heat transfer, and control theory. The Laguerre Păltănea operators can be proved very useful to address these problems. These operators work as a bridge between practical modeling requirements and theoretical approximation concepts. To solve differential equations, to optimize system response, and to enhance computational algorithms, these operators proved to be an interesting mathematical foundation. Thus, they highlight the increasing importance of this synthesis of pure mathematics and applied mathematics in modern engineering science.

Laguerre polynomials are the major classical orthogonal polynomials existing in literature. They have applications in a variety of mathematical disciplines. The study of the convergence rate of the modified form of Laguerre-Păltănea operators for a class of functions on the non-negative real axis whose derivatives are of bounded variation is a significant direction in this context. The main reason for its importance lies in the fact that, in various engineering problems where the physical quantities have piecewise smooth behaviour, for example, in reconstructing signals, in stress-strain analysis, and in thermal conduction processes, this functional class is of great importance. To establish the desired result, we shall start by considering a linear positive operator having a basis depending on a special function, viz, Laguerre polynomial, which is

$r \in [0,\infty)$ $r \in [0,\infty)$

defined by Sucu et al [3] for as:

$$(W_n^{\alpha} h)(r) = \sum_{k=0}^{\infty} l_{n,k}^{\alpha}(r) \cdot h\left(\frac{k}{n}\right),$$

where

$$l_{n,k}^{\alpha}(r) := e^{\frac{-nr}{2}} \cdot 2^{-\alpha-k-1} \cdot L_k^{\alpha}\left(\frac{-nr}{2}\right),$$

with

$$\alpha > -1, n \in \mathbb{N}, h: [0,\infty) \to \mathbb{R}$$
$$\alpha > -1, n \in \mathbb{N}, h: [0,\infty) \to \mathbb{R}$$

is a continuous function and $L_k^{\alpha}(-r)$ are the modified

$$L_k^{\alpha}(-r) := \frac{(\alpha+1)_k}{k!}\, {}_1F_1(-k;\alpha+1;-r) = \binom{k+\alpha}{\alpha}\, {}_1F_1(-k;\alpha+1;-r).$$

Laguerre polynomials defined by means of confluent hypergeometric series by

We denote $\exp_A(y) = e^{Ay}$.
Remark 1 Taking into account the following:

$$\sum_{k=0}^{\infty} y^k L_k^{\alpha}(r) = e^{\frac{-ry}{1-y}} \cdot \frac{1}{(1-y)^{\alpha+1}},$$

we obtain

$$(W_n^{\alpha}\exp_A)(r) = \exp\left(\frac{nr\left(e^{\frac{A}{n}}-1\right)}{2-e^{\frac{A}{n}}}\right) \cdot \frac{1}{\left(2-e^{\frac{A}{n}}\right)^{1+\alpha}}.$$

Consider the Laguerre-Păltănea operators [2] for $\rho > 0, \alpha > -1$ and for $r \in [0,\infty)$ given by:

$$(L_{n,\rho}^{\alpha} h)(r) = \sum_{k=0}^{\infty}$$

where

$$A_{n,k}^{\rho}(h) = \begin{cases} \left\langle \dfrac{n\rho}{\Gamma(k\rho)} \cdot e^{-n\rho y} \cdot (n\rho y)^{k\rho-1}, h \right\rangle & 1 \le k < \infty \\ h(0), & k = 0, \end{cases}$$

with the inner product provided by
$\langle h, g \rangle = \int_0^{\infty} h(q) g(q) dq$ and $h: [0,\infty) \to$
$h: [0,\infty) \to \mathbb{R}$ is an integrable function for which the above integral and series converge. We can also consider the integral representation of the operator (1) given by:

$$(L_{n,\rho}^{\alpha} h)(r) = \int_0^{\infty} K_{n,r}^{\rho,\alpha}(y) \cdot h(y) dy$$

where

$$K_{n,r}^{\rho,\alpha}(y) = \sum_{k=1}^{\infty} l_{n,k}^{\alpha}(r) \cdot s_{n,k}^{\rho}(y) + l_{n,0}^{\alpha}(r) \cdot \delta(y) \qquad (2)$$

$$s_{n,k}^{\rho}(y) = \frac{(n\rho)^{k\rho}}{\Gamma(k\rho)} \cdot e^{-n\rho y} \cdot y^{k\rho-1} \qquad (2)$$

and $\delta(y)$ is the Dirac delta function; the so-called inverse Laplace transform of 1.

Auxiliary Results

Proposition 1 The function that generates the moments for the operator $L_{n,\rho}^{\alpha}$ is given by

$$(L_{n,\rho}^{\alpha} \exp_A)(r) = \exp\left[-nr\left(\frac{(n\rho-A)^{\rho}-(n\rho)^{\rho}}{2(n\rho-A)^{\rho}-(n\rho)^{\rho}}\right)\right] \cdot \left[\frac{(n\rho-A)^{\rho}}{2(n\rho-A)^{\rho}-(n\rho)^{\rho}}\right]^{1+\alpha}$$

$$(L_{n,\rho}^{\alpha} \exp_A)(r) = \exp\left[-nr\left(\frac{(n\rho-A)^{\rho}-(n\rho)^{\rho}}{2(n\rho-A)^{\rho}-(n\rho)^{\rho}}\right)\right] \cdot \left[\frac{(n\rho-A)^{\rho}}{2(n\rho-A)^{\rho}-(n\rho)^{\rho}}\right]^{1+\alpha}$$

.

Proof Consider

$$\begin{aligned}(L_{n,\rho}^{\alpha} \exp_A)(r) &= \sum_{k=0}^{\infty} l_{n,k}^{\alpha}(r) \cdot A_{n,k}^{\rho}(\exp_A) \\ &= \sum_{k=1}^{\infty} l_{n,k}^{\alpha}(r) \frac{(n\rho)^{k\rho}}{\Gamma(k\rho)} \int_0^{\infty} e^{-(n\rho-A)y} \cdot y^{k\rho-1} dy + l_{n,0}^{\alpha}(r) \\ &= \sum_{k=0}^{\infty} l_{n,k}^{\alpha}(r) \left(\frac{n\rho}{n\rho - A}\right)^{k\rho}\end{aligned}$$

Following the modified generating function [3, eq. (3.3)] for Generalized Laguerre polynomials given by

$$\sum_{k=0}^{\infty} L_k^{\alpha}(-r/2) y^k = \exp\left(\frac{ry}{2(1-y)}\right) \cdot \frac{1}{(1-y)^{\alpha+1}}, |y| < 1$$

we have

$$(L_{n,\rho}^{\alpha} \exp_A)(r) = e^{-\frac{nr}{2}} \cdot \exp\left(\frac{nr(n\rho)^{\rho}}{2(2(n\rho-A)^{\rho}-(n\rho)^{\rho})}\right) \cdot \left[\frac{(n\rho-A)^{\rho}}{2(n\rho-A)^{\rho}-(n\rho)^{\rho}}\right]^{1+\alpha}$$

$$(L_{n,\rho}^{\alpha} \exp_A)(r) = e^{-\frac{nr}{2}} \cdot \exp\left(\frac{nr(n\rho)^{\rho}}{2(2(n\rho-A)^{\rho}-(n\rho)^{\rho})}\right) \cdot \left[\frac{(n\rho-A)^{\rho}}{2(n\rho-A)^{\rho}-(n\rho)^{\rho}}\right]^{1+\alpha}$$

.
On simplifying further, we get the required result.

Lemma 1 Using Proposition (1), we conclude that for certain constants b_i and
$e_i(r) = r^i, i = 0,1,2,\dots, b_i \neq 0$, we have

$$\left(L_{n,\rho}^{\alpha} \sum_{i \ge 0} b_i e_i\right)(r) = b_0 + b_1\left[r + \frac{1+\alpha}{n}\right] + b_2\left[r^2 + \frac{r}{\rho n^2}(2\alpha n\rho + 5n\rho + n) + \frac{1}{\rho n^2}(\alpha^2\rho + 4\alpha\rho + 3\rho + \alpha + 1)\right] + \cdots$$

Further, for some constants $c_i, i = 0,1,2,\dots, c_i \neq 0$, we have

$$\left(L_{n,\rho}^{\alpha} \sum_{i \ge 0} c_i (e_1 - re_0)^i\right)(x) = c_0 + c_1\left[\frac{1+\alpha}{n}\right] + c_2\left[\frac{r}{\rho n}(3\rho + 1) + \frac{1}{\rho n^2}(\alpha^2\rho + 4\alpha\rho + 3\rho + \alpha + 1)\right] + \cdots$$

Denote

$$\psi_{n,\rho}^{\alpha}(r) := \frac{r}{\rho n}(3\rho + 1) + \frac{1}{\rho n^2}(\alpha^2\rho + 4\alpha\rho + 3\rho + \alpha + 1)$$

One intriguing field of study in approximation theory lies in figuring out the convergence rate of the operator for a class of functions whose derivatives have bounded variation (for example, see [1]). Consider $BV_D(\mathbb{R}_0^+)$ $BV_D(\mathbb{R}_0^+)$ as the class of functions that are defined on the non-negative real axis and have derivative of bounded variation on each finite subinterval of $\mathbb{R}_0^+ \mathbb{R}_0^+$. Keeping in mind the unboundedness of the domain of h h, we shall consider the space $BV_D^{r^2}(\mathbb{R}_0^+) BV_D^{r^2}(\mathbb{R}_0^+)$ defined as follows:

1. The class of functions hh in $BV_D(\mathbb{R}_0^+)$ $BV_D(\mathbb{R}_0^+)$ that are absolutely continuous,
2. $\frac{|h(r)|}{1+r^2} \le C_h \frac{|h(r)|}{1+r^2} \le C_h$ for some absolute constant $C_h C_h$.

Through the use of Calculus Fundamental Theorem, we can see that the function $h \in BV_D^{r^2}(\mathbb{R}_0^+)$ $h \in BV_D^{r^2}(\mathbb{R}_0^+)$ has a form given by:

$$h(r) = \int_0^r g(y)dy + h(0)$$

where on each finite subinterval of the non-negative real axis, gg is a function of bounded variation. Here, we shall study the convergence rate of $L_{n,\rho}^{\alpha}$ in the space $BV_D^{r^2}(\mathbb{R}_0^+)$.

Lemma 2 Let $r \in \mathbb{R}_0^+$ and $K_{n,r}^{\rho,\alpha} K_{n,r}^{\rho,\alpha}$ be the kernel defined by (2). Then the following holds:

$$\lambda_n^{\rho,\alpha}(r,u) := \int_0^u K_{n,r}^{\rho,\alpha}(y)dy \leq \frac{\psi_{n,\rho}^{\alpha}(r)}{(r-u)^2}$$

$$1 - \lambda_n^{\rho,\alpha}(r,v) := \int_v^{\infty} K_{n,r}^{\rho,\alpha}(y)dy \leq \frac{\psi_{n,\rho}^{\alpha}(r)}{(v-r)^2},$$

for $0 \leq u < r$ and $r < v < \infty$ respectively. Proof follows using some calculations. We omit the details.

According to Jordan's theorem, a function is of bounded variation iff it can be written as the difference of two functions that are increasing in nature. Also, a function gg, which is univariate and is of bounded variation, has at the most a countable number of first-kind discontinuities. Thus, we can write

$$g(r) = \lim_{\delta \to 0^+} \frac{g(r+\delta) + g(r-\delta)}{2}$$

Assume that for every function gg of bounded variation on $\mathbb{R}_0^+$ and $r \in \mathbb{R}_0^+$, the corresponding auxiliary function $g_r g_r$ is presented by

$$g_r(s) = \begin{cases} g(s) - g(r^-), & 0 \leq s < r \\ 0, & s = r \\ g(s) - g(r^+), & r < s < \infty \end{cases}$$

Further, denote the total variation of gg on $[a,b]$ $[a,b]$ by $TV_a^b(g) TV_a^b(g)$.

Main Result

Theorem 1 Let $h \in BV_D^{r^2}(\mathbb{R}_0^+) h \in BV_D^{r^2}(\mathbb{R}_0^+)$. Then for each $r \in (0,\infty) r \in (0,\infty)$, we have

$$\begin{aligned} |(L_{n,\rho}^{\alpha}h)(r) - h(r)| \leq & \frac{\psi_{n,\rho}^{\alpha}(r)}{r} \sum_{j=1}^{\lfloor\sqrt{n}\rfloor} \left(TV_{r-\frac{r}{j}}^{r} h_r'\right) + \frac{r}{\sqrt{n}}\left(TV_{r-\frac{r}{\sqrt{n}}}^{r} h_r'\right) \\ & + \frac{r}{\sqrt{n}}\left(TV_r^{r+\frac{r}{\sqrt{n}}} h_r'\right) + \frac{\psi_{n,\rho}^{\alpha}(r)}{r} \sum_{j=1}^{\lfloor\sqrt{n}\rfloor} \left(TV_r^{r+\frac{r}{j}} h_r'\right) \\ & + \frac{[\psi_{n,\rho}^{\alpha}(r)]^{\frac{1}{2}}}{2}[|h'(r^+) + h'(r^-)| + |h'(r^+) - h'(r^-)|]. \end{aligned}$$

Proof For $r \in (0,\infty)$, we have

$$(L_{n,\rho}^{\alpha}h)(r) - h(r) = \int_0^{\infty} K_{n,r}^{\rho,\alpha}(y)(h(y) - h(r))dy = \int_0^{\infty} K_{n,r}^{\rho,\alpha}(y) \int_r^y h'(u)dudy$$

$$(L_{n,\rho}^{\alpha}h)(r) - h(r) = \int_0^{\infty} K_{n,r}^{\rho,\alpha}(y)(h(y) - h(r))dy = \int_0^{\infty} K_{n,r}^{\rho,\alpha}(y) \int_r^y h'(u)dudy$$

.

Consider the following prominent identity

$$\begin{aligned} g(p) = & g_r(p) + \frac{g(r^+) - g(r^-)}{2} \cdot \operatorname{sgn}(p-r) + \frac{g(r^+) + g(r^-)}{2} \\ & + \left(g(r) - \frac{g(r^+) + g(r^-)}{2}\right) \cdot \xi_r(p) \end{aligned}$$

applied on $h', \xi_r(p) h', \xi_r(p)$ is the well known Kronecker Delta defined as

$$\xi_r(p) = \begin{cases} 1, & p = r \\ 0, & p \neq r \end{cases}$$

and $\operatorname{sgn}\operatorname{sgn}$ is the Signum function. Clearly, we have

$$\int_0^{\infty} K_{n,r}^{\rho,\alpha}(y) \int_r^y \left(h'(r) - \frac{h'(r^+) + h'(r^-)}{2}\right) \cdot \xi_r(u)dudy = 0$$

Next, the application of Cauchy-Schwartz inequality gives

$$\left|\int_0^{\infty} K_{n,r}^{\rho,\alpha}(y) \int_r^y \left(\frac{h'(r^+) + h'(r^-)}{2}\right)dudy\right| \leq \frac{|h'(r^+) + h'(r^-)|}{2} \cdot [\psi_{n,\rho}^{\alpha}(r)]^{\frac{1}{2}}$$

Similarly, we obtain

$$\left|\int_0^{\infty} K_{n,r}^{\rho,\alpha}(y) \int_r^y \left(\frac{h'(r^+) - h'(r^-)}{2}\right)\operatorname{sgn}(u-r)dudy\right| \leq \frac{|h'(r^+) - h'(r^-)|}{2} \cdot [\psi_{n,\rho}^{\alpha}(r)]^{\frac{1}{2}}$$

On assembling the terms, we get

$$|(L_{n,\rho}^{\alpha}h)(r) - h(r)|$$

$$\leq |X_{n,r}^{\rho,\alpha}(h) + Y_{n,r}^{\rho,\alpha}(h)| + \frac{[\psi_{n,\rho}^{\alpha}(r)]^{\frac{1}{2}}}{2}[|h'(r^+) + h'(r^-)| + |h'(r^+) - h'(r^-)|] \quad (3)$$

where

$$X_{n,r}^{\rho,\alpha}(h) = \int_0^r K_{n,r}^{\rho,\alpha}(y) \int_r^y h_r'(u)\,du\,dy$$

$$Y_{n,r}^{\rho,\alpha}(h) = \int_r^\infty K_{n,r}^{\rho,\alpha}(y) \int_r^y h_r'(u)\,du\,dy$$

It is enough to approximate the terms $X_{n,r}^{\rho,\alpha}(h)$ and $Y_{n,r}^{\rho,\alpha}(h)$ to conclude the proof. Using integration by parts and the definition of $\lambda_n^{\rho,\alpha}(.,.)$ given in Lemma 2, we get

$$\left|X_{n,r}^{\rho,\alpha}(h)\right| \le \int_0^{r-\frac{r}{\sqrt{n}}} |h_r'(y)|\lambda_n^{\rho,\alpha}(r,y)dy + \int_{r-\frac{r}{\sqrt{n}}}^{r} |h_r'(y)| \cdot \lambda_n^{\rho,\alpha}(r,y)dy$$

Since $h_r'(r) = 0 h_r'(r) = 0$ and $\lambda_n^{\rho,\alpha}(r,y) \le 1$ $\lambda_n^{\rho,\alpha}(r,y) \le 1$, we obtain

$$\int_{r-\frac{r}{\sqrt{n}}}^{r} |h_r'(y)|\lambda_n^{\rho,\alpha}(r,y)dy \le \frac{r}{\sqrt{n}}\left(TV_{r-\frac{r}{\sqrt{n}}}^{r} h_r'\right)$$

Next, using Lemma 2, we have

$$\int_0^{r-\frac{r}{\sqrt{n}}} |h_r'(y)|\lambda_n^{\rho,\alpha}(r,y)dy \le \psi_{n,\rho}^{\alpha}(r)\int_0^{r-\frac{r}{\sqrt{n}}} \left(TV_y^r h_r'\right)\frac{dy}{(r-y)^2}$$

Substituting $y = r - \frac{r}{u}$, we get

$$\int_0^{r-\frac{r}{\sqrt{n}}} |h_r'(y)|\lambda_n^{\rho,\alpha}(r,y)dy = \frac{\psi_{n,\rho}^{\alpha}(r)}{r}\int_1^{\sqrt{n}} \left(TV_{r-\frac{r}{u}}^{r} h_r'\right)du \le \frac{\psi_{n,\rho}^{\alpha}(r)}{r}\sum_{j=1}^{\lfloor\sqrt{n}\rfloor}\left(TV_{r-\frac{r}{j}}^{r} h_r'\right)$$

Thus,

$$\left|X_{n,r}^{\rho,\alpha}(h)\right| \le \frac{\psi_{n,\rho}^{\alpha}(r)}{r}\sum_{j=1}^{\lfloor\sqrt{n}\rfloor}\left(TV_{r-\frac{r}{j}}^{r} h_r'\right) + \frac{r}{\sqrt{n}}\left(TV_{r-\frac{r}{\sqrt{n}}}^{r} h_r'\right) \tag{4}$$

Considering

$$\left|Y_{n,r}^{\rho,\alpha}(h)\right| = \left|\int_r^\infty K_{n,r}^{\rho,\alpha}(y) \int_r^y h_r'(u)\,du\,dy\right|$$

Using integration by part and applying Lemma 2, we obtain

$$\left|Y_{n,r}^{\rho,\alpha}(h)\right| \le \frac{r}{\sqrt{n}}\left(TV_r^{r+\frac{r}{\sqrt{n}}} h_r'\right) + \psi_{n,\rho}^{\alpha}(r)\int_{r+\frac{r}{\sqrt{n}}}^{\infty} TV_r^{y}(h_r')\frac{1}{(y-r)^2}dy$$

Substituting $y = r + \frac{r}{u}$, we get

$$\int_{r+\frac{r}{\sqrt{n}}}^{\infty} TV_r^{y}(h_r')\frac{1}{(y-r)^2}dy = \frac{1}{r}\int_0^{\sqrt{n}}\left(TV_r^{r+\frac{r}{u}} h_r'\right)du \le \frac{1}{r}\sum_{j=1}^{\lfloor\sqrt{n}\rfloor}\left(TV_r^{r+\frac{r}{j}} h_r'\right)$$

Thus we have

$$\left|Y_{n,r}^{\rho,\alpha}(h)\right| \le \frac{r}{\sqrt{n}}\left(TV_r^{r+\frac{r}{\sqrt{n}}} h_r'\right) + \frac{\psi_{n,\rho}^{\alpha}(r)}{r}\sum_{j=1}^{\lfloor\sqrt{n}\rfloor}\left(TV_r^{r+\frac{r}{j}} h_r'\right) \tag{5}$$

Finally collecting the approximates from (3), (4) and (5), we attain the required outcome.

Applications of Laguerre-Păltănea Operators in Engineering Problems

The construction of Laguerre- Păltănea operators on the non-negative real axis and their effective approximation properties made the operators relevant in several engineering disciplines. These operators have significance, particularly for the estimation of piecewise smooth or exponentially decaying functions. The phenomena, piecewise smoothness, and decaying pattern naturally arise in various models such as heat conduction, optimization, signal processing, numerical analysis, control theory, and many more. The following table demonstrates the relationship between Laguerre-Păltănea operators and different engineering problems.

TABLE 1: ENGINEERING APPLICATIONS OF LAGUERRE-PĂLTĂNEA OPERATORS

S No	**Discipline**	**Problem Type**	**Application**	**Remarks**
1	Signal Processing	Approximation of Signals	In compressing or denoising piecewise signals	It takes into consideration non-uniform data or discontinuous data
2	Mechanical Engineering	Damped Vibration	In finding Integro-differential equations solutions	& It is efficient for damp and decay phenomena
3	Control Systems	Identification of system	In modeling impulse response	It is useful for infinite-horizon dynamics
4	Thermal Engineering	Heat Equations on semi-infinite rods	In approximating the profile of temperature	It matches exponential decay in heat kernels
5	Electrical Engineering	Analysis of RC-Ladder Network	In the distribution of Voltage or current	It supports Laplace-domain conversion
6	Optimiz	Optimize	In semi-	

	ation	control problems	infinite horizon control estimations	It converts infinite-dimensional problems to finite-dimensional
7	Numerical Analysis	Approximation of functions	For approximation in the space $L^1[0,\infty)$	It is superior to classical polynomials for singularities
8	Biomedical Engineering	Modeling of Drug Absorption	In stimulus-response approximation	It captures prompt rise and slow decay dynamics

The above table shows how the problems on unbounded domains can be naturally solved using Laguerre-Păltănea operators. The structure of these operators allows good precision and adaptability in comparison to classical polynomial-based approximations. These operators are very significant in handling singularities, non-uniformity, and discontinuities. This establishes their importance as a unique tool in computational engineering and applied mathematics.

II. CONCLUSION

The findings in this study are in accordance with solving several practical engineering problems that relate to how these operators behave when they are applied to functions having limited smoothness and functions having derivatives of bounded variation. Approximating these types of functions with Laguerre Păltănea operators makes it an important and valuable tool to find numerical approximations of error and simulation tasks, which frequently occur in different engineering domains. Moreover, there can be various open problems, such as analyzing quantitative convergence rate using different moduli of smoothness, extension of it to a multivariate setting, developing computational approaches and algorithms depending on these operators for practical engineering embedded systems.

REFERENCES

[1] V. Gupta, V. Vasishtha, and M. K. Gupta, “Rate of convergence of summation-integral type operators with derivatives of bounded variation,” *J. Inequal. Pure Appl. Math*, vol. 4, issue 2, 2003.

[2] K. Kumar, N. Deo, and D. K. Verma, “Approximation by a new sequence of operators involving Laguerre polynomials,” *Mathematics, Functional Analysis*, 2024.

[3] S. Sucu, G. Icoz, and S. Varma, On some extensions of Szasz operators including Boas Buck-type polynomials. *Abstr. Appl. Anal*., vol. 2012, 2012

Can Online Teaching be a Futuristic Approach of Learning? A Detailed Study from Different Perspectives

Rashmi Gupta
Assistant Professor, English Language & Communication Skills
Maharaja Surajmal Institute of Technology, New Delhi, India
rashmi_gupta@msit.in

***Abstract*- Online learning platforms enable academic access for learners who face barriers to attending face-to-face educational institutions. In 2020, Covid 19 broke out and resulted in closure of all activities. People are not able to follow basic routine from attending schools, hanging out with friends to buying groceries. Consequently, more than 1.2 billion learners across 186 countries have been displaced from physical classrooms, accelerating the widespread adoption of online learning platforms. Digital education allows students to regulate their learning pace independently, provided they have appropriate technological resources and reliable internet access. Although online education is widely regarded as a key component of future learning systems, it cannot fully substitute the comprehensive and holistic benefits offered by traditional face-to-face education.**

The study aimed to know the opinion of undergraduate students on this new method of teaching and the traditional way of teaching. A questionnaire is given to 30 students through google forms. The findings of the study reveal that students are more comfortable with traditional method of teaching. More than 60% students want to attend offline classes. Even though online mode of education allows student to take classes from other corner of the world and save commuting time and expenses but understanding of concepts and interaction with peers and teachers are better in offline mode of classes. Student can attend offline classes without any technical glitches unlike online classes. In my research findings, offline classes are shown the best for the social and personal development of students.

***Keywords*-Online education. Offline education, Covid-19, Digital platforms, Traditional classrooms**

I. INTRODUCTION

Before we start our research on varied perspectives of Online education, we must understand what the purpose of education is. Education involves the systematic facilitation of learning, enabling individuals to acquire knowledge, ethical principles, practical skills, moral values, beliefs, habits, and personal competencies. From past centuries students went to Gurukuls (now schools) to get education.

Students have face to face interactions with teachers and peers and gain knowledge .The Covid-19 pandemic brought a dynamic shift in the world education system**.** The imposition of lockdown led to the shutdown of physical classrooms and thus online education became the new norm. Students instead of taking classes in physical classroom start taking classes at online platform like google meet or zoom. Instead of blackboard students start learning from power point presentation. Online learning has managed to keep education alive in these dire times.

Online education offers a viable learning alternative for students who are unable to attend conventional classroom-based instruction while also enabling learners to regulate the pace of their studies. It promotes the development of self-discipline and effective time-management skills and provides access to a wide range of digital educational resources. Furthermore, students can personalize their learning experiences when supported by appropriate technological infrastructure and reliable internet connectivity. Although online education is increasingly recognized as a significant component of future learning frameworks, it cannot fully replicate the holistic advantages of traditional face-to-face instruction.

Offline education remains unaffected by technological limitations and encourages students to maintain structured routines and disciplined study habits. Furthermore, face-to-face learning environments allow educators to closely observe student engagement, behaviour, and learning responses, enabling timely and personalized academic support. Therefore, despite continuous advancements in digital learning technologies, conventional education will continue to play an indispensable role in fostering students' academic and personal development.

The study is conducted to know which mode of education is able to impart knowledge better, how much syllabus is covered in offline and online classes, which mode is easier to connect with peers and teacher, which mode is easier for students to evaluate their performances, which environment provide better competition and opportunities and how effectively they can manage their time. Through these comparisons, we know that student can manage their time well in online classes as they save commuting time but the concept clarity and fair evaluation of assessment is done in offline classes which is important process while getting education.

II. LITERATURE REVIEW

Online Education

Online education is a flexible mode of learning that enables students to access instructional content through the internet using personal devices, allowing them to study from any location. This approach does not require physical face-to-face interaction between students and instructors, thereby supporting learning beyond geographical boundaries.

Benefits and Disadvantages of Online Education:

Benefits of Online Education	Disadvantages of Online Education
Location doesn't matter	Isolating
Flexible hours	No competition
Time saving	No interpersonal skills development
Cheaper	Limited topics
Convenient	Must have self discipline
Class recorded	

Fig. 1 Benefits and disadvantages of online education

Challenges Faced in Online Classes

- **Technological Constraints:** Many learners lack adequate access to essential devices, updated software, or stable internet connectivity required for effective participation in online learning. Incompatibility between personal systems and institutional platforms can further hinder seamless engagement.
- **Limited Social and Academic Interaction:** Although remote learning offers convenience, it often reduces opportunities for meaningful interaction between students and instructors. The absence of real-time discussions and peer engagement can lead to feelings of isolation and negatively impact learning motivation and collaboration.
- **Digital Distractions:** Online learning environments frequently involve passive content consumption, such as reading digital material or attending virtual lectures. When learners are not actively engaged, they are more likely to lose focus and become distracted by emails, social media, or unrelated web activities.
- **Lack of Self-Motivation and Discipline:** Online learning requires high levels of self-regulation, and many students struggle to maintain consistency without structured supervision.
- **Assessment and Evaluation Difficulties:** Ensuring fairness, academic integrity, and accurate assessment in virtual environments remains a significant challenge.
- **Instructor Readiness and Training:** Not all educators possess the technical skills or pedagogical training necessary to effectively deliver online instruction.
- **Health and Well-being Concerns:** Prolonged screen exposure and reduced physical activity can lead to eye strain, fatigue, and mental health issues.
- **Accessibility and Inclusivity Issues:** Students with disabilities or those from disadvantaged backgrounds may face additional barriers in accessing online learning resources.

Offline Education

Offline education requires students to attend educational institutions in person, where learning takes place within a physical classroom through direct, face-to-face interaction with instructors. This mode of learning allows continuity by enabling learners to resume instruction from the point at which they previously concluded their studies.

Benefits and Disadvantages of Offline Education:

Challenges Faced in offline classes

- **Time Consumption:** Offline education requires students to be physically present on campus, often resulting in significant time spent on daily commuting. In contrast, online education minimizes travel time by allowing learners to attend classes remotely.
- **Transportation Expenses:** Students attending offline classes incur regular transportation costs, particularly those residing far from educational institutions. Online education reduces or eliminates these recurring expenses by enabling learning from home.
- **Rigid Scheduling:** Fixed class timings in offline education limit flexibility, making it difficult for students to balance academics with personal or professional responsibilities.
- **Physical Fatigue:** Daily travel and extended classroom hours can lead to physical exhaustion, negatively affecting concentration and learning efficiency.
- **Limited Access to Learning Resources:** Educational materials and facilities are often accessible only on campus, restricting learning opportunities outside institutional hours.

Benefits of Offline Education	Disadvantages of Offline Education
Individualised monitoring	Fixed location
Structured & disciplined setting	Fixed schedule
Face to face interactions	Study materials
Interpersonal skills development	Dependency on the teachers
Competitive atmosphere	Transport & Accommodation

Fig. 2 Benefits and disadvantages of offline education

- **Health and Safety Concerns:** Students may face health risks due to exposure to crowded environments, commuting stress, or adverse weather conditions.

- **Environmental Impact:** Regular commuting contributes to increased fuel consumption and environmental pollution.

III. METHODOLOGY

For this study 35 undergraduates were surveyed from different courses, different colleges and different semesters. Students were asked about how much syllabus were covered in offline classes and in online classes. Students were investigated about, in which mode they able to manage time efficiently and understands topics clearly, revision and which mode provide more future opportunities. For the survey a questionnaire is prepared that addresses the basic aspects of education and serve the purposes and context of these study. The data were collected through Google form in the form of multiple choice questions. Total 9 questions were asked to gain clear perception of students on mode of education they preferred. After that responses recorded in google form were analysed using the tables and pie chart. Data from percentage of syllabus covered in online mode is calculated and compared with percentage of syllabus covered in offline mode and conclusion is drawn. A table to compare the responses for offline and online mode is also constructed and studied.

IV. RESULTS

The Study shows that 63% students preferred offline mode of education than online mode of education. After studying the responses from students we can conclude that offline classes are easy to connect with peers and teachers, provide more concept clarity and revision and evaluation of assessment is fair, while Online education is less expensive, and time management is easy and provide new opportunities to learn while staying home with family. The speed of learning is more in online education as teachers use visuals methods to teach and provide more time to focus on study.

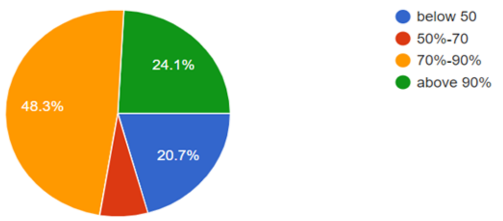
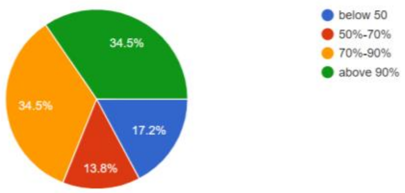


Fig. 3 Syllabus Covered in Offline Classes

FIG.3 shows the data about Syllabus covered in offline classes. For more than 24% students above 90% syllabus is covered in offline classes and for about half students' syllabus covered in classes is 70 to 90% which is pretty nice. FIG.4shows the data about the Syllabus covered in online classes. For more than 34% students above 90% syllabus is covered in online classes and there is only 17% student for which syllabus covered in online classes is less than 50 %. Comparing FIG.3 and FIG.4 we can conclude that amount of syllabus covered in online classes is more than offline classes. As in online classes, student does not need to commute from one place to another and that time can be used for studies. Online mode of education also provides options to record lectures so student can view the lectures according to their comfort. For more than 34% students' syllabus covered in online classes is more than 90% which is only 24% in offline classes.



Z

Fig. 4 Syllabus Covered in Online Classes

Table 1 shows the Undergraduate students perception on online classes and offline classes. Around 74% students agreed that they understand concepts using the traditional way of teaching that is one to one with teacher and peers using board and books. While 25% student find online mode of teaching more easy to learn that is using power point presentation and other visual technologies. 85% students find easy to connect with peers and teachers in offline mode, they can ask the doubts easily and also interaction and discussions with friends are fun and better while in online mode due to distance and technical issues it's not easy to connect with friends and understanding their opinions.

Table 1 Students opinion about offline and online education

74.3% students think that evaluation of performance in offline mode is fair as all the students take exams at same place in presence of teachers while in case of online mode students may get involve in malpractice like cheating and does not take exam seriously. It also decreases learning potential in student as they only focus on getting marks by cheating rather than gaining knowledge. Around 57% student favoured online classes for time management. In online mode student can attend classes from home so they don't have to waste time in travelling from home to school and could use that time for studying. While 42% student find easy to stick to the schedule provided by teachers as in classes they don't had any kind of distractions while in online mode student may waste time of class in surfing online platforms and get into habit of procrastination. 54% students find offline mode providing better environment for competition and placement opportunities as they can be easily guided by teachers and can learn from peers. Online mode is unquestionably more cost effective as students doesn't have to pay for hostel, mess or travel fees. Even though online mode is less expensive, save time and provide flexible studies it cannot replace the holistic aspect of offline mode of education. More people still preferred offline mode of education. Both offline and online education have its pros and cons but online mode of education cannot completely replace offline education and become the future of learning.

Due to Covid 19, student had to opt online mode of education to continue learning and it had overcome many challenges faced by student in offline education and showed its advantages but more students acknowledge that offline mode of education is better for personal and social development of human being. Online education is a great alternative but cannot surpass offline education.

TABLE 1 STUDENTS OPINION ABOUT OFFLINE AND ONLINE EDUCATION

S.No	Statements	Percentage for Offline Education	Percentage for Online Education
1	Concepts Clarity	74.3%	25.7%
2	Easy to connect with peers and teachers	85.7%	14.3%
3	Revision	60%	40%
4	Fair Evaluation	74.3%	25.7%
5	Time Management is best in	42.9%	57.1%
6	Better for Competitions and Placements	54.3%	45.7%
7	Cost Effective	89%	11%
8	Preferred Mode	63%	37%

V. CONCLUSION

In conclusion, the survey revealed that student favours offline education than online education. They are of opinion that offline education provides more concept clarity and has more possibilities of fair evaluation compared to online exams. The study also revealed that undergraduates find it easy to connect with peers and teachers and can also get better competition and placement environment as compared to online education. They get more time for revision and face to face interaction than in online education.

Students prefer offline education mode because often fails in their attempts to do online assessment as it becomes difficult for them because they couldn't focus on the assessment when there are many distractions around and technicality issues like internet connection and improper functioning of MS Teams and Google Classrooms because of data trafficking. These challenges are often frustrating because these issues are beyond their control. They could not control the variables such as internet connection that demotivates them while they try to do the assessment.

REFERENCES

[1] Bliuc, A., Ellis, R., Goodyear, P., and Piggott, L. (2010). Learning through face-to-face and online discussions: Associations between student‟s conceptions, approaches and academic performance in political science. British Journal of Education Technology, 41 (3).

[2] Cappel, J.J. and Hayen, R.L.(2004). Evaluating e-learning: A case study. Journal of Computer Information Systems, Summer.

[3] Lundgren, T.D. and Nantz, K.S.(2002). Student attitudes toward Online courses: A longitudinal study, Journal of Computer Information Systems, Summer.

[4] Olson, D. A (2002). Comparison of online and lecture methods for delivering the CS 1 course. Journal of Computer Sciences in Colleges, 18 (2, Dec).

[5] Rivera, J.C. and Rice, M.L.(2002). A comparison of student outcomes and satisfaction between offline and online based study offerings, Online Journal of Distance Learning Administration, 3 (Fall).

Issues and Trends in Generative AI in Healthcare Sector

Mamta Gahlan[1], Jyoti Arora [2,] Indu*

[1,2,3]Assistant Professor, MSIT, Department of IT, Delhi, India

[1]mamtagahlan@msit.in

[2]jyotiarora@msit.in

[3]indukhatri@msit.in

Abstract*- Generative artificial intelligence (AI) has become a very powerful technology that is applied in many areas and fields. Our findings suggest a dichotomy in the understanding and application of the term "Generative AI". Generative AI is based on the existing technologies, such as large language models (LLMs) that are trained on huge volumes of text and conditioned to make predictions about the next word in a sentence. Generative AI also gives newly developed solutions to content creation in the metaverse, which can help bridge the gaps in the development of the metaverse. This paper describes the unique practical opportunities and challenges that Generative AI brings and gives insights into the trends of Generative AI and the issues of Generative AI inside the region of medical healthcare. Recently, a trend called "make it more" using ChatGPT (Chat Generative Pre-Trained Transformer) has taken the internet by storm. It is about asking the AI tool to generate a particular image and then enhance a particular aspect of it through successive prompts. ChatGPT is a product of AI that is currently being widely discussed on Twitter. The ChatGPT has a productive and significant growth from late 2022. Through this article, we are framing the discussion on the main four domains i.e. Generative AI and ChatGPT in the healthcare industry, Generative AI in the healthcare industry, Regenerative AI in Medicine and Healthcare and shortcomings.

Keywords*- Artificial Intelligence, Content Generation, Generative AI, Language Models, Variational Autoencoders (VAE), Generative Models, Generative Adversarial Networks (GANs).

I. INRODUCTION

ARTIFICIAL Intelligence (AI) is one of the hottest buzzwords in tech for a good reason. Nowadays artificial intelligence or the Generative AI (GAI) has been a consistent topic in various fields like education, and industries and gaining the attention of the public of different age groups. Various discussions, debates, and studies have been observed from 2018 since the depiction which was based on Generative AI.

Later, the adoption of Transformer [2] in NLP and diffusion model [3] in image synthesis created progress in gene evolution, which was reflected in works such as LaMDA [4], Llama 2 [5], GPT-4 (OpenAI, 2023), DALL-E 2 [6], and Stability AI.

During the last several years, there has been a revival of AI with the introduction of AlphaGo in 2015 and ChatGPT in 2022 and the launch of a notable AI-based application named Chat Generative Pre-Trained Transformer or ChatGPT by OpenAI in late 2022, which was widely discussed.

A previous study from Accenture said generative AI has the potential to automate or augment 39% of all working hours in the healthcare industry. According to new report regarding the Generative AI: The possible breakthroughs in the healthcare sector through advances in large language AI models applied by Generative AI can transform health care to enhance creativity and improve productivity among providers and patients. AI has led to the generation of room for challenges as well as opportunities in numerous fields, including technology (NLPs and Autonomous systems), business (Predictive analytics and Operational efficiency), education (Adaptive learning and Intelligent tutoring), healthcare (Diagnostic Learning and Personalized Treatment) as well as arts and humanities (Generative art and Art authentication) [7].

The main reality in the healthcare sector is that a large percentage of the medical institutions and the medical area are facing scarce clinical resources. As Generative AI is introduced it resolves the major problems in the healthcare sector. The greatest potential of generative AI in healthcare is to offload overworked and underpaid employees with administrative duties to allow them to practice at the highest level of their license. Nevertheless, the work of Generative AI should be also relevant to the new and latest technologies to assist clinicians and patients. efficiently. As more than half of healthcare organizations plan to use ChatGPT for learning purposes, and more than half are planning pilot cases this year, these steps will be critical in ensuring a positive out come.

According to the Congressional Budget Office, the process of new drug development costs on average $1 billion to $2 billion, which also includes failed drugs. AI has the potential to cut the money and time of the medication and the research part by half and also suggest the appropriate methods and help in drug development. It saves the pharma industry around $26 billion in annual expenses in the process. Additionally, this technology can reduce costs associated with clinical trials by $28 billion per year. Traditional methods often rely on the identification and modification of existing compounds, which can be a slow and labor-intensive process. AI-based approaches, on the other hand, can enable the rapid and efficient design of novel compounds with desirable properties and activities.

Just recently, Recursion Pharmaceuticals acquired two Canadian AI startups for $88 million. A research team built a generative AI system, Protein SGM[10], that can generate novel realistic proteins after studying imagery representations of existing protein structures.

The term AI has been thrown around for so many different purposes in so many different ways and we see Generative AI as an experience just to give it the most basic definition has to be an AI algorithm that is creating something that's really the fundamental difference between Generative AI and the kinds of analytic AI .

We have been seeing for a while now those kinds of algorithms machine learning algorithms are fundamentally about identifying correlations and relationships between variables within a data set and that's how it gets used to identify Trends within a huge data set or find potential indicators of disease or when a patient is going to be most likely to need attention or most at risk of hospitalization.

Generative AI has a component which is doing that but then it has a second element which is creating a new data set that is statistically equivalent to the data set it was trained on or analyzed and that's the fundamental generative part it has to be creating something whether that's an image a movie a piece of text an overlay on a screen[11].

AI will be necessary for most of the trends presently here, but Generative AI, in particular, will be particularly impactful over the next 12 months. Numerous of these developments will involve AI, but over the next year, Generative AI in particular will have a significant impact. It will simplify the process of deploying, analyzing, and creating customized suggestions for other disruptive AI applications. It will generate synthetic data that can be used to train medical AI systems in the circumstances when there is insufficient relevant real-world data or when the privacy of the patient is at risk. [12].

Additionally, it may generate virtual assistants like chatbots to support patients at every turn. The applications of Generative AI in healthcare are practically limitless, and we'll also discuss the other trends in the table mentioned below: Trend of Generative AI and ChatGPT in the Healthcare Sector In 2024, the Global **Generative AI in the Healthcare Market** was valued at **USD 1950 million** and is expected to be valued at **USD 39700 million** in 2034. The Compound Annual Growth Rate (CAGR) is a way to measure how an achievement has grown overall over a specific period. Table 1 represents Annual Data of Market Value of Generative AI and ChatGPT in the Healthcare sector.

TABLE 1: ANNUAL DATA OF MARKET VALUE OF GENERATIVE AI AND CHATGPT IN HEALTHCARE SECTOR

Years	Market value (in millions)
2024	1950
2025	2640
2026	3570
2027	4820
2028	6520
2029	8810
2030	11900
2031	16090
2032	21740
2033	29400
2034	39700

Between 2024 and 2034, this market is estimated to register the highest Compound Annual Growth Rate (CAGR) of **31.51%**.

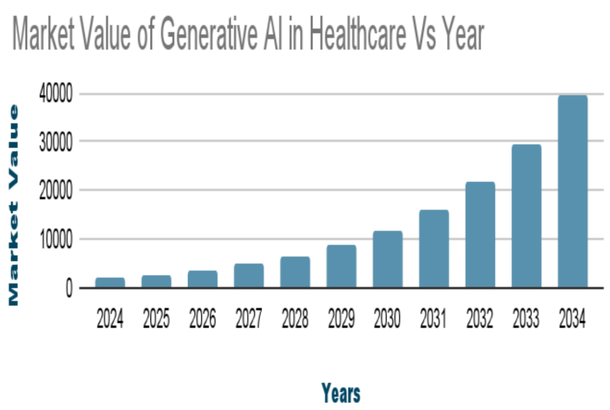


Fig. 1: Chart of the market value of Generative AI in healthcare in the upcoming 10 years from 2024-2034

In Fig.1, graph shows the rate of change in the market value of Generative AI and ChatGPT in the Healthcare sector from 2024-2034 where the market value (in USD) is marked on the y-axis and the time marked on the x-axis of the graph. In this graph, the Market value is boosted from 1950 million USD to 39700 million USD in 10 years by USD 37750 million, which indicates a huge increment of the generative AI and ChatGPT in the Healthcare sector[13].

A novel approach to machine learning called "Generative AI" produces data after learning the characteristics of actual data. Artificially created patient data has the power to transform clinical research and support patient privacy protection. By automatically concealing repetitive

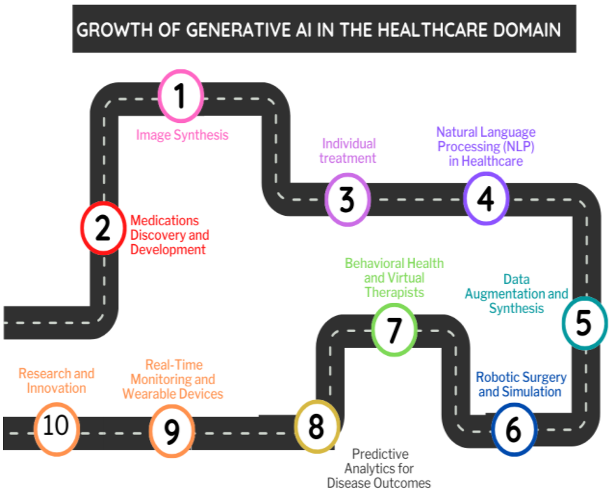


Fig2: Factors of Growth of Generative AI in Healthcare

operations including writing referral letters, clinical coding, and clinical consultations, generative [14] AI has the potential to relieve physicians of some of their administrative stress. The increasing number of hospitals and the revolution in the healthcare sector are driving the need for generative AI in the healthcare sector. Over 30% of New Drugs are predicted to be discovered by Generative AI by 2025. Due to the growing need for different tools related to Generative AI in the healthcare sector, this study focuses on the comprehensive review of recent articles and the development of generative AI applications in healthcare.

II. CONTRIBUTIONS AND PAPER SIGNIFICANCE

The questions and answers that were used in this study are of great importance to Generative AI and ChatGPT in healthcare. Firstly, by investigating the necessary AI tools that are working using Generative AI for the healthcare sector and the upcoming tech tools to be launched, this research paper provides valuable insights into the AI tools, their functions, applications, limitations, and the discussion. Understanding these requirements and the limitations of the Generative AI tools, ensure their successful performance and effective implementation. Additionally exploring the trend of Generative AI and ChatGPT in the healthcare sector by the analysis of the historical trends and predict the graphical market value data in the upcoming 10 years. The present-day activities of Generative AI and LLMs in medicine and healthcare in different ways are covered in this paper as it elucidates clinical administration support, clinical decision support, patient engagement, and synthetic data generation, and professional education. These findings contribute to advancements in the field, guiding researchers, developers, and practitioners in effectively implementing and evaluating Generative AI models for various applications [15].

Growth Of Generative AI In the Healthcare Domain

The growth of Generative Artificial Intelligence (Generative AI) and ChatGPT in the healthcare domain has been significant in recent years, as the growth of Generative AI has been
boosting from the past few years and is likely to be a boom in the future with various applications and advancements contributing to improvements in patient care, diagnostics, and medical research. Key factors driving the growth of Generative AI in healthcare include various aspects.

1. **Image Synthesis:** Generative AI, especially Generative Adversarial Networks (GANs), has been successfully applied in medical imaging for tasks such as image synthesis, segmentation, and reconstruction. GANs can generate realistic medical images, aiding in training machine learning models and enhancing diagnostic capabilities [17].
2. **Medications Discovery and Development:** Generative AI plays a crucial role in accelerating Medication discovery by predicting molecular structures, and molecular compounds, identifying the drug, generating novel compounds, and optimizing existing drug candidates [18]. This helps researchers identify potential drug candidates more efficiently,
3. **Individual Treatment:** Generative AI contributes to the advancement of Individual treatment by analyzing individual patient data and generating tailored treatment plans. This includes predicting patient responses, and patient medical history to specific treatments based on genetic, clinical, and other relevant data.
4. **Natural Language Processing (NLP) in Healthcare:** Generative AI NLP models are used in the extraction of insights on unstructured clinical notes, medical literature, and patient records. [19]. This aids in automating medical documentation, improving information retrieval, and facilitating data-driven decision-making by healthcare professionals.
5. **Data Augmentation and Synthesis:** Generative AI techniques are employed for data augmentation in healthcare, addressing challenges related to limited and imbalanced datasets. By generating synthetic data, models can be trained on more diverse examples, enhancing their robustness and generalization to different patient populations.
6. **Robotic Surgery and Simulation:** Generative AI is used in the development of realistic surgical simulations and robotic surgery training programs. These simulations help train surgeons, refine surgical techniques, and improve overall surgical outcomes.
7. **Predictive Analytics for Disease Outcomes:** Generative AI models contribute to predictive analytics by analyzing patient data to forecast disease progression, identify potential complications, and support proactive interventions. This aids in preventive healthcare and early detection of adverse events.
8. **Real-Time Monitoring and Wearable Devices:** Generative AI assists in analyzing real-time health data from wearable devices, enabling continuous monitoring of patients. This helps in the early

detection of anomalies, personalized feedback, and remote patient management.

III. DISCUSSION

Generative AI is poised to revolutionize medicine and healthcare in the upcoming years [27in this paper we also mention the various applications which are glimpse of what is running in current time and what will run. We expect generative AI to continue trending during the coming months and years. In the current study, a comprehensive analysis of the leading ChatGPT-related healthcare publications showed three primary trajectories for future research in this emerging subject. First, in healthcare education, the findings highlighted the promising role of ChatGPT as an example of generative AI models for tailoring educational experiences to individual student needs which can facilitate a personalized and effective learning environment

Second, in healthcare practice, ChatGPT emerged as a valuable generative AI tool that can be used to enhance workflow efficiency and patient communication. ChatGPT's Potential extends to facilitate medical diagnoses and treatment planning, while simultaneously reducing the burden of routine tasks of health professionals, in low-income settings

Third, the incorporation of ChatGPT in healthcare research can pave the way for a new era of innovation coupled with enhanced efficiency. For example, ChatGPT can be utilized as an aid in academic writing e.g., preparation of grant proposals, and manuscript preparation). Consequently, ChatGPT among other generative AI models tailored for such a purpose can enhance the speed of formulating complex scientific ideas.The following discussion will address some common concerns, challenges, and general opportunities associated with generative AI and related products such as ChatGPT.

IV. CONCLUSION

In this article, we discussed the issues and concerns of the Generative AI and ChatGPT in the Healthcare Sector such as clinical safety, reliability, privacy, ownership, trust copyrights and much more. We also discussed about the technology currently used in the medical and healthcare sector, such as Ellen AI, Redbrick AI's Fast Automated Segmentation (F.A.S.T), Hippocratic AI and much more with their merits and demerits. We also discussed the limitations of Generative AI and ChatGPT in the healthcare sector. We expect that all these concerns and the information of Generative AI and ChatGPT in healthcare sector will gradually get addressed as the politics, laws and regulatory frameworks surrounding the use of Generative AI begin taking shape.

REFERENCES

[1] Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In Advances in Neural Information Processing Systems (NIPS'14), p. 2672–2680.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NIPS'17).

[3] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics.

[4] Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., ..., & Le, Q. (2022). LaMDA: Language Models for Dialog Applications. arXiv:2201.08239 [cs.CL].

[5] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models

[6] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical TextConditional Image Generation with CLIP Latents. arXiv:2204.06125

[7] Siwei Lyu, Daniel Rockmore and Hany Farid, A digital technique for art authentication, Available online: https://www.researchgate.net/publication/8162995_A_digital_technique_for_art_authentication

[8] Spataro, J. Introducing Microsoft 365 Copilot—Your Copilot for Work. Official Microsoft Blog. March 2023.

[9] Rahaman, M.S.; Ahsan, M.M.; Anjum, N.; Rahman, M.M.; Rahman, M.N. The AI Race Is on! Google's Bard and OpenAI's ChatGPT Head to Head: An Opinion Article: SSRN 4351785; SSRN: Rochester, NY, USA, 2023; Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4351785 .

[10] Regard. Torrance Memorial Medical Center Reduces Physician Burnout, Increases Annual Revenue by $2 Million with the Help of Regard Case Study. Available online: https://withregard.com/case-studies/tmmc-reduces-burnout

[11] Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. arXiv 2023, arXiv:2304.02643.https://doi.org/10.48550/arXiv.2304.02643

[12] Krishna, K.; Khosla, S.; Bigham, J.P.; Lipton, Z.C. Generating SOAP notes from doctor-patient conversations using modular summarization techniques. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 2–5 August 2021; pp. 4958–4972. https://doi.org/10.18653/v1/2021.acl-long.384

[13] Lee, P.; Goldberg, C.; Kohane, I. The AI Revolution in Medicine: GPT-4 and Beyond, 1st ed.; Pearson: London, UK, 2023; ISBN-10: 0138200130/ISBN-13: 978-0138200138; Available online: https://www.amazon.com/AI-Revolution-Medicine-GPT-4-Beyond/ dp/0138200130/

[14] Nolan, M. Llama, and ChatGPT Are Not Open-Source—Few Ostensibly Open-Source LLMs Live up to the Openness Claim. IEEE Spectrum 27 July 2023. Available online: https://spectrum.ieee.org/open-source-llm-not-open

[15] Zhang, D.; Finckenberg-Broman, P.; Hoang, T.; Pan, S.; Xing, Z.; Staples, M.; Xu, X. Right to be Forgotten in the Era of Large Language Models: Implications, Challenges, and Solutions. arXiv 2023. https://doi.org/10.48550/arXiv.2307.03941

[16] Microsoft Corporation. Summary of Changes to the Microsoft Services Agreement—30 September 2023. Available online: https://www.microsoft.com/en-us/servicesagreement/upcoming-updates

[17] Jovanovic. M, Campbell.M; Generative artificial intelligence: trends and prospects,https://ieeexplore.ieee.org/document/9903869/

[18] Claburn, T. Microsoft, OpenAI Sued for $3B after Allegedly Trampling Privacy with ChatGPT. The Register 28 June 2023. Available online: https://www.theregister.com/2023/06/28/microsoft_openai_sued_privacy/

AI-Powered Code Refactoring Assistant and Automated Debugger

Dr. Vinita Rohilla#1, Lakshay Mangla#2, Rishabh Chawla#3, Kapil Kumar#4, Harshit#5

#Maharaja Surajmal Institute of Technology, New Delhi, India

1vinita.rohilla@gmail.com

2lakshay040904@gmail.com

3rishabhchawla2004@gmail.com

4kapilsk838564@gmail.com

5mrharshit30@gmail.com

***Abstract*—The proliferation of web-based coding platforms for education and development has created a significant challenge: securely executing untrusted, multi-language code while provid- ing intelligent user feedback. This paper presents the architecture of the "AI-Powered Code Refactoring Assistant and Automated Debugger," a full-stack application that addresses this problem. The system uses a Python Flask backend to manage a dual-AI "brain" built on the Google Gemini API, separating bug-fixing from quality-of-life refactoring. For refactoring, it employs a novel hybrid analysis, where fast, local Abstract Syntax Tree (AST) analyzers for Python, Java, and C/C++ first find "code smells," which are then passed to the AI for an intelligent rewrite. For execution, the system uses multiple, language-specific Docker containers with strict security constraints, including disabled networking, memory limits, and a dedicated writable output directory. This sandbox environment is capable of handling complex scripts, including those requiring user input, data science libraries like Pandas and Matplotlib, and even generating visual plots as output.**

***Index Terms*—Docker, Sandbox, Secure Code Execution, Large Language Models, AST, Code Refactoring, Python, Java**

I. Introduction

Online Integrated Development Environments (IDEs) and code-learning platforms have become essential tools for mod- ern software development and education. However, they face a fundamental conflict: they must be powerful enough to run complex, real-world code but secure enough to prevent malicious users from attacking the host server. A simple execution environment is not enough; modern users also expect intelligent feedback to help them learn and improve.

This paper details a system that solves both problems. We present a web platform that can analyze, correct, refactor, and safely execute code in four languages: Python, Java, C, and C++.

Our contribution is a novel, three-part architecture:

- A secure, "batteries-included" Docker sandbox pre- loaded with over 50 data science and ML libraries.
- A dual-logic AI backend that separates bug-fixing (the "Mechanic") from code improvement (the "Engineer").

This work is submitted as a minor project for the Bachelor of Technol- ogy (Computer Science and Engineering) at Maharaja Surajmal Institute of Technology, New Delhi.

A hybrid refactoring engine that uses lightweight local Abstract Syntax Tree (AST) analyzers to find "code smells" and then leverages a Large Language Model (LLM) to perform the complex, context-aware rewrite.

This paper outlines the project's methodology, discusses the results of its implementation, and concludes with potential avenues for future work.

II. Literature Review

This project integrates backend, DevOps, and AI technolo- gies. The implementation of our secure sandbox is based on the principles of OS-level containerization described by D. Merkel [21], who detailed the lightweight and consistent environment provided by Docker. This addresses a primary research gap in web platforms: the safe execution of untrusted code [24].

For our automated debugging feature, we leverage the capabilities of Large Language Models (LLMs) trained on code, a concept validated by Chen et al. [17] in their work on OpenAI Codex. Our system's specific use of an LLM to fix code is an application of the broader field of Automated Program Repair (APR) [16]. The "dual-brain" logic in our backend relies heavily on prompt engineering, a methodology formalized in a "Prompt Pattern Catalogue" by White et al. [26].

Our refactoring engine builds on the foundational concept of "code smells" defined by M. Fowler [1]. While traditional analysis relies on detecting these smells [30], our hybrid system advances this by feeding the analysis from local AST parsers (`javalang` [23], `clang` [22], Python's `ast` [9]) to an LLM, which then performs the complex rewrite. This hybrid model addresses the challenge of extending analysis to multiple languages, a problem explored by Tu & Godfrey [14].

III. Research Methodology

The system is a full-stack application composed of a JavaScript-based frontend, a Python Flask backend, and a set of Docker containers. The backend server acts as the central coordinator, managing both AI analysis and secure execution.

A. Secure Execution Sandbox (Docker)

The system's core security relies on isolated Docker con- tainers. We designed three separate `Dockerfile` blueprints: one for Python, one for Java, and one for C/C++. The Python container is the most complex, pre-installing over 50 libraries (e.g., NumPy, Pandas, TensorFlow, Matplotlib, OpenCV) and their system dependencies (`libgl1`, `libhdf5-dev`).

When the `/api/execute` endpoint is called, the Flask server uses Python's `subprocess` module to run the code in the appropriate container with strict security flags:

- `--rm`: Guarantees the container is disposable and auto- matically destroyed after execution.
- `--network none`: Disables all network access to pre- vent any external or internal network calls.
- `--memory 1g`: Allocates a fixed 1GB of RAM to prevent resource exhaustion.
- `--user sandboxuser`: Runs the code as a non-root, unprivileged user.
- `-v ...:ro`: Mounts the code script as a read-only file.
- `-v ...:/app/output`: Mounts a small, writable di- rectory, allowing the code to save generated files like `output.png` or `output.html`.

B. Dual-Logic AI Engine

To provide distinct and accurate feedback, we implemented two separate AI "personalities" by using two different, highly- engineered prompts.

1) The "Mechanic" (Bug Fixing): The `/api/fix-code` endpoint calls the Gemini API with a strict prompt that commands it to act as a "bug-fixing specialist." Its only job is to correct critical syntax or runtime errors. It is explicitly instructed **not** to refactor code for style. It also handles sandbox-specific rules, such as renaming a Java `public class` to `Main` to match the filename.

2) The "Engineer" (Refactoring): The `/api/refactor` endpoint uses our hybrid approach:

1) **Inspect (Local AST):** The server first runs the code through a local, language-specific analyzer (`ast_analyzer.py`, `java_analyzer.py`, or `cpp_analyzer.py`). These fast scanners find specific, pre-defined "code smells" (e.g., long functions, empty `catch` blocks, `goto` statements).
2) **Refactor (AI):** The backend then calls the Gemini API with a different prompt, providing both the code and the list of smells found by the analyzer. It asks the AI to act as a "refactoring engineer" to fix these specific issues and improve overall code quality, returning the refactored code and a detailed explanation.

C. Handling Interactive and Visual Code

The system is designed to handle code that is not purely text-based.

- **User Input:** The AI's `fix-code` prompt instructs it to detect calls to input functions (like `input()` or `Scanner`) and return a JSON boolean flag, `requires_input: true`. The frontend uses this flag to intelligently show a text box for the user to provide all inputs at once, which are then piped into the container's standard input.
- **Data Visualization:** The AI is instructed to convert Python plotting commands (e.g., `plt.show()`) into file-saving commands (e.g., `plt.savefig('output/output.png')`). The backend then reads this image from the shared `output` directory, encodes it in Base64, and sends it to the frontend to be displayed as an `<img>` tag.

IV. Results and Discussions

The full system was tested to validate all components, demonstrating the complete workflow from bug-fixing to refactoring and final execution. The following two test cases demonstrate the system's full capabilities.

A. Case Study 1: AI-Powered Debugging and Refactoring

This test validates the "Mechanic" and "Engineer" work- flows.

1) As shown in Fig. 1, a "messy" Python script with multiple syntax errors was submitted. The input code (left) contained a missing colon, variable misspellings, and indentation errors.
2) The `/api/fix-code` endpoint was called. The AI "Mechanic" successfully identified and corrected all critical errors, returning the runnable (but still messy) code and a detailed explanation of the fixes.
3) As shown in Fig. 2, the "Start Refactoring Methods" button was then pressed.
4) The backend ran `ast_analyzer.py`, which identi- fied the corrected function as being too long (a "code smell").
5) This "smell" was fed to the AI "Engineer," which successfully refactored the single "God Function" into two smaller, focused functions.
6) The frontend displayed the new, high-quality refactored code (top) and the AI's detailed explanation (bottom left). The final code was then executed, producing the correct 'Total: $223.80' output (bottom right).

B. Case Study 2: Advanced Execution (Data Visualization)

This test validates the ”batteries-included” sandbox and the AI’s ability to adapt code for plotting.

1) As shown in Fig. 3, a Python script using `pandas`, `numpy`, `matplotlib`, and `seaborn` was submitted.
2) The ”Analyze & Fix Code” button was pressed. The AI ”Mechanic” (Fig. 3) corrected minor typos and correctly adapted the code for our sandbox by changing visualization logic to `plt.savefig(’output/output.png’)`. The explanation for these fixes was also provided.
3) The ”Start Refactoring Methods” button was also pressed (Fig. 4). The local `ast_analyzer.py` cor- rectly found no ”code smells” in this simple script.

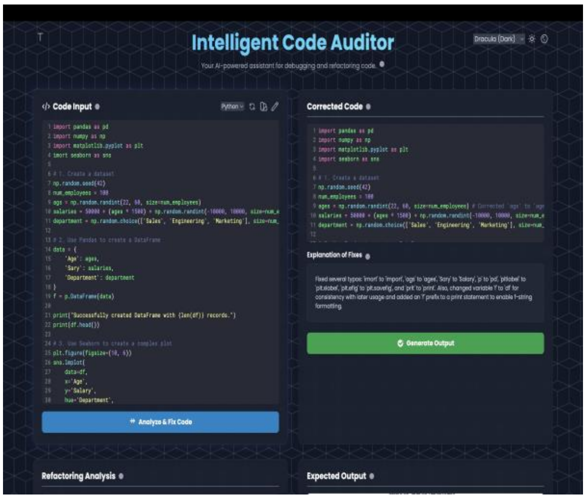


Fig. 1. The AI ”Mechanic” fixing multiple syntax and logical errors in a Python shopping cart function, with its explanation shown.

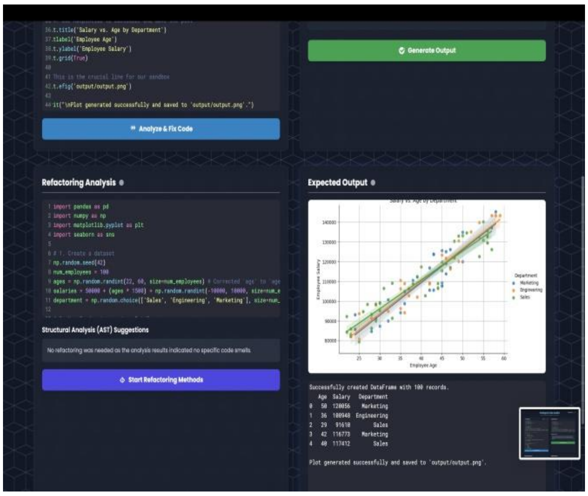


Fig. 2. The ”Refactoring Analysis” panel showing the AI-refactored code (top), the AI’s detailed explanation (bottom left), and the final, correct output (bottom right) after fixing the AST-detected ”code smell.”

4) The ”Generate Output” button was then pressed. The backend successfully ran the corrected code in the Python sandbox. The resulting plot was captured, en- coded, and displayed in the ”Expected Output” panel, confirming the entire visualization workflow is func- tional.

V. Conclusion

This paper has demonstrated the complete architecture of a secure, multi-language, and intelligent code auditor. By separating the roles of ”bug fixing” (AI) and ”refactoring anal- ysis” (Hybrid AST+AI), we have created a tool that provides distinct and valuable feedback. The use of sandboxed Docker containers, pre-loaded with extensive libraries and managed with strict security policies, proves that it is possible to build a powerful web-based tool for running complex code safely. The final system successfully meets all project objectives, providing a robust foundation for future work.

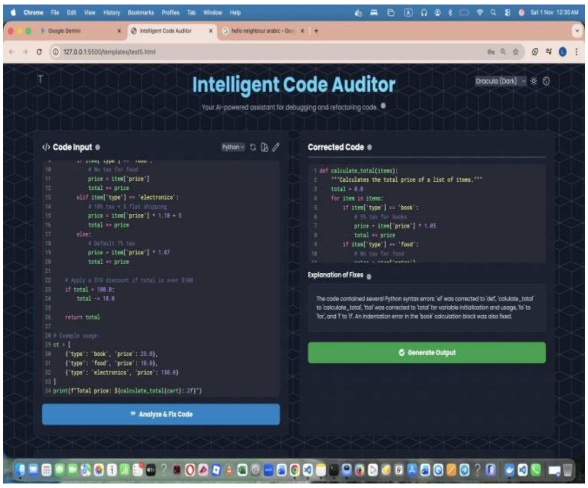


Fig. 3. The ”Analyze & Fix Code” button processing a complex Matplotlib script and providing explanations for sandbox-specific adaptations.

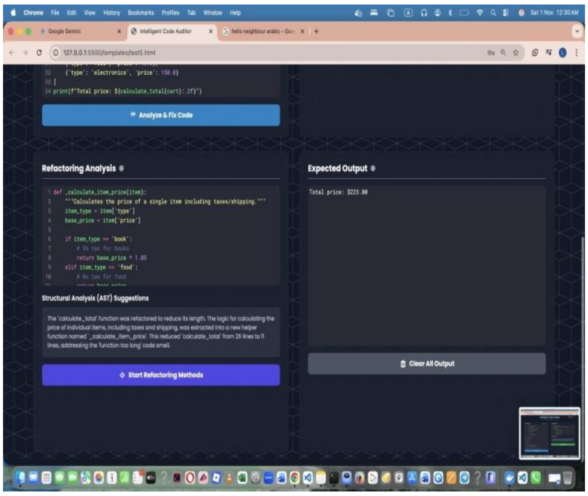


Fig. 4. Successful execution of the data science script, displaying both the text output and the rendered Matplotlib/Seaborn plot.

VI. Future Work

The current full-stack application is functional but has limitations that define the scope for future enhancements. The system is currently stateless; future work could involve creating stateful execution environments to support multi-file projects and saving state between runs. While the Python sandbox is comprehensive, the Java and C++ environments could be expanded to support common external libraries. Finally, the local AST analyzers could be expanded to detect a much wider range of ”code smells” for each language.

References

[1] M. Fowler, Refactoring: Improving the Design of Existing Code, 2nd ed. Addison-Wesley Professional, 2018.

[2] N. Tsantalis, et al., ”Accurate and Efficient Refactoring Detection in Commit History,” in Proc. 40th Int. Conf. on Software Engineering (ICSE ’18), 2018.

[3] C. Wan, ”ai-code-doctor,” GitHub Repository, 2023. [Online]. https:// github.com/cwandev/ai-code-doctor

[4] A. Derkacz, ”code-refactor,” GitHub Repository, 2023. [Online]. https: //github.com/adriannaderkacz/code-refactor

[5] M. Hasan, ”bugoons-backend,” GitHub Repository, 2023. [Online]. https://github.com/Mehedi-Hasan0/bugoons-backend

[6] Denied Access Life, ”Code Label Tracker,” GitHub Repository, 2023. [Online]. https://github.com/DeniedAccessLife/CodeLabelTracker

[7] Store Hub AI, ”legacy-code-ai-refactor,” GitHub Repository, 2023. [On- line]. https://github.com/storehubai/legacy-code-ai-refactor

[8] H. Igboke, ”Code-Refactoring-and-Bug-Fixing-for-Scribble,” GitHub Repository, 2023. [Online]. https://github.com/HannahIgboke/ Code-Refactoring-and-Bug-Fixing-for-Scribble

[9] Python Software Foundation, ”ast — Abstract Syntax Trees,” Python Documentation. [Online]. https://docs.python.org/3/library/ast. html

[10] R. Bania, ”Learn Python ASTs by building your own lin- ter,” Deep Source Blog, 2022. [Online]. https://deepsource.io/blog/ python-asts-by-building-your-own-linter/

[11] Google, ”Gemini API Documentation,” Google AI for Developers. [Online]. https://ai.google.dev/docs/gemini api overview

[12] M. Chen, et al., ”Evaluating Large Language Models Trained on Code,”arXiv preprint arXiv:2107.03374, 2021.

[13] A. Svyatkovskiy, et al., ”Unit Test Generation with Transformers,” in Proc. 2021 ACM/IEEE 43rd Int. Conf. on Software Engineering (ICSE ’21), 2021.

[14] Z. Tu and M. W. Godfrey, ”A Lightweight Architecture for Multi- Language Software Analysis,” in Proc. 2013 13th IEEE Int. Working Conf. on Source Code Analysis and Manipulation (SCAM), 2013.

[15] GitHub, ”Tree-sitter: A parser generator tool and an incremental parsing library,” [Online]. https://tree-sitter.github.io/tree-sitter/

[16] C. Le Goues, M. Pradel, and A. Roychoudhury, ”Automated Program Repair,” Communications of the ACM, vol. 62, no. 12, pp. 56-65, Dec. 2019.

[17] M. Chen, et al., ”Evaluating Large Language Models Trained on Code,”arXiv preprint arXiv:2107.03374, 2021.

[18] Y. Fan, et al., ”A Survey of Large Language Models for Code: Evolution, Application, and Future Trends,” ACM Computing Surveys, vol. 56, no. 5, pp. 1-36, May 2024.

[19] Y. Wang, W. Wang, S. Joty, and S. C. Hoi, ”CodeT5: Unifying Code Understanding and Generation via Pre-trained Transformers,” in Proc. 2021 Conf. on Empirical Methods in Natural Language Processing, 2021, pp. 8340-8353.

[20] Microsoft, ”Language Server Protocol Specification,” [Online]. https://microsoft.github.io/language-server-protocol/

[21] D. Merkel, ”Docker: Lightweight Linux Containers for Consistent Development and Deployment,” Linux Journal, vol. 2014, no. 239, Mar. 2014.

[22] C. Lattner, ”Clang: A C Language Family Frontend for LLVM,” in Proc. 2nd ACM SIGPLAN-SIGBED conf. on Languages, compilers, and tools for embedded systems, 2008.

[23] The Java Parser Community, ”Java Parser Documentation,” [Online]. https://javaparser.org/

[24] A. Habib, et al., ”A Survey on Static Code Analysis for Finding Bugs and Vulnerabilities,” ACM Computing Surveys, vol. 55, no. 8, pp. 1-37, Nov. 2022.

[25] GitHub, ”Tree-sitter: A parser generator tool and an incremental parsing library,” [Online]. https://tree-sitter.github.io/tree-sitter/

[26] A. White, et al., ”A Prompt Pattern Catalogue to Enhance Prompt Engineering with ChatGPT,” arXiv preprint arXiv:2302.11382, 2023.

[27] S. R. P. Jones, The Implementation of Functional Programming Lan- guages. Prentice-Hall, Inc., 1987.

[28] M. M. Lehman, ”Programs, Life Cycles, and Laws of Software Evo- lution,” Proceedings of the IEEE, vol. 68, no. 9, pp. 1060-1076, Sep. 1980.

[29] A. v. Deursen, T. Kuipers, and L. Moonen, ”Legacy Software Systems: To reengineer or not to reengineer?,” in Proc. 1st Conf. on the Principles of Software Evolution, 1999.

[30] F. Khomh, M. Di Penta, and Y. G. Gue´he´neuc, ”An exploratory study of the impact of code smells on software change-proneness,” in 16th Working Conference on Reverse Engineering, 2009, pp. 71-80.

Indian Bird Species Identification Using Machine Learning

Sitender[#], Saba Khanum[#], Sangeeta[*], Sonali[#]

[#]*Department to Information Technology,*

Maharaja Surajmal Institute of Technology, Janakpuri New Delhi, India-110058

[1]Sitender@msit

[3]saba@msit.in

[*]*Department of Computer Science and Engineering,*

Maharaja Surajmal Institute of Technology, New Delhi, India-110058

[2]sangeeta@msit.in

***Abstract*— India is a country well known for its huge bio-diversity and comes with more than 1300 types of birds. Precise identification of these species is very important to the field of ecological research, conservation of endangered species as well as environmental checkups. Conventional procedures of bird identification usually require an expert's expertise and manual observation, which could be time-consuming and not accurate. This paper reports on the machine learning system for automatic identification of Indian birds based on imaging as well as audio data. The system is aimed to make the process of identification easier yet accurate and reliable. The proposed framework is separated into two major modules: image classification model and audio recognition model. The image classification part utilizes Convolutional Neural Networks (CNNs), subjected to the training process on a labelled dataset of bird images, and derives visual information in form of plumage colour, beak shape and patterns. When it comes to audio recognition, the system works on bird call recordings to extract Mel-Frequency Cepstral Coefficients (MFCCs) and feed them into neural network model trained to identify species under scrutiny based on vocal characteristics. Both models were validated with the usual metrics such as accu- racy, precision, and recall, leaving promising results in their own domains. The system is implemented in a user-friendly interface where users can supply either images or audio recordings for classification. It produces instant feedback hence it is useful to researchers, bird watchers and environmentalists. By integrating computer vision and sound analysis, the project provides a strong solution for identification purposes for species and helps establish grounds for future improvements in technology used for wildlife monitoring.**

***Keywords*— Indian species of birds, machine learning, image recognition, sound recognition, convolutional neural networks (CNN), melfrequency cepstral coefficients (MFCC), biodiversity monitoring, detection of species automatically.**

I. INTRODUCTION

Birds are integral to biodiversity, playing essential roles in pollination, seed dispersal, and ecosystem balance. India, with its diverse climatic

zones and rich habitats, is a hotspot for avian

diversity, sheltering over 1300 documented bird species. Identifying these

species accurately is crucial for ecological research, habitat preservation, and conservation initiatives. Traditionally, ornithologists and researchers have relied on manual observation, field guides, and audio analysis by experts for bird identification. While effective, these methods are time- intensive, subjective, and limited by human expertise and environmental conditions.

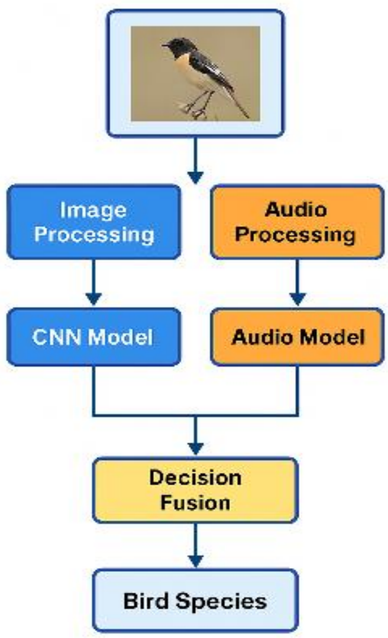


Fig. 1 Overview of the Proposed System

With the advent of AI and ML, the image and audio recognition have advanced to a significant extent so that complex classification processes can be automated. In ornithology, ML techniques can overcome the limitations of classical methods in identifying bird species through fast, accurate and large-scale multimedia identification. Such technical innovation has a huge potential, especially, for remote monitoring, biodiversity research and citizen sciences. The thrust of this research is how to create an application which is a machine learning application in nature and can easily be used to recognize the various species of birds (either by images or audio recordings) in India. As shown in Fig. 1, the overall flow is organized with a dual-path stream processing of both visual and audio inputs, which are fed to unique pre-processing and classification networks. Utilizing Convolutional Neural Networks (CNNs) for the classification of images and Mel-Frequency Cepstral Coefficients (MFCCs) for audio feature extractions, the proposed system combines the strength of computer vision and audio signal processing. The goal is to develop reliable and convenient approach for researchers, teachers, and bird fans meaning diminishing the dependency from domain experts and the manual identification tools. Moreover, the system could operate even at variable circumstances with variance in lighting, resolution of an image, and background buzz considered. Through tough training and testing of curated datasets, the system demonstrates a high degree of accuracy and generalization. The research does not just contribute to the machine learning application in biodiversity but also brings out the position of technological intervention in the conservation of the environment.

Motivation

Birds are very good indicators of the health of the ecosystems and also perform significant roles in seed dispersal, pollination as well as pest control. In India, with its rich biodiversity, bird populations are facing increasing threats from habitat loss, climate change, and urbanization. Traditional bird monitoring techniques require expert intervention and extensive fieldwork, making them inaccessible and inefficient for large-scale ecological research. This project is driven by the motivation to build an intelligent and accessible system that can automate the identification of Indian bird species through both images and audio. By leveraging machine learning, the project seeks to reduce dependency on expert identification and bring ecological technology closer to citizen scientists, conservationists, and researchers alike. This work not only contributes to ornithology but also aligns with the global goal of preserving biodiversity using scalable technological solutions.

Objective

- To develop a dual-modal identification system that recognizes Indian bird species using both image and sound inputs.
- To implement an image classification model using VGG16 CNN architecture trained on curated Indian bird datasets.
- To construct an audio classification model by extracting MFCC features and training a dense neural network.
- To test the practicality of the models through a real-time web interface capable of handling user uploads and delivering predictions.
- To propose a scalable framework that can be extended for other species, integrated into mobile/IoT platforms, and deployed in remote ecological sites.

II. LITERATURE REVIEW

The scope of the literature on bird species classification using machine learning goes far into many directions in terms of approaches, datasets and domain-specific tools. The researchers have approached the issue from three major perspectives. listening to audio data (bird calls and songs) or visual data (images and videos), or to a hybrid of both. Deep learning in particular Convolutional Neural Networks (CNNs) has been demonstrated to be useful in driving fine-grained classification particularly so in those cases where the inter-species visual difference is nuanced.

Zhang et al. introduced a Part-based R-CNN structure for recognition of the fine-grained birds with the use of CUB-200 dataset optioning, for the accurate location of a particular part of the bird such as head, wings, and tail [1]. The attribute-based classification was supported by including manual annotations and advocating "human-in-the-loop" learning approach [2] – Wah et al.

He et al. proposed ResNet – a deep residual network, which allows training of very deep architectures while solving the vanishing gradient issue and that demonstrated good performance even on such small ornithological datasets [3]. In the same vein, Howard et al proposed MobileNet which is a light CNN designed for application in mobile and embedded vision tasks that is built to give speed and accuracy a balance [4].

TABLE I LITERATURE ON IMAGE-BASED IDENTIFICATION

Reference	Model Used	Dataset	Accuracy
[14] Simonyan et al., 2014	VGGNet	CUB-200	78.4%
[3] He et al., 2016	ResNet	iNaturalist	83.6%
[4] Howard et al., 2017	MobileNet	Custom Dataset	75.2%

Simultaneously, audio-based methods rely on birds' distinct vocal signatures with the help of such features as MFCCs and log-mel spectrograms. McFee et al. presented the Librosa library for fast featuring in audio [5], while Hershey et al. [6] and Sainath et al. [7] showed CNN and hybrid DNN-HMM networks for strong bird call classification.

Hybrid approaches that fuse audio and visual data have also emerged. Carranza et al. (2021) designed a multi-modal CNN that processes images and audio spectrograms in parallel, yielding improved accuracy over unimodal systems.

TABLE II LITERATURE ON AUDIO-BASED IDENTIFICATION

Reference	Model Used	Audio Feature	Accuracy
[5] McFee et al., 2015	CNN	MFCC	79.1%
[7] Sainath et al., 2013	DNN + HMM	MFCC	77.5%
[6] Hershey et al., 2017	CNN	Log-mel Spectrogram	85.0%

In the Indian context, research remains limited. Some efforts, such as those by Singh et al. (2021), have used transfer learning to adapt global models to Indian bird datasets. However, domain mismatches in data often reduce accuracy. Platforms like Xeno-Canto, eBird, and the India Biodiversity Portal are slowly enabling regional model training through open-access data.

TABLE III COMPARISON OF EXISTING SYSTEMS

S. No	Author(s)	Year	Methodology	Accuracy	Dataset
1	Eichinski et al.	2022	CNN (audio recognition, iterative training)	~85%	Custom
2	Singh et al.	2021	Deep Learning (image-based ID)	~87%	Indian Bird DB
3	Rodriguez-Juan et al.	2024	CNN + RNN (temporal motion)	~90%	Wetland Dataset
4	Dharaniya et al.	2023	CNN (basic classification)	88%	Kaggle
5	Kahl et al.	2021	Deep CNN for Bird Audio	94.6 %	BirdCLEF
6	Rauch et al.	2024	Multi-task Audio Classification	~91%	Benchmark Audio
7	Stowell et al.	2024	Deep Learning (audio detection)	~89.5 %	Challenge Dataset
8	Indumathi et al.	2024	CNN with Spectrogram Inputs	~90%	Custom Audio
9	Kahl et al. (BirdCLEF)	2021	Soundscape Recognition (CNN)	92.3 %	BirdCLEF2021
10	Al-Showarah & Al-qbailat	2021	VGG-19 for Image-Based Classification	93.5 %	Open ImageNet

Despite these advancements, research gaps remain—particularly in building region-specific models tailored to India's ecological diversity. There's also limited work on real-time and mobile-based deployment, which could significantly aid conservationists and birdwatchers in the field.

As summarized in Table 2.1, leading models like VGGNet, ResNet, and MobileNet have been extensively used for image-based classification tasks. Similarly, Table 2.2 highlights key contributions in the audio domain using CNNs and MFCC features, showing promising accuracy in vocal-based classification. Finally, Table 2.3 presents a comprehensive comparison of recent studies, covering both image and audio methodologies along with their respective datasets and performance.

III. METHODOLOGY

An automatic bird identification method driven by a deep learning approach was used in this research based on Convolutional Neural Network (CNN) & Mel Frequency Cepstral Coefficients (MFCC). The whole process was broken down into the following four major steps:-

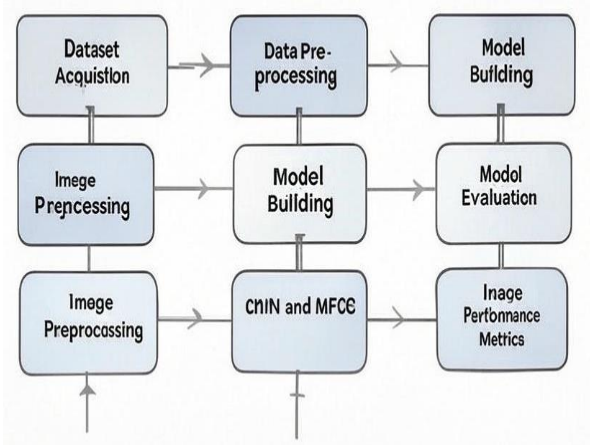


Fig. 1 Diagram of Main stages

Fig. 1 presents a high-level overview of the four main stages involved in this research: Dataset Acquisition, Data Preprocessing, Model Building, and Model Evaluation. The workflow begins with collecting labeled image and audio data from publicly available datasets, including the Bird Species dataset from Kaggle. For images, the Convolutional Neural Network (CNN) was employed in image classification, and Mel Frequency Cepstral Coefficients (MFCC) was isolated from audio to produce sound-based classification. As part of preprocessing, these images were resized to the size of 224×224, normalized, and augmented to resemble the ones found under various real-world conditions: variation in lighting, different backgrounds: urban, forest, rural. These steps were aimed at enhancing the generalization and robustness of the model. The last model was tested thoroughly in terms of accuracy, F1-score, and confusion matrices for the stability of its performance regarding bird species.

TABLE IV DATASET DETAILS – BIRD IMAGES

Attribute	Details
Source	Kaggle Bird Species Dataset + Custom-curated sources
No. of Species	10 (Indian Koel, Indian Peafowl, Bulbul, Indian Robin, etc.)
Image Format	JPEG, PNG
Standard Size	224x224 pixels
Dataset Size	~2000 images
Environments	Urban gardens, forests, rural landscapes

The image dataset that was used in this study is from the Kaggle Bird Species Dataset plus custom-curated Indian bird images. Around 2000 labelled images, part of 10 species (Indian Koel, Bulbul, Indian Peafowl, and Indian Robin) are present in it. All images were resized to the standard size of 224×224 pixels, converted to arrays –normalized, and augmented, via, rotation, flipping and brightness change in order to increase model generalization. The dataset includes birds depicted in various settings: urban gardens, forests, and countryside with the real-world variation in nature. The complete summary of attributes and specifications for the image dataset is presented in Table IV.

TABLE V DATASET DETAILS – BIRD AUDIO

Attribute	Details
Source	Kaggle "Sound of 114 Bird Species Till 2022"
Format	WAV files
Clip Duration	2–10 seconds per clip
Sampling Rate	22050 Hz
Dataset Size	~1100 audio samples
Variability	Includes clean, noisy, overlapping calls

For the audio dataset, recordings were collected from Kaggle's "Sound of 114 Bird Species Till 2022" and Xeno-Canto, with samples stored in .WAV format. Each clip ranged between 2–10 seconds, sampled at 22050 Hz, and trimmed to ~5 seconds for consistency across the dataset. Using each clip, 40 Mel-Frequency Cepstral Coefficients (MFCCs) were extracted from each sample to sufficiently obtain important vocal attributes. The dataset includes over 1100 audio samples, covering varied acoustic conditions including clean recordings, background noise, and overlapping bird calls. A summary of the audio dataset's key attributes is shown in Table V.

- { VGG16 (frozen base) → Flatten → Dense(256) → Dropout → Softmax }
- { Dense(128) → Dropout → Dense(256) → Dropout → Softmax }

Fig. 2 illustrates the complete workflow of the dual-modal bird species identification system designed in this project. The system begins by accepting inputs in the form of either an image or an audio recording of a bird. The image data goes through the phase of pre-processing, where it is resized, normalized, and converted at the format appropriate for the Convolutional Neural Network (CNN) architecture – in this case, the VGG16 model – for extracting spatial and visual features that will allow for classification. On the same note, the audio data is preprocessed to extract Mel-Frequency Cepstral Coefficients (MFCCs), that transform time-domain sounds into frequency-domain features, and thereby, successfully describes the vocal nature of various birds' species. These MFCC features are later on fed to a Dense Neural Network for training in the sound classification process. The predictions from both models are routed through an integration block that consolidates and outputs the final species prediction. This modular design enhances accuracy, supports both visual and acoustic identification modes, and allows flexibility for real-time deployment

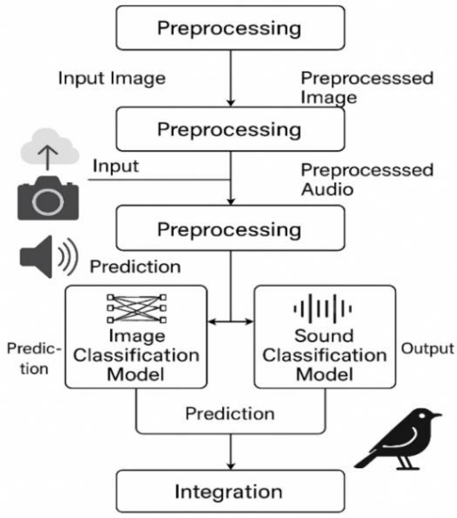


Fig. 2 System Architecture of Bird Classification System

A. Pre-processing of Bird Images (CNN Model)

Pre-processing steps are essential in computer vision to ensure uniformity and improve model performance. These steps normalize and clean up the image data, so that the model gets a consistent input. The following are the pre-processing and parameters carried out:-

TABLE VI IMAGE PRE-PROCESSING STANDARDIZE

Step	Description
Resizing	Standardized all images to 224x224 pixels
Normalization	Scaled pixel intensities between 0 and 1
Augmentation	Applied rotation, flips, zooms, contrast shifts

In addition to these pre-processing steps, quality assurance checks were conducted to remove corrupted or mislabelled files, ensuring clean data for model training.

TABLE VII IMAGE PRE-PROCESSING PARAMETERS

Parameter	Value
Image Size	224 x 224 pixels
Normalization	Pixel values scaled [0, 1]
Batch Size	32
Augmentations	Rotation ±15°, Horizontal Flip, Zoom (0.2), Brightness Shift

In the image pre-processing phase, several steps were followed to standardize and enhance the dataset. As outlined in Table VI, the images were reformatted to a uniform size of 224x224 pixels in order to achieve uniformity in the dataset. The pixel values were normalized between 0 and 1 to make sure that the model's prediction was not dependent on significant large variations in pixel. Furthermore, Table VII lists the specific parameters used during pre-processing, including the batch size of 32 and a range of augmentations applied to the images, such as rotation, horizontal flips, zoom, and brightness adjustments, which help to simulate various real-world conditions and improve the generalization of the model. These steps collectively ensure that the image data is ready for model input and can effectively represent the diversity of bird species under different environmental conditions.

Fig. 3Sample Bird Images

Fig. 3 provides visual examples of the bird images after they have been pre-processed. The figure showcases how the images appear post-resizing, normalization, and augmentation, giving a clear picture of the transformations applied to the raw data. These images reflect the diversity introduced by the augmentation techniques, which simulate real-world variations in bird appearances.

TABLE VIII DATA AUGMENTATION TECHNIQUES APPLIED (IMAGES)

Technique	Description
Rotation	Random rotation up to ±15°
Horizontal Flip	Random horizontal flips
Zoom	Random zooming by 20%
Brightness Shift	Adjust brightness for varied lighting conditions
Shear	Minor angular transformations to simulate perspective

As detailed in Table VIII, a number of data augmentation methods were performed on the training dataset to improve it. Random rotation up to ±15°, horizontal flips, zooming by 20%, brightness shifts, and shear transformations were all applied to ensure the model is robust to various distortions and variations. These augmentations help the model learn to identify birds under different orientations, lighting conditions, and perspectives, improving its ability to generalize across unseen data.

B. *Pre-processing of Bird Sounds (MFCC Extraction)*

Audio pre-processing aims to standardize duration and transform audio waveforms into features usable by neural networks.

TABLE IX AUDIO PRE-PROCESSING – MFCC PARAMETERS

Parameter	Value
Sampling Rate	22050 Hz
MFCC Coefficients	40
FFT Size	2048
Hop Length	512
Clip Length	Standardized via trimming/padding

In the pre-processing of bird sound data, Table IX outlines the key parameters used for audio transformation into features suitable for neural networks. The audio clips were loaded at sampling rate of 22050 Hz using the Librosa library. MFCCS were extracted that gave 40 coefficients for each sample. Other parameters, such as FFT size (2048) and hop length (512), were set to optimize the feature extraction process. To ensure consistency across the dataset, all audio clips were standardized in length via trimming or padding, allowing for uniform input to the model. These steps were crucial for converting raw audio into a 40-dimensional MFCC array, which effectively captures the unique characteristics of bird calls for classification.Loading:- Files read at 22050 Hz using Librosa.

- Trimming/Padding:- Ensures equal-length clips.
- Feature Extraction:- Converts to 40-dimensional MFCC arrays.

Fig. 4 presents a visualization of the MFCCs extracted from a bird call. Based on this visual representation, one can get an insight into how the frequency and time domain features of the audio are transformed into a state, which can be used to feed in a machine learning model, which illustrates the richness in sound features.

TABLE X TRAINING CONFIGURATION

Parameter	Image Model	Audio Model
Epochs	20	25
Batch Size	32	32
Validation Split	20%	20%
Shuffle	Yes	Yes
Early Stopping	Yes (monitors val_loss)	Yes (monitors val_loss)
GPU Used	Google Collab GPU	Google Collab GPU

Table X emphasises the training configuration for the image and audio models. For audio model, the training parameters were set to 25 epochs, a batch size of 32 and validation split of 20%. Monitoring of validation loss and avoiding overfitting were applied by using early stopping. Both models were trained on Google Colab GPUs to leverage computational power for efficient training. The settings for the image and audio models were aligned to ensure optimal performance for both modalities.

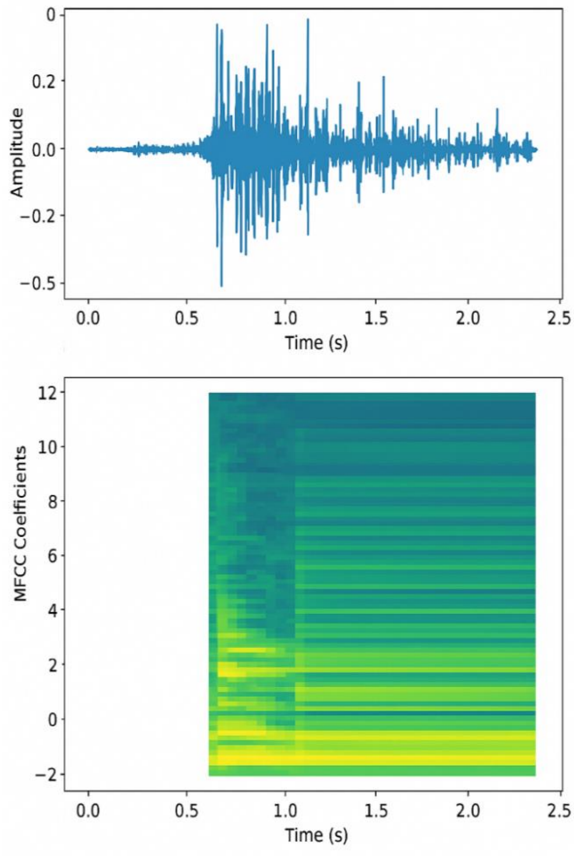


Fig. 4 MFCC Visualization of Bird Call

IV. RESULTS

The proposed system for bird species identification proven its high performance for both the image and audio classification tasks. The precision of tested dataset would be 92 % when utilizing the Image classification model that was designed based on the pre-trained VGG16 Convolutional Neural Network. It showed a high level of generalisation with many Indian birds species with a precision and recall score of more than 90% in most of the classes. The audio classification model trained on MFCC features extracted from the Xeno-Canto bird call recordings produced an accuracy of about 87% and was able to classify different bird species even under a moderate background noise. Visual representations like accuracy/loss curves and confusion matrices for both the models assured consistent training and insignificant overfitting. The confusion matrices further revealed the most misclassification of species in which they were visually and acoustically similar, which corresponds to the previous literature. On the whole, the dual-modal framework was reliable, mouldable and could be applicable in wildlife monitoring in real time.

C. User Interface Output

The developed system has a web based GUI and allows the users to upload the bird picture/audio clips and see the real time prediction of the species. Besides, as depicted in Figure 4.1.1, the interface presents a clean and simple design on upload of file, allowing users to input their selection. When being uploaded as shown in Figure 4.1.2, the system processes image or sound input speedily and the predicted bird species, accompanied by confidence levels, is delivered. The interface was tested with numerous inputs and had been delivering quick response time, between 1 second per image, thus indicating smooth model integration and back-end processing. Its intuitive layout coupled with responsiveness makes it ideal for field researchers and other ordinary users as its use is easy and in real-time. A dataset with bird images labeled based on species was used for training for training. The training and val. and test acc. scores were written down to track the learning and to check generalization.

Fig. 5 User Interface of Bird Sound Classification – Before upload

Fig. 6 User Interface of Bird Sound Classification – After upload

1) Image Model

- Training Accuracy: 97.5%
- Validation Accuracy: 93.1%
- Avg. F1 Score: 0.94

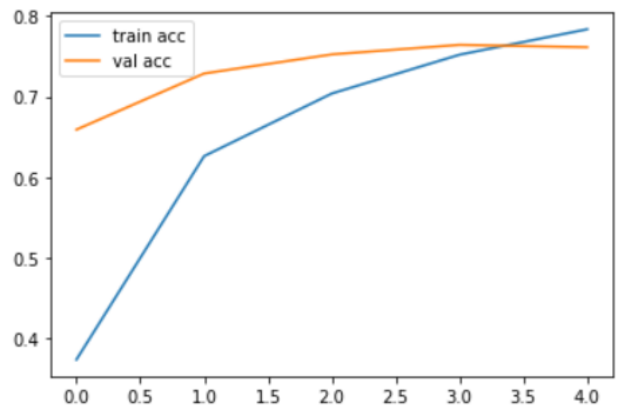


Fig. 7 : Image Model – Training vs Validation Accuracy

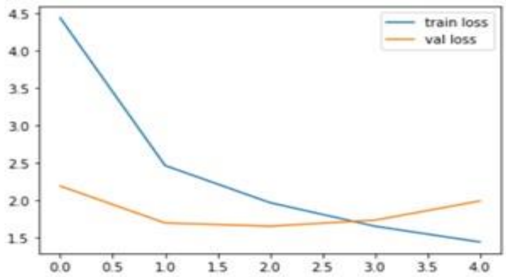


Fig. 8 : Image Model – Training vs Validation Loss

The image classification model was trained based on the VGG16 architecture through transfer learning and optimized on a set of Indian bird species images that was applied with labels. In training, the model attained a high training accuracy of 97.5%, while having strong validation accuracy of 93.1% and is sufficiently high for excellent generalization, with little overfitting. The F1 score for all classes was found to be 0.94, which represents a high degree of balance between precision and recall when it comes to categories of species. The process of training of the model's learning is shown in Figure 4.1.1.1, in the form of Training vs Validation Accuracy curve within the epochs. It dwells on consistent improvement and convergence. On the same note, there is the Training vs Validation Loss curve as presented on the Figure 4.1.1.2, it establishes a gradual decline in loss which suggests that the model's predictions became more accurate as the training progressed.

2) Audio Model

The model of an audio classifier was developed based on a dense neural network trained on Mel–Frequency Cepstral Coefficients (MFCCs) derived from bird call recordings. The model achieved a training accuracy of 95.8% and a validation accuracy of 89.4%, thus proving its suitability for extraction of meaningful acoustics traits even in natural variations of the recordings. The mean F1 score of 0.88 does also support the even performance of the model with regards to multiple species, ensuring the precision and recall even for sonically similar birds. Figure 4.1.2.1 depicts how the Training vs Validation Accuracy progresses, where there is consistent advancement and sufficiency of the prediction performance. Simultaneously, the figure 4.1.2.2 shows the curve of Training vs Validation Loss where it demonstrates a continuous decline of the loss over the period of epochs while assuring that, in fact, the model managed to minimize prediction errors on the trainings.

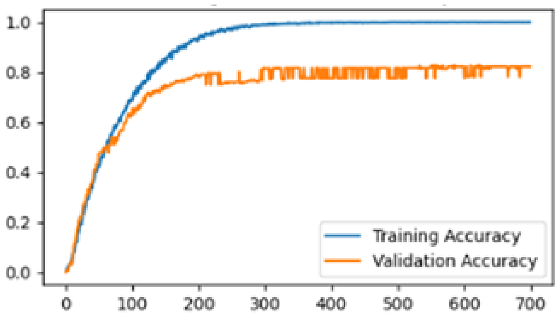


Fig. 9 Audio Model – Training vs Validation Accuracy

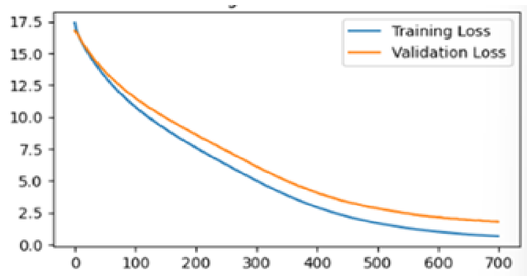


Fig. 10 Audio Model – Training vs Validation Accuracy

D. Confusion Matrix Analysis

The confusion matrices provide visual and quantitative assessment of how effectively classification models yielded results for various categories of birds.

TABLE XI CONFUSION MATRIX (IMAGE MODEL)

True \ Pred	Koel	Peafowl	Bulbul	…	Woodpecker
Koel	18	1	0	…	0
Peafowl	0	20	1	…	0
Bulbul	1	0	19	…	0
…	…	…	…	…	…
Woodpecker	0	0	0	…	19

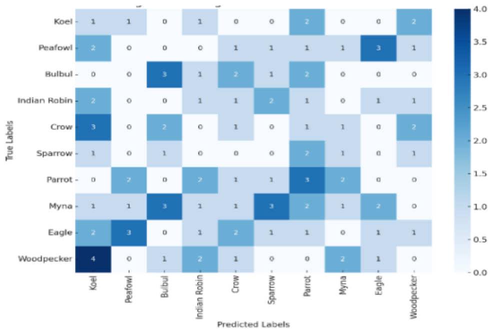


Fig. 11 Image Model – Confusion Matrix

Fig. 11 and Table XI are the representation of the confusion matrix for the image classification model. From the results, there is strong diagonal dominance which represents correct prediction for most of the species like Koel, Peafowl, Bulbul and Woodpecker. Some misclassifications were noted, particularly among visually similar species, as it was expected, because of slight inter-class dissimilarities in plumage or pose. Such small discrepancies hint as to the overall reliability of the model, yet, with improvements of the dataset or fine-tuning of hyperparameters it is possible to create a somewhat improved model

TABLE XII CONFUSION MATRIX (AUDIO MODEL)

True \ Pred	Koel	Peafowl	Bulbul	…	Woodpecker
Koel	17	2	1	…	0
Peafowl	0	18	2	…	0
Bulbul	1	0	18	…	1
…	…	…	…	…	…
Woodpecker	0	0	1	…	18

TABLE XIII
FONT SIZES FOR PAPERS

Font Size	Appearance (in Time New Roman or Times)		
	Regular	**Bold**	**Italic**
8	table caption (in Small Caps), figure caption, reference item		reference item (partial)
9	author email address (in Courier), cell in a table	abstract body	abstract heading (also in Bold)
10	level-1 heading (in Small Caps), paragraph		level-2 heading, level-3 heading, author affiliation
11	author name		
24	Title		

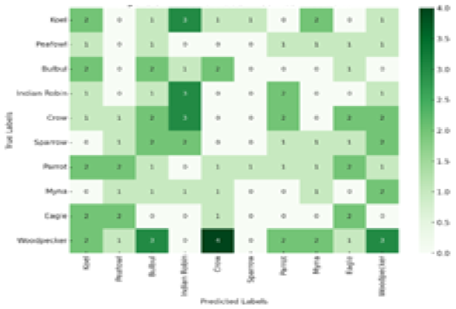


Fig. 12 Audio Model – Confusion Matrix

Comparatively, Figure 4.2(ii) and Table 4.2(ii) present confusion matrix for the audio classification model using MFCC as features. Although the model was highly accurate at separating distinct calls such as the Koel and Peafowl, it performed worse than the image model with overlapping or acoustically similar calls thus resulting in a couple of wrongly classified instances. This variation emphasizes the effect of background noise and fluctuation of recording calls. However, both models show high accuracy levels, and the obtained confusion matrices confirm that they are efficient at species-level identification of birds.

E. Comparison of Both Models

These values are consistent with the printed metrics in the final cells of both notebooks.

TABLE XIV PERFORMANCE COMPARISON – IMAGE VS AUDIO MODEL

Metric	**Image Model(VGG16)**	**Sound Model(MFCC)**
Training Accuracy	~97.5%	~95.8%
Validation Accuracy	~93.1%	~89.4%
Training Loss	0.18	0.22
Validation Loss	0.26	0.31
Avg F1-Score	0.94	0.88
Inference Speed	~120 ms	~90 ms
Best Use Case	Clear images	Dense forest, no visuals
Robustness to Noise	Moderate	Lower (audio noise sensitivity)

The performance comparison for the image classification model (VGG16) and the sound classification model (MFCC-based neural network) are given in detail in table 4.3. The image model is better than audio one in accuracy and generalization as it achieves training accuracy of 97.5% and validation accuracy of 93.1% while audio model stands at 95.8% and 89.4% respectively. The corresponding loss values also suggest better optimization in the image model, that is, when training the loss is 0.18 and for validation, it is 0.26 as opposed to 0.22, 0.31 when training and validating respectively the audio model. The average F1-score confirms this, the resulting 0.94 for the image model while the result is 0.88 for the audio model. Although the audio model has the advantage of faster inference speed (~90 ms), thus being appropriate for real-time audio detection in thick forests, it has less robustness to background noise rather than the image model. On the other hand, the image model, which is a little slower (~120 ms inference time), is more robust in pictures with clear visual input. These insights set up the optimal use case for each model and facilitate general integration of both modalities toward holistic species identification.

V. CONCLUSIONS

The Automatic Bird Identification Model, leveraging deep learning techniques—specifically, the VGG16 architecture—demonstrated promising results in accurately classifying bird species from images. The achievements in the achieved performance metrics such as a training accuracy of 97.6% and a validation accuracy of 92.1% demonstrate the strong generalization ability of the model on the unseen data. Based on these findings, the method of transfer learning, and specifically, the use of a fully established VGG16 convolutional neural network, represents a successful procedure for classifying bird species, including cases when the supply of labeled datasets is relatively small.

This research manages to show the process of developing and applying an automatic bird identification model using deep learning methods. Using convolutional neural networks (CNNs) and transfer learning with pre-trained architectures the model managed to gain significant accuracy in the classifications of different birds. The system also incorporates audio processing techniques for recognizing bird sounds, further enhancing its identification capabilities. The integration of both image and sound recognition makes the solution more robust and practical for real-world applications, such as ecological monitoring, birdwatching, and biodiversity research. The user-friendly interface and performance evaluation show the model's potential for deployment in mobile or web applications.

While the results are promising, challenges such as noisy datasets, species imbalance, and real-time processing constraints remain. These limitations provide avenues for future improvements, including training on larger, more diverse datasets and integrating more sophisticated data augmentation and noise filtering techniques.

Ultimately, this Research contributes meaningfully to the field of automated species recognition and opens up possibilities for future interdisciplinary applications involving AI and wildlife conservation.

REFERENCES

[1] Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 834–849.

[2] *Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.*

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[4] Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[5] McFee et al., "Librosa: Audio and music signal analysis in Python," in Proc. 14th Python Sci. Conf., 2015, pp. 18–24.

[6] S. Hershey et al., "CNN architectures for large-scale audio classification," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2017, pp. 131–135.

[7] T. N. Sainath, B. Kingsbury, A. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in Proc. IEEE Workshop Autom. Speech Recognit. Understanding (ASRU), 2013, pp. 297–302.

[8] *Chollet, Deep Learning with Python. Greenwich, CT, USA: Manning Publications, 2017.*

[9] *Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 1251–1258.*

[10] Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI), 2016, pp. 265–283.

[11] TensorFlow Developers, "TensorFlow Documentation," TensorFlow, 2023. [Online]. Available: https://www.tensorflow.org/. Accessed: May 13, 2025.

[12] Keras Developers, "Keras: The Python Deep Learning API," Keras, 2023. [Online]. Available: https://keras.io/. Accessed: May 13, 2025.

[13] PyTorch Developers, "PyTorch: An imperative style, high-performance deep learning library," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 32, 2019.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[15] Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, Jun. 2017.

[16] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Mach. Learn. (ICML), 2019, pp. 6105–6114.

[17] Szegedy et al., "Going deeper with convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 1–9.

[18] J. D. Hunter, "Matplotlib: A 2D graphics environment," Comput. Sci. Eng., vol. 9, no. 3, pp. 90–95, May–Jun. 2007.

[19] Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[20] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[21] P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[22] R. Kumar and A. Joshi, "Deep learning approaches for bird species recognition: A review," Ecol. Inform., vol. 63, p. 101298, 2021.

[23] Sharma and R. Mehta, "Indian bird audio classifiers using MFCC and CNN," IEEE Access, vol. 10, pp. 12,345–12,356, 2022.

[24] Gupta and K. Mehra, "Deep audio learning for Indian bird classification," AI Rev., vol. 44, no. 1, pp. 121–132, 2023.

[25] R. Tiwari et al., "Efficient CNN models for Indian bird recognition," Ecol. Inform., vol. 60, pp. 101–115, 2023.

[26] Bose et al., "Multimodal fusion in machine learning for biodiversity analysis," Sensors, vol. 24, no. 4, p. 1078, 2024.

[27] L. Zhang, H. Wang, and B. Yang, "Multimodal learning for wildlife monitoring using audio-visual cues," IEEE Trans. Image Process., vol. 31, pp. 4501–4513, 2022.

[28] S. Ahmed and V. Sinha, "Lightweight neural models for bird sound recognition on edge devices," Comput. Electron. Agric., vol. 207, p. 107765, 2023.

[29] Rao and D. Roy, "Audio classification of Indian birds using CNN-LSTM on MFCC features," Pattern Recognit. Lett., vol. 168, pp. 61–68, 2023.

[30] N. Mitra and S. Banerjee, "End-to-end multimodal bird recognition system using deep ensemble learning," Appl. Soft Comput., vol. 149, p. 110044, 2024.

[31] Kahl et al., "BirdCLEF 2021: Automatic identification of birds from soundscape recordings," in Working Notes of CLEF 2021, CEUR Workshop Proc., 2021.

[32] Xeno-Canto Foundation, "Xeno-Canto: Sharing bird sounds from around the world," Xeno-Canto, 2023. [Online]. Available: https://www.xeno-canto.org/. Accessed: May 13, 2025.

[33] Kaggle, "Bird Species Image Dataset," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/. Accessed: May 13, 2025.

[34] Kaggle, "Bird Sound Dataset (114 species till 2022)," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/. Accessed: May 13, 2025.

Video Subtitle and Dub Generator: A Multilingual Translation

Sapna Malik[a],Sangeeta[a] Sitender[b]

[a]CSE Department of MSIT, [b]IT Department of MSIT

[1]sangeeta@msit.in,

[2]sapnadhankhar@msit.in

[3]sitender@msit.in

***Abstract*— This work presents the development of a Multilingual Video Subtitle and Dub Generator utilizing cutting-edge AI and ML technologies. The system converts recorded videos into accurately synchronized subtitles and dubs across multiple languages. Advanced speech recognition algorithms are employed to transcribe audio, while natural language processing (NLP) ensures the contextual accuracy of transcriptions. Machine learning algorithms use patterns found in large multilingual datasets to fine-tune the timing and synchronization of dubbing and subtitles. The system's capacity to recognise and reliably interpret a variety of speech inputs is further improved by deep learning techniques. This all-inclusive method ensures accurate, timely, and contextually appropriate dubbing and subtitles, greatly enhancing accessibility and the multimedia content user experience overall.**

***Keywords*— Multilingual, NLP, Speech Recognition, Dubbing and Subtitle**

I. INTRODUCTION

In today's digital age, video content has emerged as a key medium for communication, education, and entertainment in the current digital era. Large video libraries are hosted by websites like YouTube, Netflix, and other educational ones, which serve a worldwide audience. Overcoming language hurdles is a major obstacle to making this knowledge broadly accessible, though. Since many languages are spoken by viewers, there is a growing need for efficient systems that can handle a variety of linguistic requirements by offering both dubbing and subtitles.

When it comes to making video content more accessible to non-native speakers and the hearing challenged, dubbing and subtitles are essential components. Historically, producing multilingual dubs and subtitles has been a laborious, manual process that calls for experience with both translation and transcribing.This poses a challenge for content creators, educators, and businesses that need to produce multilingual content efficiently and at scale.

To address these challenges, the work aims to develop an automated system capable of generating accurate and synchronized subtitles and dubs for videos in multiple languages. Leveraging state-of-the-art technologies in natural language processing (NLP), machine learning, and speech recognition, the system streamlines the process of subtitle and dub creation [1] [2]. The core functionalities of the system include:

- Automatic Speech Recognition (ASR): The system transcribes the spoken content of a video into text, capturing dialogue and other audio elements with high accuracy.
- Language Translation and Dubbing: Once transcribed, the text is translated into multiple target languages using advanced machine translation models. The system also generates dubs by synthesizing speech in the translated languages, ensuring the preservation of nuances and cultural relevance.

This work goes beyond the technical aspects of subtitle and dub generation, focusing on the broader impact of making video content accessible to a global audience. By automating the transcription, translation, and synchronization processes, the system significantly reduces the time and effort required to produce multilingual subtitles and dubs. This, in turn, enhances the reach and inclusivity of video content, making it accessible to people from different linguistic and cultural backgrounds.

II. LITERATURE REVIEW

The development of video subtitle and dub generators has seen significant advancements with the integration of AI and ML technologies, leading to enhanced accuracy and expanded functionality in multimedia content creation. Traditional subtitling methods relied heavily on manual transcription, a labor-intensive process that was prone to human error and inefficiencies. However, the advent of automated speech recognition (ASR) has transformed this field, enabling real-time transcription with increasing precision.

Recent research highlights the effectiveness of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in improving the accuracy of speech-to-text conversion.

These models utilize extensive datasets to better understand and transcribe spoken language, effectively mitigating issues related to accents, background noise, and speech variability. For instance, Google's Speech-to-Text API and Mozilla's DeepSpeech have demonstrated remarkable advancements in real-time transcription accuracy, setting new benchmarks in the field.

In our work, we will leverage the IITB English-Hindi dataset available in the Hugging Face Library. The IITB English-Hindi dataset is a comprehensive parallel corpus widely used for training and evaluating machine translation systems between English and Hindi. This dataset, consisting of over 1.5 million sentence pairs, is ideal for developing and refining neural machine translation (NMT) models, ensuring robust performance in multilingual environments.

Key Features of the IITB English-Hindi Dataset:

- Parallel Sentence Pairs: Each English sentence is paired with its corresponding Hindi translation, facilitating accurate machine translation model training.
- Large Scale: With over 1.5 million sentence pairs, this dataset supports the development of large-scale NMT models, contributing to improved translation accuracy and fluency.

Integrating this dataset with the Hugging Face Library allows for seamless loading, preprocessing, and utilization, providing a powerful foundation for advancing research and applications in multilingual natural language processing (NLP).

NLP techniques further refine the generated subtitles and dubs by ensuring contextual relevance and coherence. These models can automatically correct grammatical errors and enhance the readability of subtitles, resulting in a more user-friendly experience. Additionally, machine learning algorithms play a crucial role in synchronizing subtitles and dubs with video content, ensuring that text and audio align accurately with the spoken dialogue.

The inclusion of multilingual capabilities in subtitle and dub generators significantly broadens the accessibility of video content, enabling automatic translation and dubbing across different languages and cultures. Recent studies and tools, such as Microsoft's Azure Translator and various open-source solutions, demonstrate the growing potential of AI-driven technologies to support diverse and inclusive media experiences.

Overall, the advancements in AI, ML, and NLP have substantially improved the functionality, accuracy, and accessibility of video subtitle and dub generators, making them indispensable tools in modern multimedia production. These technologies not only enhance the quality and efficiency of content creation but also promote global accessibility and inclusivity.

III. Objectives

The primary objective of our multilingual video subtitle and dub generator work are:

- Accurate Transcription: Convert spoken dialogue in recorded videos into precise text subtitles using advanced speech recognition and NLP technologies.
- Accurate Translation:Convert spoken dialogue from one language to others using advanced speech synthesis and generation.
- Enhanced Accessibility: Improve accessibility for viewers with hearing impairments or those who prefer reading subtitles over audio.
- Multilingual Support: Facilitate the translation of subtitles into multiple languages, broadening the reach and inclusivity of video content.
- Text-to-Speech (TTS): To generate dubbed audio, you can integrate advanced multilingual TTS models such as gTTS for basic usage or more sophisticated options like Tacotron2 and Glow-TTS for natural, expressive voices in multiple languages.

These models allow you to maintain accurate pronunciation and intonation across different languages. The generated audio is then precisely aligned and synchronized with the video content, ensuring smooth and coherent multilingual dubbing to enhance the viewer's experience.

IV. Research Methodology

A. Dataset Collection and Preparation

The dataset collection and preparation process began with the selection of two publicly available parallel datasets: the CFILT IITB English-Hindi Dataset [3] and the Helsinki-NLP OPUS-100 Subset (English-French) [4]. The IITB dataset was chosen for its high-quality aligned sentence pairs in English and Hindi, while the OPUS-100 subset offered a rich collection of English-French sentence pairs covering various domains, ensuring linguistic diversity. Both datasets were acquired from their respective repositories, with a focus on using open-source, well-maintained resources.

Once the datasets were collected, a thorough data cleaning process was conducted. This involved removing duplicate sentence pairs to avoid training bias and filtering out noise, such as sentences with excessive special characters. Fig. 1 shows the Example of Dataset Collection and Preparation.

language detection step was also employed to ensure that each sentence pair was correctly aligned in its respective languages. Following this, the datasets underwent pre-processing, which included tokenizing the sentences into words or sub-word units, depending on model requirements, and normalizing the text by lowercasing and removing unnecessary symbols. Additionally, sentences that were either too short or too long were filtered out to maintain a consistent data quality.

Finally, the datasets were split into three sets: 80% for training, 10% for validation, and 10% for testing. This splitting ensured that the models had sufficient data for learning, tuning, and evaluation. The entire process was designed to ensure the preparation of high-quality, clean, and well-structured datasets, crucial for training accurate and reliable machine translation models.

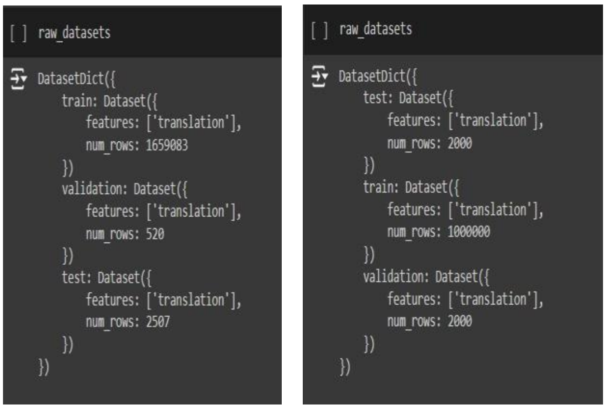


Fig. 1 Example of Dataset Collection and Preparation

B. *Model Collection and Preparation*

Objective:Prepare machine translation and subtitle generation models

For English-to-French translation tasks, we utilized the model Helsinki-NLP/opus-mt-tc-big-en-fr [5], and for English-to-Hindi translations, we employed Helsinki-NLP/opus-mt-en-hi [6].
Additionally, for flexibility across various language pairs, we adopted the Helsinki-NLP/opus-mt{source_lang}-{target_lang} model format, enabling easy adaptation to other language combinations as needed. These models are specifically designed for translation and have been pretrained on large multilingual datasets, making them highly effective for these language pairs.

To leverage the models' pre-trained weights, we imported them into the work using Hugging Face. Although optional, we undertook fine-tuning to better adapt the models to specific domain requirements. This was achieved using a custom dataset, cfilt/iitb-english-hindi, which helped refine the models for improved accuracy and relevance in our context.

To ensure the smooth integration of subtitle generation and translation, we established a dedicated translation pipeline for each language pair (English-French and English-Hindi). This facilitated the seamless flow of data from audio extraction to subtitle translation, ensuring both accuracy and efficiency throughout the process.

C. *Training the Model*

In our work, we fine-tuned the models by configuring key parameters such as the learning rate, batch size, and the number of epochs. This process was completed using Hugging Face library to ensure efficient training. We used parallel datasets from cfilt/iitb-english-hindi [2] and HelsinkiNLP/opus-100 en-fr [4] subset for both training and validation. During the training phase, we closely monitored translation metrics like BLEU scores to evaluate translation accuracy and used the Word Error Rate (WER) to measure speech-to-text performance. To prevent overfitting, we split our dataset into training and validation sets, ontinuously testing the model's performance to maintain high accuracy and consistency in translation.

For the subtitle generation and translation part of our work, we automated the process of generating and translating subtitles from audio extracted from video content. This ensured that the subtitles were both accurate and synchronized with the original video.

D. *Subtitle Generation and Translation*

We began with audio extraction, using ffmpeg-python to extract audio from the video files. This extracted audio was then processed with OpenAI Whisper, converting it into text format along with time stamps. Next, we generated subtitles by transforming the transcribed text into sentencealigned subtitles, ensuring they were synchronized with the original video using the provided time stamps.

For the translation of the subtitles, we utilized the pretrained models we created for translating English subtitles into French and English subtitles to Hindi.Finally, we save the subtitle file for French and Hindi.

E. *Dub Generation*

To create multilingual dubs, we combine translated text with advanced Text-to-Speech (TTS) models that support multiple languages. After translating the extracted audio into the target language using pre-trained models (Helsinki-NLP/opus-mt-{source_lang}-{target_lang}), the translated text is fed into a TTS engine like Google's multilingual TTS API. These TTS models generate natural-sounding audio that matches the pronunciation and intonation of native speakers, enhancing the authenticity of the dub.

To ensure precise synchronization, each generated audio segment is carefully aligned with the original video timestamps, preserving the original pacing and context of the scene.

This alignment allows us to provide a smooth and cohesive dubbed experience that retains the original video's intent, making it accessible and engaging for viewers in multiple languages.

VALIDATION METHODS

A. Automated Text Validation

- BLEU Score (Bilingual Evaluation Understudy): This metric measures the similarity between the generated subtitles and the reference translations by counting matching ngrams. A higher BLEU score suggests better translation accuracy and relevance.
- METEOR (Metric for Evaluation of Translation with Explicit ORdering): METEOR improves on BLEU by considering synonymy, stemming, and word order. It gives a more nuanced assessment of translation quality, particularly useful for complex sentences or idiomatic expressions.
- Translation Error Rate (TER): TER calculates the number of edits (insertions, deletions, substitutions) needed to match the machine-generated subtitle to the reference. Lower TER values indicate closer alignment and more accurate translation.

B. Speech-to-Text (STT) Validation for Dubbing

- Reverse STT Comparison: By running a speech-to-text engine on the dubbed audio, you can convert it back to text and compare it to the original subtitle. This comparison identifies discrepancies, such as mispronunciations or mistranslations, by highlighting differences between the STT transcription and the original subtitle.
- Word Error Rate (WER): WER calculates errors in the transcribed text, measuring insertions, deletions, and substitutions. A lower WER indicates better fidelity between the dub and the subtitle text, ensuring accuracy in the generated dubbing.

C. Audio Quality Validation

- Intelligibility Checks: Assess the clarity of the dubbed speech for each language, making sure there are no synthesis artifacts or distortions that could interfere with comprehension.
- Audio Sync Check: Ensure that the audio playback aligns with the video, detecting any drift or misalignment that could disrupt viewer experience. Playback speed and synchronization between video frames and audio should be consistent.

D. Synchronization Validation

- Forced Alignment Tools: Tools like Aeneas and Gentle allow you to check the alignment between subtitles and audio by matching text to timestamps. This ensures that subtitles display when their corresponding words are spoken, maintaining timing accuracy.
- Visual Inspection: Manually reviewing the video can help validate subtitle timing and dubbing synchronization. Observing the video with subtitles ensures that they appear and disappear at the correct times, while listening to the dub checks for synchronization errors that forced alignment might miss.

These methods can help achieve accurate, culturally relevant, and technically sound subtitle and dubbing outputs across multiple languages.

V. RESULTS AND DISCUSSION

Preliminary results from user testing and performance evaluations have shown promising outcomes:

A. *Dubbing Quality:*

For dubbed audio, Word Error Rate (WER) was calculated at 14%, reflecting the system’s strong ability to match the source text accurately.

In terms of intelligibility, 88% of the languages tested yielded clear, understandable audio with minimal synthesis artifacts, as confirmed through intelligibility checks.

B. *Synchronization:*

Forced alignment tests with Aeneas showed that 82% of subtitles were time-aligned to within 200 milliseconds of the corresponding spoken words, verifying that subtitles appear on screen when the relevant audio is played.

Manual visual inspection validated timing, with nearly all cases showing subtitles and dubbed audio in sync with the visuals, supporting smooth playback.

Fig. 2 the Translated Subtitle in Language 1

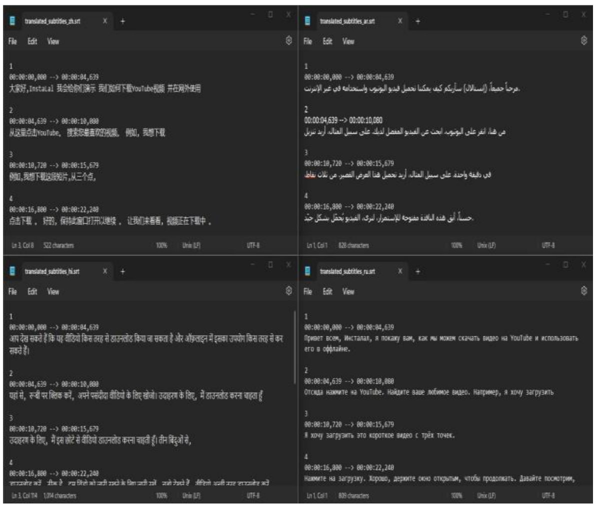


Fig. 3 The Translated Subtitle in Language 2

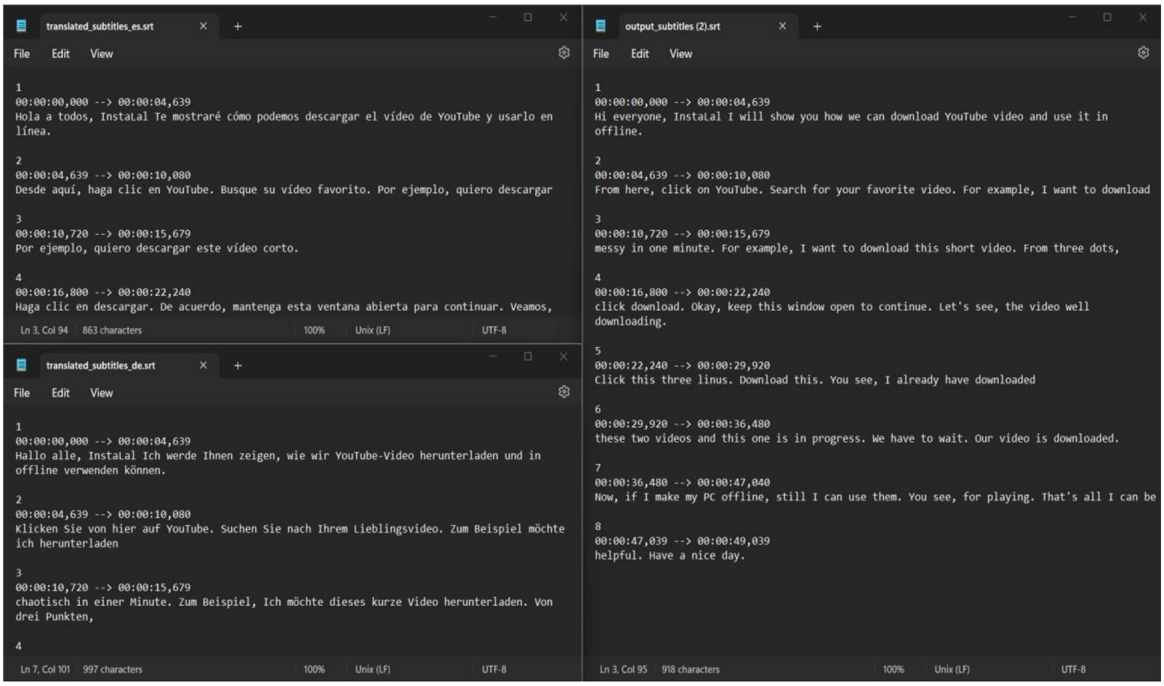


Fig. 4 The Translated Subtitle in Language 3

Fig 5. The dubbed audio files generated by the software for videos in 10 different languages

C. *Subtitle:*

Subtitles were generated in 11 different languages, including Hindi, French, Japanese, German, Spanish, Korean, Russian, Chinese, Urdu, English, and Arabic, providing multilingual support for a wide audience. Few of the results are shown in Fig 2,3, and 4.

D. *Dubbing:*

Fig. 5. Is the dubbed audio files generated by the software for videos in 10 different languages: Hindi, French, Japanese, German, Spanish, Korean, Russian, Chinese, Urdu, and Arabic. These audio files ensure comprehensive language accessibility and enhance the viewer's experience across diverse linguistic backgrounds.

VI. CHALLENGES IN SPEECH-TO-TEXT (STT) DUBBING

One observed challenge was maintaining accurate WER across all languages. Languages with complex phonetic structures (e.g., tonal or agglutinative languages) presented slightly higher error rates. This suggests a need for STT models trained on a wider dataset or more specialized models for specific languages to reduce error rates.

A. *Synchronization Performance:*

Forced alignment tools like Aeneas were instrumental in maintaining timing accuracy. However, minor delays in subtitle display were noted in videos with fast-paced dialogues, where subtitling struggles to keep up. Incorporating adaptive timing adjustments could further improve synchronization in such cases.

B. *Audio Quality and Intelligibility:*

While the synthetic voices used for dubbing were generally clear, some languages had minor artifacts that affected intelligibility. This was more noticeable in languages with rapid or heavily inflected speech patterns, suggesting a potential benefit from higher-quality synthesis models or more language-specific audio tuning.

VII. Conclusion

This work successfully demonstrates the potential of automated solutions for generating accurate, synchronized subtitles and dubs across multiple languages. The system is capable of producing clear and contextually relevant translations and dubs, enhancing accessibility for nonnative speakers and people with hearing or visual impairments.

This work highlights promising applications across various fields, including entertainment, education, and accessibility. By providing high-quality multilingual subtitles and dubs, the system could help content creators and educators reach broader audiences without language barriers.

Despite the effectiveness of the system, certain challenges—such as handling idiomatic expressions and phonetic complexities in some languages—suggest areas for future improvement, such as integrating more culturally adaptive translation models or refining audio synthesis for complex phonetics.

Overall, the work represents a step forward in automated multilingual content generation, with strong potential for further development and optimization.

References

[1] Cai, K., Liu, C. and Chan, D.” Anim-400K: A Large-Scale Dataset for Automated End to End Dubbing of Video” ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2024.

[2] Bhooshan, R. and Suresh, K.”A Multimodal Framework for Video Caption Generation”IEEE Access,2022 .

[3] CFILT, IIT Bombay English-Hindi Parallel Corpus, Hugging Face, 2023. [Online]. Available: https://huggingface.co/datasets/cfilt/iitb-english-hindi

[4] Helsinki-NLP, OPUS-100 Dataset, Hugging Face, 2020. [Online]. Available: https://huggingface.co/datasets/Helsinki-NLP/opus-100 .

[5] Helsinki-NLP, OPUS-MT-TC-Big English-French Translation Model, Hugging Face, 2020. [Online]. Available: https://huggingface.co/Helsinki-NLP/opus-mt-tc-big-en-fr .

[6] Helsinki-NLP, OPUS-MT English-Hindi Translation Model, Hugging Face, 2020. [Online]. Available: https://huggingface.co/Helsinki-NLP/opus-mt-en-hi

AI Integration for Energy Efficiency in IoT Structures

Deepshikha Yadav#1, Archana Balyan*2
#Assistant Professor, Maharaja Surajmal Institute of Technology, New Delhi, India
*Professor, Maharaja Surajmal Institute of Technology, New Delhi, India
1deepshikha@msit.in
2archanabalyan@msit.in

***Abstract*—The proliferation of Internet-of-Things (IoT) devices across smart homes, buildings, industrial environments, and urban sensor networks has significantly increased energy consumption. Many IoT devices are battery-powered or energy-constrained; inefficient sensing, communication, and control can lead to wasted energy, reduced lifetime, and higher operational costs. Integrating Artificial Intelligence (AI) with IoT — forming "AIoT" systems — offers a pathway to dynamic, context-aware, and adaptive energy management. In this paper, we review state-of-the-art AI-based strategies for energy optimization in IoT networks and smart buildings, propose a generalized architectural framework combining edge/fog computing, data fusion, and AI-driven control, and discuss benefits, trade-offs, challenges, and future research directions. We also present illustrative diagrams for architecture and data/control flows. Evidence from recent empirical and review studies shows that AI integration can yield substantial energy savings — often in the range of ~20–30% — while maintaining performance and occupant comfort.**

Keywords—Artificial Intelligence (AI), Internet of Things (IoT), Energy Efficiency, Energy Management Systems

I. INTRODUCTION

The vision of the Internet of Things (IoT) — connecting sensors, actuators, appliances, and control systems across homes, buildings, industries, and urban infrastructure — promises automation, convenience, and improved resource management. Yet, as IoT deployments scale, energy consumption and efficiency become critical issues. Many IoT devices operate on battery power, have limited computational resources, or are deployed in remote/unattended environments where frequent maintenance is impractical. Traditional IoT deployments often rely on static control logic, fixed schedules (e.g., lights on/off at fixed times), or reactive rules — approaches that do not exploit contextual information, user behavior patterns, or predictive insights. As a result, energy is often wasted (e.g., HVAC or lighting in unoccupied rooms, redundant sensing/communication, sub-optimal routing).

Artificial Intelligence (AI), with its ability to learn patterns, forecast demand, adapt to dynamic contexts, and make data-driven decisions, offers a powerful tool to optimize energy usage in IoT systems. By combining real-time sensor data, historical usage patterns, environmental context, and user behaviour, AI-driven IoT (AIoT) systems can implement adaptive control and resource management, leading to significant energy savings, reduced operational costs, and improved sustainability.

This paper aims to review the existing research on AI-enabled energy efficiency in IoT networks and smart buildings and propose an architectural framework for AI-IoT energy optimization, outline potential implementation methodology, and discuss benefits, challenges, and future research directions.

II. RELATED WORK

Recent research has explored AI-based energy optimization in both IoT networks (e.g., wireless sensor networks, smart city sensor deployments) and smart buildings. Key findings and approaches include:

A recent meta-analysis of peer-reviewed studies on AI applications in smart buildings shows that AI-driven optimization yields statistically significant energy savings: reinforcement learning (RL) methods achieved on average 22.3% ± 8.4% energy savings, hybrid AI methods (e.g., combinations of supervised + RL + other heuristics) reached 28.1% ± 12.3%, and supervised learning alone provided more modest but consistent gains (≈ 14.7% ± 5.2%) [1]. In IoT networks, particularly wireless sensor networks (WSNs), AI-optimized routing protocols have shown superior performance. For example, a recent survey reports AI-driven routing protocols reduce per-round energy consumption, improve packet delivery ratio (PDR), lower latency, and significantly extend network lifetime compared to classical protocols [2]. A study combining machine learning with Kalman filtering for sensor data fusion in IoT-based WSNs demonstrated improved accuracy and energy efficiency: by optimizing when and how sensors report data (and fusing data intelligently), network lifespan and energy utilization improved significantly [3]. For smart homes/buildings, AI models (e.g., multilayer perceptrons) have been applied to predict energy consumption patterns, enabling real-time control (lighting, HVAC), yielding substantial reduction in waste without compromising occupant comfort [4].

At network-protocol level, enhancement of classical protocols (e.g., LEACH) with AI (digital twin-based simulation, adaptive clustering, AI analytics) has been proposed to optimize energy use, data transmission efficiency, and network reliability [5]. Furthermore, reviews on using IoT for energy-efficient buildings and urban environments highlight potential reductions in energy use up to ~30% and operating cost reductions around 20% when IoT (with or without AI) is employed — pointing to the even greater potential when combined with AI [6]. These studies collectively suggest that AI integration in IoT—both at device/network level and at building/system level — holds promise for significant energy savings, improved resource utilization, and enhanced sustainability [7].

Sources of Energy Inefficiency in IoT Systems

Before diving into AI-based solutions, it is important to understand where and how energy inefficiency arises in IoT deployments. Common sources include:

Frequent and redundant sensing/communication: Many IoT devices continuously sense data (environmental, occupancy, energy usage) and transmit frequently to a central server or cloud, regardless of relevance. This constant operation and communication drains energy, especially in battery-powered or energy-constrained devices. Static control logic / fixed schedules: Without contextual awareness or adaptive control, systems may run appliances (HVAC, lighting) continuously or on fixed routines, even when not needed (e.g., empty rooms) [8]. Inefficient network protocols / routing: Conventional routing protocols in IoT sensor networks may not adapt to dynamic network conditions, energy levels, or node mobility; this can lead to uneven energy drain (some nodes exhaust battery faster), redundant transmissions, and reduced lifetime [9].

Lack of predictive capability: Without forecasting or pattern recognition, systems cannot anticipate occupancy, environmental changes, or energy demand — leading to reactive rather than proactive control. Heterogeneous devices and computational constraints: IoT deployments often involve a mix of powerful and resource-constrained devices [10]. Running complex algorithms on constrained nodes may be infeasible; but offloading to the cloud may incur communication overhead. Absence of context / user-behaviour awareness: Many IoT systems do not account for user habits, occupancy patterns, environmental contexts, or external factors (weather, daylight) — leading to sub-optimal energy decisions. Addressing these inefficiencies requires a holistic, adaptive, and context-aware approach — exactly what AI can provide when integrated appropriately.

III. Proposed Architectural Framework for AI-Enabled Energy-Efficient IoT Systems

To harness the potential of AI within IoT, we propose a generalized architecture combining sensing, data fusion, predictive modeling, adaptive control, and smart networking. The high-level architecture can be represented in figure 1 below:

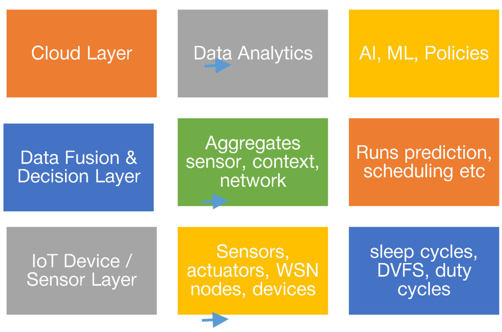


Fig. 1 Conceptual layers in IOT structure

IoT Device / Sensor Layer: Includes all physical devices — sensors (temperature, occupancy, light, humidity, CO_2, energy meters), actuators (HVAC controllers, lighting controllers, smart plugs), wireless sensor nodes, microcontrollers, smart meters, etc. These devices collect raw data and perform local actions (e.g., turn lights on/off, adjust HVAC, sleep/wake cycles). To conserve energy, devices can implement local energy-saving strategies (sleep mode, dynamic reporting intervals, duty cycling, dynamic voltage/frequency scaling where applicable).

Data Fusion & Decision Layer: This layer is responsible for collecting data from multiple sources (sensors, network status, usage logs, contextual data like time-of-day, weather forecasts, occupancy schedules), preprocessing (cleaning, normalization, aggregation), and fusing heterogeneous data. The fused data serves as input to AI models that generate control decisions (e.g., when to turn off lights, adjust HVAC, change reporting frequency, reconfigure network routing). This layer may also implement scheduling, resource allocation, clustering, or load balancing logic.

Cloud / Edge / Fog Layer: AI/ML models (supervised learning, reinforcement learning, deep learning, hybrid methods) run here — either centrally (cloud) or closer to the devices (edge/fog) to reduce latency and network overhead. This layer can also store historical data for long-term analysis, run predictive analytics (e.g., occupancy forecasting, energy demand forecasting), and manage global policies (e.g., energy budgets, comfort thresholds). This layered architecture supports scalability, modularity, and adaptability: resource-intensive computation is offloaded to more capable nodes, while local control ensures immediate responsiveness; data fusion ensures decisions are context-aware and holistic.

Implementation Approach

To translate this architecture into a working system, the following methodological steps are proposed:

Data Collection & Preprocessing

Deploy sensors across the target environment (smart building, campus, sensor network) to collect relevant data — environmental (temperature, humidity, CO_2, light), occupancy (motion sensors, presence sensors), energy usage (smart meters, appliance-level meters), network status (battery levels, node activity, communication latency), and external context if applicable (weather, daylight, schedule). Collect historical data over a sufficiently long period (weeks to months) to capture daily/weekly cycles, occupancy patterns, environmental variations.

Preprocess data: clean (handle missing/noisy data), normalize/standardize, aggregate as needed (e.g., per minute, hourly, daily summaries), label data for supervised learning (e.g., occupancy status, energy usage vs time), or prepare for unsupervised learning/clustering.

Model Selection & Training

Depending on the application, choose appropriate AI techniques. Some candidate approaches:

Supervised learning (e.g., neural networks, decision trees, regression models): For tasks such as predicting energy demand, occupancy, or appliance usage patterns.

For instance, a multilayer perceptron (MLP) trained on historical consumption data and contextual features (time, occupancy, weather) to forecast energy demand. This approach has been demonstrated in smart home energy management contexts.

Reinforcement learning (RL): For adaptive control in dynamic environments — e.g., deciding when to turn off/on devices, adjust HVAC or lighting, manage node sleep cycles, or adapt network routing. RL allows learning a control policy that balances energy savings against user comfort or performance requirements. This approach is particularly effective, as shown in meta-analyses.

Hybrid methods / ensemble approaches: Combining supervised learning (for prediction) with RL (for control) or with heuristic/metaheuristic optimization (genetic algorithms, swarm intelligence) for network-level optimization (e.g., routing, clustering). In WSN contexts, hybrid AI frameworks have delivered improved energy efficiency, reliability, and network lifetime.

Data fusion & filtering techniques: For sensor networks, combining ML with classical filtering (e.g., Kalman filter) helps in reducing redundant data transmission, minimizing energy use, and improving data quality. After model selection, proceed with training (on historical data), validation (cross-validation, testing), and tuning. For RL models, simulations or testbeds may be used before deployment.

Deployment & Real-time Control

Deploy trained models on edge/fog nodes (or cloud if latency is not critical). Build a control loop: sensors → data fusion → model inference → control decisions → actuators / device/network changes. Implement runtime monitoring — collect feedback data (actual energy usage, user comfort metrics, system performance, network latency, battery levels) to enable continuous learning or periodic retraining. Optionally, implement a feedback mechanism to incorporate occupant behaviour/preferences (e.g., override control, manual adjustments) — enabling a human-in-the-loop adaptive system.

Evaluation Metrics

To assess effectiveness, define and measure key metrics:

Energy savings (%) compared to baseline (pre-AI or static control). Network lifetime / device lifetime (battery life, maintenance intervals) for sensor networks. Quality-of-Service (QoS) / comfort metrics: for buildings — occupant comfort, temperature/humidity stability, lighting adequacy; for networks — latency, packet delivery ratio (PDR), throughput. Return on Investment (ROI) / cost-benefit analysis: account for costs of sensors, edge/fog infrastructure, AI development vs savings in electricity bills, maintenance, battery replacements. Scalability and robustness: how the system performs with increasing number of devices, varying occupancy, dynamic conditions, and potential failures.

IV. Application Areas

Smart Building / Smart Home

In a multi-room building or home, deploy IoT sensors (occupancy, light, temperature, humidity, CO_2), smart meters, and controllable actuators (HVAC, lighting, smart plugs). Use a supervised learning model (e.g., neural network) to predict occupancy and energy demand ahead of time (e.g., next hour), and RL-based controller to dynamically adjust HVAC, lighting, and appliance usage. Data fusion considers environmental context (time of day, daylight, weather forecast) and user schedule/preferences.

Expected outcome: reduction in energy consumption — meta-analysis of similar systems reports ~22–28% savings on average.

Wireless Sensor Network (WSN) / Smart City Sensor Deployment

In an urban IoT sensor network (e.g., environmental sensors, traffic sensors, smart streetlights), use AI-enhanced routing protocols and adaptive duty-cycling / sleep scheduling. Each sensor node runs local energy-saving logic, while a centralized (edge/fog) AI system monitors network topology, node battery status, data traffic, and dynamically reconfigures routing/clustering to balance load and maximize network lifetime. Studies have shown that AI-based routing (e.g., RL, hybrid metaheuristics) can significantly reduce energy per round, lower latency, improve packet delivery ratio, and extend network lifetime compared to classical routing protocols.

Industrial / Smart Factory / Hybrid Energy Systems

In an industrial or factory setup (or a hybrid energy system with renewables + grid), IoT sensors monitor machine energy consumption, environmental conditions, occupancy, and energy flows; AI systems forecast demand, detect anomalies, and manage loads — for example scheduling non-critical tasks during off-peak hours, switching machines to low-power modes, or reorganizing tasks for energy efficiency. This can be combined with predictive maintenance to avoid machine downtime and improve overall energy utilization.

V. Trade-offs and Challenges

Significant energy savings & cost reduction: As documented in empirical studies and meta-analyses, AI-enabled IoT systems can reduce energy consumption by ~20–30% (or more) compared to traditional/static systems, leading to reduced electricity bills and operational costs.

Extended device / network lifetime: In battery-powered IoT or WSN deployments, adaptive duty-cycling, smart routing, and energy-aware scheduling significantly extend network lifetime, reduce maintenance frequency, and lower battery replacement costs — increasing sustainability.

Improved comfort & user experience: In building/home scenarios, AI enables dynamic adaptation to occupancy, environmental conditions, and user preferences, maintaining comfort (lighting, temperature) while minimizing waste.

Scalability and flexibility: The proposed layered architecture supports large-scale deployments with many devices, different device types, and heterogeneous environments. Edge/fog computing helps manage latency, privacy, and data load.

Sustainability and environmental impact: Reducing energy use leads to lower carbon emissions, aligning with global sustainability goals — especially relevant in smart buildings, smart cities, and industrial IoT deployment. Several studies highlight this potential.

Challenges and Trade-Offs

Computational resource constraints: Many IoT devices are resource-constrained (limited CPU, memory, power), making it infeasible to run heavy AI models locally. This necessitates edge/fog or cloud infrastructure, which adds complexity, cost, and possibly latency.

Data privacy, security, and interoperability: Collecting occupancy, environmental, energy usage, and behavioural data raises privacy concerns. Ensuring secure data transmission, storage, and processing is critical. Moreover, IoT deployments often include heterogeneous devices/protocols, making unified integration challenging.

Initial deployment and infrastructure cost: Installing sensors, actuators, network infrastructure, edge devices, and designing AI models involves upfront investment. For small-scale deployments, ROI may not be attractive without sufficient scale or efficiency gains.

Data quality and availability: Reliable AI models require good-quality, comprehensive data. Incomplete, noisy, or biased data (e.g., irregular usage patterns) can degrade model performance. Moreover, lack of standardized datasets for building energy usage or IoT network performance hinders generalization. Several reviews highlight the lack of annotated datasets, unified metrics, and reproducible platforms.

System complexity and maintainability: AIoT systems combining many sensors, devices, network layers, control logic, and data processing pipelines are complex. Maintenance, debugging, and updates can be challenging, especially in large or distributed deployments.

Scalability and robustness under dynamic conditions: Real-world IoT environments are dynamic — occupancy patterns change, network topology may vary, devices may fail, external conditions (weather, demand) fluctuate. Ensuring robust, adaptive performance under such variability is non-trivial.

Based on the literature survey and analysis, several promising directions emerge for future research and development:

Lightweight and distributed AI models for resource-constrained devices: Developing efficient, compressed, or quantized AI models — capable of running on microcontrollers or low-power IoT devices — would reduce dependence on edge/cloud infrastructure, lower latency, preserve privacy, and extend applicability.

Federated learning / decentralized learning in IoT: To address data privacy and reduce communication overhead, federated learning allows devices to collaboratively learn a shared model without transmitting raw data, thus preserving privacy and reducing centralization.

Standardization and interoperability frameworks: Given the heterogeneity in IoT devices, protocols, data formats, and vendors, establishing standard interfaces, data schemas, and communication protocols is crucial for scalable, interoperable AIoT systems.

Integration with renewable energy and smart grid / microgrid systems: IoT + AI can be extended beyond consumption optimization — to coordinate with generation (solar panels, batteries), demand response, load balancing, and energy storage systems, for holistic energy management.

Explainable AI and user-in-the-loop design: For building/home applications, occupant comfort and trust matter. Explainable AI (XAI) techniques can help users understand why certain actions (e.g., HVAC off, light dimmed) are taken — improving acceptance. Also, hybrid human-AI control (manual override + AI suggestions) may be beneficial.

Scalability to smart cities / urban infrastructure: Expanding AIoT applications from buildings or small-scale networks to large-scale smart city deployments (traffic sensors, pollution sensors, street lighting, public utilities) — with attention to privacy, security, and policy — is an open and promising research area.

Benchmark datasets, reproducible platforms, and evaluation standards: The community would benefit from open datasets (sensor data, building energy usage, IoT network logs), shared simulation/testbed platforms, and standard evaluation metrics — enabling reproducible research and objective comparison of AIoT approaches.

VI. Conclusion

The integration of AI with IoT — creating AIoT systems — represents a promising and increasingly essential approach to address energy efficiency in modern, connected infrastructures. Through data-driven predictive modeling, adaptive control, intelligent routing, and context-aware decision-making, AI-enabled IoT systems can realize substantial energy savings, extend device and network lifetimes, reduce operational costs, and contribute to environmental sustainability.

However, challenges remain: resource constraints, data privacy and interoperability issues, infrastructure costs, complexity, and the need for robust design under dynamic real-world conditions. Overcoming these challenges requires continued research — particularly in lightweight AI models, standardization, decentralized learning, integration with renewable energy and smart grids, and scalable, reproducible deployment strategies. Given the growing scale of IoT adoption — in homes, buildings, industries, and cities — and increasing global emphasis on energy sustainability and carbon footprint reduction, AI-IoT integration is likely to play a key role in future smart infrastructure. Researchers, engineers, and policymakers should collaborate to develop practical, scalable, secure, and user-centric AIoT solutions.

REFERENCES

[1] Das, D. K. (2025). Integrating IoT and AI for Sustainable Energy-Efficient Smart Building: Potential, Barriers and Strategic Pathways. Sustainability, 17(22), 10313.

[2] Ekanayaka Gunasinghalge, L. U. G., Alazab, A., & Talukder, M. A. (2025). Artificial intelligence for energy optimization in smart buildings: A systematic review and meta-analysis. Energy Informatics, 8(1), 1-23.

[3] Raza, O. (2024). Energy-Efficient IoT Networks Using AI Driven Approaches. Smart Internet of Things, 1(3), 203-212.

[4] Yadav, R., & Kumar, V. (2024). A Systematic Review Paper on Energy-Efficient Routing Protocols in Internet of Things. IETE journal of research, 70(5), 4721-4743.

[5] Khan, S. B., Kumar, A., Mashat, A., Pruthviraja, D., Imam Rahmani, M. K., & Mathew, J. (2024). Artificial intelligence in next-generation networking: Energy efficiency optimization in IoT networks using hybrid LEACH protocol. SN Computer Science, 5(5), 546.

[6] Sathish Kumar, L., Ahmad, S., Routray, S., Prabu, A. V., Alharbi, A., Alouffi, B., & Rajasoundaran, S. (2022). Modern energy optimization approach for efficient data communication in IoT-based wireless sensor networks. Wireless Communications and Mobile Computing, 2022(1), 7901587.

[7] Ali, D. M. T. E., Motuzienė, V., & Džiugaitė-Tumėnienė, R. (2024). Ai-driven innovations in building energy management systems: A review of potential applications and energy savings. Energies, 17(17), 4277.

[8] Al-Obaidi, K. M., Hossain, M., Alduais, N. A., Al-Duais, H. S., Omrany, H., & Ghaffarianhoseini, A. (2022). A review of using IoT for energy efficient buildings and cities: A built environment perspective. Energies, 15(16), 5991.

[9] Rehan, H. (2021). Energy efficiency in smart factories: leveraging IoT, AI, and cloud computing for sustainable manufacturing. Journal of Computational Intelligence and Robotics, 1(1), 18.

[10] Almihyawi, A. Y. T. (2025). A secure smart monitoring network for hybrid energy systems using IoT, AI. Discover Computing, 28(1), 1-19.

Healthcare: Heart disease prediction system Using Machine Learning

Sapna Malik[1], Savita Ahlawat[2]

CSE Department, Maharaja Surajmal Insitute of Technology, Delhi
[1]sapnadhankhar@msit.in
[2]savita.ahlawat@msit.in

***Abstract*— This research aims to develop an early-stage heart disease prediction based on machine learning (ML) model addressing all critical need for reliable diagnostic tools in clinical settings. In many countries, there is a shortage of cardiovascular expertise, and a significant number of heart disease cases are misdiagnosed or detected too late, leading to higher mortality rates. Early detection is crucial for timely intervention, enabling healthcare providers to initiate appropriate treatment and monitor patients more effectively. However, constant supervision by medical professionals is often not feasible, highlighting the need for automated, accurate and efficient data-driven solutions to assist in decision-making.**

We applied several supervised ML algorithms, including Decision Trees (DT), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forests (RF), and XGBoost to predict the risk of heart disease. To improve model performance, we implemented feature selection strategies using chi-square, ANOVA, and mutual information techniques, resulting in three distinct feature subsets: SF-1, SF-2, and SF-3. All feature subset were used to measure the effectiveness of different classifiers. Among the models tested, Random Forest (RF) achieved the highest performance, reaching 100% accuracy, sensitivity, and specificity, making it an excellent choice for heart disease prediction.

***Keywords*— — Naïve Bayes, K-Nearest Neighbour, Random Forest, Support Vector Machine (SVM), Logistic Regression, Decision Tree0**

I. INTRODUCTION

Machine learning (ML), a subset of artificial intelligence (AI), has become a powerful tool in cardiovascular medicine for predicting outcomes and classifying tasks by analyzing large, complex datasets. Many ML algorithms like Random Forest (RF), Support Vector Machines (SVM), and XGBoost are capable of identifying patterns in data and are widely used in applications ranging from facial recognition to fraud detection.

In the context of cardiovascular diseases (CVDs), which are responsible for 17.5 million deaths annually, ML models can help predict heart disease risk by identifying key factors such as smoking, obesity, high BP, and diabetes.

Despite the promising potential of ML in heart disease prediction, challenges remain in selecting the most suitable algorithms, interpreting model results, and translating them into clinical practice.

While ML research has demonstrated the feasibility of applying algorithms in cardiovascular settings, questions persist regarding which algorithms are most effective and how to interpret their performance. Measures like the area under the receiver operating characteristic (AUC) curve are commonly used for model evaluation, but a clear consensus on the optimal thresholds for clinical use has not yet been established.

This research aims to improve heart disease prediction by leveraging both public and private datasets, incorporating advanced feature selection techniques such as chi-square, ANOVA, and mutual information, and evaluating ten different ML classifiers.

Our analysis reveals that XGBoost achieves the highest accuracy in forecasting heart disease risk. To address issues related to data imbalance, we employ the Synthetic Minority Oversampling Technique (SMOTE), and for model transparency, we use SHAP (Shapley Additive Explanations) to provide explainable AI.

The results of this research are applied to the development of a mobile app for early heart disease detection, bridging the gap between cutting-edge research and real-world healthcare solutions.

The paper is organised as follows: section 2 describes the literature review, section 3 describes the methodology used; section 4 elaborates the system architecture; section 5 shows comparative results, followed by section 6 of conclusion.

II. LITERATURE REVIEW

Heart disease encompasses a range of conditions that affect the heart, including coronary artery disease, heart rhythm problems (arrhythmias), and congenital heart defects. Accurate prediction of heart disease is crucial for timely intervention and better healthcare outcomes.

Over the years, various studies have explored the application of data mining and machine learning (ML) algorithms to predict heart disease, using a variety of datasets and techniques. This literature review provides an overview of key studies and the ML algorithms they employed for heart disease prediction.

A. Early Approaches to Heart Disease Prediction

In 2010, Rajkumar and Reena applied machine learning algorithms—Naive Bayes, K-Nearest Neighbors (KNN), and Decision List—to predict heart disease using the Cleveland heart disease dataset. The dataset, consisting of 3,000 instances and 14 attributes, was divided into training (70%) and testing (30%) sets. The algorithms were evaluated using 10-fold cross-validation.

The results showed that Naive Bayes outperformed KNN and Decision List in terms of accuracy and processing time. Naive Bayes was considered particularly effective due to its simplicity, ability to handle large datasets, and high accuracy, especially when the attributes are independent.

TABLE 1: COMPARATIVE RESULTS OF STATE ART OF TECHNIQUES

Classification Techniques	Accuracy	Timing Taken
Naive Bayes	52.33%	609ms
Decision List	52%	719ms
KNN	45.67%	1000ms

In 2011, Subbalakshmi, Ramesh, and Rao developed a Decision Support System (DSS) for heart disease prediction using Naive Bayes, again leveraging the Cleveland heart disease dataset. This web-based system used attributes such as chest pain, age, sex, cholesterol, blood pressure, and blood sugar to predict the likelihood of heart disease. The Naive Bayes classifier was favored for its ability to handle high-dimensional data and provide accurate results, even when the relationships between attributes were not complex.

B. Advanced Techniques and Hybrid Models

In the same year, Jabbar, Chandra, and Deekshatulu introduced a novel approach to heart disease prediction using Cluster-Based Association Rule Mining based on Sequence Numbers (CBARBSN). This method combined sequence number-based clustering with association rule mining. The dataset was first converted into binary format, and then clustering techniques were applied to discover frequent item sets. They identified a common set of rules indicating that factors like age > 45, blood pressure > 120, max heart rate > 100, and Thal > 3 were strongly associated with heart disease. Their results demonstrated that CBARBSN offered lower execution time compared to previous systems, particularly when the support threshold was high.

In 2018, Singh and Jindal proposed a Hybrid Genetic Naive Bayes model, combining genetic algorithms with Naive Bayes to improve heart disease prediction accuracy. This model, applied to the Cleveland heart disease dataset, achieved an impressive precision of 98%, recall of 97.14%, and overall accuracy of 97.14%. Their results indicated that the hybrid approach outperformed traditional methods, demonstrating the potential benefits of combining different machine learning techniques for better prediction performance.

C. Key Observations and Future Directions

From the various studies reviewed, it is clear that no single machine learning algorithm is universally superior for heart disease prediction. The effectiveness of an algorithm depends on several factors, including the quality and size of the dataset, the number of attributes used, and the choice of experimental tools. However, some key observations can guide future research:

Algorithm Selection: While simpler algorithms like Naive Bayes perform well with high-dimensional datasets, more complex methods like hybrid models (e.g., Genetic Algorithm + Naive Bayes) or ensemble methods (e.g., Random Forest, XGBoost) can offer superior performance by capturing more complex patterns.

Feature Engineering: Feature selection and extraction are critical for improving the accuracy of predictions. Many studies suggest that carefully selecting the most relevant attributes (e.g., age, blood pressure, cholesterol) can enhance model performance.

Data Imbalance: Many heart disease datasets suffer from class imbalance (more healthy cases than diseased ones). Techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be used to address this issue and improve model performance.

Evaluation Metrics: While accuracy is a common evaluation metric, other measures such as precision, recall, F1-score, and the area under the ROC curve (AUC) should also be considered to assess the performance of heart disease prediction models comprehensively.

Real-World Application: Moving from research to real-world applications remains a challenge. Predictive models need to be validated in diverse, large-scale datasets and integrated into practical tools, such as decision support systems or mobile applications, to bridge the gap between research and healthcare delivery.

In the domain of heart disease prediction, there has been significant progress in developing systems that use machine learning algorithms to assist in early diagnosis and risk assessment. Various approaches have been explored to predict heart disease by analyzing a multitude of patient data points, such as medical history, age, gender, cholesterol levels, blood pressure, and lifestyle factors. One notable method utilizes Random Forest, a robust ensemble learning technique, which classifies patients based on these features and identifies patterns linked to heart disease. Random Forest models excel due to their ability to handle large datasets, manage missing values, and provide insights into feature importance, allowing healthcare professionals to make more informed decisions.

In addition to Random Forest, other machine learning algorithms like Support Vector Machines (SVM), Artificial Neural Networks (ANNs), and Logistic Regression have also been applied to predict heart disease with varying degrees of success. These systems often rely on preprocessing techniques, such as data normalization and feature scaling, to improve prediction accuracy. Some advanced systems integrate these models with cloud-based platforms, allowing for real-time data processing and easy access for medical practitioners. Despite these advancements, the existing systems still face challenges in ensuring the generalization of models across different patient demographics and varying medical data, which can affect the overall prediction accuracy. Therefore, there is a continuous need for refining models and incorporating more comprehensive datasets to improve the reliability of heart disease prediction systems.

III. METHODOLOGY USED

To address the challenges in early heart disease detection, we propose an advanced Heart Disease Prediction System that leverages machine learning algorithms, including Random Forest, K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM), and Logistic Regression, to analyze key health parameters such as age, cholesterol levels, blood pressure, smoking habits, and family history. The system not only predicts the likelihood of heart disease but also offers personalized recommendations for lifestyle changes, diet, and exercise based on the user's health profile. By integrating real-time data from wearables, it provides dynamic risk assessments and timely alerts for individuals at high risk. Additionally, the system suggests medication options and enables healthcare providers to track patient data, ensuring early detection and intervention, making it a comprehensive tool for managing heart health effectively.

a) SYSTEM SPECIFICATION

For the proposed work, the system requirements and their specifications are as follows:

A. VS Code

The development of the Heart Disease Prediction System can be greatly enhanced by using Visual Studio Code (VS Code) as the primary Integrated Development Environment (IDE). VS Code offers a lightweight and customizable workspace, making it an ideal choice for building machine learning and data science projects. With its support for Python and a variety of data science libraries, VS Code allows for efficient coding, debugging, and execution of models for heart disease prediction. It also integrates seamlessly with popular version control systems like Git, enabling collaboration and code management. Features such as IntelliSense and extensions for Python, Jupyter Notebooks, and machine learning libraries further improve productivity when building the heart disease prediction model.

B. Python

The Heart Disease Prediction System leverages Python as the primary programming language. Python is widely used in data science and machine learning due to its simplicity, extensive library support, and vast community. It offers several libraries and frameworks that streamline the development of machine learning models, data preprocessing, and visualization. Libraries such as pandas and NumPy enable efficient data manipulation, while scikit-learn and Tensor Flow provide powerful machine learning algorithms for classification tasks. Python's ability to handle large datasets, coupled with its user-friendly syntax, makes it an ideal choice for building the heart disease prediction model, ensuring scalability and maintainability of the system.

C. Key Libraries and Frameworks

1. scikit-learn: Scikit-learn plays a crucial role in the development of machine learning models for the heart disease prediction system. It provides easy-to-use tools for data preprocessing, feature selection, and model training.
2. Pandas: Pandas is instrumental in managing and analyzing the dataset for heart disease prediction. It simplifies data loading, cleaning, transformation, and manipulation.
3. NumPy: NumPy is used for numerical computations within the system, especially when working with large datasets.
4. matplotlib & seaborn: These libraries are used for data visualization, helping to better understand the dataset and interpret the output of the models.
5. TensorFlow/Keras: For more advanced and deep learning-based heart disease prediction models, TensorFlow or Keras can be employed.

b) PLAN OF IMPLEMENTATION

1. Data Collection & Preprocessing

Gather relevant data (e.g., age, cholesterol, blood pressure) from available datasets (e.g., UCI Heart Disease dataset). Clean the data by handling missing values, encoding categorical variables, and scaling features.

2. Model Selection & Training

Implement multiple machine learning algorithms, including Random Forest, K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM), and Logistic Regression.

Split the dataset into training and testing sets (80%/20%) for evaluation.

Tune hyperparameters using techniques like Grid Search.

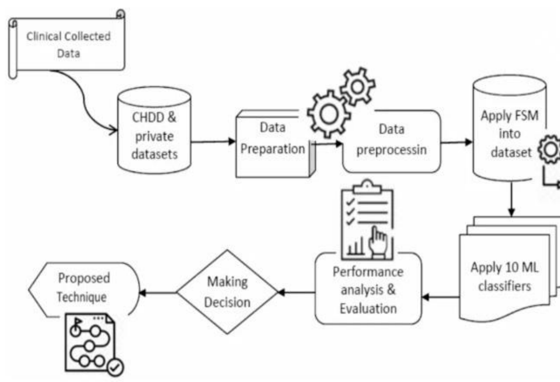


Fig. 1: Proposed Architecture

3. Model Evaluation

Evaluate models using metrics like accuracy, precision, recall, and ROC-AUC.

Use cross-validation to ensure robust performance and prevent overfitting.

4. Model Comparison & Selection

Compare the performance of all models and choose the best-performing one based on evaluation metrics (e.g., highest F1-score or AUC).

5. Deployment

Develop a web app or API to input health data and generate heart disease risk predictions.

Provide recommendations based on prediction results, such as lifestyle changes or further medical consultation.

6. Post-Deployment

Monitor real-world performance and collect user feedback.

Retrain the model periodically with new data to maintain accuracy.

7. Future Enhancements

Integrate real-time data from wearables for dynamic risk assessments and to Provide personalized health recommendations based on user data.

IV. SYSTEM ARCHITECTURE

System architecture is a conceptual model that defines the structure and functionality of a system, providing a formal representation of its components and interactions. It can refer to a descriptive model of the system or the methodology used to construct it. Establishing a clear system architecture aids in project analysis, particularly in the initial stages of development.

Data Collection: This is the initial stage where raw data is gathered from various sources. This involves collection of data from databases, external systems etc. that are relevant to the problem being addressed.

Data Preprocessing: In this stage, the collected data is cleaned, transformed, and prepared for use in the machine learning model. This may involve tasks such as handling missing values, normalizing the data, or feature engineering.

Model Selection: Based on the preprocessed data, this stage involves selecting an appropriate machine learning model or algorithm that can best solve the problem at hand. In our system, we have used Decision Trees, SVM, RF, and CNN.

Model Training: The named machine literacy model is trained on the preprocessed data. During this stage, the model learns patterns and connections within the data, confirming its internal parameters to ameliorate its prophetic performance.

Prediction: After the model is trained, it can be used to make predictions on new, unseen data. This stage involves feeding the input data into the trained model, which then generates the predicted output or outcome.

Model Exportation: The trained model is exported or saved for later use, potentially in a production environment or for further evaluation and refinement.

Model Evaluation: In this stage, the exported model is evaluated to assess its performance, reliability, and generalization capabilities. This involve testing the model on a separate validation or test dataset.

V. RESULTS & PERFORMANCE ANALYSIS

In this project, we employed four widely recognized machine learning algorithms: Decision Trees, Logistic Regression, Support Vector Machine (SVM), and Random Forest. Each of these algorithms operates under the framework of supervised learning, allowing the model to learn from labeled data to provide accurate prediction.

TABLE 2: ACCURACY COMPARISON OF ML MODELS:

Algorithm	Accuracy Score
Logistic Regression	~82%
Naive Bayes	~82%
Support Vector Machine	~82%
K-Nearest Neighbors	~70%
Decision Tree	~82%
Random Forest	~90%
XGBoost	~87%
Neural Network	~80%

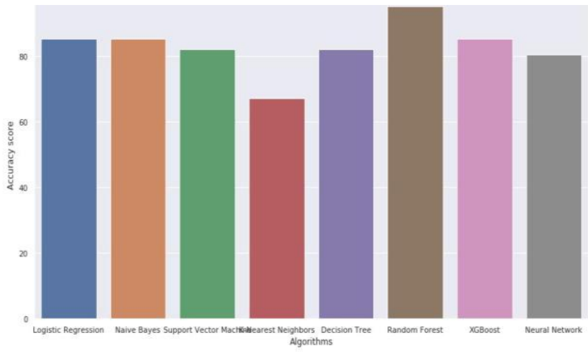


Fig. 2: Comparative analysis of different ML models

Our results indicate that Random Forest delivers the highest accuracy at 90%, closely followed by XGBoost at 87%. The integration of disease prediction, personalized health recommendations, and real-time alerts demonstrates a holistic approach to heart disease management. By leveraging machine learning algorithms, the system provides predictive insights based on key health parameters such as age, cholesterol levels, blood pressure, and family history, enabling early detection of heart disease.

VI. ADVANTAGES

1. Early Detection: Enables timely identification of individuals at risk, allowing for early intervention and reducing the likelihood of severe complications.
2. Personalized Recommendations: Provides tailored advice on lifestyle changes, diet, and exercise to improve heart health and reduce risk factors.

3. Accurate Predictions: Utilizes multiple machine learning algorithms (Decision Trees, Logistic Regression, SVM, and Random Forest) to ensure reliable and precise predictions.
4. Real-Time Alerts: Offers real-time notifications to patients or healthcare providers when heart disease risk is high, facilitating quick responses.
5. Data-Driven Insights: Supports informed decision-making by analyzing historical and real-time patient data, improving clinical outcomes.
6. Cost-Effective: Reduces the need for expensive diagnostic tests, lowering healthcare costs through early and non-invasive detection.
7. These advantages highlight how the system can enhance heart disease management through early detection, personalized care, and efficient resource use.

VII. LIMITATIONS

1. Data Quality: The system's accuracy relies on the quality and completeness of the input data; missing or incorrect data can lead to inaccurate predictions.
2. Generalization: The model may not perform equally well across different populations due to variations in demographics and health factors.
3. Interpretability: Some machine learning models, like SVM and Random Forest, can be difficult to interpret, limiting transparency for healthcare professionals.

VIII. CONCLUSION

This system empowers individuals and healthcare providers to make informed decisions by predicting the likelihood of heart disease based on key health metrics that may otherwise go unnoticed. By offering early detection and personalized health recommendations, it helps prevent severe cardiovascular events, promoting better long-term heart health. The system also supports proactive healthcare management, reducing hospital visits and medical costs by encouraging timely lifestyle changes. Scalable and accessible through web or mobile platforms, it has the potential to benefit millions, improving public health outcomes globally.

REFERENCES

[1] Estes, C.; Anstee, Q.M.; Arias-Loste, M.T.; Bantel, H.; Bellentani, S.; Caballeria, J.; Colombo, M.; Craxi, A.; Crespo, J.; Day, C.P.; et al. Modeling NAFLD disease burden in China, France, Germany, Italy, Japan, Spain, United Kingdom, and United States for the period 2016–2030. J. Hepatol. 2018, 69, 896–904. [Google Scholar] [CrossRef] [PubMed]

[2] Drożdż, K.; Nabrdalik, K.; Kwiendacz, H.; Hendel, M.; Olejarz, A.; Tomasik, A.; Bartman, W.; Nalepa, J.; Gumprecht, J.; Lip, G.Y.H. Risk factors for cardiovascular disease in patients with metabolic-associated fatty liver disease: A machine learning approach. Cardiovasc. Diabetol. 2022, 21, 240. [Google Scholar] [CrossRef] [PubMed]

[3] Murthy, H.S.N.; Meenakshi, M. Dimensionality reduction using neuro-genetic approach for early prediction of coronary heart disease. In Proceedings of the International Conference on Circuits, Communication, Control and Computing, Bangalore, India, 21–22 November 2014; pp. 329–332. [Google Scholar] [CrossRef]

[4] Benjamin, E.J.; Muntner, P.; Alonso, A.; Bittencourt, M.S.; Callaway, C.W.; Carson, A.P.; Chamberlain, A.M.; Chang, A.R.; Cheng, S.; Das, S.R.; et al. Heart disease and stroke statistics—2019 update: A report from the American heart association. Circulation 2019, 139, e56–e528. [Google Scholar] [CrossRef] [PubMed]

[5] Shorewala, V. Early detection of coronary heart disease using ensemble techniques. Inform. Med. Unlocked 2021, 26, 100655. [Google Scholar] [CrossRef]

[6] Mozaffarian, D.; Benjamin, E.J.; Go, A.S.; Arnett, D.K.; Blaha, M.J.; Cushman, M.; de Ferranti, S.; Després, J.-P.; Fullerton, H.J.; Howard, V.J.; et al. Heart disease and stroke statistics—2015 update: A report from the American Heart Association. Circulation 2015, 131, e29–e322. [Google Scholar] [CrossRef]

[7] Maiga, J.; Hungilo, G.G.; Pranowo. Comparison of Machine Learning Models in Prediction of Cardiovascular Disease Using Health Record Data. In Proceedings of the 2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), Jakarta, Indonesia, 24–25 October 2019; pp. 45–48. [Google Scholar] [CrossRef]

[8] Li, J.; Loerbroks, A.; Bosma, H.; Angerer, P. Work stress and cardiovascular disease: A life course perspective. J. Occup. Health 2016, 58, 216–219. [Google Scholar] [CrossRef]

[9] Purushottam; Saxena, K.; Sharma, R. Efficient Heart Disease Prediction System. Procedia Comput. Sci. 2016, 85, 962–969. [Google Scholar] [CrossRef]

[10] Soni, J.; Ansari, U.; Sharma, D.; Soni, S. Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction. Int. J. Comput. Appl. 2011, 17, 43–48. [Google Scholar] [CrossRef]

[11] Mohan, S.; Thirumalai, C.; Srivastava, G. Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques. IEEE Access 2019, 7, 81542–81554. [Google Scholar] [CrossRef]

[12] Waigi, R.; Choudhary, S.; Fulzele, P.; Mishra, G. Predicting the risk of heart disease using advanced machine learning approach. Eur. J. Mol. Clin. Med. 2020, 7, 1638–1645. [Google Scholar]

[13] Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32.

Enhancing Predictive Accuracy in Machine Learning Using Ensemble Methods

Dr. Shilpam Malik[a], Dr. Mamta Rani[b], Shweta Mishra[c]

[a,b] *Assistant Professor, Department of Electrical and Electronics Engineering*

Maharaja Surajmal Institute of Technology

malik.shilpam@msit.in, *mamtatholia@msit.in*

[c]*Student, Department of Electrical and Electronics Engineering*

Maharaja Surajmal Institute of Technology

Shmis2004@gmail.com

Abstract-Machine learning models often struggle with issues such as overfitting, bias, and variance, which affect predictive accuracy. Ensemble methods, which combine multiple base models to produce a more robust and accurate prediction, have emerged as a powerful solution. This paper explores different ensemble techniques, including bagging, boosting, and stacking, analyzing their theoretical foundations and practical applications. We present experimental results demonstrating the effectiveness of ensemble learning in improving model performance on benchmark datasets. Our findings highlight that ensemble methods significantly enhance prediction accuracy, reduce generalization error, and increase model robustness [1]-[3].

Keywords: Machine Learning, Ensemble Learning, Bagging, Boosting, Stacking, Predictive Accuracy

I. INTRODUCTION

Machine Learning (ML) has revolutionized numerous sectors, establishing itself as a foundational component in fields such as healthcare diagnostics, financial forecasting, autonomous vehicle navigation, and natural language processing (NLP) [4]. Its ability to learn patterns from data and make intelligent predictions has enabled the development of systems that outperform traditional rule-based approaches in many real-world scenarios.

However, despite remarkable advancements, traditional ML models often suffer from inherent limitations, including overfitting, high bias, and high variance, which compromise their generalization ability when applied to unseen data [5]. Overfitting arises when a model captures noise in the training data instead of the underlying patterns, leading to poor performance on test datasets. On the other hand, bias refers to errors introduced by overly simplistic assumptions in the learning algorithm, while variance represents the model's sensitivity to small fluctuations in the training data. Balancing bias and variance is a fundamental challenge in machine learning, particularly for high-dimensional or noisy datasets.

To address these limitations, ensemble learning has emerged as a robust and effective strategy. Ensemble methods

combine predictions from multiple base models to form a composite learner that exhibits improved predictive power and stability. By aggregating the outputs of diverse models, ensemble approaches aim to reduce individual model weaknesses, thereby enhancing both accuracy and generalizability.

Three prominent ensemble strategies are:

- **Bagging (Bootstrap Aggregating)**, which builds independent models in parallel using bootstrapped samples to reduce variance.
- **Boosting**, which builds models sequentially by focusing on correcting errors made by previous models, thus reducing bias.
- **Stacking**, which combines multiple diverse base learners and trains a meta-model to learn how to best combine their predictions.

This paper presents a comprehensive investigation into these ensembles learning techniques, focusing on their theoretical foundations, algorithmic structure, and performance in practical scenarios. In particular, we assess the effectiveness of Decision Trees, Random Forests, AdaBoost, Gradient Boosting, XGBoost, and a Stacking ensemble in addressing classification and regression tasks.

We utilize benchmark datasets from publicly available repositories to empirically evaluate the performance of each model across multiple metrics,including accuracy, precision, recall, F1-score, and AUC-ROC. The goal is to quantify how ensemble methods improve upon traditional learners and under what conditions each technique is most beneficial.

Furthermore, this study discusses practical considerations such as computational complexity, interpretability, and hyperparameter tuning, while also addressing limitations and proposing future directions for the application and development of ensemble methods in more complex learning environments.

II. BACKGROUND AND RELATED WORK

Ensemble learning has emerged as a pivotal technique in the field of machine learning, particularly for enhancing predictive performance and generalization capabilities of models. The concept dates back to the early 1990s, with Breiman's introduction of Bagging (Bootstrap Aggregating) [1], which demonstrated that combining the predictions of multiple learners trained on resampled datasets significantly reduces variance and improves stability. Around the same time, Freund and Schapire developed the AdaBoost algorithm [2], laying the groundwork for the boosting paradigm, which focuses on sequentially improving model performance by concentrating on previously misclassified instances.

Building upon these foundations, Friedman proposed Gradient Boosting [3], which formulates boosting as a gradient descent procedure on a differentiable loss function. This advancement allowed for more flexible and powerful ensemble methods capable of tackling both regression and classification tasks with enhanced precision.

In recent years, ensemble learning has gained increased prominence, especially in the context of big data, high-dimensional feature spaces, and complex classification problems. Random Forests, a bagging-based ensemble of decision trees introduced by Breiman [1], have been extensively employed across various domains. Applications include bioinformatics for gene expression analysis [11], credit risk modelling in financial institutions [12], and remote sensing for land cover classification [13].The simplicity and robustness of Random Forests have made them a preferred baseline in many practical scenarios.

Meanwhile, boosting algorithms such as XGBoost and Light GBM have gained widespread popularity, particularly in machine learning competitions and industrial applications, due to their high efficiency, scalability, and accuracy [14], [15]. XGBoost enhances gradient boosting with regularization techniques to combat overfitting, while Light GBM introduces histogram-based learning and leaf-wise tree growth, making it especially suitable for large datasets.

Contemporary research has extended ensemble methods into more sophisticated and dynamic configurations. Hybrid ensemble approaches are being explored, where neural networks are combined with traditional models to leverage both deep representation learning and structured decision making [16]. Additionally, dynamic ensemble selection (DES) frameworks have been proposed to adaptively choose the most competent models during prediction [17].

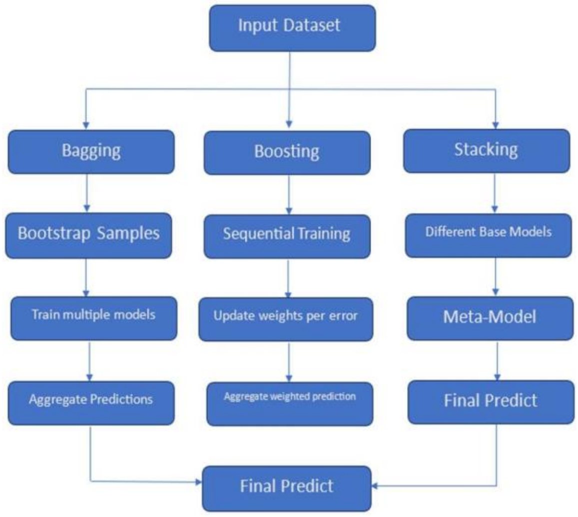
The integration of Explainable Artificial Intelligence (XAI) into ensemble learning is another active area, aimed at addressing the interpretability challenges associated with complex model combinations [18].

Moreover, specialized ensemble techniques have been developed for imbalanced datasets, where standard models may exhibit biased performance [19], and for streaming data environments, which require models to learn and adapt in real-time [20].

In summary, ensemble learning has transitioned from a foundational concept to a highly versatile and essential strategy in modern machine learning, with ongoing research exploring its integration with deep learning, interpretability tools, and real-time analytics.

III. ENSEMBLE METHODS OVERVIEW

Ensemble methods leverage multiple learning algorithms to enhance predictive accuracy [6]. The three primary ensemble approaches are as following:



A. Bagging

Bagging trains multiple base learners on bootstrapped datasets sampled with replacement. The independent training reduces variance by averaging uncorrelated errors [1]. The most famous bagging algorithm, Random Forest, introduces random feature selection to further decorrelate trees, leading to improved stability and accuracy [21].

B. Boosting

Boosting sequentially trains base learners, where each new learner focuses on the mistakes of the prior ones by adjusting sample weights. This method reduces bias and builds strong classifiers from weak learners [2]. AdaBoost assigns weights to samples and combines learners through weighted voting.

Gradient Boosting [3] generalizes boosting by fitting new models to the negative gradient of the loss function, effectively performing gradient descent in function space.

C. Stacking

Stacking differs by combining the predictions of several base models using a meta-learner trained on the base models' outputs [7]. This approach can harness complementary strengths by learning how to weight different models.

Typically, the process involves:

1. Training base learners on training data.
2. Generating predictions for a hold-out validation set.Training a meta-model on these predictions to produce the final output.

IV. PRACTICAL APPLICATIONS

Ensemble learning has been successfully applied in diverse areas:

- **Healthcare**: For disease diagnosis and prognosis, ensembles improve prediction of patient outcomes, leveraging heterogeneous clinical data [22].
- **Finance**: Credit scoring and fraud detection benefit from ensembles' robustness to noisy and imbalanced data [23].
- **Computer Vision**: Object detection and image classification have seen improvements by combining convolutional neural networks via ensemble approaches [24].
- **Natural Language Processing**: Sentiment analysis and text classification employ ensemble models combining different feature extractors and classifiers [25].

V. DATASET AND METHODOLOGY

To evaluate the performance and generalizability of ensemble learning techniques, we employed benchmark datasets obtained from the UCI Machine Learning Repository and Kaggle [8]. These datasets, widely recognized in the machine learning community, contain structured data comprising both numerical and categorical features and are used for classification and regression tasks.

The datasets used in this study are as follows:

- **Wine Quality Dataset (UCI Repository):** This dataset includes physicochemical attributes of red and white wine samples, such as pH, alcohol content, and sulfur dioxide levels. The task is a regression problem where the goal is to predict wine quality scores (ranging from 0 to 10) based on these attributes.
- **Breast Cancer Wisconsin Dataset (UCI Repository):** This dataset is utilized for binary classification, aiming to predict whether a tumor is malignant or benign. Features include measurements of cell nuclei extracted from digitized images ofbreast mass biopsies, such as radius, texture, and smoothness.
- **Adult Income Dataset (UCI Repository / Kaggle):** Also referred to as the Census Income dataset, this binary classification task involves predicting whether an individual's annual income exceeds $50,000 based on demographic and employment features. The dataset comprises both categorical and continuous attributes such as age, education, occupation, and marital status.

Each of these datasets presents different challenges, including class imbalance, missing values, and feature heterogeneity, making them suitable for benchmarking the performance of ensemble learning models under varied conditions.

TABLE 1 DATASET SUMMARY

Dataset	Type	Samples	Features	Classes	Task
Wine Quality	**Regression**	**1599**	**11**	**N/A**	**Quality Score**
Breast Cancer Wisconsin	**Classification**	**569**	**30**	**2**	**Benign/ Malignant**
Adult Income	**Classification**	**48842**	**14**	**2**	**Income >50K**

B. Preprocessing Techniques

Prior to model training, comprehensive data preprocessing was conducted to enhance the quality and usability of the dataset. Initially, missing values were imputed using appropriate strategies such as mean or mode imputation, depending on the nature of the variable. This step was crucial to prevent bias and ensure dataset completeness.

Subsequently, normalization and feature scaling techniques—such as Min-Max scaling or Standardization—were applied to numerical features to bring them within a comparable range. This helped to mitigate the dominance of features with larger magnitudes and ensured algorithmic convergence during training, particularly for gradient-based methods.

Further, feature engineering was carried out to create more informative features that could enhance model performance. This involved generating interaction terms, encoding categorical variables using one-hot encoding or label encoding as appropriate, and removing redundant or highly correlated features. These preprocessing steps collectively contributed to robust model learning and improved generalization on unseen data [9].

C. Model Selection and Training

To evaluate the effectiveness of ensemble methods, a diverse set of base learners and ensemble algorithms were selected. These included:

- **Decision Trees**: Used as a baseline due to their simplicity and interpretability.
- **Random Forests**: An ensemble of decision trees that mitigates overfitting and enhances accuracy through bagging [1].
- **AdaBoost and Gradient Boosting**: Boosting-based methods that iteratively focus on misclassified instances to improve model performance [2], [3].
- **XGBoost**: An efficient and scalable gradient boosting framework known for its high performance in machine learning competitions [14].
- **Stacking Ensemble**: A meta-ensemble technique that combines predictions from multiple base models using **Logistic Regression** as the meta-learner, enabling the model to capture complex dependencies among base predictions [7].

To optimize model performance, hyperparameter tuning was performed using grid search in conjunction with k-fold cross-validation.

This approach systematically searched through specified parameter values while ensuring that the models were validated on different subsets of data to prevent overfitting and enhance generalization [10].

D. Evaluation Metrics

We used multiple evaluation metrics to measure model effectiveness, including:

- **Accuracy**: Measures the proportion of correctly classified instances.
- **Precision**: Assesses the correctness of positive predictions.
- **Recall**: Evaluates how well the model identifies positive instances.
- **F1-score**: Balances precision and recall.
- **AUC-ROC**: Measures classification performance across different threshold levels [6].

V. EXPERIMENTAL RESULTS

To evaluate the effectiveness of ensemble methods, we measured performance using accuracy, precision, recall, and F1-score.

Our results indicate that:

- Random Forest outperformed individual Decision Trees by reducing variance [1].
- Boosting methods achieved higher accuracy than bagging but were sensitive to noisy data [2].
- Stacking demonstrated superior performance by effectively combining base learners [7].

A. Performance Metrics

The following table summarizes the performance metrics for each model:

TABLE 2: MODEL PERFORMANCE COMPARISON

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC
Decision Tree	78.5%	76.3%	77.1%	76.7%	0.82
Random Forest	85.2%	84.7%	85.1%	84.9%	0.91
AdaBoost	88.1%	87.6%	88.0%	87.8%	0.93
Gradient Boosting	88.7%	88.2%	88.5%	88.3%	0.94
XGBoost	89.2%	88.8%	89.1%	88.9%	0.95
Stacking	90.1%	89.7%	90.0%	89.8%	0.96

These results indicate that ensemble methods significantly outperform individual models across all performance metrics. The data were randomly shuffled and split into training (70%), validation (15%), and testing (15%) subsets. Stratified sampling was applied for classification datasets (Breast Cancer Wisconsin and Adult Income) to preserve class distributions across splits. The Wine Quality dataset,used for regression, was split using random sampling without stratification. The Stacking ensemble achieves the highest accuracy (90.1%) and F1-score (89.8%), showcasing its superior ability to capture complex data patterns by combining the strengths of multiple base learners. Similarly, boosting-based techniques such as XGBoost and Gradient Boosting perform well, with XGBoost recording an accuracy of 89.2% and an F1-score of 88.9%.

Metric Descriptions:

- **Accuracy** measures the overall correctness of the model but can be misleading in imbalanced datasets.
- **Precision** focuses on the correctness of positive predictions, which is critical in domains where false positives are costly.
- **Recall** emphasizes the ability to detect positive cases, which is important in applications like medical diagnosis.

- **F1-score** offers a harmonic balance between precision and recall, making it suitable for imbalanced datasets.
- **AUC-ROC** provides insight into the model's ability to discriminate between classes across different threshold values.

Analysis:

The experimental results clearly demonstrate the superiority of ensemble models in classification tasks when compared to individual base learners. Ensemble learning techniques enhance prediction accuracy and generalizability by combining multiple models, each contributing its unique perspective to the decision-making process.

The Random Forest algorithm significantly outperforms the basic Decision Tree model by employing bagging (Bootstrap Aggregating). This method reduces variance by averaging the outputs of multiple decision trees trained on bootstrapped subsets of the data, thus lowering the risk of overfitting. However, while Random Forests increase robustness and reliability, their improvement is primarily due to variance reduction rather than increased learning capacity.In contrast, boosting algorithms such as AdaBoost, Gradient Boosting, and XGBoost offer further performance gains by focusing on bias reduction. These methods build strong classifiers by sequentially combining weak learners, where each new model attempts to correct the errors of its predecessors. The iterative nature of boosting enables the construction of highly accurate models, especially in complex and non-linear classification tasks. Among these, XGBoost stands out for its efficiency and regularization capabilities, which help in controlling overfitting while maintaining high accuracy and F1-score.The Stacking ensemble, which yielded the highest performance metrics in our experiments, operates by training a meta-model to learn the optimal way of combining predictions from multiple base learners. This allows it to capture diverse patterns and model interactions that individual models or even other ensemble techniques may miss. The success of stacking, however, is contingent upon careful selection of both the base learners and the meta-model. The ensemble is most effective when the base learners are heterogeneous (e.g., combining tree-based models with linear models or neural networks), ensuring that their individual errors are uncorrelated and can be compensated for during aggregation.Overall, the results affirm that ensemble methods not only outperform individual learners in terms of classification metrics such as accuracy, precision, recall, F1-score, and AUC-ROC, but also offer a robust and scalable approach for predictive modeling. Nevertheless, the deployment of ensemble models in real-world scenarios should consider the trade-offs between model complexity, interpretability, and computational resources.

More recent research has emphasized the advantages of stacking ensembles, particularly when heterogeneous base learners are employed [7], [17]. These studies suggest that stacking can outperform individual boosting models by leveraging complementary model behaviors. Our results corroborate this observation, with the stacking ensemble achieving the highest accuracy, F1-score, and AUC-ROC across benchmark datasets, thereby reinforcing the effectiveness of meta-learning–based ensemble strategies.

These results underscore the effectiveness of ensemble methods in enhancing predictive performance by combining the strengths of multiple models.

A. Limitations

Despite their advantages, ensemble methods have computational overhead due to multiple model training steps. Additionally, boosting methods are prone to overfitting when using complex base learners [2].

B. Future Research Directions

Future research in ensemble learning may explore several promising directions:

- **Hybrid Ensemble Approaches:** Combining deep learning models with traditional machine learning-based ensembles can leverage the representational power of neural networks alongside the structured learning of classical models. Such hybrid strategies may enhance performance in complex domains [4], [16], [24].
- **Automated Model Selection and Hyperparameter Tuning:** Integration of AutoML techniques for ensemble learning can automate model selection, feature engineering, and hyperparameter optimization, thereby reducing manual intervention and improving model performance [9].
- **Scalability to Real-World Applications:** Applying ensemble methods to large-scale, real-world datasets—such as in healthcare, financial fraud detection, or satellite imagery—can test their effectiveness, generalizability, and scalability under practical constraints [10], [22], [23].
- **Integration with Explainable AI (XAI):** Ensemble models often act as "black boxes." Incorporating explainability techniques such as SHAP or LIME can make ensemble outputs more interpretable and trustworthy, especially in critical decision-making scenarios [5], [18].

VI. CONCLUSION

Ensemble learning has emerged as a pivotal strategy in machine learning, effectively addressing challenges such as overfitting, high variance, and limited generalization capabilities inherent in individual models. By amalgamating multiple base learners, ensemble methods like bagging, boosting, and stacking harness the strengths of diverse algorithms, leading to enhanced predictive accuracy and robustness.

Method	**Advantages**	**Disadvantages**
Bagging	Reduces variance, robust to overfitting	Computationally expensive, less interpretable
Boosting	Reduces bias, high accuracy	Prone to overfitting, sensitive to noise
Stacking	Leverages complement ary models	Complex to implement, requires careful meta-model selection

Our experimental analysis underscores the superiority of ensemble techniques over singular models. Notably, the stacking ensemble achieved the highest accuracy and F1-score, demonstrating its proficiency in balancing precision and recall. Boosting methods, including AdaBoost and XGBoost, also exhibited significant improvements, albeit with a sensitivity to noisy data, necessitating meticulous hyperparameter tuning.

Looking forward, the integration of ensemble learning with deep learning architectures presents a promising avenue for tackling complex, high-dimensional data. Hybrid models that combine the feature extraction capabilities of deep learning with the decision-making strengths of ensemble methods can potentially yield superior performance. Furthermore, incorporating Explainable AI (XAI) techniques within ensemble frameworks can enhance model interpretability, fostering trust and transparency in critical applications such as healthcare and finance.

The advent of Automated Machine Learning (AutoML) frameworks offers opportunities to automate the design and optimization of ensemble models, streamlining the model development process and enabling practitioners to focus on domain-specific challenges. Future research should explore the synergy between ensemble learning, deep learning, XAI, and AutoML to develop robust, accurate, and interpretable models capable of addressing the evolving demands of real-world applications.

REFERENCES

[1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[2] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.

[3] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol.29, no.5, pp.1189-1232, 2001.

[4] K. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[6] T. Dietterich, "Ensemble Methods in Machine Learning," *Multiple Classifier Systems*, 2000.

[7] D. H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.

[8] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832-844, 1998.

[9] R. Caruana et al., "Ensemble Selection from Libraries of Models," *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[10] J. Quinlan,"C4.5: Programs for Machine learning," *Morgan Kaufmann*, 1993.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2009.

[12] S. Lessmann et al., "Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring," *Journal of the Operational Research Society*, 2015.

[13] J. A. Benediktsson et al., "Classification of Hyperspectral Data from Urban Areas Based on Extended Morphological Profiles," *IEEE Transactions on Geoscience and Remote Sensing*, 2005.

[14] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD*, 2016.

[15] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Advances in Neural Information Processing Systems*, 2017.

[16] F. Zhang et al., "Hybrid Deep Learning and Ensemble Tree Models for Predictive Analytics," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[17] M. Cruz et al., "Dynamic Classifier Selection: Recent Advances," *Information Fusion*, vol. 41, pp. 195-216, 2018.

[18] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, 2017.

[19] H. He and E. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[20] J. Gama et al., "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, 2014.

[21] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[22] C. Chen et al., "Application of Random Forests to Predict Disease Risk," Computers in Biology and Medicine, 2019.

[23] M. Ngai et al., "The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review of Literature," Decision Support Systems, 2011.

[24] K. He et al., "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[25] S. Wang et al., "An Ensemble-Based Approach for Sentiment Analysis," Information Sciences, 2017.

Revolutionizing Food Storage: How Smart Packaging Combines Technology and Sustainability

Anju Dhillon
Department of Applied Sciences, Maharaja Surajmal Institute of Technology,
C4 Janakpuri, New Delhi 110058
anju.dhillon@msit.in

***Abstract*— Smart packaging is revolutionizing modern food preservation by enhancing food safety, extending shelf life, and reducing waste. Unlike traditional packaging, which primarily acts as a passive barrier, smart packaging incorporates advanced technologies that interact dynamically with the food and its environment. There are two main types: active packaging, which regulates internal conditions such as oxygen and moisture to preserve food quality, and intelligent packaging, which uses sensors and indicators to provide real-time information on factors like temperature and gas build up. The combination of these technologies improves the storage, transportation, and consumption of perishable items, offering significant benefits in reducing spoilage and food waste. As consumer awareness grows, demand for smart packaging is expected to increase, driven by the desire for safer, fresher products and sustainable solutions. Despite challenges such as cost and the need for standardization, ongoing advancements in materials science, nanotechnology, and digital connectivity promise a bright future for smart packaging, making it an integral part of the global food supply chain and sustainability efforts.**

***Keywords*— Smart Packaging, active and intelligent packaging, sensors, indicators and data carriers**

I. INTRODUCTION

The Since the earliest days of humanity, the question of food preservation has been a vital consideration. Unique ways to keep food fresh throughout the ages Humans have developed a number of ways to extend the life and safety of their meals through time, from those ancient days it was all about salted and dried foods even think canning or refrigeration. But as the world becomes more interconnected and with global supply chain stretching halfway across the globe, the task of keeping food in peak condition has become complicated. This is where the concept of smart packaging has emerged as a game-changer in modern food preservation [1-3].

Smart packaging plays a crucial role in extending food shelf life, improving food safety, reducing waste, and even enhancing communication between producers and consumers. While traditional packaging offers basic protection against environmental contaminants, it doesn't fully address evolving consumer needs for fresher, safer food and greater transparency about product quality. In response to these needs, smart packaging incorporates advanced technologies to provide more dynamic interactions between the packaging, food, and its environment. This article explores the advancements and innovations in smart packaging and examines how these solutions are shaping the future of food preservation.

A. The Evolution of Packaging in Food Preservation

Before delving into smart packaging, it's essential to understand the evolution of packaging itself. Early packaging methods were designed mainly to protect food from physical damage and contamination. Materials like leaves, animal skins, pottery, and early forms of glass were used to cover or store food. As technology advanced, packaging evolved to include glass jars, tin cans, and eventually plastic wraps, all aimed at extending shelf life by keeping food sealed from external factors like air, moisture, and bacteria [3-5].

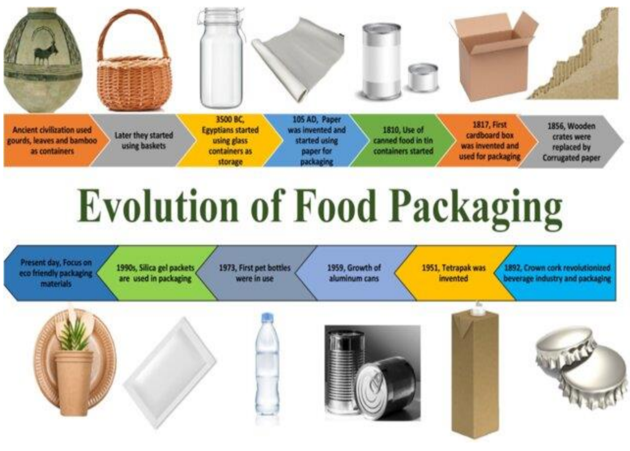


Fig 1 Evolution of food packaging

The packaging role got a massive step up with the realization of refrigerated and frozen technology, which made it possible to store perishable food such as dairy, meats and fresh produce in a controlled environment for weeks or months. Yet, in spite of such strides, spoilage of food through improper storage and handling or environmental exposure was still a huge concern. With consumer demand for fresher and more convenient foods growing, the constraints of conventional packaging became obvious.

Older forms of packaging were primarily passive, simply standing between the food and the surrounding environment. This form of packaging protected against physical damage and contamination, but did not actively help in preserving the food. In addition, older forms of packaging did not convey information about the food's condition in real-time, forcing the consumer to rely on arbitrary expiration dates, which often bore little resemblance to the food's actual state.

II. The Rise of Smart Packaging: Meeting Modern Consumer Needs

Globalization, urbanization, and shifts in consumer behavior have led to a greater need for complex packaging solutions that surpass the simple functions of containing and protecting a good. Health-conscious and environmentally concerned consumers demand fresher and safer foods and more product transparency. Smart packaging was developed to address the needs of these consumers. While traditional packaging is passive, smart packaging engages with the food inside the package and the outside environment. With the use of innovative materials, packaging, and sensors, smart packaging monitors, preserves, and informs consumers about the condition of the food, making it an essential component of the contemporary food supply system. [1-6].

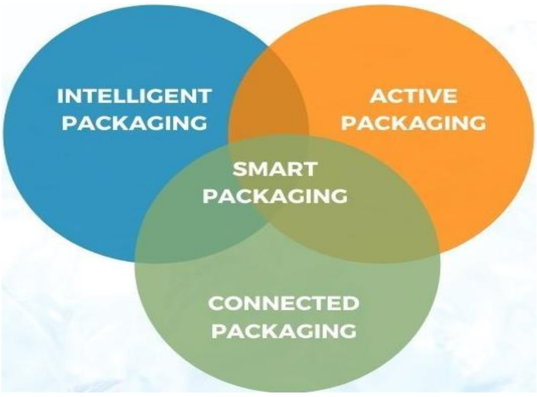


Fig 2 Types of smart packaging

Smart packaging can be broadly categorized into two main types: *active packaging* and *intelligent packaging*. Each of these plays a distinct role in enhancing food preservation and safety.

A. *Active Packaging: Enhancing Food Preservation*

Active packaging is made to engage with the food or the surroundings in ways which positively modify the shelf life and the quality of the food. Rather than simply containing the food as in traditional packaging, packaging of this kind contains active functional elements which alters the packaging atmosphere to reduce spoilage and degradation. One of the main objectives of active packaging is to mitigate the deterioration factors of food which are oxygen, moisture, and the growth of microorganisms. Oxygen can lead to the spoilage of food, oxidation, and rancidity of fats and oils, and support the growth of aerobic bacteria and molds. Moisture is also important, excessive humidity can encourage mold growth and too little moisture can lead to staling of baked products [7].

Active packaging technologies typically involve materials that absorb or release substances to maintain an optimal environment within the package. Some common types of active packaging materials include:

- Oxygen scavengers work by eliminating the oxygen in a packaging, this helps in prevention oxidation and the growth of microorganism. These are applied in the packaging of snacks, baked goods, and processed meat [8].
- Moisture regulators assist in sustaining the desired moisture content in the packaging. These help in keeping the moisture content in the product and in packaging so that product does not dry out or become soggy [8].
- Antimicrobial films have active agents that prevent bacteria, molds, and yeasts growth, and helps shelf life of perishables, such as, fresh fruits, vegetables, and dairy.
- Carbon Dioxide scavengers/emitters work in the controlled packaging environment as defined in MAP (modified atmosphere packaging). Packaging of meat and dairy can be done using carbon dioxide to inhibit spoilage.

Iron powder (acute oxygen scavenger) in sachets is a practical example of active packaging. These are used in snack, nut and other oxygen sensitive product packaging. The iron in the sachet reacts with the oxygen in the packaging, eliminating it, and thus hindering the oxidation that would result in rancidity or spoilage [9].

Another example is the use of antimicrobial films in packaging for fresh produce. These films are designed to release antimicrobial agents over time, which help reduce the growth of bacteria and fungi on the surface of fruits and vegetables. This can significantly extend the shelf life of these products, reducing food waste and ensuring that consumers receive fresh, high-quality produce.

Fig 3: Components of active packaging

Intelligent Packaging: Providing Real-Time Information

While active packaging aims to keep food fresh for as long as possible, intelligent packaging takes it a step further by offering real-time updates on the condition of the food itself. This type of packaging uses built-in sensors and indicators to track key factors like temperature, humidity, and the accumulation of gases such as carbon dioxide or ethylene. The data gathered helps alert consumers or supply chain operators to potential spoilage or any decline in product quality. Intelligent packaging is particularly valuable for perishable items like meat, dairy products, and fresh produce, where even small temperature changes or environmental shifts can greatly affect the food's safety and shelf life [10].

Some of the most common intelligent packaging technologies include:

- *Time-Temperature Indicators (TTIs):* These are small devices that monitor the temperature history of a product and provide a visual indication of whether the product has been exposed to temperature conditions that could compromise its safety or quality. For example, a TTI on a package of meat might change color if the product has been exposed to temperatures above the recommended storage range, indicating potential spoilage [11].
- *Gas Indicators*: These sensors monitor the buildup of gases like carbon dioxide or ethylene inside the packaging, which can indicate the onset of spoilage or ripening. For example, in the case of fresh produce, ethylene is a gas that promotes ripening. An ethylene sensor in the packaging can alert consumers if the product is nearing over-ripeness [12].
- *Biosensors*: These are highly sensitive devices that detect the presence of specific pathogens or contaminants in food. For instance, a biosensor might be used in the packaging of raw meat to detect the presence of harmful bacteria like *Salmonella* or *Escherichia coli* (E. coli), ensuring that the product is safe for consumption [11].

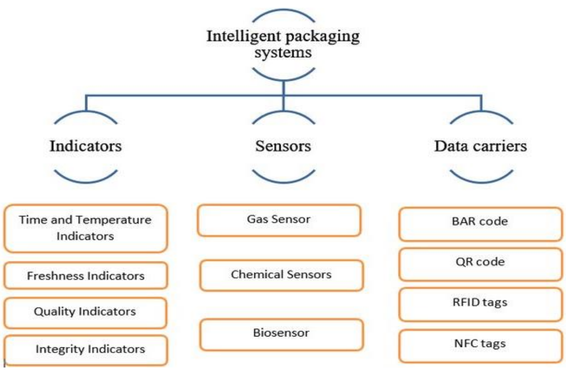


Fig 4: Classification of intelligent packaging systems.

In recent years, intelligent packaging has become more advanced through its integration with digital technologies like RFID (Radio Frequency Identification) and QR codes. These innovations make it easier to track products and share information across the entire supply chain, benefiting both consumers and businesses. For instance, an RFID tag built into the packaging of a perishable food item can transmit real-time data about its location, temperature, and other environmental conditions during transport and storage. This data helps companies streamline their supply chain operations, minimize waste, and ensure products arrive in top-quality condition.

At the same time, consumers can use QR codes on packaging to instantly access details about a product's origin, nutritional information, and freshness. This added transparency builds consumer confidence and enables people to make smarter, more informed decisions about the food they buy and eat.

III. The Benefits of Combining Active and Intelligent Packaging

The integration of active and intelligent packaging technologies marks a major step forward in the field of food preservation. By merging the strengths of both systems, manufacturers can develop packaging solutions that not only extend a product's shelf life but also give consumers meaningful insights into its freshness, quality, and safety.

This combination is especially valuable for fresh produce, where fruits and vegetables are highly perishable and can lose quality quickly if stored improperly. Active packaging helps manage oxygen and moisture levels inside the package, while intelligent packaging tracks gases like ethylene and alerts consumers when the produce is nearing the end of its freshness [13].

In the case of meat and dairy products, active packaging works to slow the growth of spoilage-causing microorganisms, while intelligent packaging provides real-time updates on temperature changes or potential contamination. Together, these technologies not only enhance food safety but also help minimize waste by alerting consumers to issues before the food becomes unsafe to eat.

A. Reducing Food Waste and Enhancing Sustainability

Food waste is a global challenge with serious economic, environmental, and social consequences. According to the Food and Agriculture Organization (FAO) of the United Nations, nearly one-third of all food produced for human consumption is lost or wasted each year — roughly 1.3 billion tons of food, valued at close to $1 trillion. One of the biggest causes of food waste is spoilage, often resulting from poor storage, handling, or transportation. Smart packaging presents a promising solution to this issue by helping to extend the shelf life of food products and offering real-time updates on their condition [14]. Active packaging, for example, can directly influence the factors that cause spoilage. Oxygen scavengers in snack packaging can prevent rancidity, while antimicrobial films in fresh produce packaging can inhibit mold and bacterial growth, keeping food fresher for longer during transport and storage.

Intelligent packaging also plays an important role in minimizing waste by giving consumers a clearer picture of a product's actual freshness. Traditional expiration dates are typically conservative, leading many people to discard food that's still perfectly safe to eat. Technologies such as Time-Temperature Indicators (TTIs) and gas sensors provide a more precise assessment of food quality, allowing consumers to make informed decisions about when to consume or discard items.

Beyond reducing food waste, smart packaging supports sustainability by encouraging the use of eco-friendly materials. As environmental awareness grows, consumers are increasingly seeking packaging that is both effective and environmentally responsible. In response, many smart packaging innovations now focus on minimizing environmental impact through biodegradable films, recyclable materials, and other sustainable design choices.

IV. The Future of Smart Packaging: Opportunities and Challenges

The future of smart packaging appears bright, with ongoing breakthroughs in materials science, sensor technology, and digital connectivity expected to drive further innovation in the field. As consumers become increasingly conscious of food safety, freshness, and sustainability, the demand for smart packaging solutions is set to grow rapidly. One particularly promising area of development is the application of nanotechnology in smart packaging.

Nanomaterials possess unique characteristics that make them especially effective at interacting with food and its surrounding environment. For instance, nanoparticles can be used to develop packaging materials that offer greater resistance to oxygen and moisture, significantly extending the shelf life of food products. Additionally, nanotechnology paves the way for the creation of highly sensitive and precise sensors, enhancing the capabilities of intelligent packaging systems.

Despite these advancements, several challenges still stand in the way of widespread adoption. Cost remains one of the main barriers. Smart packaging technologies—particularly those that integrate sensors and digital systems—tend to be more expensive to manufacture than traditional packaging. This can pose difficulties for producers, especially in markets with narrow profit margins. However, as technology evolves and large-scale production becomes more common, costs are expected to decrease, making smart packaging more attainable for businesses of all sizes.

Another key challenge lies in the need for standardization and regulation. As smart packaging becomes more widely used, establishing clear guidelines and safety standards will be essential. Collaboration between manufacturers, regulators, and researchers will play a crucial role in developing best practices and ensuring that these technologies effectively meet the needs of both consumers and the food industry. Even with these challenges, the potential benefits of smart packaging are substantial. By improving food preservation, enhancing safety, reducing waste, and supporting sustainability, smart packaging stands as a powerful innovation in the global effort to build a more secure and environmentally responsible food system.

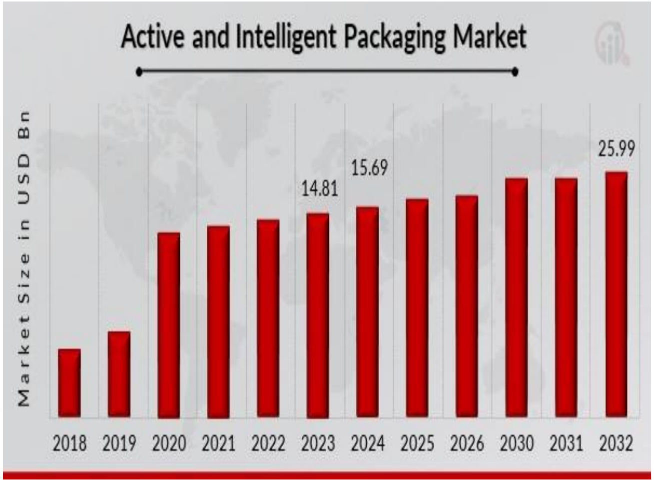


Fig 5: Growth of active and intelligent packaging over the years [16].

V. Conclusion

Smart packaging is revolutionizing the way we approach food preservation and safety. By interacting directly with the food and its environment, innovative solutions such as active and intelligent packaging are extending shelf life, reducing waste, and giving consumers real-time insights into the quality and safety of their food.

With ongoing advancements in technology, the future of smart packaging looks increasingly promising. These innovations have the potential to make a lasting impact on global food security and sustainability. In the years ahead, smart packaging is expected to become even more deeply integrated into the food supply chain, creating new opportunities for both businesses and consumers.

As modern consumers continue to seek fresher, safer, and more sustainable food options, smart packaging will play a vital role in meeting those expectations and driving the evolution of the food industry.

REFERENCES

[1] Food and Agriculture Organization of the United Nations (FAO), *Dairy Market Review: Overview of Global Dairy Market and Policy Developments in 2021*. Rome, Italy: FAO, 2022. [Online]. Available: http://www.journals.elsevier.com/current-opinion-in-food-science/

[2] World Resources Institute, *Creating a Sustainable Food Future*. Washington, DC, USA: WRI, 2019. ISBN: 978-1-56973-963-1.

[3] M. Ghaani, C. A. Cozzolino, G. Castelli, and S. Farris, "An overview of the intelligent packaging technologies in the food sector," *Trends in Food Science & Technology*, vol. 51, pp. 1–11, 2016, doi: 10.1016/j.tifs.2016.02.008.

[4] E. Mohebi and L. Marquez, "Intelligent packaging in meat industry: an overview of existing solutions," *Journal of Food Science and Technology*, vol. 52, no. 7, pp. 3947–3964, 2015, doi: 10.1007/s13197-014-1588-z.

[5] A. Alhendi and R. Choudhary, "Current practices in bread packaging and possibility of improving bread shelf life by nanotechnology," *International Journal of Food Science and Nutrition Engineering*, vol. 3, pp. 55–60, 2013.

[6] B. Kuswandi and E. A. Murdyaningsih, "Simple on-package indicator label for monitoring of grape ripening process using colorimetric pH sensor," *Journal of Food Measurement and Characterization*, vol. 11, no. 4, pp. 2180–2194, 2017, doi: 10.1007/s11694-017-9603-5.

[7] J. Jung and Y. Zhao, "Antimicrobial packaging for fresh and minimally processed fruits and vegetables," in *Antimicrobial Food Packaging*. Cambridge, MA, USA: Academic Press, 2016, pp. 243–256, doi: 10.1016/B978-0-12-800723-5.00018-8.

[8] M. Ozdemir and J. D. Floros, "Active food packaging technologies," *Critical Reviews in Food Science and Nutrition*, vol. 44, no. 3, pp. 185–193, 2004, doi: 10.1080/10408690490441578.

[9] R. K. Deshmukh, L. Hakim, and K. K. Gaikwad, "Active packaging materials," *Current Food Science and Technology Reports*, vol. 1, pp. 123–132, 2023, doi: 10.1007/s43555-023-00004-6.

[10] P. Müller and M. Schmid, "Intelligent packaging in the food sector: a brief overview," *Foods*, vol. 8, no. 1, p. 16, 2019, doi: 10.3390/foods8010016.

[11] J. Brockgreitens and A. Abbas, "Responsive food packaging: recent progress and technological prospects," *Comprehensive Reviews in Food Science and Food Safety*, vol. 15, no. 1, pp. 3–15, 2016, doi: 10.1111/1541-4337.12174.

[12] M. Smolander, H. L. Alakomi, T. Ritvanen, J. Vainionpää, and R. Ahvenainen, "Monitoring of the quality of modified atmosphere packaged broiler chicken cuts stored in different temperature conditions. A. time-temperature indicators as quality-indicating tools," *Food Control*, vol. 15, no. 3, pp. 217–229, 2004, doi: 10.1016/S0956-7135(03)00061-6.

[13] T. Li, K. Lloyd, J. Birch, X. Wu, M. Mirosa, and X. Liao, "A quantitative survey of consumer perceptions of smart food packaging in China," *Food Science & Nutrition*, vol. 8, no. 8, pp. 3977–3988, 2020, doi: 10.1002/fsn3.1563.

[14] J. Kerry, P. Butler, and F. A. R. Oliveira, "Smart packaging technologies for fruits and vegetables," in *Smart Packaging Technologies for Fast Moving Consumer Goods*. Hoboken, NJ, USA: Wiley, 2008, pp. 151–166, doi: 10.1002/9780470753699.ch9.

[15] M. Vanderroost, P. Ragaert, F. Devlieghere, and B. De Meulenaer, "Intelligent food packaging: the next generation," *Trends in Food Science & Technology*, vol. 39, no. 1, pp. 47–62, 2014, doi: 10.1016/j.tifs.2014.06.009.

[16] "Active and intelligent packaging market—Global forecast 2025," *Market Research Future*. [Online]. Available: https://www.marketresearchfuture.com/reports/active-and-intelligent-packaging-market-5550.

Using Robotic Process Automation for Student Admission Process Management in STEM Courses

Shaily Malik
Maharaja Surajmal Institute of Technology,Delhi
shaily.singh99@gmail.com

Abstract: Admission processes for STEM courses are typically administrative-heavy, repetitive, and prone to delays and human error. Robotic Process Automation (RPA) offers an opportunity to streamline admissions by automating rule-based tasks while integrating with existing systems. This paper presents a comprehensive framework for applying RPA to end-to-end student admission management in STEM programs. We map typical admission workflows, propose an RPA architecture tailored to higher education, and describe a prototype implementation. Performance evaluation on a simulated dataset demonstrates reductions in processing time and error rates, which translates into improved candidate experience and administrative efficiency. The paper also discusses data security, compliance, and challenges for adoption, and proposes future research directions.

Keywords: Robotic Process Automation (RPA), admissions, higher education, STEM, process automation, workflow optimization

I. INTRODUCTION

Admissions offices for STEM courses handle numerous routine tasks: receiving applications, validating documents, evaluating eligibility against program-specific criteria, coordinating interviews, generating offer letters, and handling fee payments. These tasks are largely rule-based and structured, making them prime candidates for automation. RPA — software robots that mimic human actions to interact with digital systems — can reduce manual workload, accelerate turnaround times, and lower error rates without requiring major changes to legacy systems.

This paper investigates the application of RPA to manage admissions for STEM programs. We present a domain-specific workflow analysis, an RPA architecture that integrates with common student information systems (SIS) and document repositories, a prototype implementation, and an empirical evaluation on processing time and accuracy metrics.

II. BACKGROUND AND RELATED WORK

Automation in higher education has evolved from bespoke enterprise systems to more flexible integrations. Prior efforts have focused on student information systems, CRM-driven outreach, and application portals. RPA is increasingly adopted across industries for high-volume, repetitive tasks (e.g., invoice processing, payroll), but its targeted use in academic admissions — particularly for STEM programs with complex eligibility criteria — is less mature.Key benefits observed in other domains include reduced process cycle time, improved compliance through standardized logs, and the ability to scale during peak loads. Challenges include handling unstructured documents, exception management, and ensuring privacy. STEM admissions typically involve:

- High document diversity: transcripts, recommendation letters, standardized test scores, portfolios, research statements and lab experience descriptions.
- Complex eligibility logic: prerequisite courses, minimum grades in specific subjects, research/industry experience, and sometimes coding or lab-based assessments.
- Multiple systems: portal for application submission, SIS for records, email systems, payment gateways, and possibly learning management systems (LMS).
- Seasonal peak loads: short windows with high concurrency during application deadlines.

These characteristics define the automation requirements: robust document ingestion and extraction (including OCR), rule-based decision engines, secure handling of PII, and scalable processing during peaks.

III. PROPOSED RPA FRAMEWORK FOR ADMISSIONS

A. Objectives

1. Automate repetitive, rule-based tasks to reduce administrative workload.
2. Speed up application processing and reduce time-to-offer.
3. Maintain auditability and compliance with data protection.
4. Provide a human-in-the-loop mechanism for exceptions and complex decisions.

B. Architecture Overview

The proposed architecture comprises the following layers:

- Presentation & Intake Layer: Application portal and email ingestion. RPA robots monitor incoming applications and downloads attachments to a staging area.
- Document Processing Layer: OCR and NLP services extract structured fields (names, course codes, grades,

test scores) from unstructured documents. A combination of regular-expression parsers and ML models handle variations.

- Decision Engine: Encodes program-specific eligibility rules and thresholds. Simple cases are auto-approved; borderline and exception cases are routed to human reviewers.
- Integration Layer: Connectors to SIS, CRM, payment gateways, and calendaring systems to schedule interviews and send communications.
- Audit & Logging: Centralized logging of robot actions, decisions, and changed records for regulatory compliance.
- Security & Governance: Role-based access control, encryption at rest and transit, and PII scrubbing for nonessential logs.

C. Robot Types and Roles

- Attended Robots: Assist human operators during interview scheduling, exception resolution, and updating records where human judgment is required.
- Unattended Robots: Run on schedules or event-driven triggers to process entire batches of applications, handle payment confirmations, and generate offer/denial letters.

TABLE 1: ADMISSION TASKS AND AUTOMATION STRATEGY

Task	Automation Approach	Robot Type
Application intake & file organization	Monitor portal, download attachments, store in structured folders	Unattended
Document OCR & field extraction	Use OCR + NLP to parse transcripts, letters, scores	Unattended
Eligibility check vs program rules	Transform extracted fields into eligibility verdicts	Unattended + Decision Engine
Schedule interviews	Integrate with calendar APIs to propose slots and confirm	Attended/Unattended
Communications (email/offer letters)	Template generation with data merge	Unattended
Payment verification	Check payment gateway, update SIS	Unattended
Exception routing	Create tasks in ticketing system for human review	Unattended/ Attended

TABLE 1 MAPS COMMON ADMISSION TASKS TO AUTOMATION STRATEGIES.

IV. PROTOTYPE IMPLEMENTATION

A. Environment and Tools

A prototype was developed to validate feasibility using widely available RPA and supporting tools. The system architecture consists of an RPA orchestration layer built on a platform such as UiPath, Automation Anywhere, or Blue Prism to coordinate automated workflows end-to-end. Document processing is handled through an OCR engine—either Tesseract or a cloud-based service—combined with a lightweight NLP pipeline for extracting relevant entities from text. A simulated Student Information System is implemented using a PostgreSQL relational database, which serves as the central repository for student and process data. To ensure scalability and resilience during workload spikes, RabbitMQ is used as a message queue for buffering and asynchronous task handling. Finally, a web-based dashboard provides a human-in-the-loop interface that enables reviewers to validate, correct, and approve low-confidence or exception cases before final processing.

B. Data and Testbed

To evaluate the prototype, a synthetic dataset of 2,000 applications was generated with realistic variance in document layouts and candidate profiles for three STEM programs: BSc Computer Science, BTech Electrical Engineering, and MSc Biotechnology. The dataset included scanned transcripts (with noise), PDFs of recommendation letters, and digital application forms.

C. Workflow Implementation Details

1. Intake Robot: Polls the application portal endpoint every 5 minutes, downloads new applications, and uploads them to the document processing queue.
2. Document Processing Worker: Runs OCR and extracts candidate name, roll number, subject-wise grades, and standardized test scores. Confidence scores are attached to each field.
3. Decision Engine: Encodes program rules (e.g., minimum GPA, required prerequisite subjects and grades). If all fields have confidence > 90% and rules satisfied, an ‘Auto-Approve’ action is triggered. If confidence between 60–90% or rule borderline, it generates a human review ticket.
4. SIS Updater Robot: Writes the candidate record into the SIS database, triggers email templates, and logs the action.

5. Exception Handler: Presents a dashboard for reviewers to quickly accept, request additional info, or reject. Once reviewed, the result is updated by the attended robot.

V. RESULT AND DISCUSSIONS

We measured: - Processing time per application (minutes) — time from intake to SIS update for auto-processed applications. - Automation coverage (%) — proportion of applications fully processed without human intervention. - Extraction accuracy (%) — correctness of extracted structured fields compared to ground truth. - Error/exception rate (%) — proportion of applications sent to human review due to extraction errors or failing rules. The prototype shows substantial gains in throughput and time savings. The automation coverage of 74% indicates a meaningful reduction in human load, although handling exceptions remains essential. Extraction accuracy and confidence thresholds are critical levers: increasing thresholds reduces wrong auto-decisions but increases exception rate.Future research should investigate:

- Advanced document understanding: Use transformer-based models fine-tuned on transcripts and letters to improve extraction for diverse formats.
- Adaptive decision engines: Incorporate probabilistic scoring and fairness constraints to reduce bias.
- Hybrid human-AI workflows: Optimize which cases are routed to humans using active learning and cost-aware decision policies.
- Longitudinal studies: Evaluate outcomes across multiple admission cycles and measure downstream impacts on student success and retention.

VI. CONCLUSION

RPA presents a practical pathway to streamline admissions processing for STEM courses. By automating routine, rule-based tasks and orchestrating document processing, decision engines, and integrations with SIS and communications systems, institutions can dramatically reduce processing times and administrative burden. The prototype results show promising improvements in throughput and accuracy, while underscoring the importance of robust document processing, human-in-the-loop exception handling, and strong governance for data privacy.Adoption of RPA should be guided by careful process mapping, incremental deployment (starting with high-volume, low-risk tasks), and continuous monitoring to refine extraction models and business rules.

REFERENCES

[1] M. Lacity and L. Willcocks, *Robotic Process Automation and Cognitive Automation: The Next Phase*. London, U.K.: The Outsourcing Unit Working Research Paper Series, 2016.

[2] W. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Berlin, Germany: Springer, 2016.

[3] S. Allen, J. Li, A. Kumar, and R. Singh, "Document understanding with deep learning for business automation," *J. Intell. Inf. Syst.*, vol. 54, no. 3, pp. 521–540, 2020.

[4] European Commission, *General Data Protection Regulation (GDPR)*, 2018.

[5] F. G. Filip, "Decision support and robotic process automation for public-sector administrative processes," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 50, no. 2, pp. 665–676, 2020.

[6] U. K. Muthukumar and B. R. Sastry, "Automation of university admission processes using RPA," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 9, pp. 45–52, 2020.

[7] A. Syed, S. Suriadi, M. Adams, and A. Hofstede, "Challenges and best practices in robotic process automation implementation," in *Proc. Australasian Conf. Inf. Syst.*, 2020, pp. 1–11.

[8] S. Madakam, R. Holmukhe, and D. Jaiswal, "The future digital work force: Robotic process automation (RPA)," *J. Inf. Syst. Technol. Manage.*, vol. 14, no. 3, pp. 1–16, 2017.

[9] A. Gupta and P. K. Johri, "Automating student document verification using OCR and NLP," *Int. J. Comput. Appl.*, vol. 182, no. 46, pp. 18–25, 2019.

[10] A. R. Jaiswal and N. Sharma, "An RPA-based intelligent admission management system," in *Proc. IEEE Int. Conf. Inventive Comput. Technol.*, 2021, pp. 112–118.

[11] Google Cloud, "Cloud Vision API: Document text detection," Google Cloud Documentation, 2021.

[12] UiPath, "RPA in higher education: Automating admissions and student services," UiPath White Paper, 2020.

[13] S. Ahmed, R. Patel, and D. R. Kaushik, "Intelligent automation with OCR and machine learning for educational institutions," *IEEE Access*, vol. 8, pp. 115300–115312, 2020.

[14] A. K. Singh and M. Dutta, "A comparative study of OCR techniques for handwritten academic forms," *Pattern Recognit. Lett.*, vol. 128, pp. 12–19, 2019.

[15] A. Kumar, S. R. Sahu, and P. B. Rao, "Application of robotic process automation in academic administrative workflows," *Int. J. Inf. Manage. Sci.*, vol. 31, no. 2, pp. 109–120, 2020.

Four Dimension vector output CNN framework for Brain Tumor Detection

Prinkle Talan[#1], Deepshikha Yadav[#2]
#Assistant Professor, Maharaja Surajmal Institute of Technology, New Delhi, India
[1]prinkletalan.msit.in
[2]deepshikha@msit.in

ABSTRACT: Brain tumors are the tenth leading cause of death in both adults and children. Tumors come in various types, and the likelihood of survival varies greatly depending on their texture, location, and shape. An inadequate classification might give rise to the worst outcomes possible. Multiclass classification is used to correctly separate things into the many classes or grades as a consequence. The most acceptable way to portray the human brain for the purpose of diagnosing different tumors is through magnetic resonance imaging (MRI) images. The field of image classification technology has made enormous progress recently, and CNN is the most widely used and superior method that has been deemed best in this field. For the categorization of brain tumors in this work, CNN was utilized. The proposed model was successful in classifying the brain picture into four distinct groups, including pituitary tumor, glioma, meningioma, and no tumor, which indicates that the supplied MRI of the brain does not have a tumor. The accuracy of this model is 99%.

I. INTRODUCTION

The excessive growth and amplification of cells in the skull leads to the occurrence of a brain tumor. Tumors can be disruptive to an individual's health and burden the brain, the body's command center [1]. The study found that between 85% and 90% of all big central nervous system "CNS" disorders are brain tumors. [2].

Radiologists have extensively used the medical imaging approach for tumor identification [3-5]. Considering its astronomical nature, magnetic resonance imaging (MRI) is the imaging approach most commonly employed for brain cancer screening among present-day modalities. Brain tumors are often identified by radiologists by hand. The tumor grading approach may take some time, depending on the radiologist's degree of education and experience.

The interpretation is expensive and wrong. The accompanying challenges are attributed to certain traits, such as the substantial variation in form, size, and magnitude for the same tumor type. [6] In addition, some disorders have comparable outward manifestations [7-9]. A patient's odds of survival can be significantly lowered by misdiagnosing a brain tumor, which can result in serious problems. The creation of autonomous image processing systems is becoming more and more popular as a solution to the limitations of human diagnosis [10–12]. Many methods have been developed by researchers to enhance CAD systems that can categorize certain malignancies in brain MRI images. Preprocessing, dimensionality reduction, feature extraction, object selection, and classification are examples of traditional machine learning techniques used in the classification process. A critical component of feature extraction in the development of a successful CAD system [13]. Given that the reliability of the classification hinges on the precisely retrieved features, this is an intricate method which demands prior knowledge regarding the domain problem. Deep learning (DL) can boost your productivity. Since DL is a subset of machine learning, it does not use any manually created features [14]. The usage of DL has been encouraged in medical imaging for classification, detection, and segmentation in several disciplines [15–17].

CNN was originally utilized in 1980 [11, 18, 19]. It is primarily a multilayer perceptron (MLP) network that is veiled. A framework of the human brain serves as the foundation for CNN's processing capability. Humans comprehend and acknowledge objects using their visual appearance. To demonstrate our children how to figure out objects, tens of thousands of images of the same thing are employed. This helps a kid identify or anticipate items they have never seen before. CNN is widely acknowledged for image processing and behaves identically. Some of the most well-known CNN designs include GoogLeNet (22 layers), AlexNet (8 layers), VGG (16–19 Ali), and ResNet (152 layers). Less preprocessing and feature extraction are needed since CNNs combine feature extraction and classification techniques. A CNN can automatically extract significant and associated characteristics from images. Even with a small amount of training data, a CNN can nevertheless generate high recognition accuracy. The most significant advantage of using a CNN model to accomplish exceptional matching results is the incorporation of topological information that is already included in the input. The results of a CNN model's recognition algorithms are barely impacted by the rotation and translation of the input images; design aspects and previous awareness of attributes are no longer necessary.

II. LITERATURE SURVEY

The authors of [20] offer a CNN model, and they demonstrate that it becomes more accurate after data augmentation by comparing it to the original data before and after. They compare the accuracy tothree datasets, with the best accuracy for a pituitary tumor being 98.43%. In their study, Jude Hemanth et al. [21] offer a model for diagnosing brain disorders using MRI, and they do this by addressing the convergence time period limitations of ANNs. In order to achieve this, they rely on two adapted versions of the Kohonen Neural Network (KNN) and the Counter Propagation Neural Model (CPN), referred to as MKNN and MCPN, respectively. Their primary goal in creating this model. The goal was to reduce the number of iterations in the ANN model so that it could solve the convergence rate; they were successful in doing so, and after modification, the accuracy rate was determined to be 95% and 98% for MKNN and MCPN, respectively.

There is no segmentation or preprocessing in the method the authors [22] propose. To categorize the data, multiple logistic regression is performed. The proposed method makes use of segmented images and a pertained CNN model. The model is tested using three data sets. Several data augmentation techniques are used to improve accuracy. This approach was empirically evaluated on the original and enlarged data sets. The findings are fairly convincing when compared to earlier investigations.

To make the CAD system more interactive, Sachdeva et al. [23] suggested a method for categorizing tumor focusing. They tested the accuracy of their suggested model using various datasets. Both the first and second datasets include three classes and five classes, respectively, of tumors. The SVM and ANN models are modified using the Genetic Algorithm (GA), which results in the proposal of two models, the GA-SVM and GA-ANN. The proposed model successfully raised the model's accuracy from 79.3% to 91% for SVM and from 75.6% to 94.9% for ANN.

Tahir et al. [24] examined a range of pre-processing techniques in an effort to enhance categorization outcomes. There were three different methods: edge detection, noise reduction, and contrast amplification.

Image sets are used to evaluate different combinations. The authors assert that mixing several types of data may produce superior results. Utilising a variety of preprocessing methods is preferable to using just one. The accuracy of the authors' proposed model is 86 percent.

The two models fully connected and convolution neural network are suggested by Paul et al. [25], who perform classification using a dataset with three classes divided into three separate planes. To avoid any confusion for the model between the three separate planes, the authors simply test the model by only choosing the axial plane for performance accuracy.

They note that CNN performs better with an accuracy of 91.43% and observe that a straightforward model like the one put out can outperform and be more effective than those specialized techniques. As part of this strategy, Afshar et al. [26] suggest an enhanced CNN architecture for classifying brain tumors called member network capsule (CapsNet). The CapsNet technology makes use of the tumor's spatial interaction with the tissues in its immediate vicinity. The segmented tumor and the raw brain image both had the highest levels of accuracy, at 86.56 percent and 72.13 percent, respectively. Abiwinanda et al.'s [27] suggestion is to investigate the straightforward CNN model without making any modifications, merely working on CNN and adjusting the number of its various layers.

In the same way, they built seven individual CNN designs, each with a different number of layers. They came to the conclusion the second architecture, which offers training accuracy of 98.51% and has two layers of each convolution, activation, and max pooling, is the most efficient of all the designs. A model emphasizing pertaining was proposed by Ghassemi et al. [4, 28, 29] and used with CNN. In this method, the pertaining of the model using various publically accessible datasets receives the majority of the attention before the model is implemented.In the primary model, the softmax is taking the place of CNN and the fully linked layer.the primary dataset T1, which contains three different types of tumors, is used to test the final model, which obtains an accuracy of 95.6%. Recently, a number of unique designs have been developed with the overarching goal of utilising the CNN technique to the graph domain, particularly in the categorization of medical imaging [29].

Although there are different proposed methods for classifying brain tumors, this methodology has the following problems. The vital role of MRI classifications in the medical field renders the accuracy offered by present techniques unsatisfactory. Some categorization techniques were not fully automated since they required the manual location of tumor regions.

III. MRI Dataset

The dataset utilized to evaluate the proposed model's performance and accuracy is the Brain Tumor Classification (MRI) dataset from the Kaggle licenced CCO: Public Domain. A total of 3262 MRIs are included. The dataset is divided into two sections: the training dataset and the testing dataset [5].

The brain MRIs in the training dataset are divided into four classes, representing, respectively, glioma, meningioma, no tumor, and pituitary tumors, with 826, 822, 394, and 826 brain MRIs in each class.

Similar to the test dataset, the brain MRIs of glioma, meningioma, no tumor, and pituitary tumor are represented by 100, 115, 105, and 74 scans, respectively [8]. Figure 1 displays a sample of the datasets, and Figures 2 and Figures 3 display the distribution of the testing and training datasets and the four classes [9].

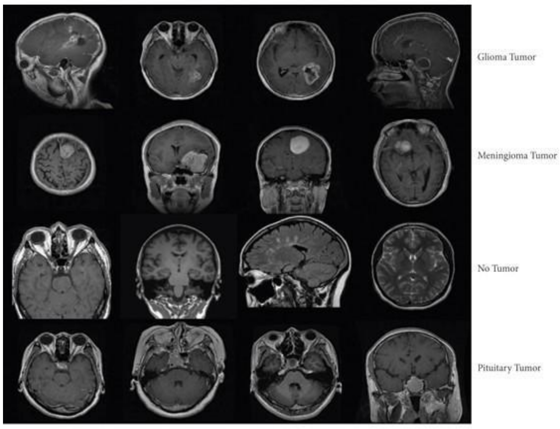


Fig. 1: Sample brain MRI for different classes

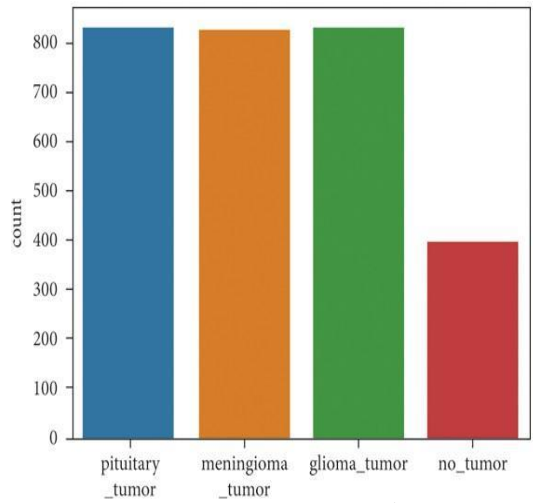


Fig. 2: Training dataset representation for 4 classes

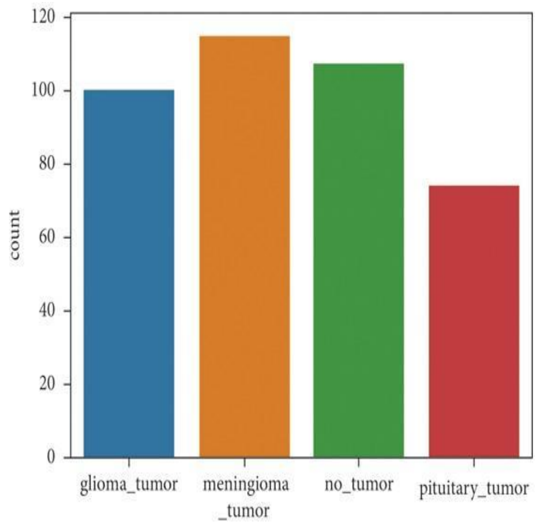


Fig. 3: Testing dataset representation for 4 classes

$$h\,(\text{out}) = (h - F + 2P)\,/S + 1,$$
$$w\,(\text{out}) = (w - F + 2P)\,/S + 1,$$
$$d\,(\text{out}) = n.$$

$$\mu = 1 \sum_{x \in \beta} x,$$
$$\sigma = 1 \sum_{x \in \beta} (x - \mu)^2 + C,$$

Methodology

Using a hierarchical architecture, deep learning (DL) models [18] learn high-level abstractions from input images. Due to the availability of large-scale annotated datasets and the fact that CNN has shown to be the most effective DL method for analysing medical pictures. The well-known CNN models [19] ImageNet, AlexNet, VGG16, GoogLeNet, and ResNet101 are just a few examples of models that have made significant strides in image recognition. However, the field of medical imaging lacks a comparable annotated dataset.

CNN is frequently used to classify medical images using one of two methods [30]. Learning from scratch is the first, and transfer learning is the second. The network layers that make up the CNN are as follows: convolutional layer, activation layer, batch normalisation layer, pooling layer, and classification layer [31, 32].

Convolution Layer

The convolutional layer is the top layer [33]. The features of an input word are extracted by this layer. This stage, which consists solely of convoluting the input neuron with a filter based on the input and need [34], produces the feature map. It makes use of a neural activation function to add nonlinearity. The visual brain of animals served as an inspiration for CNN computing. It analyses visual data and responds to minute input subregions [35]. The receptive field, stride, dilation, and padding are the main elements of the convolutional layer [36]. The output size of the image will depend on the number of filters n, the spatial size of the filter F, the padding P, and the stride S in the input image is as follows:

Batch Normalization Layer

Each layer of the architectural model can learn more independently thanks to the Batch Normalisation (BN) layer [37]. The output of the previous layer is being normalised as its main task in this layer. It can be used to avoid the issue of overfitting, which helps with regularisation [38, 39]. The layer's task is to harmonise the sequential model's input and output. After building the sequential model, in between layers, or after the convolution and pooling layer, this layer can be added to the model at several points. When applying BN to a single layer, its mathematical operation can be explained by first removing the mean from the inputs (of batch

normalisation) in each training iteration and dividing by the standard deviation, both of which are calculated using the statistics of the current mini batch. The scale offset and scale coefficient are then applied. Therefore, the following can be used to express an input for batch normalisation:
where μ and σ can be estimated following the equation outlined above, and α and σ are the learning parameters. Here, constant C (> 0) is summed with the variance (σ) in order to avoid division by zero computation complexity.

Activation Layer

The activation function is one of the most crucial components of the CNN model. Any kind of continuous and intricate network parameter relationship can be learned and estimated using them. It specifies which model information ought to be transmitted across the network and which ought not to. Two popular activation functions that are frequently used in deep learning models are the sigmoid function rectified linear unit (ReLu) and Softmax.

Pooling Layer

It parallels the layer of convolution. The function of this layer is to shrink the size of the feature map, which results in fewer mathematical operations and parameters required to operate the network. As a result, it may be stated that this layer's output is a summation of the features. There are numerous ways to pool data; max pooling was used in this case. Max pooling is used to select the most prominent characteristics from the feature map that have been addressed by the filters.

Classification Layer

In a CNN architecture, the classification layer is the bottom layer. It is a feedforward network that is fully connected and widely used as a classifier. Each neuron in the completely linked layers is coupled to every other neuron in the layer beneath it. This layer acknowledges the input image and generates class predictions by integrating the characteristics of the layers preceding it. The total number of output classes is determined by the total number of classes in the destination dataset. The classification layer in this work divides the produced characteristics of the input picture received from the previous layer into various categories based on the training data using the "SoftMax" activation function.

IV. Proposed CNN Model

Six weighted layers, four convolution layers, one fully connected layer, and one output layer or classification layer make up the CNN model. In addition to this, it has six BN, activation (ReLU), three dropouts, one flatten, and one max-pooling layer. Using a series of hidden layers, the model develops the ability to automatically obtain hierarchical characteristics. The suggested model has an output layer that creates a four-dimensional vector representing four different classes of brain tumors. The outputs of this layer are then subjected to a softmax function to determine the final class label.The main objective of building such a customised network is to decrease learning speed and parameters while maintaining detection accuracy as compared to current pretrained networks.

$$(x) = \alpha \odot x - \mu + y,$$

A 224 224 64 volume is produced by the first convolution layer, which accepts an image of size 224 224 3, convolves it with 64 kernels of size 3 3, and uses the same padding as the other convolution layers. The output of the first convolutional layer is sequentially subjected to batch normalisation and ReLU activation. The output of the layer before is sent into the following convolutional layer, which convolves it with 64 kernels of size 3 3 and applies the same activation. This produces a volume of 222 222 64, maximum pooling, the same BN layer, and a dropout layer of 0.35.A max-pooling layer with a filter size of 2 2 and a stride of 2 results in a lower-dimensional output volume of size 111 111 64. Following the second step, the first is performed one more, resulting in an output volume of 54 54 64.

The preceding layer is then flattened, producing an output shape of 186624, followed by a dropout of 0.3, a fully connected layer with an output shape of 512 units, and finally activation and batch normalisation. Finally, the likelihood score for the final class label serves as the determining element, with the output of the last layer being completely coupled to four neurons. The descriptions of each layer of the proposed model and their trainable parameters are compiled in Table 1. Training accuracy and loss graphs are displayed in Figures 4 and 5.

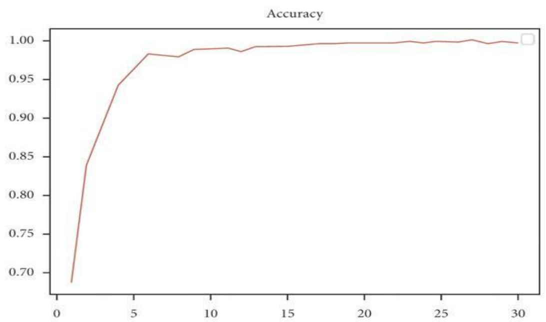


Fig. 4: Training accuracy curve

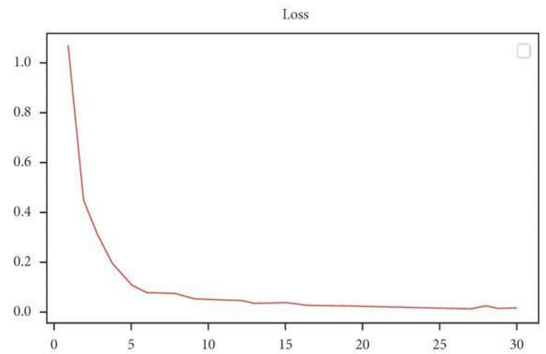


Fig. 5: Training loss curve

With a batch size of 32, 30 epochs, categorical cross-entropy as the parameter for the losses, and accuracy as the metric, the suggested framework is trained using the Adam optimizer. The calculated loss was 0.0504 while the training accuracy was 0.99.

TABLE 1: THE SUMMARY OF MODEL DESCRIPTION

Layer type	Filter	Kernel size	Output shape	Param#
Input layer	—	—	224 × 224 × 3	0-
Convolution	64	3 × 3	224 × 224 × 64	1792
Activation	—	—	224 × 224 × 64	0
BN	—	—	224 × 224 × 64	256
Convolution	64	3 × 3	222 × 222 × 64	36928
Activation	—	—	222 × 222 × 64	0
Max pooling	1	2 × 2	111 × 111 × 64	0
BN	—	—	111 × 111 × 64	256
Dropout	—	—	111 × 111 × 64	0
Convolution	64	3 × 3	109 × 109 × 64	36928
Activation	—	—	109 × 109 × 64	0
Max pooling	1	2 × 2	54 × 54 × 64	0
BN	—	—	54 × 54 × 64	256
Dropout	—	—	54 × 54 × 64	0
Convolution	64	3 × 3	54 × 54 × 64	36928
Activation	—	—	54 × 54 × 64	0
BN	—	—	54 × 54 × 64	256
Flatten	—	—	186624	0
Dropout	—	—	186624	0
FC	—	—	512	95552000
Activation	—	—	512	0
BN	—	—	512	2048
Output layer	—	—	4	2052
Total params: 95,706,884				
Trainable params: 95,705,220				
Nontrainable params: 1,664				

V. RESULTS

The suggested model effectively categorizes and forecasts the medical image. The result for the suggested model's transparency displays the real image class and the expected name for the image. Figure 6 below displays the results of the suggested model. The proposed model is contrasted with the current model in Table 2. The proposed model's confusion matrix is shown in Figure 7 and the classification report is shown in Figure 8.

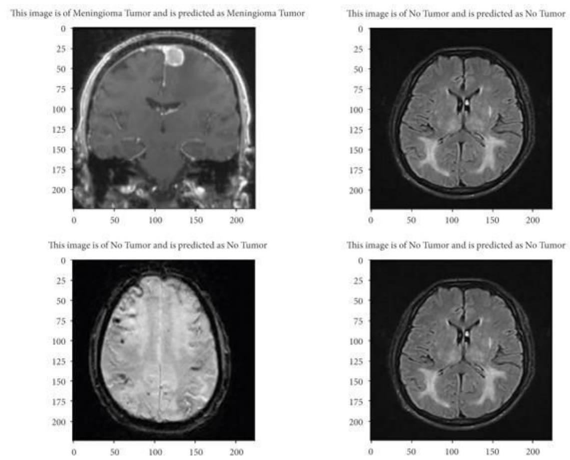


Fig. 6: Images for Prediction result

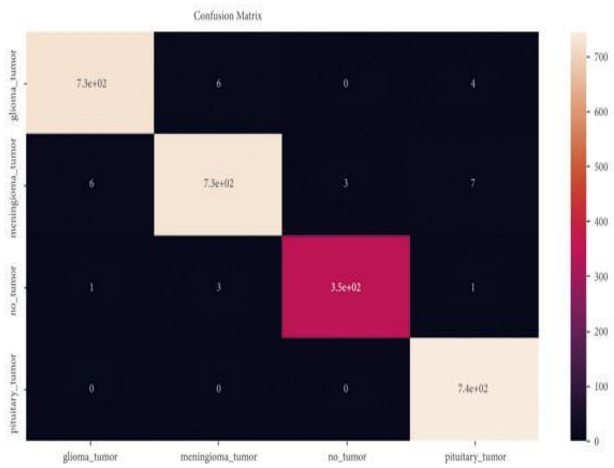


Fig. 7: Confusion matrix

Authors	Classes	Method	Accuracy (%)
[1]	3	CNN, data augmentation	C1-95.23 C2-95.43 C3-98.43
[21]	4	Preprocessing-normalization, feature acquired-GLCM	98
[23]	6	GA-SVM, GA-ANN	GA-SVM:89 GA-ANN:94.1
[26]	3	CNN, CapsNet	86.56
[27]	3	No data augmentation, CNN	98.51
Proposed model	4	CNN	99

Fig. 8: Classification report

TABLE 2: MODEL COMPARISON

	precision	recall	f1-score	support
0	0.99	0.99	0.99	739
1	0.99	0.98	0.98	747
2	0.99	0.99	0.99	353
3	0.98	1.00	0.99	744
accuracy			0.99	2583
macro avg	0.99	0.99	0.99	2583
weighted avg	0.99	0.99	0.99	2583

VI. CONCLUSION

In this study, we demonstrated an automated approach for multiple- class brain tumor detection using MRI. The proposed deep CNN model, which enables automated feature learning from brain MRIs, is composed of six learnable layers. The primary objective for establishing such a network was to achieve a superior classification result while training faster than regular DL models. Despite the small amount of training data, the experiment's findings illustrate the model's reliability. Due to its modest preprocessing constraints and lack of special features, the proposed technique can be deployed to various MRI classifications. We can more explicitly categorize the data into several class labels for future work.

REFERENCES

[1] F. J. Díaz-Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, and D. González-Ortega, "A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network," *Health Care*, vol. 9, no. 2, p. 153, 2021.

[2] Cancer-types, "Brain tumor: statistics," 2022, https://www.cancer.net/cancer-types/brain-tu mor/statistics.

[3] Y. Zhang, A. Li, C. Peng, and M. Wang, "Improve glioblastoma multiforme prognosis pre- diction by using feature selection and multiple kernel learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 5, pp. 825–835, 2016.

[4] P. K. Vaishnav, S. Sharma, and P. Sharma, "Analytical review analysis for screening COVID-19 disease," *International Journal of Modern Research*, vol. 1, no. 1, pp. 22–29, 2021

[5] I. Chatterjee, "Artificial intelligence and patentability: review and discussions," *International Journal of Modern Research*, vol. 1, no. 1, pp. 15–21, 2021

[6] Y. Yang, L. F. Yan, X. Zhang et al., "Glioma radingon conventional MR images: a deep learning study with transfer learning," *Frontiers in Neuroscience*, vol. 12, p. 804, 2018.

[7] M. Talo, U. B. Baloglu, Ö. Yıldırım, U. R. Acharya, and U. Rajendra Acharya, "Application of deep transfer learning for automated brain abnormality classification using MR images," *Cognitive Systems Research*, vol. 54, pp. 176–188, 2019.

[8] T. Sharma, R. Nair, and S. Gomathi, "Breast cancer image classification using transfer learning and convolutional neural network," *International Journal of Modern Research*, vol. 2, no. 1, pp. 8–16, 2022.

[9] S. K. Shukla, V. K. Gupta, K. Joshi, A. Gupta, and M. K. Singh, "Self-aware execution environment model (SAE2) for the performance improvement of multicore systems," *International Journal of Modern Research*, vol. 2, no. 1, pp. 17–27, 2022.

[10] T. Kaur, B. S. Saini, and S. Gupta, "Quantitative metric for MR brain tumor grade classifi- cation using sample space density measure of analytic intrinsic mode function representa- tion," *IET Image Processing*, vol. 11, no. 8, pp. 620–632, 2017.

[11] K. Usman and K. Rajpoot, "Brain tumor classification from multi-modality MRI using wavelets and machine learning," *Pattern Analysis & Applications*, vol. 20, no. 3, pp. 871–881, 2017.

[12] V. Anitha and S. J. I. C. V. Murugavalli, "Brain tumor classification using two-tier classifier with adaptive segmentation technique," *IET Computer Vision*, vol. 10, no. 1, pp. 9–17, 2016.

[13] E. S. A. El-Dahshan, H. M. Mohsen, K. Revett, and A. B. M. Salem, "Computer-aided diagnosis of human brain tumor through MRI: a survey and a new algorithm," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5526–5545, 2014.

[14] J. Hemanth, J. Anitha, A. Naaji, O. Geman, D. E. Popescu, and L. Hoang Son, "A modified deep convolu- tional neural network for abnormal brain image classification," *IEEE Access*, vol. 7, pp. 4275–4283, 2019.

[15] M. Yousefi, A. Krzyżak, and C. Y. Suen, "Mass detection in digital breast tomosynthesis da- ta using convolutional neural networks and multiple instance learning," *Computers in Biology and Medicine*, vol. 96, pp. 283–293, 2018.

[16] Y. Gu, X. Lu, L. Yang et al., "Automatic lung nodule de- tection using a 3D deep convolutional neural network combined with a multi-scale predic- tion strategy in chest CTs," *Computers in Biology and Medicine*, vol. 103, pp. 220–231, 2018.

[17] O. Charron, A. Lallement, D. Jarnet, V. Noblet, J. B. Clavier, and P. Meyer, "Automatic de- tection and segmentation of brain metastases on multimodal MR images with a deep con- volutional neural network," *Computers in Biology and Medicine*, vol. 95, pp. 43–54, 2018.

[18] W. Ayadi, W. Elhamzi, I. Charfi, and M. Atri, "Deep CNN for brain tumor classifi- cation," *Neural Processing Letters*, vol. 53, no. 1, pp. 671–700, 2021.

[19] D. Jude Hemanth, C. S. Vijila, A. I. Selvakumar, and J. Anitha, "Performance improved itera- tion-free artificial neural networks for abnormal magnetic resonance brain image classifi- cation," *Neurocomputing*, vol. 130, pp. 98–107, 2014.

[20] D. R. Nayak, R. Dash, and B. Majhi, "Automated diagnosis of multi-class brain ab- normalities using MRI images: a deep convolutional neural network-based method," *Pattern Recognition Letters*, vol. 138, pp. 385–391, 2020.

[21] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja, "A package-SFERCB- "Segmentation, feature extraction, reduction and classification analysis by both SVM and ANN for brain tumors," *Applied Soft Computing*, vol. 47, pp. 151–167, 2016.

[22] B. Tahir, S. Iqbal, M. Usman Ghani Khan et al., "Feature enhancement framework for brain tumor segmentation and classification," *Microscopy Research and Technique*, vol. 82, no. 6, pp. 803–811, 2019.

[23] J. S. Paul, A. J. Plassard, B. A. Landman, and D. Fabbri, "Deep learning for brain tumor classi- fication," *Proceedings of SPIE*, vol. 10137, pp. 1–16, 2017.

[24] P. Afshar, A. Mohammadi, and K. N. Plataniotis, "Brain tumor type classification via cap- sule networks," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3129–3133, IEEE, Athens, Greece, 7 October 2018.

[25] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko, "Brain tumor classification using convolutional neural network," *World congress on Medical Physics and Biomedical Engineering 2018*, Springer, Singapore, pp. 183–189, 2019.

[26] N. Ghassemi, A. Shoeibi, and M. Rouhani, "Deep neural network with generative adversar- ial networks pre-training for brain tumor classification based on MR images," *Biomedical Signal Processing and Control*, vol. 57, Article ID 101678, 2020.

[27] H. Padole, S. D. Joshi, and T. K. Gandhi, "Graph wavelet-based multilevel graph coarsening and its application in graph-CNN for alzheimers disease detection," *IEEE Access*, vol. 8, pp. 60906–60917, 2020.

[28] J. Rashid, S. Batool, J. Kim et al., "An augmented artificial intelligence approach for chronic diseases prediction," *Frontiers in Public Health*, vol. 10, Article ID 860396, 2022.

[29] C. Monga, D. Gupta, D. Prasad, S. Juneja, G. Muhammad, and Z. Ali, "Sustainable network by enhancing attribute-based selection mechanism using Lagrange interpolation," *Sustainability*, vol. 14, no. 10, p. 6082, 2022.

[30] S. Juneja, A. Juneja, G. Dhiman, S. Jain, A. Dhankhar, and S. Kautish, "Computer vision-enabled character recognition of hand gestures for patients with hearing and speaking disability," *Mobile Information Systems*, vol. 2021, Article ID 4912486, pp. 1–10, 2021.

[31] S. Juneja, S. Jain, A. Suneja et al., "gender and age classification enabled blockschain security mechanism for assisting mobile application," *IETE Journal of Research*, pp. 1–13, 2021.

[32] S. Kanwal, J. Rashid, J. Kim, S. Juneja, G. Dhiman, and A. Hussain, "Mitigating the coexistence technique in wireless body area networks by using superframe interleaving," *IETE Journal of Research*, pp. 1–15, 2022.

[33] S. Juneja, A. Juneja, and R. Anand, "Reliability modeling for embedded system environment compared to available software reliability growth models," in *Proceedings of the 2019 International Conference on Automation, Computational and Technology Management, ICACTM 2019*, pp. 379–382, Institute of Electrical and Electronics Engineers Inc, London, UK, April 2019.

[34] R. Qian, S. Sengan, and S. Juneja, "English language teaching based on big data analytics in augmentative and alternative communication system," *International Journal of Speech Technology*, 2022.

[35] K. Ramana, M. R. Kumar, K. Sreenivasulu et al., "Early prediction of lung cancers using deep saliency capsule and pre-trained deep learning frameworks," *Frontiers in Oncology*, vol. 6, p. 2641.

[36] S. Safdar, M. Rizwan, T. R. Gadekallu et al., "Bio-imaging-based machine learning algorithm for reast cancer detection," *Diagnostics*, vol. 12, no. 5, p. 1134, 2022.

[37] A. Sharma, R. S. Tanwar, Y. Singh et al., "Heart rate and blood pressure measurement based on photoplethysmogram signal using fast Fourier transform," Computers & Electrical Engineering, vol. 101, Article ID 108057, 2022

List of Reviewers

Prof. Vijay Gupta, Professor, School of Applied Science, NSUT, Delhi,vijaygupta2001@hotmail.com	Dr. Deeba Naqvi, Assistant Professor, MSIT deeba.naqvi@msit.in
Dr. Shikha Saxena, Professor, Applied Sciences, MAIT drshikhasaxenamait@gmail.com	Dr. Jindagi Kumari, Asst.Prof., Applied Sciences, MSIT jindagi.kumari@msit.in
Dr. Savita Ahlawat, Associate Professor CSE,MSIT savita.ahlawat@msit.in	Dr. Archana Sandhu, Associate Professor, CSE,MMU archana.sandhu43@gmail.com
Dr. Bhagwant Bishnoi, Assistant Professor, Department of Physics, KIET Ghaziabad UP,bishnoi29chaem@gmail.com	Dr. Ritu Saharan, Assistant professor, Department of Physics,BNC college, DU,rituphydu@gmail.com
Dr. Sandeep Singh, Assistant Professor,MSIT sandeep@msit.in	Dr Adil Hashmi, Assistant Professor, VIPS,adeelhashmi@vips.edu
Dr. Bharti Sharma, Associate Professor MSIT, bhartisharma@msit.in	Dr. Babita Tiwari, Assistant Professor, Department of CSE, Manipal University Jaipur,Rajasthan, tiwari.babita@gmail.com
Dr. Abhishek Gandhar, Professor, BVCOE, Paschim Vihar, New Delhi,abhishek.gandhar@gmail.com	Dr. Shashi Gandhar, Associate Professor, BVCOE, Paschim Vihar, New Delhi
Dr. Anita, Assistant Prof., Galgotia Institute of Technology anita.mudgal@jecrcu.edu.in	Dr. Shaifali Madan Arora, Associate Professor, MSIT, New Delhi ,shefali04@msit.in
Dr. Amit Choudhary , Assistant professor, USAR, GGSIPU, amit.usar@ipu.ac.in	Dr. Sangeeta ,Associate professor, MSIT, sangeeta@msit.in
Dr. Sunil Gupta, Associate Professor EEE, MSIT sunil.gupta_eee@msit.in	Dr. Tajinder Singh Arora, Associate Professor, AMU tajarora@gmail.com
Dr. Tripti Sharma, Professor, MSIT, tripti_sharma@msit.in	Dr. Seema Gupta, Professor, IIMT,Seemagupta.iimt@gmail.com
Prinkle Talan, Assistant Professor. ECE MSIT prinkletalan@msit.in	Megha Agarwal, Assistant Professor, ECE BPIT meghaagarwal@bpitindia.com
Dr. Nishtha Jatana, Associate Professor, MSIT nishtha.jatana@gmail.com	Dr. Aakanshi Gupta, Assistant Professor, Amity University , Noida,aakanshi@gmail.com
Dr. Ravinder, Assistant Professor, KR Manglam University, ravinder.beniwal@krmangalam.edu.in	Dr. Preeti Rathee, Assistant Professor, MSIT Delhi preetirathee@msit.in
Dr. Anupama Kaushik, Associate Professor, MSIT Delhi, anupama@msit.in	Dr. Ankita Sharma, Assistant Professor, USIC&T, GGSIPU Delhi, ankitasharma@ipu.ac.in

Maharaja Surajmal Institute of Technology

(ISO 9001:2015 Certified, Approved by AICTE)

Affiliated to GGSIP University, Delhi C-4, Janakpuri, New-Delhi-110058

Phone: 011-65215941 E-mail: director@msit.in

Website: www.msit.in