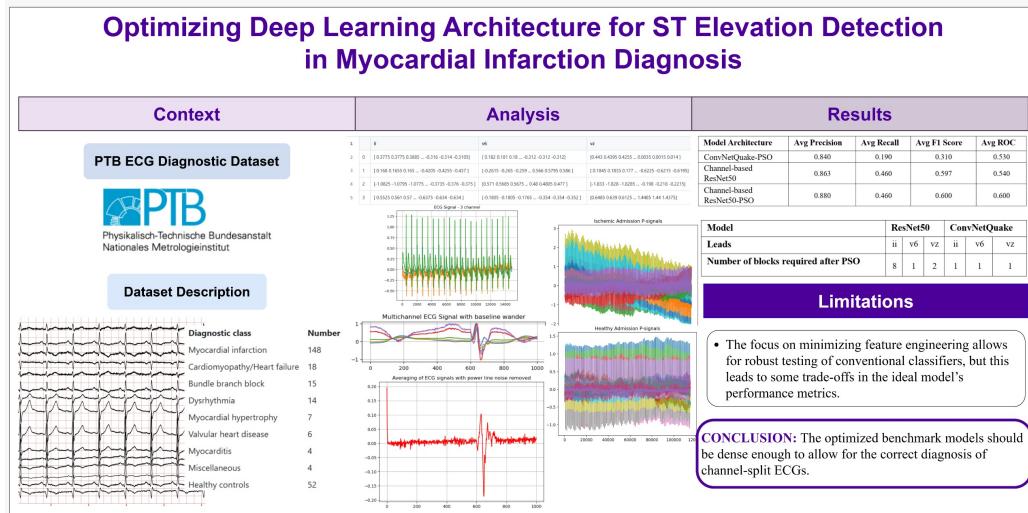


Graphical Abstract

Optimizing Deep Learning Architecture for ST Elevation Detection in Myocardial Infarction Diagnosis

Walaa H. Elashmawi, Sohayla I. Hamed



Highlights

Optimizing Deep Learning Architecture for ST Elevation Detection in Myocardial Infarction Diagnosis

Walaa H. Elashmawi, Sohayla I. Hamed

- Myocardial Infarction evolves from a range of hardly diagnosable, possibly fatal ischemic electrocardiogram cases
- Present benchmark deep neural networks are not tested on images to classify cases into either healthy or unhealthy
- Particle Swarm Optimization was used to improve present deep neural networks by forecasting the exact density required for successful diagnosis
- Issues with attaining successful diagnosis include; multi-computer set-up, statistical model engineering or workflow administration

Optimizing Deep Learning Architecture for ST Elevation Detection in Myocardial Infarction Diagnosis

Walaa H. Elashmawi^{a,b}, Sohayla I. Hamed^c

^a*Faculty of Computers & Informatics, Suez Canal University, Ismailia, Egypt*

^b*Faculty of Computer Science, Misr International University, Cairo, Egypt*

^c*Computer Engineering, Ain Shams University, Cairo, Egypt*

Abstract

Myocardial infarction (MI) is the ischemic death of myocardial tissue, often causing acute coronary syndrome (ACS). ST-elevation myocardial infarction (STEMI), diagnosed by ECG elevation, is the most fatal heart attack and increases heart arrhythmia and heart failure risk. Deep learning (DL) techniques are used in medical diagnoses to categorize electrocardiogram (ECG) delineations into feature maps to detect disease-causing abnormalities. The training of features depends on the diagnosis, and severe episodes of ischemic heart disease and STEMI increase the risk of physician error. This chapter presents a diagnostic DL model using Particle Swarm Optimization (PSO), a natural-inspired optimization algorithm, to improve the identification of unhealthy classes in data. Variations of DL architectures are examined, such as ConvNetQuake, channel-based, and transfer-based ResNet50. The results demonstrated that channel-split PSO ResNet50 has achieved the maximum results with a precision of 88%. This provides insights into enhancing cardiovascular health monitoring and diagnosis.

Keywords: ECG, Deep Learning, Convolution Neural Network (CNN), Signal Processing, Particle Swarm Optimization (PSO), S-T Elevation Myocardial Infarction (STEMI), Ischemia

1. Introduction

The heart goes through many stages of electrical activation as it pumps blood to the entire body. Historically, electrocardiograms were created as galvanometers to measure the heart's electrical activation at the body's extremities. Later innovations applied multiple electrodes, or leads, placed

throughout the body. The acquired data is plotted as waves, called electrocardiograms (ECGs), showing the difference in voltage potential by time. ECGs are widely used for efficient and accurate monitoring of cardiac health. They are used by various healthcare professionals and caretakers, including cardiologists, physicians, emergency-room paramedics, and nurses.

The accuracy of physician inspection by ECG is approximately 54% and increases to 67% with extra training [1]. Computerized ECG interpretation proved an accuracy of 88.0% for as long as 20 years, proving it is a long-standing method of cardiac health assessment that will continue in the future [2]. Professionals use manual methods to diagnose heart problems, even life-threatening ones such as Myocardial Infarction (or colloquially known as "heart attacks"), from ECG signals. The main problem with manual diagnosis of heart attacks, like other time-series data, is the difficulty of detecting and categorizing different waveforms and morphology in the signal. Another less severe difficulty is the understanding of certain "hidden" patterns. For a human, a proper diagnosis is extensively time-consuming and prone to errors. Consider life-threatening conditions that account for approximately one-third of all deaths around the globe. Therefore, obtaining the best categorization and diagnosis from ECGs is paramount.

ECG features are divided into two main types, waves, and intervals [3]. Waves are segments of the ECG that represent the heart muscle circulatory cycle. Intervals represent locations between each of the waves. Five main waves exist (P, Q, R, S, T) and are shown in Figure 2. P wave reflects the spread of electrical activation (depolarization) through the heart's right and left atria as shown in figure 1. The T wave represents the heart muscle relaxation, known as ventricular repolarization. PR interval lies between the beginning of the P wave and the beginning of the Q wave (i.e., initial depolarization or contraction of the ventricle, the heart is about to "pump"). The QRS complex represents the total depolarization of the ventricles ("the pump is working"). The QT interval starts at the beginning of the Q wave and ends at the T wave. It represents the entire period of contraction and relaxation of the ventricles (i.e., the pump starts at the bottom chambers, the ventricles, where the pressure is highest, and then fluid flows up to the top chambers). Finally, the ST segment represents the interval between the end of the S wave (ventricular depolarization) and the beginning of the T wave (ventricular repolarization). It represents the period when the ventricles are depolarized ("contracted, will relax").



Figure 1: Heart muscle CT scan showing coronary vessels and coronary circulation

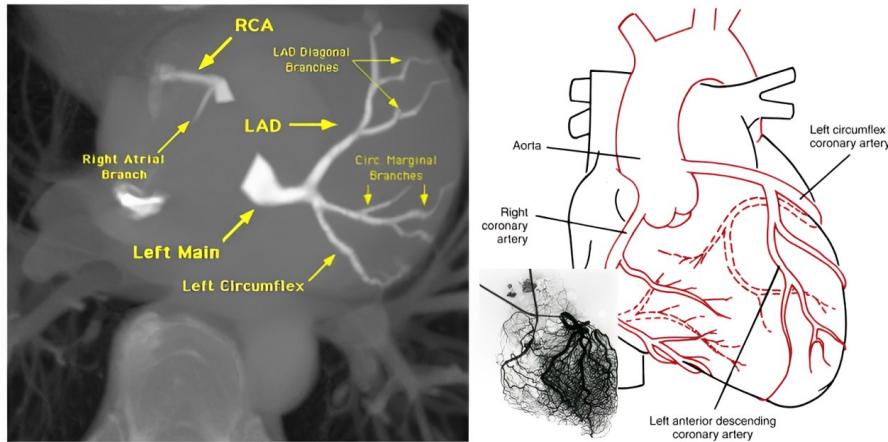


Figure 2: Normal and Anterior ST-segment Elevation pattern of ECG signal. Variations of ST elevation.

Due to its possibly missable patterns of risk factors ([4]), diagnosing myocardial infarction (MI), usually known as a heart attack, often relies on the use of an electrocardiogram (ECG or EKG). These miss-able signs are usually ones of acute coronary syndrome (ACS), which, in approximately 80% of patients, will lead to MI. Physicians usually look for abnormal patterns in the ECG waves or intervals detected by electrodes in multiple locations, called leads. The values of the leads are the members in the courtroom of body ionization. Basic acute patterns are required for confirmatory diagnosis of MI, regardless of the type of atypical activity in the ECG itself. The most common cases constituting MI data of cardiac tissue occlusion (blockage) and necrosis (death) are:

- **Anterior STEMI:** Anterior, meaning nearer to the front, ST-segment elevation causes the majority of deaths if not treated urgently. This type of STEMI is caused by the blockage of the left anterior descending artery (LAD). This type of STEMI is not straightforward to identify due to the multitude of ECG patterns and leads required to be investigated for successful diagnosis.
- **Inferior STEMI:** Inferior, meaning lower in position, ST-segment elevation accounts for 40-50% of all MI cases. In most cases, it involves blockage of the right coronary artery (RCA), while in rarer cases, it may involve the circumflex artery (CX). The former blockage potential ECG pattern was demonstrated in Figure 2.
- **Posterior STEMI:** Posterior, meaning nearer to the hind, ST elevation involves the blocking of the posterior descending artery (PDA) or posterior interventricular artery (PIA). This vessel branches from the RCA, meaning that inferior STEMI can cause this type of blockage and further result in posterior STEMI.

The previous types of STEMI are all caused by blockages to certain coronary (of the heart) blood vessels. If undetected, each blockage will cause a cessation of oxygen and blood flow to the myocardium or middle heart muscle. If this repeats, it can cause irreversible muscle damage and sudden death. To minimize life-threatening consequences, it is crucial to gather clean patient ECGs to identify case symptoms and features. Obtaining those features from patient data requires a multi-step process for extracting, transforming and loading high-quality ECG recordings [5]:

1. **Baseline ECG:** A baseline ECG is taken as an initial evaluation of the patient suspected to have MI or has had a heart attack. This step aims to detect pre-existing anomalies that could influence subsequent findings' validity.
2. **Serial ECGs:** Serial ECGs involve taking multiple ECGs at different time points, usually at regular intervals, to monitor changes in patient ECG data. This step plays a vital role in facilitating continuous monitoring and disease progression tracking, particularly in situations with a high risk of disease exacerbation.
3. **ST-Segment Changes:** This is the key feature observed on an ECG to diagnose MI. There are several patterns to ST changes, such as

elevations and depressions (i.e., the diagnosis of STEMI usually focuses on elevations).

Additionally, some features are used to confirm the diagnosis of MI, such as T-wave changes and Q-waves in case of significant damage to the heart muscle. Figure 3 summarize an example of serial monitoring of STEMI.



Figure 3: Serial ECGs of a patient throughout the day. Baseline wander and power line interference of an ECG.

Achieving an accurate diagnosis is contingent upon the acquisition of high-quality ECG recordings, free from artifacts and electrical noise. Any bio-signal acquisition falls under many interferences from the external environment that affect the quality and accuracy of that signal. ECGs are no exception. According to [6], there are many types of interference:

1. **Muscle Artifacts:** Muscle artifacts occur as high-frequency noise on the ECG, often due to the movement of muscles near the electrodes. They can make actual cardiac signals indiscernible.
2. **Baseline Drift:** Baseline drift is a slow, gradual shift of the baseline of the ECG recording over time, as shown in Figure 3. It is usually caused by poor electrode contact and can mislead the identification of abnormalities.
3. **Power Line Interference:** Power line interference occurs as harmonic spikes in the ECG, as shown in Figure 3, often due to electrical noise from nearby power sources.

To minimize the influence of artifacts and ensure accurate signal interpretation, appropriate filtering techniques such as band-pass or band-stop filters are recommended during ECG acquisition. However, even with meticulous physician efforts in diagnosing ST-elevation myocardial infarction (STEMI), misdiagnosis remains a potential clinical challenge. In light of these considerations, the primary objective of this paper as follow:

- Explore different architectures for the diagnosis of STEMIs.
- Propose a cardiologist-level optimized deep learning model to detect unhealthy patients from stochastic batches.

This paper is organized as follows: Section 2: Background, delves into the latest advancements in MI diagnosis models, exploring how researchers are leveraging technology to improve diagnostic accuracy and efficiency. Also, this section highlights the difference between major architectures. Section 3: Transfer Learning, refers to the neural network designs that are used to process the ECGs and provide the required diagnosis. Section 4: PTB ECG Diagnostic Database, outlines the dataset mainly utilized with the proposed architecture. Section 5: Proposed Methodology, focuses on this paper's methodology and major optimization technique. Section 6: Experimental Results and Performance Analysis, compares the results attained among the different architectures and their optimized versions. Section 7: Limitations of the Proposed Methodology, summarizes some limitations faced while working on the models discussed in this paper. Finally, section 8: Future Research Directions and section 9: Conclusion, wrap up all the conclusive details of this study and propose other ways to improve it.

2. Background

Accurate diagnosis of MI is paramount for improving patient outcomes. This section explores the evolving landscape of MI diagnosis models, highlighting how researchers harness technology's power to achieve faster and more precise detection. We will delve into recent advancements in artificial intelligence (AI), specifically machine learning (ML) and deep learning (DL) approaches used to analyze patient data including, ECG signals, vectorcardiogram (VCG) signals and biomarkers.

2.1. Categorization Based Models

Categorization based models make the most use of categorizing different modes of patient data into classes, thus presenting mainly a healthy and a unhealthy class. Class-distinguishing features are drawn using statistical methods, mathematical transforms, ML techniques or a combination of either. While they are not highlighted in the works cited in this subsection, other intermediate classes can be detected as well.

Building upon the work of [7], several feature extraction techniques such as: Local Binary Pattern (LBP), Higher Order Spectra (HOS), and Discrete Wavelet Transform (DWT), were leveraged into a method that provides an adaptable, more personalized, real-time healthcare solution. In addition, optimization heuristics were capitalized alongside the mainly statistical techniques mentioned.

To address potential limitations in existing diagnostic machine learning methods, the authors of [8] proposed a novel approach that leverages dimensionality reduction using Linear Discriminant Analysis (LDA) in conjunction with hybrid feature selection algorithms. Their study aimed to enhance the decision-making capabilities of ML algorithms, specifically Support Vector Machines (SVM) and Decision Trees (DT). The effectiveness of their approach was further evaluated using Artificial Neural Networks (ANN). The results demonstrated significant improvements across various performance metrics, including F1 score, specificity, and sensitivity, in all test cases.

In comparison to other literature works that usually used morphological images or bio-signals, [9] proposed the diagnosis of Acute Myocardial Infarction (AMI) by identifying risk factor biomarkers from ECG metadata. The results proposed a five feature model that included cardiac troponin I, HDL cholesterol, HbA1c, anion gap, and albumin, which achieved comparable results to other models with more features.

[10] first subject ECGs to Flexible Analytic Wavelet Transform (FAWT) to output ECG sub-bands. Each reconstructed sub-band FAWT coefficient is input into a sample entropy framework which calculates that entropy of that particular sub-band. Finally, a classification is yielded after features are subjected to conventional ML algorithms.

[11] proposes a method that uses both temporal (intervals between beats) and morphological (beat patterns) to categorize ECGs. Because this analysis requires complex, multi-dimensional relationships that are modeled by support vectors, multiple SVMs are used to generate feature descriptors. Each

feature descriptor has a corresponding machine, where the feature descriptors were divided among morphological and temporal features. Higher accuracy and performance of heart beat categorization are advantages of this method.

According to [12] and [13], classification and regression-based architectures are trialed on complimentary diagnostic wave and heart morphological features, ECGs and VCGs, along with feed-forward networks. In addition, features like chest pain or other clinical features may be incorporated into tests. At first, features are extracted via windows or retraced via transforms. Then, a neural network with numerous dense blocks, such as VGG-16, is used to refine the traced features. This feed-forward technique serves the classification problem. After classification, machine learning techniques such as logistic regression (LR) and decision tree (DT) are used to calculate the probability of a new case being of a certain class, which serves the regression problem. The main advantage of their proposed architecture is its applicability in clinical settings despite its potentially difficult code.

2.2. Deep Learning Based Models

Deep Learning (DL) based models are those that exclusively use neural networks to draw feature maps for each class. The loading, transforming and pre-processing of data can utilize no neural networks, but the classification and optimization processes primarily use neural networks. Unlike last subsection, this subsection outlines studies that have relied on ANNs rather than statistical categorization of class-distinguishing features.

[14] compared the performance of processed ECGs using CNN-Dense with different optimizers such as Adadelta, Adam and Nadam, where each optimizer performance is ranked by the number of data used to compute the gradient descent. The system processes one-dimensional input ECG signals and passes them through multiple layers for diagnosis. The proposed CNN architecture includes layers for input, convolution, max-pooling, dropout, dense, and Softmax. During testing, different neural network architectures like EchoState, ECG-Net, CNN-Adadelta, Block-based neural networks, FC-ANN, VGGNet, GoogleLeNet, AlexNet, ANN, and RNN are presented.

The authors in [15] employed deep transferable representations to accurately categorize heartbeats for monitoring cardiovascular health. They proposed a method using deep convolutional neural networks to classify arrhythmias according to the Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard and transfer this knowledge to the classification of MI. The methodology involves pre-processing ECG signals,

training a classifier, and utilizing learned representations for MI prediction. Their study demonstrated competitive accuracy results compared to other existing methods and showed effective separation of data points for both classifying arrhythmia and MI (thus confirming the medical claim of the correlation between both cardiac conditions).

Applying deep learning techniques along with entropy, specifically artificial neural networks with one-dimensional convolutional layers for signal processing tasks, is employed by [16]. To mitigate the vanishing gradient problem and enhance model capacity, the convolutional layers within the proposed neural networks incorporate residual connections, as described in the cited work. The networks were trained using the Adam optimizer with a learning rate adjustment during training to prevent overfitting. Entropy-based features such as Shannon entropy, Approximate entropy, and Sample entropy improved the accuracy of heart disease classification. The main investigation in their study is the integration of SincNet layers for low-level feature extraction in future work. [17] proposed an architecture that classifies heart attacks without machine learning steps. Unlike [16], focal loss was used to handle data imbalances and constitute focus in difficult classes.

[18] focuses on short segments of ECG signals collected through bio-metric recognition. The study includes an analysis of identification and verification performance using bio-metric recognition processes, with scenarios for one-to-many matching (identification) and one-to-one matching (verification). The main aim of the method in [18] is to improve recognition performance over long periods and under different signal changes. Various deep learning models were tested for biometric recognition, showing different levels of accuracy based on the segmentation method and session length of the signal.

While a different application of ConvNetQuake architecture is presented by [19]. This architecture was originally designed to identify earthquakes. Certain ECG leads are input into a multi-channel neural network comprising convolutional layers, flattening layers, and a fully connected layer.

Deep learning models, despite their impressive capabilities, can often face challenges in optimization due to complex architectures and high dimensionality. This is where natural-inspired meta-heuristic algorithms come in. These algorithms mimic natural processes, such as swarm intelligence (like ants or bees) or evolutionary selection, to guide the training process of deep learning models. By drawing inspiration from these natural phenomena, meta-heuristics can effectively search for optimal solutions within the vast parameter space of deep learning models. The authors in [20] propose a neu-

ral network whose architecture and hyperparameters can be optimized using Marine Predators Algorithm. After reliable ECG features are extracted, a random solution is initialized, and then it is trained with a neural network to update the prey position. The diagnosis is successful if the network converges; if not, a solution is re-established, and then another block of the CNN is trained. In addition, a search for the best learning rate parameter value is implemented before a neural network can be improved [21]. Some improvements can be enabled by the python compute engine CUDA which works by parallelizing an algorithm or a neural network using multiple GPUs.

The existing literature has mostly focused on diagnosing myocardial infarction through ECG categorization or image-based deep-learning methods. Even though many of the architectures in literature perform well in terms of accuracy, precision, and efficiency, they are mostly too complicated to be featured in portable systems, such as traditional web applications. In addition, the models featured in this section involve complex designs that heavily rely on feature engineering and reduction, such as in [12].

In the current research, almost no swarm optimization techniques were tested with traditional, benchmarked neural networks.

In contrast, the proposed model introduces a less-reliant feature-engineering version of optimized traditional neural networks that are lightweight enough to be embedded into the backend of portable applications. The model seeks to optimize traditional deep neural networks with stochastic sampled, feature-reduced ECGs. We aim to design an optimized model that can be embedded along with models of other data modalities in a web app, for example, rather than a singular non-optimized model in a particular application (web or desktop or medical server). Overall, the results for this architecture are promising enough, especially with unhealthy patients, to make way for later or heavier implementations with several rounds of feature engineering.

3. Transfer Learning: ALEXNET, RESNET, CONVNETQUAKE

Transfer learning is a technique that applies a trained model intended for a task as a basis for another related task. This approach employs the knowledge obtained from the source task to improve the learning or performance of the target task. In this work, the main aim is to classify and detect patterns of MI in ECGs, thus leading to a diagnosis, so image and wave-based neural networks will be used to differentiate between the required classes (healthy or unhealthy). Transfer learning neural networks applied here are

AlexNet(2012), which was primarily used for the ImageNet dataset comprising of several type of animals, ResNet(2015), which was used to shortcut AlexNet and ConvnetQuake(2018), which was used to detect changes in seismic waves. The following section explains how each of the neural networks was built and how was each applied to MI diagnosis.

3.1. *AlexNet*

The authors [22] first trained a neural network to classify 1.2 million images in the ImageNet LSVRC-2010 contest into 1000 different classes, with the goal to advance the then-state of the art computer vision image classification performance. On test data, the top-1 and top-5 error rates were 37.5% and 17% which is better than previous neural networks. To outline AlexNet, approximately 600,000 neurons were used leading to 60 million parameters, all accelerated with non-saturating neurons and a multi-CPU (or multi-GPU) implementation.

3.1.1. *Key Architecture of AlexNet*

In this subsection, we outline precisely how the architecture of AlexNet was built to accommodate for this work’s proposed model. A classic AlexNet was used to experiment with optimized MI diagnosis.

- **Layers:** AlexNet consists of 8 layers, 5 of which are 2-dimensional convolutional layers(originally used 2 GPUs, [22]), followed by a series of maxpooling layers and 3 fully connected layers with a final 2 class softmax layer. Experiments with AlexNet intially featured 15 classes, as per the dataset in section 4, but due to a suboptimal learning rate and the risk of saturation, subsequent experiments only featured either healthy or unhealthy patients (2 classes). Feature graphs were verified to work correctly after passing through convolutional layers, and label generation was tested after saving the entire architecture. AlexNet employed large receptive fields(large stride) in the first layer which gradually decreased as the layers progressed. Max pooling layers use a pool size of s pixels (with $s = 2$) among a neighborhood of $z \times z$ ECG image pixels (where $z = 3$). This setup introduces overlapping pools, which contrary to traditional practice, decreases the error rate by a standard 0.4%, as in [22], and decreases over-fitting, as demonstrated in both standard practice and our experiments.

- **Activation Function Non-linearity and Saturation:** While a neural network operates, it outputs pre-defined metrics like accuracy, error rate, loss, or bias according to a function such as sigmoid or tanh. A fault occurs when the metrics cease to change at all for a considerably long period of time, causing the points in that curve or line stop changing which leads to a zero derivative. The phenomenon is called saturation. If a neural network output saturates quickly, its gradient vanishes over time as the derivative becomes zero. AlexNet used the novel Rectified Linear Unit (ReLU) which accelerates training and addresses the vanishing gradient problem. In terms of training time and gradient descent (calculated using a regular cost function), ReLU’s nonlinearities saturate much slower than other nonlinear activation functions like sigmoid or tanh.
- **Normalization:** The ReLU used in both standard and our AlexNet experiments does not require input normalization to prevent saturation. As long as some training ECG sample images produce a positive input to the ReLU, learning metrics will be yielded. This normalization strategy aids generalization and normalizes activations of each common kernel of neurons (local response normalization, LRN). Although batch normalization was not employed in standard or this paper’s experiments, it can be used to stabilize and accelerate training.
- **Reducing Overfitting and Dropout:** To reduce the risk of overfitting, AlexNet employs a dropout parameter in the fully connected layers. Dropout is when random neurons are ignored during training which further regularizes the network. Additionally, random shuffling of batches of ECG images may help also prevent overfitting. Simple random shuffling of training ECG images, without dividing into random batches proved to be faster, more stable and showed no difference in terms of metrics, as detailed in section 6. The rotation, color shuffling, and horizontal flipping of training and testing images showed no improvement in terms of neural network efficacy and are therefore not discussed in the experimental results section 6.

3.1.2. Similar Applications in Literature

[23] uses the orthogonal leads and the different time segments from the continuous ECG waves to detect classes that correlate or equate to arrhythmia. Signal processing techniques used aim to remove noise, artifacts and

other possible distortions that can affect the accuracy of classification. ECG signal transformation into the frequency domain using Continuous Wavelet Transform (CWT) enables the feature extraction process to focus on 2-D representations of ECG signals in the frequency domain, called ECG power scalograms. For instance, different lengths of scalograms (1s, 2s and 3s of CWT ECGs) were input to the conventional AlexNet [22]. After training and testing the model, it was concluded that 3-second signal length yielded the best results. Therefore, the longer the input signal length, the better classification accuracy yielded.

In our proposed model given in section 5, a window of 10,000 milliseconds was used to find the Hermite transform of the input ECG signals, so it can be expected that our model aligns with the conclusion of the aforementioned study [23].

[24] starts with a data merging between 3 datasets to compile a new dataset with more than 5,000 multi-lead ECG recordings that clearly suffice for a well-developed deep learning classification. Pre-processing of this novel dataset focuses on Z-score normalization to ensure a consistent and a dimension scaled ECG. Similar to [23], feature extraction in [24] utilizes CWT to extract textural features from ECG scalograms. The 15-second scalograms are then input into a modified AlexNet where each epoch updates parametric values to enhance classification performance. Conclusively, [24] proves that AlexNet is highly adaptable with continuous waves without introducing extra complexities, unlike its counterparts (Discrete Wavelet Transform (DWT), Random Forest (RF), DWT Support Vector Machine, SqueezeNet and GoogleNet).

Our proposed model and our results in sections 5 and 6 do show the adaptability of all the neural networks used, but with major trade-offs in terms of model evaluation metrics. For example, despite proving to handle modifications required to deal with more complex continuous data, AlexNet falls into fast convergence and low accuracy among all types of orthogonal leads(standard, chest, Frank).

[25] uses the conventional pre-processing techniques, but with feature extraction, three transform techniques were used to extract the most reliant features for ECG classification: Fractional Discrete Cosine Transform(FDCT), Radon Wavelet Transform(RWT), and Fractional Wavelet Transform(FWT). To complete the feature selection stage, a bio-inspired optimization technique called Red Fox Optimization was introduced. The selected features are input into an i-AlexNet model to yield the final classification. To conclude,

i-AlexNet metrics outperformed all those of conventional algorithms (Bidirectional LSTM(BiLSTM) and Frequency Division Dehazing Network(FDDN)).

Our proposed model in section 5 used another bio-inspired optimization technique called Particle Swarm Optimization (PSO), but not on feature selection of the ECGs themselves, rather to find the perfect model density required to yield an oscillation-free, dimension-scaled classification.

3.2. ResNet

[26] addressed several challenges faced by AlexNet and other previous neural networks, mainly that dense networks are hard to train and that vanishing gradients may continue to appear. The main aim is to design a residual learning framework to ease the training of networks that are deeper than older ones, thus improving image recognition and computer vision tasks. The standard experiments were carried out on the ImageNet dataset where the residual network can contain up to 152 layers, all while having a lower complexity. Standard experiments have shown a 3.57% error rate on the test dataset(much lower than the 17% error rate of AlexNet).

3.2.1. Key Architecture of ResNet

Prior networks to ResNet, notably AlexNet, faced a crucial, irreversible issue of decreasing metrics and worsening performance as a very deep AlexNet for example quickly converges and saturates after many reruns and predictions. A rerun AlexNet degrades over time and as it becomes denser. This degradation has adverse effects on most aspects of a good application aspects of a network, such as usability, reliability and longevity of that network. Degradation is not caused by overfitting, so adding more layers, randomizing parameters, or even shuffling batches of input only increases training error and does not solve the issue. To combat this, it became necessary to design a very deep neural network that allows for possible varieties in parameter dimensions and feature sizes per layer without risking degradation. In this subsection, we outline precisely how the architecture of ResNet accommodated this work’s proposed model. In section 6, ResNet50 was used to train and test this paper’s proposed model among different data channels.

- **Layers:** According to [26], ResNet is comprised of a series of convolutional layers followed by parameter-increasing shortcut residual blocks. Each residual block contains at least two convolutional layers. Standard experiments used a total of between 34 to 152 layers in the residual

network; however, due to the size of the generated feature arrays or tensors, a maximum of 50 pre-trained layers were more convenient for the experiments in this paper. Max pooling has been performed at the end before the final fully connected (dense) layer to determine the final number of feature maps.

- **Residual Blocks:** In plain networks(i.e., networks without residual blocks), there are two basic design rules that should be followed: (i) for the same output and feature map size, the convolutional layers must have the same number of filters; (ii) when the feature map size is halved, the number of filters should be doubled to preserve the time complexity per layer. If a residual network is to converted to a plain network, down-sampling will need to be performed by convolutional layers with a lower value of stride. Nevertheless, the network still ends in a dense layer and a softmax layer. In a residual network, shortcut connections can be inserted via identity shortcuts which can be directly used when the input and output are of the same dimensions(as in plain networks). The target of residual blocks in any ResNet is to shortcut connections such that very deep computer vision tasks can be performed without extra computations and parameters. For instance, when the dimensions increase (for example due to larger images in a dataset), a residual function with large dimensions can either be projected by a smaller dimension function with negligible biases or be shortcut with identity mapping and padded with extra zero entries [26]. In both cases, element-wise addition between each input, bias and residual function is negligible. This method allows for a ResNet to bypass connections and mitigate issues caused by very high parameter dimensions (often characteristic to deep networks). Figure 4 below shows how each residual block works among the ResNet-50 parameters.
- **Deeper Networks and Gradient Flow:** As ResNet features the projection of large dimensions by smaller ones, it is easily scalable with many variations such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, where each number symbolizes the number of layers in that network. Starting ResNet-50, a bottleneck architecture is used to reduce computational complexity of the network. Basically, each block of the bottleneck architecture is comprised of a 3×3 convolution layer sandwiched between two 1×1 convolution layers. As

Layer (type)	output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
flatten_1 (Flatten)	(None, 100352)	0
dense_3 (Dense)	(None, 1024)	102,761,472
dense_4 (Dense)	(None, 512)	524,800
dense_5 (Dense)	(None, 2)	1,026

Total params: 126,875,010 (483.99 MB)
Trainable params: 103,287,298 (394.01 MB)
Non-trainable params: 23,587,712 (89.98 MB)

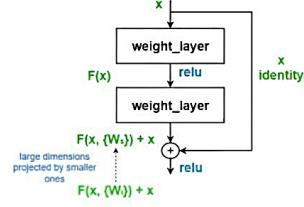


Figure 4: A summary of ResNet-50 parameters and the working of a residual block

previously mentioned, vanishing and exploding gradients are reduced because ResNet maintains the gradient flow during forward and backward passes(backpropagation). The results in section 6, incorporate a batch normalization layer after each convolutional layer, stabilizing training by normalizing inputs before they pass through each layer. In fact, it is more beneficial to output normalized feature maps after each convolutional layer.

3.2.2. Similar Applications in Literature

[27] used more advanced denoising techniques such as wavelet transforms, down-sampling to 200Hz, and QRS detection via the Pan-Tompkins algorithm. A 13-layer ML-ResNet network was employed with a single lead feature branch made up of 3 residual blocks for MI detection and localization. In order to attain better spatial information representation, the single lead network learns representative features at different levels. After the initial single lead feature experiment, multi-lead feature fusion was performed to provide a more comprehensive diagnosis in comparison to the initial experiment. Some improvements proposed in [27] include mass data utilization, inter-patient variability models and feature expansion.

Our experiments in section 6 considered some of the improvements in [27]. For instance, ResNet experiments featured a variety of leads to ensure inter-patient variability and feature expansion. Different experiments with different leads provide insights on STEMI cases (anterior, inferior, posterior). In addition, rather than a simpler ResNet-13, ResNet-50 with a bottleneck architecture was used to make use of better feature detection and classification.

[28] uses an exclusive hybrid model that combines a modified ResNet for feature extraction and a novel-trained Vision Transformer(ViT) model. The

main innovation in this paper is the focus on the improvements of the ViT architecture. Basically, a multibranch ViT network was used with a channel attention mechanism. This slim model enhances the richness of feature and information learning with a reduced complexity. The main disadvantage of this model is that it may overlook crucial local details.

3.3. ConvNetQuake

Recently, the call for improvement of seismic hazard assessment and management has grown due to induced seismicity. The frequency of earthquakes and other seismic hazards has increased and so has the amount of seismic data collected [29]. One of the artificial intelligence advances in the detection of continuous seismic waves is the highly scalable neural network, ConvNetQuake. The main advantage of ConvNetQuake is that the orders of magnitude are much faster than other feature-engineering reliant methods. The application of ConvNetQuake is not limited to earthquake detection. As explained in [19], ConvNetQuake can be used as a deep learning technique to diagnose abnormalities in ECG data (continuous wave data similar to seismic earthquake data) to yield good results without the heavy reliance on feature-engineering and feature-reduction.

3.3.1. Key Architecture of ConvNetQuake

Unlike previous transfer learning neural networks, ConvNetQuake does not provide new innovations to the technologies used in neural network design. However, in terms of MI diagnosis, experiments were carried out to provide a cardiologist level training [19]. The core limitation of [19] is that no comprehensive tests were carried out. In this subsection, we outline the major architecture of ConvNetQuake to accommodate the diagnosis of MI.

- **Preparation:** The most important difference between how ConvNetQuake and other neural networks were accommodated is the need for precise preparation of input tensors to the network. In our experiments, random batches of min-max normalized ECGs were used regardless of their corresponding classes. This was done so that most of the classification task is left to the neural network (and later to the swarm optimization as in section 5 and 6).
- **Layers:** ConvNetQuake used a series of blocks where each block is composed of a series of convolutional layers followed by flattening layers, a sigmoid layer and another series of batch normalization layers.

During backpropagation, a ReLU activation is used to call each pair of convolution and batch normalization layers . At the end, the feature maps are reshaped, flattened and a sigmoid classification layer is used. Note that ConvNetQuake does not prioritize the sensibility of the classification or would've otherwise used a softmax classification layer. This difference is ignorable because there are only 2 classes, either a healthy or an unhealthy patient.

3.3.2. Similar Applications in Literature

ConvNetQuake is a relatively novel model which was mainly used to leverage a better, more cardiologically accurate diagnosis [19], as in section 2. To review ConvNetQuake, [30] challenged conventional accuracy and sensitivity in MI diagnosis. Specifically, ConvNetQuake is among many medically applied models with a low interpretability, "black box" nature even when using area under curve (AUC) rather than accuracy. There is a lack of exploratory studies focusing on the interpretability of training and testing complex time series ECG data, which risks failed understanding of this medically critical area.

Finally, transfer learning techniques enable efficient use and reuse of pre-trained models and accelerate model development, which makes them powerful and applicable ML and DL workflows.

4. PTB ECG Diagnostic Database

Delving deeper into the realm of data resources for MI diagnosis models, we now explore the Physikalisch-Technische Bundesanstalt (PTB) Diagnostic ECG Database [31] [32] [33]. This comprehensive dataset, meticulously curated by PhysioNet, provides a cornerstone for researchers developing and evaluating automated MI detection algorithms.

A collection of electrocardiogram (ECG) data from 290 subjects is included in the database. The age range of the subjects spans from 17 to 87 years, with a mean of 57.2 years. Data for each subject is divided into 1 to 5 separate recordings. While the majority of subjects are male (209), with a mean age of 55.5 years, there are also 81 females (mean age 61.6 years). It's important to note that age data is missing for one female and 14 male subjects.

Records are provided as 15-lead ECGs, consisting of 12 conventional leads (i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6) along with the 3 Frank lead

ECGs (v_x , v_y , v_z). Table 1 provides a basic description of the ECG leads sampled, with each signal recorded at sampling rates of up to 10 kHz.

Table 1: *Description of ECG Leads*

Lead	Type	Description	Usage
i	Standard Limb Lead	Potential difference between the right arm (RA) electrode (+) and the left arm (LA) electrode (-)	Lateral MI
ii	Standard Limb Lead	Potential difference between the right arm (RA) electrode (+) and the left leg (LL) electrode (-)	Inferior MI
iii	Standard Limb Lead	Potential difference between the left arm (LA) electrode (+) and the left leg (LL) electrode (-)	Inferior MI
avr	Augmented Voltage Right	Activity with the right arm (RA) as reference and views the heart from the perspective of the left arm and left leg	Inferior MI
avl	Augmented Voltage Left	Activity with the left arm (LA) as reference and views the heart from the perspective of the right arm and left leg	Lateral MI
avf	Augmented Voltage Foot	Activity with the left leg (LL) as reference and views the heart from the perspective of the right arm and left arm	Inferior MI
$v_1, v_2, v_3,$ v_4, v_5, v_6	Chest Lead	Transverse plane view of the heart's electrical activity, labeled V1 (right sternal border) and V6 (left midaxillary line)	Anterior MI Posterior MI
v_x, v_y, v_z	Frank VCG Lead	Three orthogonal components of the vectorcardiogram representing the x (transverse), y (frontal), and z (sagittal) axes	Location of myocardial damage

Most ECG recordings within the database are accompanied by a corresponding header file (.hea) containing detailed clinical summaries for each subject. These summaries typically include demographic information (age, gender), diagnosis, and, when applicable, data on medical history, medications, interventions, coronary artery pathology, ventriculography, echocardiography, and laboratory test results.

graphy, and hemodynamics. It's important to note that clinical summaries are absent for 22 subjects. The diagnostic classifications for the remaining 268 subjects are presented in Table 2. The most focused class throughout the proposed work is the Myocardial Infarction class.

Table 2: *PTB Diagnostic Dataset classes*

Diagnostic Class	Number of subjects
Myocardial Infarction	148
Cardiomyopathy/Heart failure	18
Bundle branch block	15
Dysrhythmia	14
Myocardial hypertrophy	7
Valvular heart disease	6
Myocarditis	4
Miscellaneous	4
Healthy controls	52

5. Proposed Methodology

This study presents an automated electrocardiogram (ECG) analysis methodology to improve diagnostic accuracy non-invasively. The core of the proposed method lies in a two-pronged approach. First, it emphasizes the critical role of feature selection. By meticulously extracting the most informative and reliable morphological features from the raw ECG signal, the method aims to capture the most relevant aspects of the electrical activity within the heart. These features likely hold the key to accurately differentiating between healthy and abnormal cardiac function. Secondly, the methodology optimizes existing deep learning components within the employed neural network architecture. This optimization process seeks to enhance the network's ability to learn complex relationships between the extracted features and the corresponding patient health status. Ultimately, the combined effect of these strategies is to achieve a more robust and reliable classification of patients as either healthy or unhealthy. A detailed illustration of the overall methodology is discussed in the following subsections and presented in Figure 5, allowing for a deeper understanding of the proposed approach.

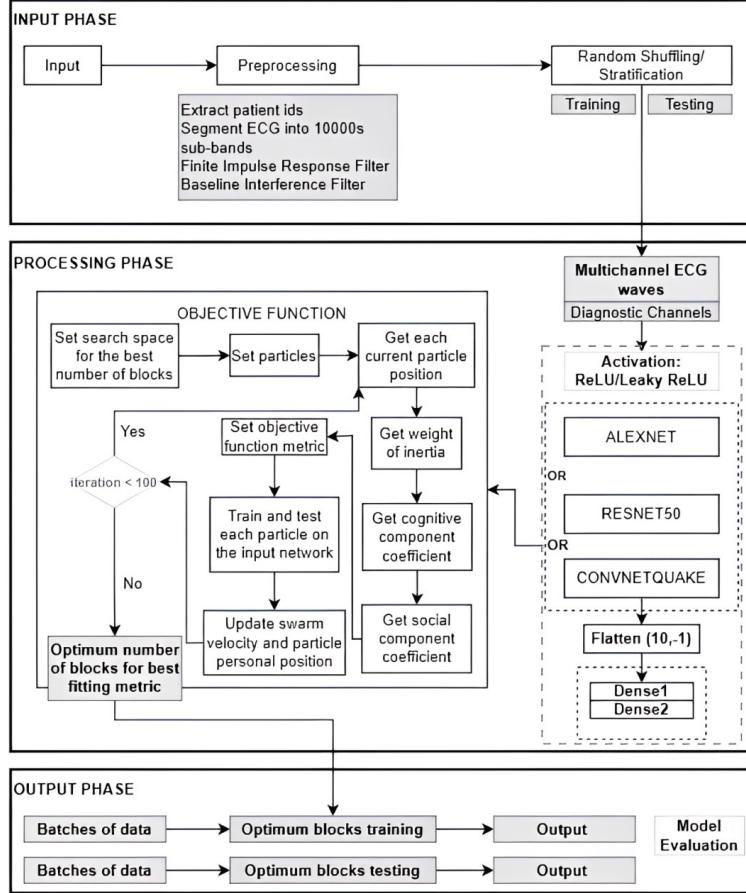


Figure 5: Overview of an optimized ischemia diagnosis model

5.1. Input Phase

The main aim of this phase is to prepare the input that allows for the optimal extraction of the best features. This enables the cleanest and best quality multi-channel ECG data. This methodology starts with data files in the form of signal and header files. This architecture requires Python waveform generation libraries to run ECG records and annotations. During the pre-processing stage, data is properly presented in the form of unique patient IDs. Then, each extracted array of ECG signal data is randomly sampled as batches with a window size of 10,000 seconds.

By then, the data will be practically ready for stratification into training and validation sets, but two final checks must be provided. The first is a

baseline interference filter. The second is a power-line interference filter. Both types of filtration utilize an FIR (Finite Impulse Response) filter. As previously explained, a baseline filter removes any potential wander from the signal, and the power-line interference filter removes potential electric noise. The FIR filter acts as a low-pass filter specifically designed to target and retain frequency components within the typical range of ECG signals. The filter utilizes a cut-off frequency of 50Hz, ensuring the elimination of higher-frequency noise that may interfere with subsequent analysis. Typically, the most beneficial ECG frequencies for diagnosing ischemia are 50Hz and below. Figure 6 shows the basic view of the FIR filter.

After pre-processing, the clean, extracted signals are stratified and shuffled randomly into the standard 80% training and the 20% validation sets. Following the pre-processing steps, both ECG recordings categorized as healthy and unhealthy are grouped together. This grouping process results in batches comprised of multichannel signals. These prepared batches are then directed towards the subsequent processing phase, as illustrated in Figure 6.

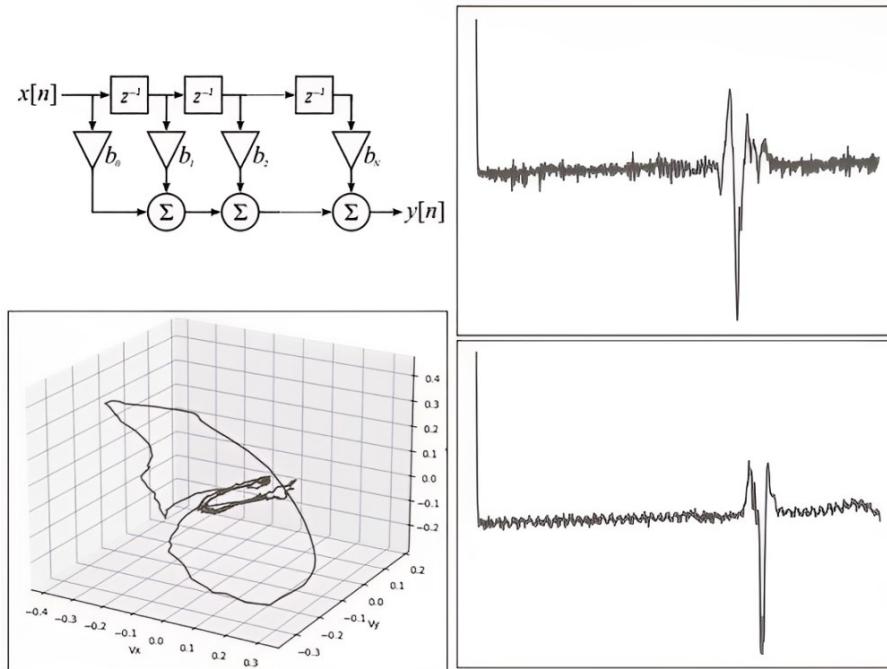


Figure 6: Myocardial Infarction patient against a Healthy patient

5.2. Processing Phase

The processing phase is divided into two main processes. The first process is passing the extracted multi-channel signals through standard CNNs, which are AlexNet, ResNet50, or ConvNetQuake [19] network with an element-wise ReLU activation, as shown in Figure 5.

After that, the output is reshaped into a flat array, then passed through two linear layers and finally a decision layer. The forward pass through this architecture is to output the best-trained number of pairs of dense blocks according to a set metric. Then, this number of pairs is used to train and test one of the standard architectures used.

The second process utilizes one of the most efficient stochastic population-based techniques: particle swarm optimization (PSO)[34]. presents a compelling optimization technique distinguished by its simplicity, efficiency, and ability to handle problems without requiring gradient information [35]. This versatility makes PSO applicable to a wide range of optimization scenarios [36]. The core principle of PSO involves a swarm of particles representing candidate solutions that are randomly distributed within the search space. Each particle's fitness is evaluated by the objective function, akin to finding food at its location. Notably, each particle possesses knowledge of its current fitness, its personal best solution encountered so far (local optimum), the best solution discovered by the entire swarm (global optimum), and its own velocity. This information guides the particles' movement through the search space, allowing them to explore promising regions iteratively and potentially converge toward the optimal solution.

The successful implementation of PSO hinges on a select set of parameters that significantly influence the algorithm's behavior and effectiveness in optimizing a particular problem. These parameters encompass:

- Particle swarm size (num_p): This defines the number of candidate solutions explored simultaneously.
- Problem dimensionality (d): The number of variables involved in the optimization problem.
- Particle velocity (V): The rate at which particles move through the search space.
- Inertia weight (ω): This factor regulates the influence of a particle's past trajectory, balancing the exploration of new regions with the exploitation of promising areas identified earlier.

- Particle velocity limits ($\min V$, $\max V$): These establish constraints on the maximum velocity of particles, preventing them from escaping the search space.
- Cognitive learning rate (c_1): This parameter determines the weight given to a particle's personal best solution, influencing its tendency to revisit promising areas it has encountered.
- Social learning rate (c_2): This parameter determines the weight given to the swarm's globally best solution, guiding particles towards areas identified as potentially optimal by the entire swarm.
- Random factors (r_1, r_2): These stochastic elements introduce a degree of variation in particle movement, promoting diversification and mitigating the risk of premature convergence on sub-optimal solutions.

The goal is to locate the single configuration that best satisfies the criteria established by the chosen problem representation (i.e., the global best position of a particle).

Within the PSO, each particle is characterized by vectors of d -dimensional, which represent its position and velocity, respectively, according to Equations 1 and 2:

$$X_i = (x_{i1}, \dots, x_{id}) \quad (1)$$

$$V_i = (v_{i1}, \dots, v_{id}) \quad (2)$$

where X_i and V_i are the position and velocity vector of a particle i in the search space, *space*.

The PSO algorithm starts by randomly initializing particles' positions and velocities. Subsequently, each particle's fitness is evaluated according to the objective function. Identify the particle with the best fitness value among all particles in the swarm and assign it the current global best solution ($Gbest$) while the local best one is assigned to the personal best ($Pbest$). This work utilizes an objective function centered on identifying the optimal number of blocks within the employed Convolutional Neural Networks (CNNs). The aim is to achieve the best performance as measured by non-optimized metrics, specifically ConvNetQuake and ResNet50.

As particles navigate the search space in their quest for the optimal solution, their velocity and position undergo continuous updates according to Equations 3 and 4, respectively. This iterative process guides their exploration and convergence towards the most promising regions.

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (Pbest_i^k - X_i^k) + c_2 r_2 (Gbest_i^k - X_i^k) \quad (3)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (4)$$

where K is the current iteration number for a particle i in a set of epochs. c_1 and c_2 are cognitive and social learning rates that affect the intensity of the random forces acting in the direction of $Pbest$ (previous best) and $Gbest$ (global previous best). r_1 and r_2 are random values ranging from 0 to 1 and ω denotes the inertia weight.

During the update process, it's customary to impose limitations on both the positions and velocities of particles within the swarm [37][38]. These bounds serve a crucial purpose, as illustrated in Algorithms 1 and 2, corresponding to the particle position and velocity clamping, respectively. By enforcing these constraints, we ensure that particles remain within the defined search space, preventing them from venturing into irrelevant areas and hindering the optimization process.

Algorithm 1 Particle position clamping

```

1: if  $x_i < minX$  then
2:    $x_i \leftarrow minX$ 
3: else if  $x_i > maxX$  then
4:    $x_i \leftarrow maxX$ 
5: end if

```

Algorithm 2 Particle velocity clamping

```

1: if  $v_i < minV$  then
2:    $v_i \leftarrow minV$ 
3: else if  $v_i > maxV$  then
4:    $v_i \leftarrow maxV$ 
5: end if

```

Specifically, the objective function plays a central role in PSO, assigning a score to each particle based on how well its corresponding model fits the

data. This score serves as a measure of the particle's "goodness". A balance between exploration and exploitation guides particle movement. The cognitive coefficient (c_1) represents the influence of a particle's personal best solution (P_{best}), while the social coefficient (c_2) reflects the influence of the swarm's globally best solution (G_{best}) discovered so far. These coefficients are further modulated by random factors (r_1 and r_2) to introduce an element of stochasticity. As illustrated in Figure 7, each particle's configuration, represented by the number of blocks in a CNN architecture, is trained on the data. The resulting performance metric, such as ROC AUC (Area Under the Curve) or accuracy, is then used to update both the particle's personal best and potentially the global best solution (G_{best}) if a better performing configuration is found. This iterative process of exploration and exploitation continues until the optimal number of blocks is identified, effectively leading the swarm toward the configuration that yields the best performance value.

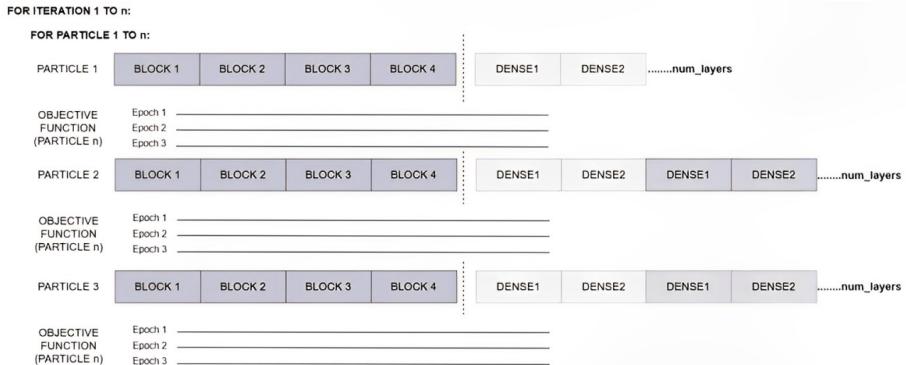


Figure 7: An example of how particles are used to optimize the number of dense layers a deep learning model has, where the optimization process ends with all particles and all iterations. Each particle is the constant CNN model with dimensions equal to model parameters.

In PSO, the inertia weight (ω) plays a crucial role in balancing exploration and exploitation during the search process. To facilitate a comprehensive update of the entire swarm, a common strategy involves linearly decreasing the weight of inertia over time. This approach allows particles to maintain some of their current velocity and direction in the early stages, promoting exploration of a wider search space. As the optimization progresses and the swarm converges towards promising regions, the decreasing inertia weight gradually reduces the influence of past movements, encouraging particles to

focus on exploiting these areas for potentially better solutions. According to existing empirical studies, as the number of iterations increases, the typical decrease range is 0.9 - 0.4 [39]. Therefore, the inertia weight can be viewed as the fluidity of the medium in which a particle travels [40]. A higher value (i.e., 0.9) indicates extensive exploration in a low-viscosity medium, so gradually decreasing the inertia to a lower value (i.e., 0.4) indicates more exploitation in a higher-viscosity medium.

In case of more exploitation, PSO naturally lacks global search ability at the end of each iteration. The swarm-based optimization technique can get stuck at a local saddle point or minima without a good global search ability. Consequently, adaptive strategies such as adjusting the weight of inertia (coefficient to the velocity) are suggested to provide results closer to the optima such as the Linear Decreasing Inertia Weight (LDIW) strategy according to Equation 5 :

$$\omega_k = (\omega_{start} - \omega_{end})\left(\frac{k_{max} - k}{k_{max}}\right) + \omega_{end} \quad (5)$$

where ω_{start} and ω_{end} are the initial and final values of inertia weight, k is the current iteration number, k_{max} is the maximum iteration number, and $\omega_k \in [0, 1]$ is the inertia weight value in the k -th iteration.

The PSO algorithm terminates after reaching a pre-defined number of iterations (k_{max}). In this instance, the stopping criterion is set to a specific number of iterations performed by a swarm of 100 particles. The successful completion of this stage yields the optimal number of dense block pairs within the CNN architecture, which is expected to achieve the best possible metric value or performance characteristic. Algorithm 3 provides a detailed illustration of the core steps involved in this optimized deep learning model.

5.3. Output Phase

Building upon the previous phase, where an optimal number of blocks was identified for the deep learning model, this final stage focuses on rigorously evaluating its performance. The chosen number of blocks is tested on batches of ECG signal data. These tests can be conducted on individual ECG channels or the complete one-dimensional ECG signal. Various metrics are employed to comprehensively assess the model's effectiveness, including accuracy, confusion matrix, and receiver operating characteristic (ROC) curve.

Algorithm 3 Optimized DL Model using PSO Algorithm

Require: Search space (space), Maximum number of iterations (k_{\max}), number of particles (num_p), number of pairs of dense blocks (num_layers), personal best score (score), personal best position (P_{best})

Ensure: G_{best} position (found at the last iteration and where the ROC AUC is calculated to attain the best number of blocks)

- 1: Initialize particle position and velocity randomly.
 - 2: Initialize global best position, best position (Initialize to zero for start)
 - 3: Initialize global best score, best score (Initialize to a large value, e.g., ∞)
 - 4: Initialize inertia weight, ω
 - 5: Set the cognitive coefficient, c_1 and the social coefficient, c_2
 - 6: **for** $k \leftarrow 1$ to k_{\max} **do**
 - 7: **for** $i \leftarrow 1$ to num_p **do**
 - 8: Set objective function $f(x_i)$ to particle position
 - 9: Clamp the position to get P_{best} of x_i
 - 10: Clamp the velocity to get the velocity v_i
 - 11: **for** j in range of num_layers **do**
 - 12: Fit the model to the current particle x_i
 - 13: Calculate the ROC AUC
 - 14: Calculate the score
 - 15: **if** score < best_score(x_i) **then**
 - 16: $G_{\text{best}}(x_i) \leftarrow x_i$
 - 17: best_score(x_i) \leftarrow score
 - 18: **end if**
 - 19: **end for**
 - 20: **end for**
 - 21: Update the particle velocity and position
 - 22: Generate random factors r_1 and r_2
 - 23: Update particle velocity using Equation (3)
 - 24: Update particle position using Equation (4)
 - 25: **end for**
 - 26: **return** the best particle position so far G_{best}
-

6. Experimental Results and Performance Analysis

This section presents a comparative analysis of various benchmark models and architectures trained on both single-channel and multi-channel electrocardiogram (ECG) data. All experiments were conducted using a 10-core CPU and a single GPU. While training outcomes were comparable for single-channel and multi-channel models, subsequent testing revealed a clear superiority of multi-channel models across various evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics are derived from the calculation of true positives (i.e., Correctly predicted positive cases), true negatives (i.e., Correctly predicted negative cases.), false positives (i.e., Incorrectly predicted as positive "Type I error"), and false negatives (i.e., Incorrectly predicted as negative "Type II error").

- Accuracy: Measures the proportion of correct predictions out of all predictions (*TotalPrediction*). It can be computed according to Equation 6.

$$Accuracy = \frac{TruePositive + TrueNegative}{TotalPrediction} \quad (6)$$

- Precision: Measures how many of the positive predictions were actually correct. It can be computed according to Equation 7.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (7)$$

- Sensitivity (Recall): Measures how many of the actual positive cases were correctly identified. It can be computed according to Equation 8.

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative} \quad (8)$$

- F1-Score: Is the weighted average of Precision and Recall. It takes both false positives and false negatives into account. It can be computed according to Equation 9.

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (9)$$

Furthermore, the confusion matrix can provide a class-by-class breakdown of the classifier's effectiveness. This breakdown includes the counts of true

Table 3: *Confusion matrix view*

	Positive	Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative

positives, false positives, true negatives, and erroneously listed false negatives as represented in Table 3.

A set of experiments is conducted to achieve promising results regarding the average accuracy of comparative architectures. Based on further experiments, including channel-base and transform-based architectures, and finally, the optimized deep learning model using particle swarm optimization, the best one is selected. These experiments are illustrated below.

6.1. Experiment on Average Accuracy

Initial experiments utilized multi-channel input data, specifically employing three primary channels. These channels were preprocessed into image format to serve as input to a preliminary non-optimized implementation of the proposed architecture. Table 4 provides a summary of the mean accuracy achieved across the various channels evaluated.

Table 4: *Average accuracy of non-optimized model architectures*

Model	AlexNet	ResNet50	ResNet101	ConvNetQuake	ConvNetQuake
Leads	ii, v6, vz	ii, v6, vz	v1, v2, v3, v4, v5, v6	ii, v6, vz	v2, v6, vz
Notes	Standard ImageNet	Feature matrices are in batches	Hermite transform of averaged leads	Physician-level deep learning	Test with ante- rior lead, v2
Avg. Accu- racy (%)	63.8	98.67	90	88	85

Among the non-optimized three-channel architectures evaluated, ResNet50 demonstrated superior performance, achieving an average accuracy of 98.67%.

ConvNetQuake followed as a close second with an average accuracy of 88%, while AlexNet exhibited the lowest performance. To conduct a more in-depth analysis, ResNet50 and ConvNetQuake, given their promising initial results, were selected for further experimentation and comprehensive metric evaluation.

6.2. Further Experiments on the Selected Non-Optimized Architectures

A ResNet50 model was retrained for single-channel ECG classification using a Hermite transform (orders 3, 4, 5) applied to data preprocessed with a low-pass filter (as depicted in Figure 6). This model achieved a 90% accuracy rate during training, as indicated in Table 4. However, subsequent testing on various ECG channels revealed unsatisfactory performance.

The first experiment performed on ConvNetQuake involved Hermite-transformed 3-channel input, where the 3-channel waves were filtered, averaged, and transformed to be input into the network. The second experiment on ConvNetQuake used stochastic random batches of untransformed, normalized waves from the 3-channels to be input into the network. While both experiments were performed over long periods of time, the second experiment yielded a better confusion matrix, as shown in figure 8. The confusion matrix shows that the experiment best works in cases of unhealthy patients, with a precision of 80% and an f1-score of 31%.

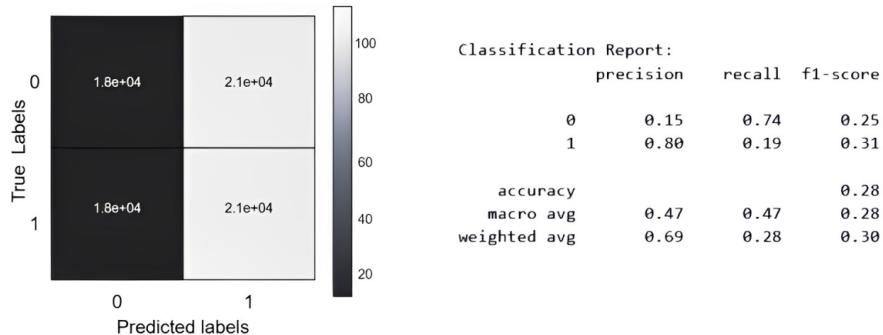


Figure 8: Channel-based ConvNetQuake confusion matrix

Similarly, on ResNet50, the channel-based experiment yielded a better confusion matrix. The confusion matrices of both experiments are shown in figure 9. From the confusion matrices, it can be seen that the channel-based ResNet50 performed better due to a higher predicted diagnosis in unhealthy patients. However, the main disadvantage is the high value of false negatives

(a person is unhealthy but is diagnosed as healthy). For this reason, even if ResNet50 performed better in terms of unhealthy patient accuracy and precision, shown in table 6, an optimization is required.

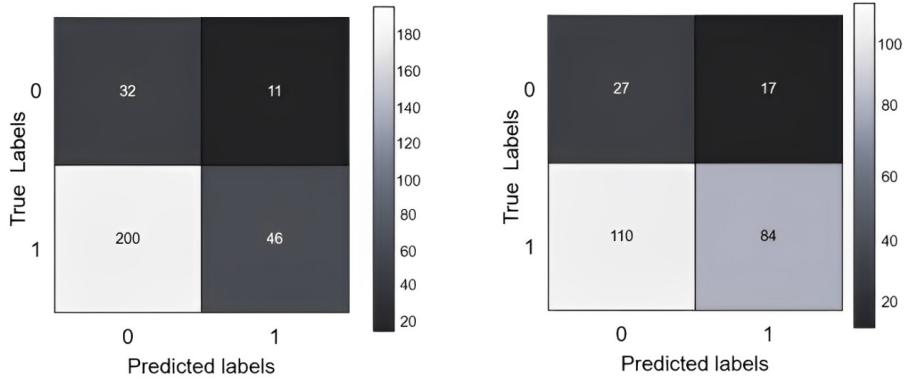


Figure 9: Transform-based ResNet50 and Channel-based ResNet50

Certain diagnosis cases were undetected by all models, such as patients with the possible existence of Anterior, Posterior, or both . In figure 10, two chest leads and a Frank lead through the ResNet-50 showed that the chest leads reported almost equal results, which in context of deep learning shows a good and consistent model, but medically shows a probable confusion between anterior and posterior STEMI cases. This is predicted to be due to the interchange in S-T segment appearance, where an Anterior case shows an elevation and a Posterior case shows a depression. If a patient was diagnosed with an Anterior case, all models failed to show that there are opposing patterns in chest ECG channels.

6.3. Experiments on Optimized Architecture using PSO

An optimization process was undertaken to determine the optimal number of blocks for each architecture, to maximize the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve according to the formula in Equation 10.

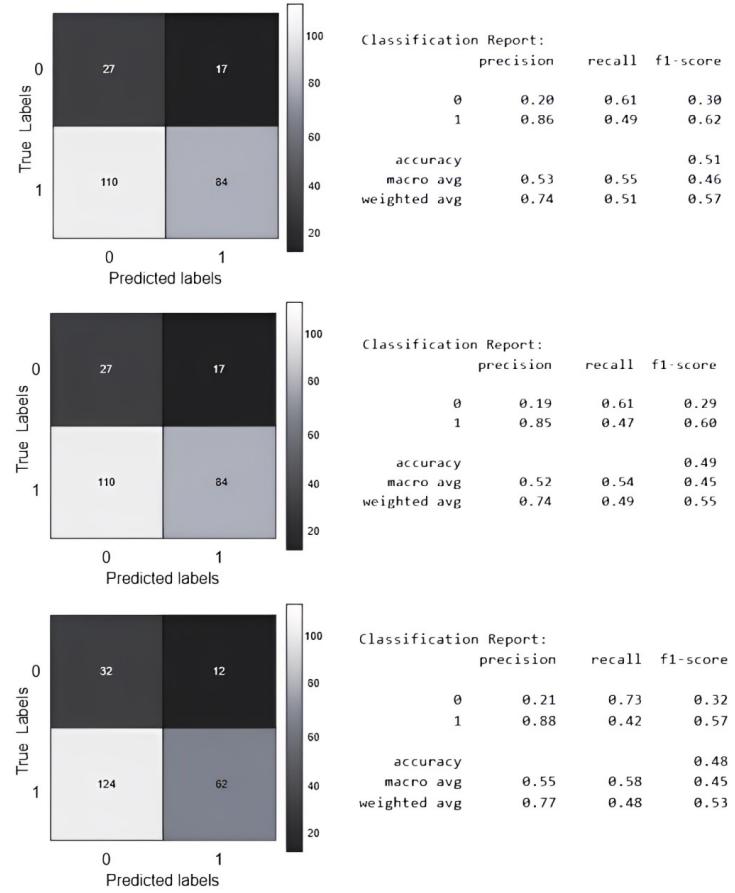


Figure 10: Classification reports of a ResNet-50 experiment with three leads: v2, v6, vz

$$\text{ROC AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(t)) dt \quad (10)$$

where TPR denotes the True Positive Rate (Sensitivity), FPR^{-1} represents the inverse of the False Positive Rate (1 - Specificity), t ranges from 0 to 1, representing the threshold used for classifying positive and negative instances.

To optimize the best-performing models (ConvNetQuake and ResNet50), the parameter settings are shown in table 5. ConvNetQuake required a lower learning rate to yield good results with a high number of iterations, since no appropriate results were yielded otherwise. Despite the lack of importance

Table 5: *Parameter settings for the two best performing models*

Parameter	ResNet50	ResNet50-PSO	ConvNetQuake	ConvNetQuake-PSO
Learning Rate	0.001	0.001	0.0001	0.0001
Batch Size	16	16	16	16
Loss	Sparse Categorical Cross Entropy	Sparse Categorical Cross Entropy	Binary Cross Entropy	Binary Cross Entropy
Optimizer	Adam	Adam	Adam	Adam
No of iterations	20	20	5,000	5,000
No of particles	-	3	-	3

Table 6: *Average metrics between ResNet50 and ConvNetQuake*

Model Architecture	Avg Precision	Avg Recall	Avg F1 Score	Avg ROC
ConvNetQuake-PSO	0.840	0.190	0.310	0.530
ConvNetQuake	0.800	0.190	0.310	0.500
Transform-based ResNet50	0.800	0.190	0.310	0.530
Channel-based ResNet50	0.863	0.460	0.597	0.540
Channel-based ResNet50-PSO	0.880	0.460	0.600	0.600

given to the calculation of loss in this paper, ConvNetQuake used Binary Cross Entropy since it only performed well when given only 2 classes, unlike ResNet50 which is much more versatile. In this paper, all experiments only feature binary class diagnosis.

Table 6 shows the main model evaluation metrics. According to the results shown in table 6 and in table 7, ConvNetQuake is quite a stable model, especially with unhealthy patients. The result also shows the number of layers, or how dense, should each good-performing 3-channel model be to attain a good operational characteristic, as shown in table 7 and figure 11. In table 7, it is shown that the maximum number of pairs of dense blocks required by ResNet50 to achieve a good ROC, is 8 pairs of blocks, where ConvNetQuake showed stability with a maximum of 1 pair of dense blocks. This proves a major disadvantage in terms of processing time, as the former takes more time and requires more resources than the latter. In figure 11, upon optimization, it is seen that there is an expected increase in the probability of true positives even if the ROC value itself is lower than the non-optimized experiment.

Transform-based ResNet50, despite having agreeable results, shows a bet-

Table 7: Number of blocks required for good characteristic curve in optimized models

Model	ResNet50			ConvNetQuake		
Leads	ii	v6	vz	ii	v6	vz
Number of blocks required after PSO	8	1	2	1	1	1

ter diagnosis in terms of false negative results only (predicted to be healthy but is actually unhealthy). As previously stated, channel-based ResNet50 showed the best confusion matrices regarding both true positive and false negative, as shown in figure 9.

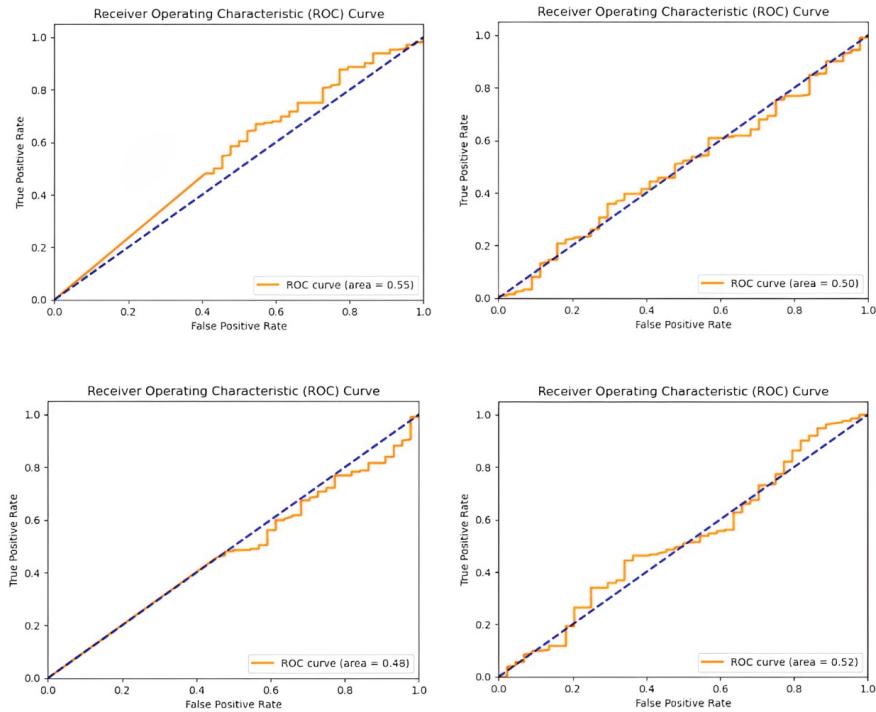


Figure 11: Left to right: Non-optimized channel-based ResNet50 ROC versus channel-based ResNet50 with PSO ROC of final particle in the final iteration; Top to bottom: Channels ii and v6 for ResNet50

In conclusion, PSO with an objective function that takes a number of layers yielded a better ROC curve, using half the epochs for 1 iteration, despite the possibility of having a lower ROC value than the non-optimized

version of a model. Given the stated results, ResNet50 with PSO is the most ideal model in terms of not only accuracy but also ROC (even if ConvNetQuake with PSO yielded better split-wise results). The main trade-offs for using ResNet50 with PSO are processing power and practical cost, which can be solved using better GPUs (Graphical Processing Unit) and more feature-reduction techniques. On the other hand, the practical cost of applying channel-based ResNet50 is much higher due to the high proportion of false negative patients(unhealthy patients diagnosed as healthy). Thus ConvNetQuake is evaluated as a better real-life model since the proportion of true positives and low-risk false positives is considerably higher.

7. Limitations of the Proposed Methodology

The main limitations encountered in this work can be summarized as limitations with throughput and design. Those limitations directly affect the applicability of the chosen architecture on software components. Throughput limitations concern the model running and output. In this work, all models are run on a multi-core CPU and a single-core GPU, leading to slow operation, quick memory clutter, and resource exhaustion per run. Parallelization of some models seemed to resolve the issue, but it led to problems with testing and software compatibility. Model design limitations deal with how complex the code or model is expected to be. This limitation affects software compatibility, model testability, and applicability. Since feature engineering is the least focused area of this work, each model design seeks to test the robustness of conventional classifiers without heavy reliance on feature engineering and precise segmentation. In this study, the most ideal model reports the best metrics. However, some trade-offs are expected when considering the ideal software model.

8. Future Research Directions

Upon reviewing the previous literature and related works, benchmark models for medical diagnosis trained and parallelized on regular GPU or even Nvidia processors are said to be reaching their end. Newer implementations of those models, such as the Internet of Things (IoT) and Quantum implementations, are becoming more popular as new research areas. Medical diagnosis without heavy reliance on feature engineering provides comparable

results to categorization models based on feature engineering and segmentation. In the future, the stages of diagnosis can be organized as phases, where each phase requires a setup and its real-time kernel (unlike the virtual kernel used in this study): segmentation, ECG segment reproduction, training on segments, appending the segments together, retraining, and optimization. Dividing the diagnosis process into phases and retraining will give physicians a more precise and spaced perspective of how exactly a patient can have a certain type of cardiovascular disease. This setup will allow researchers to bridge further the chasm between research and applying highly precise and possibly cumbersome diagnosis models.

9. Conclusion

By the end of this work, it is concluded that optimized benchmark models should be dense enough to allow for the correct diagnosis of channel-split ECGs. This paper explored many model architectures and implementations, mainly under the assumption that feature engineering and categorization principles, such as discrete wavelet or Fourier transform, are not applied. The main outlined technique in this paper starts with random batches of window-sized ECGs of selected channels that are extracted, trained, tested, and optimized. Finally, all models should be ready to be applied as separate software or web app components.

References

- [1] D. Cook, S. Oh, M. Pusic, Accuracy of physicians' electrocardiogram interpretations: A systematic review and meta-analysis, *JAMA internal medicine* 180 (11) (2020) 1461–1471, publisher Copyright: © 2020 American Medical Association. All rights reserved. doi:10.1001/jamainternmed.2020.3989.
- [2] N. M. Estes, Computerized interpretation of ecgs, *Circulation: Arrhythmia and Electrophysiology* 6 (1) (2013) 2–4, doi: url10.1161/CIRCEP.111.000097.
- [3] L. Costanzo, Physiology, Board review series, Wolters Kluwer Health, 2014.
URL <https://books.google.com.eg/books?id=A1g5zQEACAAJ>

- [4] H. M. R. K. Imran A Khan, C. K. Panda, Atypical presentations of myocardial infarction: A systematic review of case reports, *Cureus* 15 (2023). doi:10.7759/cureus.35492.
- [5] R. Jothiramalingam, A. Jude, J. D, Review of computational techniques for the analysis of abnormal patterns of ecg signal provoked by cardiac disease, *Computer Modeling in Engineering & Sciences* 128 (2021) 1–25. doi:10.32604/cmes.2021.016485.
- [6] R. Kher, Signal processing techniques for removing noise from ecg signals, *Journal of Biomedical Engineering and Research* 3 (1) (2019).
- [7] M. Hassaballah, Y. M. Wazery, I. E. Ibrahim, Ecg heartbeat classification using machine learning and metaheuristic optimization for smart healthcare systems, *Bioengineering* 10 (429) (2023). doi:10.3390/bioengineering10040429.
- [8] B. Kolukisa, H. Hacilar, G. Goy, Evaluation of classification algorithms, linear discriminant analysis and a new hybrid feature selection methodology for the diagnosis of coronary artery disease, *IEEE International Conference on Big Data* 1 (8622609) (2018). doi:10.1109/BigData.2018.8622609.
- [9] A. S. M. Faizal, W. Y. Hon, T. M. Thevarajah, A biomarker discovery of acute myocardial infarction using feature selection and machine learning [medical & biological engineering & computing (2023) 61:2527–2541], *Medical & Biological Engineering & Computing* (2023) 61 (8622609) (2023). doi:10.1007/s11517-023-02841-y.
- [10] M. Kumar, R. B. Pachori, U. R. Acharya, Automated diagnosis of myocardial infarction ecg signals using sample entropy in flexible analytic wavelet transform framework[evaluation of systems' irregularity and complexity: Sample entropy, its derivatives, and their applications across scales and disciplines, edited by dr. anne humeau-heurtier], *Journal of Entropy* 19 (488) (2017). doi:10.3390/e19090488.
- [11] V. Mondejar-Guerra, J. Novo, J. Rouco, Heartbeat classification fusing temporal and morphological information of ecgs via ensemble of classifiers, *Biomedical Signal Processing and Control* 10 (1016) (2018). doi:10.1016/j.bspc.2018.08.007.

- [12] A. M. Nastaran Jafari Hafshejani, R. Hajian, Identification of myocardial infarction using morphological features of electrocardiogram and vectorcardiogram, *IET Signal Processing* 1 (2021). doi:10.1049/sil2.12072.
- [13] C. L. Tsien, H. S. F. Fraser, W. J. Long, R. L. Kennedy, Using classification tree and logistic regression methods to diagnose myocardial infarction, *Studies in health technology and informatics* 52 Pt 1 (1998) 493–7.
URL <https://api.semanticscholar.org/CorpusID:7104966>
- [14] M. Fradi, L. Khriji, M. Machhout, Automatic heart disease class detection using convolutional neural network architecture-based various optimizers-networks, *IET Smart Cities* 3 (12003) (2021). doi:10.1049/smcc2.12003.
- [15] M. Kachuee, S. Fazeli, M. Sarrafzadeh, Ecg heartbeat classification: A deep transferable representation [version 2; under subject computers and society, last revised 12 jul 2018], arXiv:1805.00794v2 [cs.CY] 1 (1805.00794) (2018). doi:10.1109/ICHI.2018.00092.
- [16] S. Smigiel, K. Pałczynski, D. Ledzinski, Ecg signal classification using deep learning techniques based on the ptb-xl dataset, *Journal of Entropy* 23 (1121) (2021). doi:10.3390/e23091121.
- [17] M. Hammad, M. H. Alkinani, B. B. Gupta, Myocardial infarction detection based on deep neural network on imbalanced data[version 1; accepted: 1 december 2020, published online: 6 january 2021], Springer Nature: Multimedia Systems (2022) 28 (10.1007) (2021). doi:10.1007/s00530-020-00728-8.
- [18] D. A. AlDuwaile, M. S. Islam, Using convolutional neural network and a single heartbeat for ecg biometric recognition, *Journal of Entropy* 23 (733) (2021). doi:10.3390/e23060733.
- [19] A. Gupta, E. Huerta, Z. Zhao, I. Moussa, Deep learning for cardiologist-level myocardial infarction detection in electrocardiograms, in: T. Jarm, A. Cvetkoska, S. Mahnič-Kalamiza, D. Miklavcic (Eds.), 8th European Medical and Biological Engineering Conference, Springer International Publishing, Cham, 2021, pp. 341–355.

- [20] E. H. Houssein, M. Hassaballah, I. E. Ibrahim, D. S. AbdElminaam, Y. M. Wazery, An automatic arrhythmia classification model based on improved marine predators algorithm and convolutions neural networks, *Expert Systems with Applications* 187 (2022) 115936. doi:<https://doi.org/10.1016/j.eswa.2021.115936>.
- [21] W.-C. Yeh, Z. Liu, S.-Y. Tan, S.-K. Huang, Implementation of parallel simplified swarm optimization in cuda, ArXiv abs/2110.01470 (2021). URL <https://api.semanticscholar.org/CorpusID:238259284>
- [22] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Communications of the ACM* 60 (2012) 84 – 90. URL <https://api.semanticscholar.org/CorpusID:195908774>
- [23] A. S. B. Mahel, N. Harold, H. Solieman, Arrhythmia classification using alexnet model based on orthogonal leads and different time segments, 2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) (2022) 1312–1315. URL <https://api.semanticscholar.org/CorpusID:248270067>
- [24] S. T. Aarthy, J. L. M. Iqbal, Modified parametric-based alexnet structure to classify ecg signals for cardiovascular diseases, *Measurement: Sensors* (2023). URL <https://api.semanticscholar.org/CorpusID:259071514>
- [25] M. S. Kolhar, R. N. Kazi, H. Mohapatra, A. M. A. Rajeh, Ai-driven real-time classification of ecg signals for cardiac monitoring using i-alexnet architecture, *Diagnostics* 14 (2024). URL <https://api.semanticscholar.org/CorpusID:270746671>
- [26] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 770–778. URL <https://api.semanticscholar.org/CorpusID:206594692>
- [27] C. Han, L. Shi, Ml-resnet: A novel network to detect and locate myocardial infarction using 12 leads ecg, *Computer methods and programs in biomedicine* 185 (2020) 105138. URL <https://api.semanticscholar.org/CorpusID:204967424>

- [28] J. A. Wahid, M. Xu, M. Ayoub, S. Hussain, L. Li, L. Shi, A hybrid resnet-vit approach to bridge the global and local features for myocardial infarction detection, *Scientific Reports* 14 (2024).
URL <https://api.semanticscholar.org/CorpusID:267809213>
- [29] T. Perol, M. Gharbi, M. Denolle, Convolutional neural network for earthquake detection and location, *Science Advances* 4 (2) (2018) e1700578. arXiv:<https://www.science.org/doi/pdf/10.1126/sciadv.1700578>. doi:10.1126/sciadv.1700578.
URL <https://www.science.org/doi/abs/10.1126/sciadv.1700578>
- [30] P. Xiong, S. Lee, G. Chan, Deep learning for detecting and locating myocardial infarction by electrocardiogram: A literature review, *Frontiers in Cardiovascular Medicine* 9 (2022).
URL <https://api.semanticscholar.org/CorpusID:247783067>
- [31] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals, *Circulation* 101 (23) (2000) e215–e220.
URL <http://circ.ahajournals.org/content/101/23/e215.full>
- [32] R. Bousseljot, D. Kreiseler, A. Schnabel, Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet, *Biomedizinische Technik / Biomedical Engineering* (2009) 317–318 doi:10.1515/bmte.1995.40.s1.317.
- [33] Bousseljot, PtB diagnostic ecg database, PhysioNet, retrieved October 23, 2024 (September 2004).
URL <https://physionet.org/content/ptbdb/1.0.0/>
- [34] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the sixth international symposium on micro machine and human science, Ieee, 1995, pp. 39–43.
- [35] M. Arasomwan, A. Adewumi, An adaptive velocity particle swarm optimization for high-dimensional function optimization, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2352–2359. doi:10.1109/CEC.2013.6557850.

- [36] W. H. El-Ashmawi, A. F. Ali, M. A. Tawhid, An improved particle swarm optimization with a new swap operator for team formation problem, *Journal of Industrial Engineering International* 15 (1) (2019) 53–71. doi:[10.1007/s40092-018-0282-6](https://doi.org/10.1007/s40092-018-0282-6).
URL <https://hdl.handle.net/10419/267616>
- [37] M. A. Arasomwan, A. O. Adewumi, On the performance of linear decreasing inertia weight particle swarm optimization for global optimization, *The Scientific World Journal* 2013 (1) (2013) 860289. doi:<https://doi.org/10.1155/2013/860289>.
- [38] H. Liu, Z. Wen, W. Cai, Fastpso: Towards efficient swarm intelligence algorithm on gpus, in: Proceedings of the 50th International Conference on Parallel Processing, ICPP ’21, Association for Computing Machinery, New York, NY, USA, 2021. doi:[10.1145/3472456.3472474](https://doi.org/10.1145/3472456.3472474).
- [39] Y. Shi, R. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Vol. 3, 1999, pp. 1945–1950 Vol. 3. doi: [10.1109/CEC.1999.785511](https://doi.org/10.1109/CEC.1999.785511).
- [40] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization: An overview, *Swarm Intelligence* 1 (10 2007). doi:[10.1007/s11721-007-0002-0](https://doi.org/10.1007/s11721-007-0002-0).

Key Terms and Definitions

Convolutional Neural Network: A deep learning model that performs many tasks including convolution, an integral of the product of 2 functions.

Ischemia: A condition caused by an inadequate blood supply to an organ(s).

Finite Impulse Response: A frequency domain response of a signal impulse has a finite duration.

Hermite transform: An integral transform that uses classical orthogonal polynomial sequence as Hermitian wavelets.

Discrete Wavelet Transform: A transform that decomposes a signal into sets, where each set is a time series of coefficients that correspond to the changes of a signal in the frequency domain.

Fractional Discrete Cosine Transform: A transform that yields forms of a signal that interpolate between the cosine modulated form of the signal and its DCT representation.

Radon Wavelet Transform: A transform that uses the projection of a signal along a radial line oriented at a specific angle. the projection of the image intensity along a radial line oriented at a specific angle.

Frequency Division Dehazing Network (FDDN): A neural network comprising of a frequency division into a series of non-overlapping frequency bands and a mathematical signal haze effect removal.

Biography



Sohayla Hamed received her BSc in Computer Engineering and Software Systems from Ain Shams University in 2024. In addition to her experience as a Software Engineer, she also focuses her research on Data Science and Data Engineering. Currently, she plans to pursue more varied projects such SaaS workflow, API optimization or LangGraph applications.