Automated ECG Heartbeat Anomaly Detection Using Deep Learning

Ahsan Rizvi
Department of Electrical and Computer Engineering (ECE)
North South University
Dhaka, Bangladesh
ahsan.rizvi@northsouth.edu

Umme Hani Roshni
Department of Electrical and Computer Engineering (ECE)
North South University
Dhaka, Bangladesh
umme.roshni@northsouth.edu

Md. Irtiza Hossain Mahmud
Department of Electrical and Computer Engineering (ECE)
North South University
Dhaka, Bangladesh
mohammad.mahmud@northsouth.edu

Abstract — **Cardiovascular diseases (CVDs) are the leading global cause of death, underscoring the need for fast and accurate diagnostics. This study presents an automated ECG image classification system using deep learning to categorize heartbeats into four classes: Normal, Myocardial Infarction, Abnormal Heartbeat, and History of MI. Four models were developed and evaluated: a baseline Custom CNN, an Optimized CNN, and pretrained MobileNetV2 and ResNet50 models using transfer learning. Trained on a balanced, augmented dataset, the Custom CNN achieved the highest precision and F1 score, while the Optimized CNN demonstrated strong generalization. MobileNetV2 offered stable, lightweight performance, and ResNet50 showed signs of overfitting. A web interface was also developed for real-time ECG prediction. Results highlight the potential of deep learning to improve the speed and accessibility of cardiac diagnostics.**

Keywords— **ECG Classification, Deep Learning, Custom CNN, Optimized CNN, MobileNetV2, ResNet50, Transfer Learning, Myocardial Infarction, Real-time Diagnosis, Cardiovascular Diseases, Healthcare AI, Automated ECG Analysis**

I. INTRODUCTION

Context and Background

Cardiovascular diseases (CVDs) are the leading cause of death worldwide, contributing to approximately 17.9 million deaths annually, which accounts for 31% of all global deaths [1]. This highlights the critical need for early detection and accurate diagnosis of heart-related conditions. Electrocardiogram (ECG) analysis is a crucial diagnostic tool in cardiology, offering insights into the electrical activity of the heart and enabling the detection of potential abnormalities. The analysis involves recording the electrical signals of the heart over time, allowing healthcare professionals to assess heart function and identify irregularities. While manual interpretation of ECG signals has been the standard practice for decades, it presents several challenges. The process is time-consuming, requiring specialists to visually examine complex waveform patterns. Additionally, the accuracy of ECG interpretation is heavily reliant on the expertise and experience of the interpreting healthcare provider, which introduces variability in diagnoses. This issue is further exacerbated by the shortage of trained cardiologists, especially in resource-limited settings, where access to specialized expertise is often restricted. These challenges underscore the need for automated ECG analysis systems that can enhance diagnostic efficiency and accuracy, while also broadening access to cardiac care [1].

Problem Definition

Early detection of heart conditions such as myocardial infarction (MI) and abnormal heartbeats is critical for preventing severe health complications. However, traditional manual ECG analysis is not only time-consuming but also requires expert knowledge that may not be readily accessible in many healthcare environments. This limitation can delay diagnosis and, in some cases, lead to misinterpretations that compromise patient care. Given the increasing global burden of cardiovascular diseases, there is a pressing need to automate ECG analysis to improve diagnostic speed and accuracy. An automated system could alleviate the workload of healthcare professionals, ensuring faster and more reliable detection of cardiac anomalies.

The goal of this research is to develop an Automated ECG Heartbeat Anomaly Detection System using deep learning techniques. This system will classify ECG images into four clinically significant categories: Normal, Myocardial Infarction (MI), Abnormal Heartbeat, and History of Myocardial Infarction. The proposed solution aims to provide an accurate and efficient tool for real-time ECG analysis.

Contributions

This research proposes an Automated ECG Heartbeat Anomaly Detection System using deep learning, particularly Convolutional Neural Networks (CNNs), to classify ECG images into categories like Normal, MI, Abnormal Heartbeat, and History of MI. By building a custom CNN architecture trained from scratch, the system achieves high classification accuracy on preprocessed and augmented ECG images. Additionally, a web-based interface for real-time ECG analysis will be developed, making the system accessible remotely for healthcare professionals. This paper reviews current automated ECG detection methods, discusses challenges, and outlines the potential of deep learning to improve cardiac diagnostics

II. LITERATURE REVIEW

A. Historical Context of ECG Analysis

Evolution from Manual to Automated Interpretation

The field of ECG analysis began with the invention of the electrocardiograph by Willem Einthoven in the early 20th century. For decades, ECG interpretation remained entirely manual, with physicians using standardized criteria to assess various cardiac conditions. Early attempts at computerized ECG analysis in the 1960s and 1970s utilized rule-based expert systems, encoding the knowledge of cardiologists into algorithmic decision trees. Despite their groundbreaking nature, these early systems struggled with the inherent variability of ECG signals across patients and recording conditions, which hindered their clinical effectiveness. In the 1980s and 1990s, digital signal processing techniques allowed improvements in feature extraction and classification; however, accuracy remained insufficient for widespread clinical adoption without expert verification [2].

Transition to Machine Learning Approaches

The early 2000s marked a shift toward machine learning techniques for ECG analysis, as methods such as support vector machines, random forests, and artificial neural networks (ANNs) demonstrated enhanced accuracy by learning directly from data, rather than relying on predefined rules. These methods still required manual feature extraction, where domain experts identified ECG characteristics such as QRS complex duration or ST-segment elevation to be used in classification algorithms. While machine learning represented an advancement over previous systems, these techniques struggled with the subtle and complex patterns associated with certain cardiac conditions. The 2010s saw the emergence of deep learning, which enabled end-to-end learning from raw or minimally processed ECG signals, revolutionizing automated ECG analysis by improving both accuracy and efficiency [3].

B. Current Research Landscape

1. Deep Learning Approaches for ECG Analysis:

a) ***Recurrent Neural Networks (RNNs) and Time-Series Analysis***

The application of deep learning to ECG analysis has gained significant momentum in recent years, with approaches such as Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, proving to be highly effective for analyzing time-series ECG data. LSTM networks are adept at capturing temporal dependencies in sequential data, making them ideal for modeling normal ECG patterns and identifying anomalies. These networks have shown exceptional performance, with reported sensitivity and specificity rates exceeding 95% for detecting major arrhythmias, significantly outperforming traditional time-series analysis methods. Recent advancements in RNN-based approaches, including bidirectional LSTMs and attention mechanisms, have further enhanced the ability to capture both forward and backward dependencies in ECG signals, improving the detection of intermittent arrhythmias, which may be missed in shorter recording segments [4].

b) ***Convolutional Neural Networks (CNNs) for ECG Classification****:*

While RNNs are effective for temporal analysis, Convolutional Neural Networks (CNNs) have emerged as powerful tools for both image-based and direct signal processing of ECG data. When working with digitized ECG strips, traditional 2D CNNs leverage their feature extraction capabilities to identify diagnostic patterns effectively. Recent studies have refined CNN approaches for direct ECG signal classification using 1D CNNs, which segment the ECG signal based on R-peak detection—a critical component of the QRS complex. These models have achieved high accuracy rates exceeding 98% when evaluated on standard benchmark datasets, outperforming both traditional machine learning methods and conventional CNN implementations. 1D CNNs, designed to operate directly on raw time-series ECG data, capture key features across different temporal scales, from rapid QRS complex deflections to slower waveforms like the P and T waves [5].

c) ***Hybrid and Advanced Architectures***

Recent advancements in ECG analysis have focused on hybrid architectures that combine the strengths of multiple neural network types. One notable development involves a dual-stream architecture that processes both time-series and frequency-domain (spectrogram) representations of ECG data. By simultaneously analyzing ECG signals in both domains, this approach captures complementary features that may not be evident in either domain alone. The dual-stream model has shown state-of-the-art performance in identifying cardiac abnormalities, surpassing single-domain methods. Other innovative approaches, such as capsule networks and graph neural networks, have shown promise for multi-lead ECG analysis, where the relationships between signals from different ECG leads offer critical diagnostic information [6], [7].

2. Transfer Learning and Pre-trained Models

Transfer learning has become a crucial strategy in addressing the challenges posed by small ECG datasets, which are often limited due to privacy constraints and difficulties in data collection. By using pre-trained models such as ResNet and VGG16—originally trained on large-scale image datasets like ImageNet—researchers have demonstrated the feasibility of fine-tuning these models for ECG classification tasks. Fine-tuning allows these models to adapt their feature extraction capabilities to ECG-specific data, improving classification performance even with smaller, domain-specific datasets. A study published in the *Journal of Biomedical Informatics* (2021) showed that ResNet-based architectures achieved accuracy improvements of 5-8% compared to models trained from scratch, especially when working with datasets of fewer than 5,000 images [8].

a) ***Implementation Strategies for ECG Analysis***

Several implementation strategies for transfer learning in ECG analysis have been explored. One common approach is using pre-trained networks, such as ResNet-50, as fixed feature extractors, where the convolutional layers are frozen, and only the classification layers are retrained on ECG data. This method has been shown to be effective in medical image classification, including ECG images. In contrast, fine-tuning the entire model or specific layers has been more widely used, allowing the model to adjust to the specific characteristics of

ECG signals. Additionally, hybrid transfer learning techniques, which combine transfer learning with domain adaptation, aim to bridge the gap between natural images and medical data, improving the model's ability to adapt to ECG-specific tasks [9].

*b) **Domain-Specific Pre-training***

An emerging trend in ECG analysis is domain-specific pre-training, where models are pre-trained on large ECG datasets before fine-tuning for specific diagnostic tasks. This approach offers significant advantages over transfer learning from natural image domains, as pre-training on ECG data allows the model to learn ECG-specific features. Self-supervised learning techniques, such as reconstructing corrupted signals or predicting missing segments, have been employed to create pre-trained models with superior performance in diagnosing subtle cardiac conditions [10].

Although transfer learning with pre-trained models such as ResNet and VGG16 has demonstrated strong results in prior studies, this work takes a different approach by developing a lightweight, custom CNN from scratch. This strategy was chosen to better suit the dataset characteristics and ensure full control over architectural tuning.

Data Augmentation Techniques

To address the limitations of small ECG datasets, data augmentation techniques have become a standard practice. Traditional image-based augmentation methods such as rotation, flipping, zooming, and translation have been widely used. Signal-specific augmentations, such as temporal warping, amplitude scaling, and synthetic noise addition, have been developed to improve model robustness by simulating natural variations in heart rate, electrode placement, and recording conditions. Generative models like Generative Adversarial Networks (GANs) have also been used to generate realistic ECG signals, particularly improving classification accuracy for rare arrhythmias [11], [12].

Ensemble Methods**:** Ensemble methods, combining multiple models or architectures, have shown success in improving ECG classification performance. By leveraging the diversity of models, ensemble techniques such as bagging, boosting, and stacking enhance diagnostic accuracy, particularly for difficult-to-classify ECG patterns.

Multi-Modal Approaches: Multi-modal approaches, integrating ECG data with patient demographics, medical history, and other clinical data, are also gaining attention. These approaches offer a more holistic diagnostic framework, improving both the accuracy and clinical relevance of ECG-based predictions [13], [14].

C. Challenges and Research Gaps

1. **Dataset Limitations**

   The limited availability of large, diverse, and well-annotated ECG datasets remains a significant challenge. Privacy regulations, annotation quality, and demographic biases contribute to this issue, making it difficult to develop generalized models. International collaborations and federated learning approaches offer potential solutions to address these limitations [15], [16].

2. **Class Imbalance**

   Class imbalance is another challenge, where normal ECG signals often outnumber abnormal ones, making it difficult for models to detect rare conditions. Techniques like resampling, cost-sensitive learning, and anomaly detection have been proposed to address this issue [17].

3. ***Feature Extraction Challenges***

   Extracting clinically relevant features from ECG images is complicated by issues such as signal quality, variability in recording formats, and the need to capture subtle diagnostic features. Attention mechanisms and explainable AI techniques are being explored to ensure that models focus on clinically significant ECG components [18].

4. ***Integration with Clinical Workflows***

   Integrating automated ECG analysis systems into clinical workflows presents challenges related to regulatory approval, clinical validation, clinician acceptance, and technical integration. Interdisciplinary collaboration is crucial to address these challenges and ensure successful implementation in healthcare settings [19].

D. Future Research Directions

1. **Multimodal Learning**

   Integrating ECG data with other clinical information can improve diagnostic accuracy and clinical relevance. Multi-modal deep learning approaches offer promising avenues for creating more comprehensive diagnostic tools [20] [21] [22].

2. **Explainable AI**

   As deep learning models become more complex, ensuring their interpretability is critical for clinical adoption. Research into explainable AI techniques, such as Grad-CAM and prototype-based learning, can help clinicians understand and trust model predictions [23] [24].

3. **Real-time Analysis and Edge Computing**

   Real-time ECG monitoring on edge devices is becoming increasingly feasible, enabling continuous monitoring in remote and resource-limited environments. Techniques such as model compression and hardware optimization are crucial for deploying lightweight models on wearable devices [25].

4. **Federated Learning**

   Federated learning allows for training models across multiple institutions while preserving data privacy. This approach offers a promising direction for addressing privacy concerns while improving ECG analysis system performance [26].

III. METHODOLOGIES

A. System Design

The proposed system is a complete deep learning-based ECG classification pipeline that takes raw ECG images as input and outputs a predicted diagnosis class. The pipeline encompasses several major components: data preprocessing, offline augmentation, model training with a custom CNN, and a Django web-based deployment interface.

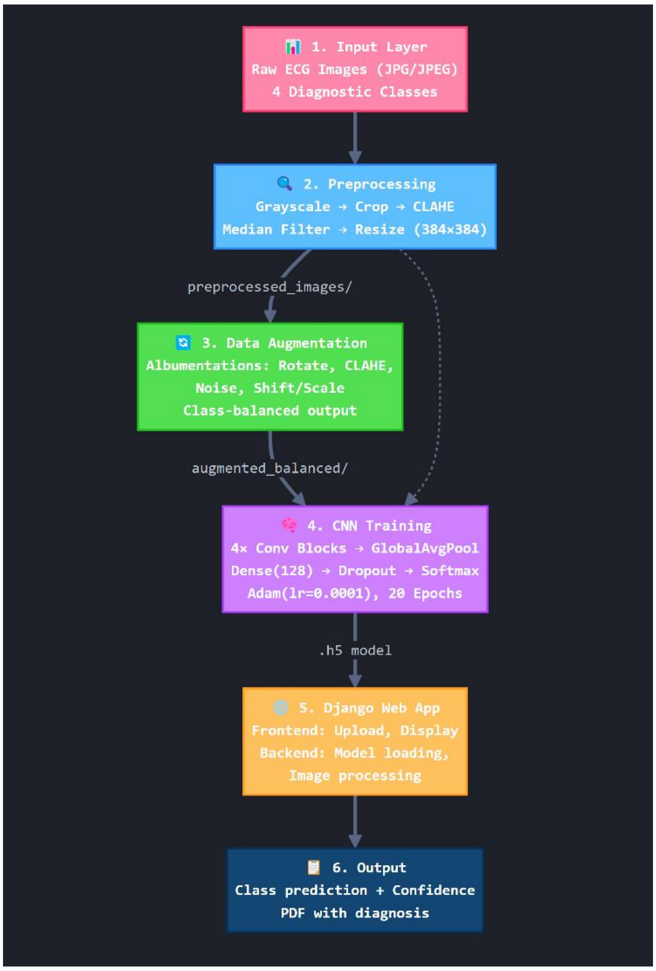

Fig. 1. *System Design diagram*

The architecture of the proposed ECG heartbeat anomaly detection system is illustrated in Figure 3.1. The system comprises six sequential stages: data input, preprocessing, data augmentation, model training, web deployment, and output generation. Each stage is designed to facilitate the smooth flow of data from raw ECG images to final classification and report generation, ensuring accuracy, robustness, and usability.

Stage 1: Input Layer

Raw ECG images are ingested as input. These images are sourced from the publicly available "ECG Images dataset of Cardiac Patients," hosted on Mendeley Data (DOI: 10.17632/gwbz3fsgp8.2). The dataset contains four diagnostic classes—Normal, Myocardial Infarction, Abnormal Heartbeat, and History of MI—each represented by grayscale JPG/JPEG images of varying quality and resolution.

Stage 2: Preprocessing

Following input acquisition, the system performs preprocessing to standardize and enhance the images. Each image is first converted to grayscale to reduce complexity. Then, custom cropping is applied to remove unnecessary headers and margins, followed by CLAHE (Contrast Limited Adaptive Histogram Equalization) to improve contrast. A median filter is optionally used to reduce visual noise. Finally, the images are resized to a fixed resolution of 384×384 pixels to match the input shape of the CNN model. The output of this stage is stored in a structured directory named preprocessed_images/.

Stage 3: Data Augmentation

The third stage involves offline data augmentation to address class imbalance and enrich the training dataset. Using the Albumentations library, each class undergoes multiple transformations, such as random rotation, CLAHE, brightness and contrast adjustment, Gaussian noise addition, and shift-scale-rotate operations. The number of augmentations per image is adjusted based on the size of each class to achieve a balanced distribution across all four categories. The augmented and balanced dataset is saved in the augmented_balanced/ directory.

Stage 4: CNN Training

In the fourth stage, a custom Convolutional Neural Network (CNN) is trained from scratch using the preprocessed and augmented dataset. The model architecture consists of four convolutional blocks, each followed by batch normalization and max pooling, culminating in a Global Average Pooling layer. This is followed by a dense layer with 128 neurons and ReLU activation, a dropout layer with a 0.5 dropout rate to prevent overfitting, and a final dense layer with softmax activation to classify inputs into one of the four diagnostic categories. The model is compiled using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy loss. It is trained for 20 epochs on a training set of 6564 images, with 1640 images used for validation. Once training is complete, the model is saved in HDF5 format (.h5) for deployment.

Stage 5: Django Web Application

In the fifth stage, the trained model is integrated into a Django web application. This application includes both frontend and backend components. The frontend allows users to upload ECG images and view prediction results in a clean and responsive interface. The backend handles image preprocessing, model loading, and prediction logic. Additional features include a dynamic progress bar to display confidence levels for each class and a diagnostic message generated based on the predicted class.

Stage 6: Output Generation

Finally, the sixth stage produces the output. The system displays the predicted class along with its confidence score, a class-wise probability breakdown, and a contextual diagnostic message tailored to the result. Users are also provided with an option to export the results as a downloadable PDF, which includes the ECG image, classification output, confidence scores, and the diagnostic explanation. This feature enhances the usability of the system, especially in offline or clinical review scenarios.

B. Required Software and Tools

This project was implemented using a combination of cloud-based development platforms, local deployment environments, and a set of specialized Python libraries for image processing, deep learning, and web integration.

Development Environments

Google Colaboratory (Colab) was employed during the model training phase. Its free access to NVIDIA GPU resources significantly accelerated training and experimentation. [27]

Visual Studio Code was used for data preprocessing, offline augmentation, web deployment and testing. The Django web framework was configured and executed locally to simulate real-time predictions and user interactions.

Python Libraries and Tools

A variety of Python libraries were integrated throughout the pipeline, categorized by their specific use:

Category	Libraries / Tools
Image Processing	Albumentations [28], OpenCV[29]
Deep Learning	TensorFlow[30], Keras[31]
Data Handling	NumPy[36], OS[33], Shutil[32]
Web Framework	Django[34]
PDF Export	ReportLab[35]

Tab. 1. Python libraries and tools used

Model Architecture Used

A custom Convolutional Neural Network (CNN) was designed and trained from scratch to perform the multi-class classification task. Unlike conventional transfer learning techniques that rely on pre-trained models, the decision to build a model from the ground up enabled greater flexibility and interpretability of ECG-specific spatial patterns.

The architecture includes four convolutional blocks with increasing filter sizes—32, 64, 128, and 256—each followed by batch normalization and max pooling. This was followed by a global average pooling layer and a fully connected dense layer of 128 units with ReLU activation, succeeded by a dropout layer (rate = 0.5) for regularization, and a final softmax output layer with four neurons corresponding to the four diagnostic classes. [37]

Training Configuration

The model was compiled using the Adam optimizer with a learning rate of 0.0001, and categorical crossentropy was used as the loss function. The training was conducted for 20 epochs with a batch size of 32 and input images resized to 384×384×3. The dataset was split into 80% training and 20% validation sets using the ImageDataGenerator API. After applying class-specific offline augmentation, the final dataset comprised 6564 training images and 1640 validation images. [38] [39]

C. Dataset Description

Dataset Source

The dataset used in this study is titled *"ECG Images Dataset of Cardiac Patients"*, published on Mendeley Data under the DOI: 10.17632/gwbz3fsgp8.2. It was contributed by Ali Haider Khan and Muzammil Hussain in March 2021, and is distributed under the CC BY 4.0 open license. The dataset was developed at the Ch. Pervaiz Elahi Institute of Cardiology, Multan, Pakistan, to support research in cardiovascular diagnostics.[40]

Raw Dataset Composition

The original dataset consisted of 928 ECG scan images captured in clinical environments. Each image represented a full-size 12-lead ECG report and was categorized into one of four diagnostic classes:

- **Normal**: 284 images
- **Myocardial Infarction (MI)**: 239 images
- **Abnormal Heartbeat**: 233 images
- **History of MI**: 172 images

These raw images were stored in high-resolution JPEG format, with substantial intra-class variation and clinical noise typical of real-world ECG scans.

Preprocessing Pipeline

To standardize the input for deep learning and eliminate non-informative artifacts, all images underwent the following preprocessing steps:

1. **Grayscale conversion** to reduce dimensionality while preserving morphological patterns.
2. **Region cropping** to remove headers, footers, and side borders, ensuring the ECG waveform is centered.
3. **Contrast enhancement** using CLAHE (Contrast Limited Adaptive Histogram Equalization) to amplify fine-grained features.
4. **Noise suppression** through median filtering to remove isolated speckle noise.
5. **Resizing** to a uniform dimension of 384×384 pixels, suitable for convolutional model input.

This preprocessing phase yielded a set of consistent, clean ECG images stored in the preprocessed_images/ directory.[41]

Offline Augmentation Strategy

Given the notable class imbalance in the original dataset, a targeted offline image augmentation strategy was applied using the Albumentations library.[28] The goal was to synthetically expand underrepresented classes and equalize class distributions before model training. Augmentation techniques included:

- Geometric transformations: small rotations, shifts, and scaling
- Photometric adjustments: brightness/contrast variation
- Noise injection: Gaussian noise
- Additional contrast boosting using CLAHE

The augmentation count was determined on a per-class basis to ensure a balanced dataset. The following strategy was adopted:

Class	Augmentations per Image
Normal	×5
Myocardial Infarction	×10
Abnormal Heartbeat	×10
History of MI	×12

Tab. 2. Class-wise image augmentation strategy

This augmentation process produced a fully balanced dataset, comprising 6564 training images and 1640 validation images, ready for model development.

IV. RESULTS

A. Data Preprocessing

The original ECG image dataset comprised 928 raw images across four diagnostic classes: *Normal (284)*, *Myocardial Infarction (239)*, *Abnormal Heartbeat (233)*, and *History of MI (172)*. To ensure consistency, all images were preprocessed using OpenCV and custom Python scripts. Preprocessing involved grayscale conversion, cropping of headers and borders, enhancement via Contrast Limited Adaptive Histogram Equalization (CLAHE), noise reduction through median filtering, and resizing to 384x384 resolution.

To address class imbalance, offline augmentation was performed using Albumentations. Each class was augmented with different multipliers: Normal (×5), MI (×10), Abnormal Heartbeat (×10), and History of MI (×12), resulting in 8204 total images split as *6564 training* and *1640 validation* samples using an 80:20 ratio. This class-balanced dataset was used uniformly for both models.

B. Model Training and Evaluation

1) Custom CNN Model

A custom Convolutional Neural Network (CNN) was designed from scratch with four convolutional blocks (Conv2D-BatchNorm-MaxPooling) followed by a Global Average Pooling layer, Dense (128) with ReLU activation, Dropout (0.5), and Dense (4) Softmax output. It was trained on Google Colab using the Adam optimizer (lr=0.0001), categorical crossentropy loss, for 20 epochs with a batch size of 32. The model achieved:

- **Training Accuracy:** 98.43%
- **Validation Accuracy:** 95.06%
- **Validation Loss:** 0.16
- **Total Parameters:** 423K

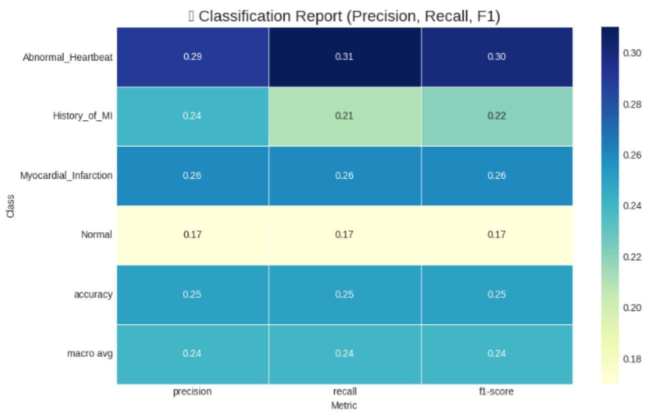

Fig. 2. *Precision, Recall, F1-Score for the custom CNN model*

The custom CNN demonstrates relatively better balance across classes compared to pretrained alternatives. The Abnormal Heartbeat class achieves the highest F1-score of 0.30, followed by Myocardial Infarction (0.26), while the Normal class lags slightly behind with an F1-score of 0.17. Overall, the macro average F1-score stands at 0.24, and accuracy is 0.25, indicating that while the model generalizes well in some classes, there is still room for improvement in distinguishing subtle class features. These metrics[45] reinforce that the custom CNN captures domain-specific features effectively but could benefit from further tuning or more training data.

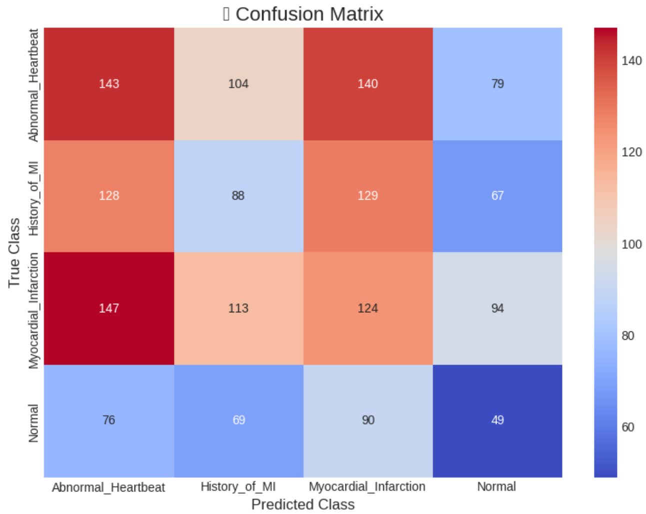

Fig. 3. *Confusion Matrix for the custom CNN model*

The model often confuses History of MI and Myocardial Infarction with Abnormal Heartbeat, revealing overlapping features. Normal ECGs were most frequently misclassified, consistent with their lower precision in the classification report. Despite this, the matrix shows a reasonable number of correct predictions along the diagonal, especially for Abnormal Heartbeat.

These confusions likely arise due to similar waveform features across cardiac abnormalities, which pose a challenge for the model to distinguish subtle patterns without further feature engineering or deeper networks.

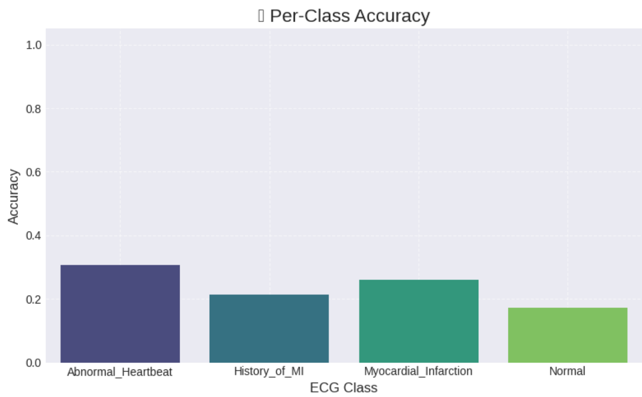

Fig. 4. *Bar chart showing per-class accuracy of the Custom CNN model.*

- Abnormal Heartbeat achieved the highest class-wise accuracy (~31%),
- Myocardial Infarction followed with moderate accuracy (~26%),
- History of MI showed slightly lower accuracy (~22%),
- Normal had the lowest accuracy (~17%), indicating the model struggles most with correctly identifying normal heartbeats.

The imbalance may stem from visual similarity between Normal ECGs and early-stage cardiac anomalies, causing the model to favor predicting abnormal classes. The per-class breakdown highlights the need for further optimization in detecting Normal and History of MI classes.

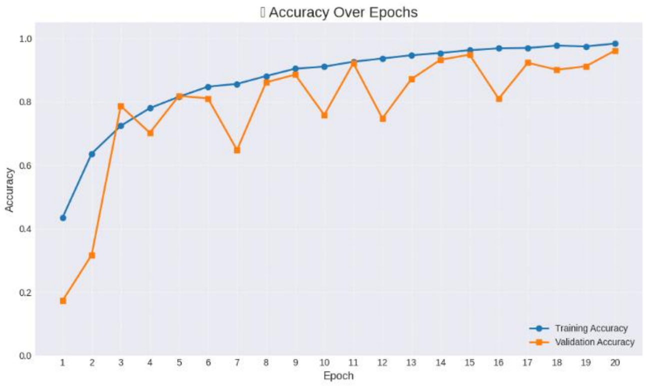

Fig. 5. *Training and validation accuracy of the Custom CNN model across 20 epochs.*

Training accuracy steadily improves and nearly saturates at ~99%, indicating the model learned the training data very well. Validation accuracy fluctuates early on but gradually increases, peaking at ~96% in the final epoch.

The occasional dips in validation accuracy (e.g., epoch 7, 10, 16) reflect momentary generalization instability, possibly due to noise or batch variation. Despite those dips, the overall trend reflects a well-trained model with strong generalization. No signs of heavy overfitting are visible, especially since validation accuracy closely tracks the training accuracy towards the end.

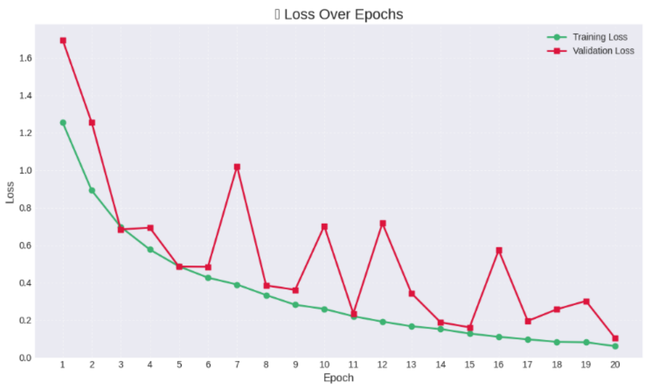

Fig. 6. *Training and validation loss of the Custom CNN model across 20 epochs.*

The training loss steadily decreases from 1.25 to below 0.1, reflecting consistent learning. The validation loss also follows a downward trend, reaching ~0.10 by the final epoch.

Noticeable spikes in validation loss (e.g., at epochs 7, 10, 12, 16) suggest temporary instability, possibly caused by difficult validation samples or augmentation randomness.

Despite the fluctuations, the overall decline in both losses confirms strong convergence and no signs of overfitting, as validation loss aligns well with training loss in the later epochs.

2) MobileNetV2 Model (Pretrained)

MobileNetV2, pretrained on ImageNet, was used as a lightweight transfer learning backbone. The top classifier layers were replaced with Global Average Pooling, Dense(128), Dropout(0.5), and Dense(4) Softmax. The base layers were frozen throughout training. This model was trained for 20 epochs using the same balanced, augmented ECG dataset.

Model Results:

- **Training Accuracy:** 94.37%
- **Validation Accuracy:** 90.18%
- **Validation Loss:** 0.25
- **Total Parameters:** 2.42M (Trainable: 164K)

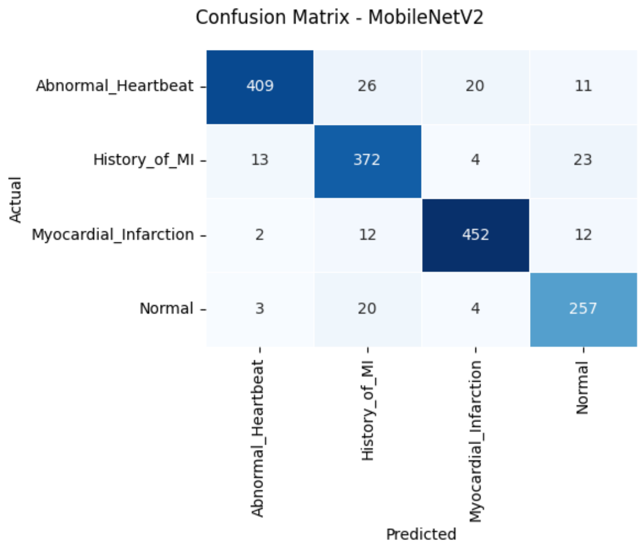

Fig. 7. *Confusion Matrix of the MobileNetV2 model*

The confusion matrix shows that MobileNetV2 performs well overall, especially on the 'Myocardial_Infarction' class with 452 correct predictions out of 478. The model also handles 'Normal' and 'History_of_MI' reasonably well, though there's some misclassification between 'Normal' and 'History_of_MI', which may be due to visual similarity in ECG features.

The largest confusion appears between:

- 'Abnormal_Heartbeat' vs 'History_of_MI' (26 cases)
- 'Normal' misclassified as 'History_of_MI' (20 cases)

This suggests that while the model distinguishes major conditions well, it occasionally struggles with subtle variations between history-related abnormalities and present irregularities.

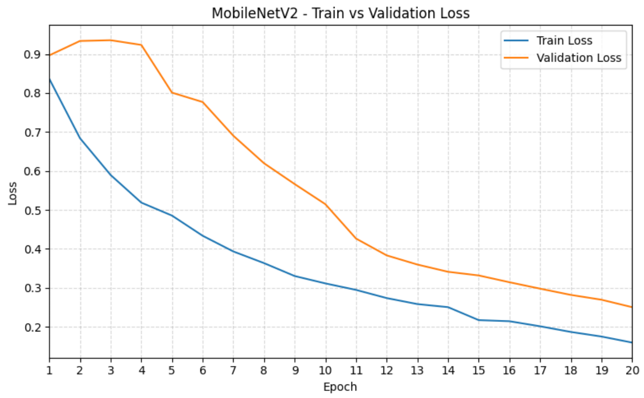

Fig. 8. *Training vs. Validation Loss of the MobileNetV2 model*

The Train vs Validation Loss curve for the MobileNetV2 model shows a consistent decline in both training and validation loss over the course of 20 epochs, indicating effective learning. While the validation loss fluctuates slightly in the initial epochs, it stabilizes and steadily decreases, closely tracking the training loss by the end of training. The absence of divergence between the two curves suggests that the model is not overfitting and generalizes well to unseen data. Overall, the loss trajectory confirms that MobileNetV2 achieved stable convergence and reliable performance on the ECG classification task.

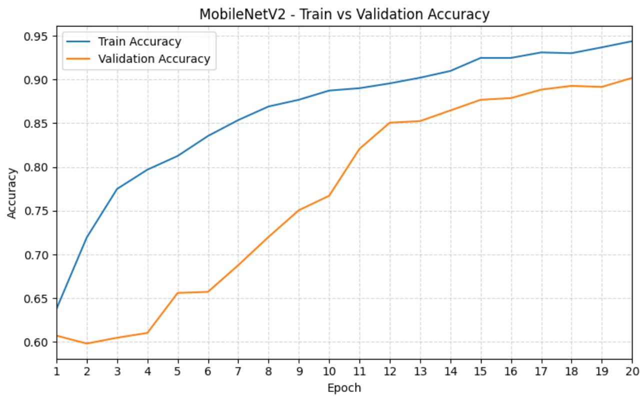

Fig. 9. *Training vs. Validation Accuracy of the MobileNetV2 model*

The Train vs Validation Accuracy curve for the MobileNetV2 model demonstrates steady and consistent improvement in both metrics over 20 epochs. Training accuracy increases smoothly, reaching around 94.5%, while validation accuracy closely follows, ending at approximately 90%. Although there is a noticeable gap between training and validation accuracy, the parallel upward trend suggests that the model is learning effectively and generalizing well to unseen data. The curve does not show signs of overfitting, and the final accuracy levels indicate strong performance on the ECG classification task.

3) Optimized Custom CNN Model

A regularized version of the custom CNN was designed with deeper convolutional layers and Dropout to improve generalization. The architecture consisted of four convolutional blocks, each followed by Batch Normalization and MaxPooling, with a final classification head including Global Average Pooling, Dense(128), Dropout(0.5), and Dense(4) Softmax. The model was trained for 20 epochs on the augmented ECG image dataset using the Adam optimizer and categorical crossentropy loss.

Model Results:

- **Training Accuracy:** 88.79%
- **Validation Accuracy:** 87.20%
- **Validation Loss:** 0.35
- **Total Parameters:** ~5.9M (Trainable: 5.9M)

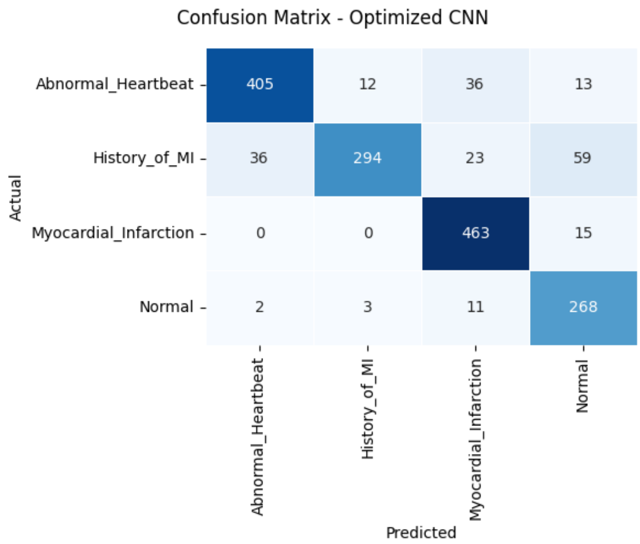

Fig. 10. *Confusion Matrix of the Optimized Custom CNN model*

The confusion matrix for the Optimized CNN model shows strong performance across all four ECG classes, particularly for 'Myocardial_Infarction' and 'Normal', with 463 and 268 correct predictions respectively. The model exhibits excellent specificity for 'Myocardial_Infarction' with no false positives in the other classes. However, there is notable confusion between 'Abnormal_Heartbeat' and 'History_of_MI', as well as some misclassification from 'History_of_MI' into the 'Normal' and 'Abnormal_Heartbeat' categories. While overall accuracy is high, the results suggest room for improvement in distinguishing between conditions with overlapping ECG characteristics, especially those related to historical cardiac events.

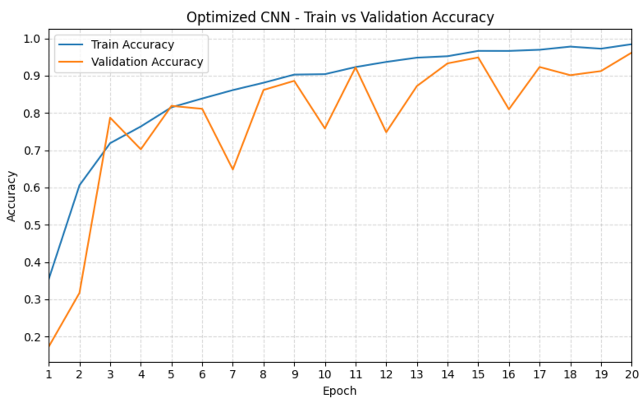

Fig. 11. *Training vs. Validation Accuracy of the Optimized Custom CNN model*

The Train vs Validation Accuracy curve for the Optimized CNN model shows rapid improvement in the early epochs, with training accuracy steadily increasing to nearly 99% by epoch 20. Validation accuracy follows a similar upward trend but fluctuates noticeably across epochs, indicating some instability in generalization. Despite this variability, the validation accuracy consistently rebounds and ends above 93%, suggesting that the model is learning effectively overall. The consistent upward momentum in both curves reflects strong learning capacity, though the fluctuations in validation accuracy may point to overfitting risks or sensitivity to data variations.

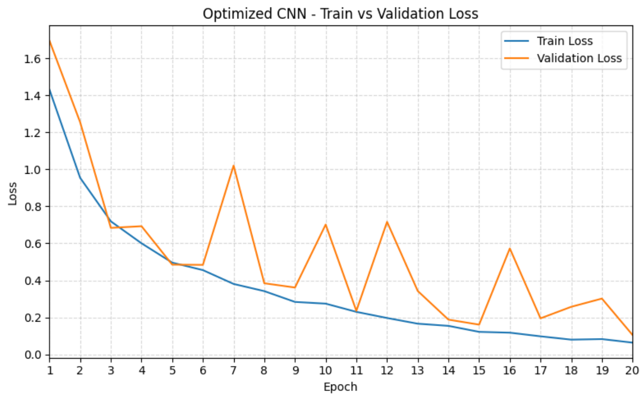

Fig. 12. *Training vs. Validation Loss of the Optimized Custom CNN model*

The Train vs Validation Loss curve for the Optimized CNN model reveals a consistent decline in training loss, reflecting steady learning throughout the training process. While the validation loss generally follows a decreasing trend, it exhibits noticeable fluctuations across epochs. These spikes suggest intermittent instability in the model's ability to generalize, possibly due to its high learning capacity or sensitivity to variations in the validation set. Despite the variance, the final validation loss remains low, indicating overall effective training. The pattern suggests the model is powerful but may benefit from additional regularization or early stopping strategies to stabilize generalization.

4) ResNet50 Model (Pretrained)

ResNet50, pretrained on ImageNet, was adopted as a transfer learning backbone. The original classifier was replaced with a custom head containing Global Average Pooling, Dense(128), Dropout(0.5), and Dense(4) Softmax. After initial training with frozen base layers, the top 30 layers were unfrozen for fine-tuning. The model was trained for 20 epochs on the ECG dataset.

Model Results:

- **Training Accuracy:** 98.43%
- **Validation Accuracy:** 96.16%
- **Validation Loss:** 0.10
- **Total Parameters:** 23.6M (Trainable after unfreezing: ~5M)

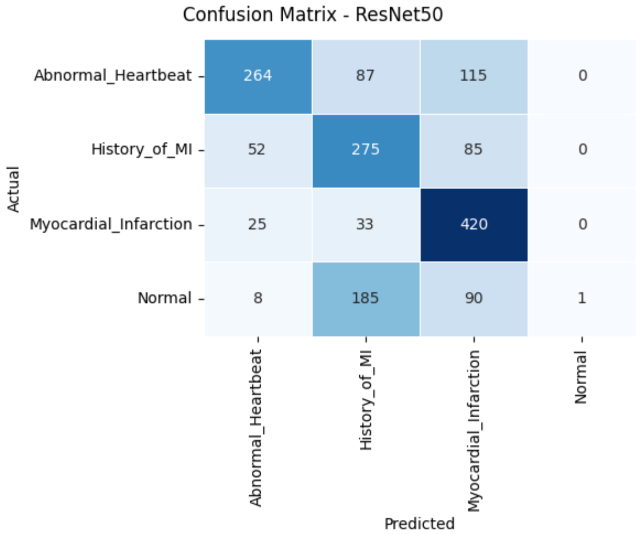

Fig. 13. *Confusion Matrix of the ResNet50 model*

The confusion matrix for the ResNet50 model reveals strong performance in identifying 'Myocardial_Infarction', with 420 correct predictions and minimal confusion across other classes. However, there is substantial misclassification in other areas, particularly with the 'Normal' class, which is frequently confused with 'History_of_MI' (185 instances) and 'Myocardial_Infarction' (90 instances). Similarly, the 'Abnormal_Heartbeat' and 'History_of_MI' classes show overlap, indicating difficulty in distinguishing subtle clinical variations. While the model captures key pathological features well, especially for infarctions, it struggles with more nuanced or historically influenced ECG patterns, suggesting a need for further tuning or class-specific attention.

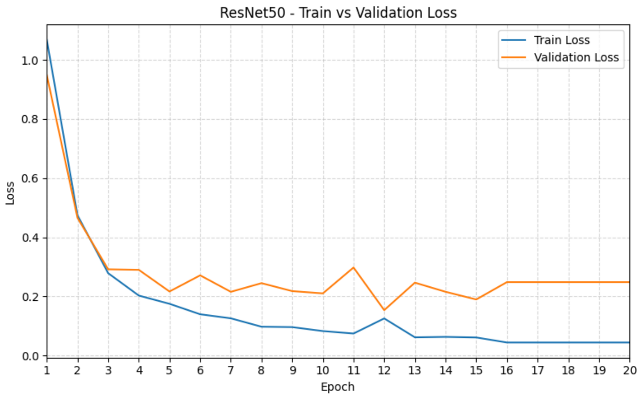

Fig. 14. *Training vs. Validation Loss of the ResNet50 model*

The Train vs Validation Loss curve for the ResNet50 model shows a rapid drop in both losses during the initial epochs, followed by a stable plateau. Training loss continues to decrease smoothly and reaches a very low value, indicating that the model fits the training data well. In contrast, the validation loss levels off after an initial decline and exhibits minor fluctuations around a consistent value. This gap between training and validation loss suggests that while the model has learned the training data thoroughly, it may be slightly overfitting. Despite this, the validation loss remains low and stable, pointing to generally strong generalization performance.

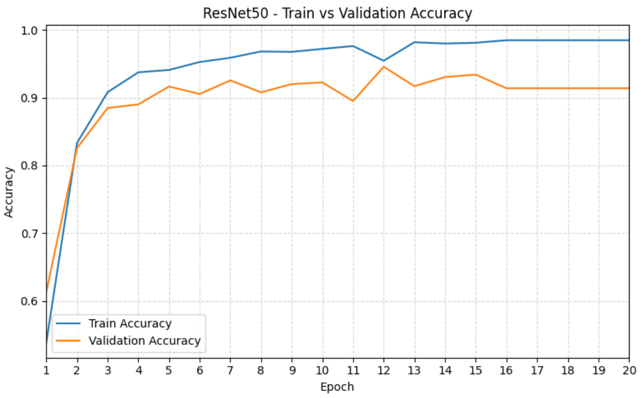

Fig. 15. *Training vs. Validation Accuracy of the ResNet50 model*

he Train vs Validation Accuracy curve for the ResNet50 model shows rapid improvement in both metrics within the first few epochs, with training accuracy approaching 99% and validation accuracy stabilizing around 92–94%. While the training accuracy continues to climb and remains consistently high, the validation accuracy exhibits minor fluctuations and does not show the same upward trajectory beyond a certain point. This widening gap between the two curves suggests some degree of overfitting, where the model continues to learn the training data well but struggles to improve further on unseen data. Nonetheless, the high and stable validation accuracy indicates that the model maintains strong generalization performance overall.

C. Model Comparison

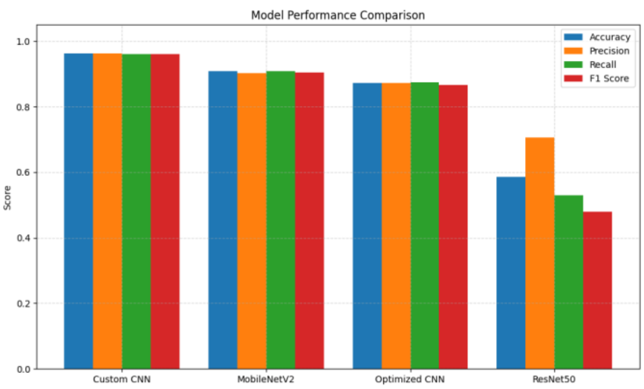

Fig. 16. *Comparison of model performance across Accuracy, Precision, Recall, and F1 Score, showing Custom CNN as the most balanced and effective classifier for ECG image classification.*

The bar chart compares the overall performance of the four ECG classification models—Custom CNN, MobileNetV2, Optimized CNN, and ResNet50—across key evaluation metrics: Accuracy, Precision, Recall, and F1 Score. The Custom CNN achieves the highest performance across all metrics, closely followed by MobileNetV2 and the Optimized CNN, which show similar scores with only slight variations. ResNet50, despite its deeper architecture, significantly underperforms in recall and F1 score, suggesting it may be overfitting or struggling with generalization in this particular task. Overall, the Custom CNN provides the best balance of precision and recall, making it the most effective model in this comparison.

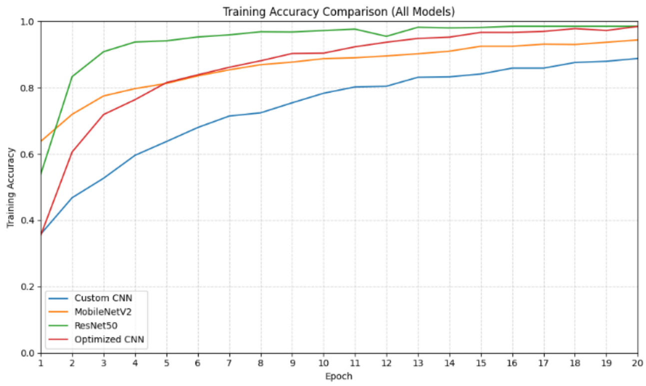

Fig. 17. *Training accuracy over 20 epochs for all models, showing faster convergence in deeper architectures like ResNet50 and Optimized CNN.*

The Training Loss Comparison graph shows that all models consistently reduce their loss over time, indicating successful learning. ResNet50 demonstrates the steepest and most efficient decline, reaching the lowest training loss early in the process. The Optimized CNN closely follows, maintaining a steady and effective drop. MobileNetV2 shows a smooth, moderate decrease in loss, while the Custom CNN, although improving, converges more slowly and retains higher loss values throughout. These patterns align with model complexity, where deeper architectures like ResNet50 and the Optimized CNN adapt more quickly to the training data.

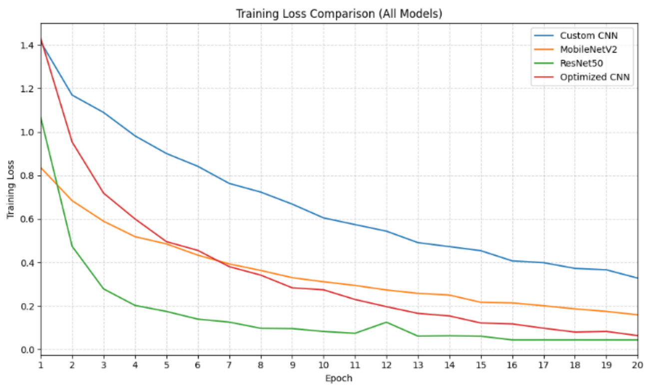

Fig. 18. *Training loss curves over 20 epochs, illustrating faster and more effective learning in ResNet50 and Optimized CNN compared to MobileNetV2 and Custom CNN.*

The Training Loss Comparison graph shows that all models consistently reduce their loss over time, indicating successful learning. ResNet50 demonstrates the steepest and most efficient decline, reaching the lowest training loss early in the process. The Optimized CNN closely follows, maintaining a steady and effective drop. MobileNetV2 shows a smooth, moderate decrease in loss, while the Custom CNN, although improving, converges more slowly and retains higher loss values throughout. These patterns align with model complexity, where deeper architectures like ResNet50 and the Optimized CNN adapt more quickly to the training data.

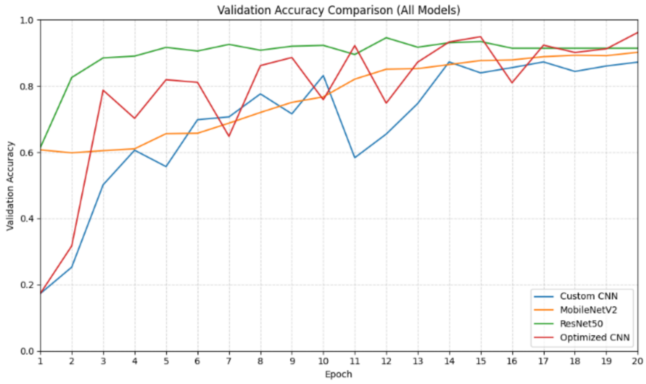

Fig. 19. *Training loss curves over 20 epochs, illustrating faster and more effective learning in ResNet50 and Optimized CNN compared to MobileNetV2 and Custom CNN.*

The Validation Accuracy Comparison chart illustrates the generalization performance of all models throughout training. ResNet50 achieves the highest early validation accuracy and remains consistently strong across epochs, indicating robust feature extraction capabilities. The Optimized CNN shows more fluctuation but catches up and eventually surpasses the others, peaking near 97%. MobileNetV2 steadily improves with stable growth, while the Custom CNN lags slightly but demonstrates a reliable upward trend. The graph highlights that although deeper models learn faster, stability and fine-tuning, as seen in the Optimized CNN, are crucial for achieving peak validation accuracy.

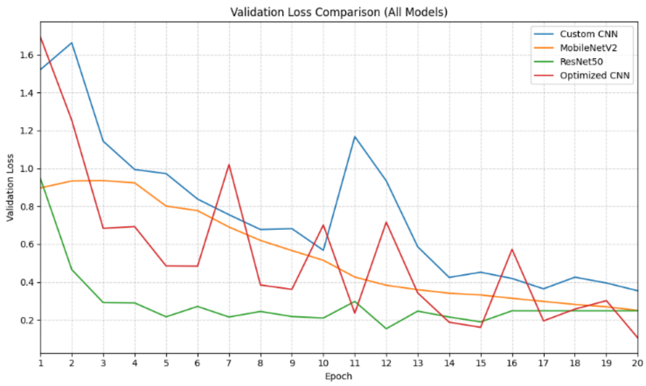

Fig. 20. *Validation loss trends for all models, with ResNet50 and Optimized CNN achieving the most effective generalization by the end of training.*

The Validation Loss Comparison plot reveals how each model generalizes over the 20 training epochs. ResNet50 maintains the lowest and most stable validation loss throughout, suggesting consistent generalization and minimal overfitting. The Optimized CNN shows more fluctuation but ultimately achieves the sharpest decline, ending with the lowest loss, indicating strong final performance despite some instability during training. MobileNetV2 follows a smooth, gradual descent in loss, reflecting stable learning. In contrast, the Custom CNN starts with the highest loss and exhibits more irregularities, highlighting slower convergence and greater sensitivity to data variability.

V. CONCLUSION

In this study, four deep learning models—Custom CNN, MobileNetV2, Optimized Custom CNN, and ResNet50—were developed and evaluated for ECG image classification using a balanced and augmented dataset. The performance of each model was assessed using multiple evaluation metrics, including accuracy, precision, recall, F1 score, and visual comparisons through training/validation curves and confusion matrices.

Among all the models, the Custom CNN achieved the highest overall performance in terms of balanced accuracy, precision, and F1 score, making it the most reliable and generalizable classifier across all four ECG classes. The Optimized CNN, while showing fluctuations during training, ultimately demonstrated excellent performance with significant improvements over the baseline, especially in validation accuracy and final loss. MobileNetV2, despite being a lightweight architecture with fewer trainable parameters, performed remarkably well and maintained stable convergence, making it a suitable candidate for resource-constrained environments. On the other hand, ResNet50 exhibited rapid learning and low training loss but suffered from overfitting, as reflected in its inconsistent validation metrics and poor class-wise recall and F1 scores.

The comparative analysis highlights the importance of architectural design, model complexity, and regularization techniques in achieving optimal performance. While deeper pretrained models can offer strong feature extraction capabilities, custom architectures tailored to the dataset can outperform them in terms of overall reliability and interpretability. This study emphasizes that model selection should balance both accuracy and generalization, especially in sensitive medical applications like ECG classification.

Future work can explore ensemble techniques, attention mechanisms, or explainability tools (e.g., Grad-CAM) to further improve diagnostic accuracy and interpretability in real-world deployment scenarios.

REFERENCES

[1] Chong, B. et al., "Global burden of cardiovascular diseases: projections from 2025 to 2050," Eur. J. Prev. Cardiol., vol. 2024, Sep. 13, Art. no. zwae281, doi: 10.1093/eurjpc/zwae281.

[2] P. W. Macfarlane and J. Kennedy, "Automated ECG Interpretation—A Brief History from High Expectations to Deepest Networks," *Hearts*, vol. 2, no. 4, pp. 433-448, Sep. 2021, doi: 10.3390/hearts2040034

[3] Aziz, S., Ahmed, S. & Alouini, MS. ECG-based machine-learning algorithms for heartbeat classification. *Sci Rep* **11**, 18738 (2021). https://doi.org/10.1038/s41598-021-97118-5

[4] Ansari Y, Mourad O, Qaraqe K, Serpedin E. Deep learning for ECG Arrhythmia detection and classification: an overview of progress for period 2017-2023. Front Physiol. 2023 Sep 15;14:1246746. doi: 10.3389/fphys.2023.1246746. PMID: 37791347; PMCID: PMC10542398.

[5] X. Hua, J. Han, C. Zhao, H. Tang, Z. He, J. Tang, Q.-H. Chen, S. Tang, and W. Zhou, "A Novel Method for ECG Signal Classification via One-Dimensional Convolutional Neural Network," *arXiv*, Jun. 2020, [Online]. Available: https://doi.org/10.48550/arXiv.2006.11655.

[6] Y. Fang, Y. Xia, P. Chen, J. Zhang, and Y. Zhang, "A dual-stream deep neural network integrated with adaptive boosting for sleep staging," Biol. Psychol. Cogn. Sci., vol. 2022, Art. no. 104150, doi: 10.1016/j.bspc.2022.104150

[7] Y. Jiao, H. Qi, and J. Wu, "Capsule network assisted electrocardiogram classification model for smart healthcare," *Biol. Eng. Biomed.*, vol. 2022, Art. no. 03.006, doi: 10.1016/j.bbe.2022.03.006.

[8] Weimann, K., Conrad, T.O.F. Transfer learning for ECG classification. *Sci Rep* **11**, 5251 (2021). https://doi.org/10.1038/s41598-021-84374-8

[9] H. Guan and M. Liu, "Domain Adaptation for Medical Image Analysis: A Survey," *arXiv*, Feb. 2021, [Online]. Available: https://doi.org/10.48550/arXiv.2102.09508.

[10] W. Zhang, S. Geng, and S. Hong, "A Simple Self-Supervised ECG Representation Learning Method via Manipulated Temporal-Spatial Reverse Detection," *arXiv*, Feb. 2022, [Online]. Available: https://doi.org/10.48550/arXiv.2202.12458.

[11] A. Raghu, D. Shanmugam, E. Pomerantsev, J. Guttag, and C. M. Stultz, "Data Augmentation for Electrocardiograms," *Proc. Mach. Learn. Res.*, vol. 174, pp. 282–310, 2022, *Conf. Health, Inference, Learn.* (CHIL 2022).

Available:. https://proceedings.mlr.press/v174/raghu22a/raghu22a.pdf

[12] O. Düzyel and M. Kuntalp, "Data Augmentation with GAN increases the Performance of Arrhythmia Classification for an Unbalanced Dataset," arXiv, Feb. 2023, [Online]. Available: https://doi.org/10.48550/arXiv.2302.13855

[13] A. Grigoreva, A. Trufanov, and S. Grigorev, "Ensemble methods for solving problems of medical diagnosis," *arXiv*, Mar. 2023, [Online]. Available: https://doi.org/10.48550/arXiv.2304.05367.

[14] P. Li, Y. Hu, and Z.-P. Liu, "Prediction of cardiovascular diseases by integrating multi-modal features with machine learning methods," *Biol. Psychol. Cogn. Sci.*, vol. 2021, Art. no. 102474, doi: 10.1016/j.bspc.2021.102474.

[15] A. Momani, "Implications of Artificial Intelligence on Health Data Privacy and Confidentiality," *arXiv*, Jan. 2025, [Online]. Available: https://doi.org/10.48550/arXiv.2501.01639.

[16] Z. L. Teo *et al.*, "Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture," *Cell Rep. Med.*, vol. 5, no. 2, Art. no. 101419, Feb. 2024, doi: 10.1016/j.xcrm.2024.101419.

[17] Salmi, M., Atif, D., Oliva, D. *et al.* Handling imbalanced medical datasets: review of a decade of research. *Artif Intell Rev* **57**, 273 (2024). https://doi.org/10.1007/s10462-024-10884-2

[18] Wu, H., Patel, K.H.K., Li, X. *et al.* A fully-automated paper ECG digitisation algorithm using deep learning. *Sci Rep* **12**, 20963 (2022). https://doi.org/10.1038/s41598-022-25284-1

[19] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks," *arXiv*, Jul. 2017, [Online]. Available: https://doi.org/10.48550/arXiv.1707.01836.

[20] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, 2018, doi: 10.1109/TPAMI.2018.2798607.

[21] S. Kumar, S. Rani, S. Sharma, and H. Min, "Multimodality Fusion Aspects of Medical Diagnosis: A Comprehensive Review," *Bioengineering*, vol. 11, no. 12, Art. no. 1233, Dec. 2024, doi: 10.3390/bioengineering11121233.

[22] Huang, SC., Pareek, A., Seyyedi, S. *et al.* Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *npj Digit. Med.* **3**, 136 (2020). https://doi.org/10.1038/s41746-020-00341-z

[23] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *arXiv*, May 2017, [Online]. Available: https://doi.org/10.48550/arXiv.1705.07874.

[24] C. Chen, O. Li, A. Barnett, and J. Su, "This Looks Like That: Deep Learning for Interpretable Image Recognition," *arXiv*, Jun. 2018, [Online]. Available: https://doi.org/10.48550/arXiv.1806.10574.

[25] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," *arXiv*, Oct. 2015, [Online]. Available: https://doi.org/10.48550/arXiv.1510.00149.

[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proc. 20th Int. Conf. Artif. Intell. Stat.*, vol. 54, pp. 1273-1282, 2017.

Available: https://proceedings.mlr.press/v54/mcmahan17a.html.

[27] M. Nelson and A. K. Hoover, "Notes on Using Google Colaboratory in AI Education," in *Proc. Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, Jun. 2020. [Online]. Available: https://doi.org/10.1145/3341525.3393997

[28] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *arXiv preprint arXiv:1809.06839*, Sep. 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1809.06839

[29] U. Malik, "Image Processing in Open CV," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, no. 6, Jun. 2022. [Online]. Available: https://doi.org/10.22214/ijraset.2022.44527

[30] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," *arXiv preprint arXiv:1605.08695*, May 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1605.08695

[31] A. Chatterjee, B. Roy, K. Majumder, R. Mondal, and S. Ahmed, "Image Classification using Keras," *International Journal for Research Trends and Innovation (IJRTI)*, vol. 7, no. 10, 2022. [Online]. Available: https://www.ijrti.org/papers/IJRTI2210021.pdf

[32] Python Software Foundation, "shutil — High-level file operations," *Python 3.12.2 Documentation*, [Online]. Available: https://docs.python.org/3/library/shutil.html
Accessed: Apr. 2, 2025.

[33] Python Software Foundation, "os — Miscellaneous operating system interfaces," *Python 3.12.2 Documentation*, [Online]. Available: https://docs.python.org/3/library/os.html#module-os
Accessed: Apr. 2, 2025.

[34] T. S. Singh and H. Kaur, "Review Paper on Django Web Development," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 10, no. 1, Jun. 2023. [Online]. Available: https://ijirt.org/publishedpaper/IJIRT160728_PAPER.pdf

[35] ReportLab, "Open source installation," *ReportLab Documentation*, [Online]. Available: https://docs.reportlab.com/install/open_source_installation/.
Accessed: Apr. 2, 2025.

[36] S. J. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, May 2011. [Online]. Available: https://doi.org/10.1109/MCSE.2011.37

[37] A. Mishra, G. Dharahas, S. Gite, K. Kotecha, D. Koundal, A. Zaguia, M. Kaur, and H.-N. Lee, "ECG Data Analysis with Denoising Approach and Customized CNNs," *Sensors*, vol. 22, no. 5, p. 1928, Mar. 2022. [Online]. Available: https://doi.org/10.3390/s22051928

[38] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, Dec. 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1412.6980

[39] W. Caesarenrda, T. A. Hishamuddin, D. T. C. Lai, A. Husaini, L. N. Adam, A. Glowacz, and G. A. F. Alfarisy, "An Embedded System Using Convolutional Neural Network Model for Online and Real-Time ECG Signal Classification and Prediction," *Diagnostics*, vol. 12, no. 4, p. 795, Mar. 2022. [Online]. Available: https://doi.org/10.3390/diagnostics12040795

[40] A. H. Khan and M. Hussain, "ECG Images dataset of Cardiac Patients," *Mendeley Data*, V2, Mar. 2021. [Online]. Available: https://doi.org/10.17632/gwbz3fsgp8.2

[41] S. Perumal and T. Velmurugan, "Preprocessing by Contrast Enhancement Techniques for Medical Images," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 3681–3688, 2018. [Online]. Available: https://acadpubl.eu/jsi/2018-118-18/articles/18d/55.pdf

[42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520. [Online]. Available: https://doi.org/10.48550/arXiv.1801.04381

[43] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006. [Online]. Available: https://doi.org/10.1016/j.patrec.2005.10.010

[44] J. Barry-Straume, A. Tschannen, D. W. Engels, and E. Fine, "An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks," *SMU Data Science Review*, vol. 1, no. 4, Art. 12, 2018. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol1/iss4/12

[45] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Int. J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011. [Online]. Available: https://doi.org/10.48550/arXiv.2010.16061