PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University of El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculty of *Sciences and Technology*

Electronics Department

THESIS

Presented to obtain

THE MASTER'S Degree

Speciality: Electronic Embedded System

By

- **Abdallah Abounacer BAMMARA**
- **Abdesselam MOUSSELMAL**

Title

Classification of ECG signals using 1D-2D transformation and convolutional neural networks (CNN)

Supported on :

Before a jury composed of:

First &Last Name	*Grade*	*Quality*	*Establishment*
Ms. Nacira DIFFELLAH	*MCA*	*President*	*Univ-BBA*
Mr. Mohamed Lamine TALBI	*Professor*	*Supervisor*	*Univ-BBA*
Ms. Fouzeia HAMMADACHE	*MAA*	*Examiner*	*Univ-BBA*

Academic year 2024 /2025

Acknowledgement

First, we thank Allah who give us the strength and wisdom to complete this modest work.

We would like also to express our sincere appreciation to our supervisor **Mr.TALBI Mohamed Lamine**, for the expertise, the huge support, And the great confidence that has certainly played a determining role in our success not only in the framework of this thesis, but throughout our university career.

Then, we would like to thank the members of Jury for agreeing to evaluate this humble work.

Finally, we extend our sincere gratitude to all our family and friends for their support and encouragement throughout our studies.

Words cannot adequately express our deep gratitude to our parents for the support and invaluable assistance they provided us during our many years of study.

Dedication

I dedicated this humble and modest work to:

My dear father, the man whom I respect the utmost in my life,

The light of my life and a guide that I follow and aspire to be like.

My dear mother, the woman who encouraged me the most in my studies,

My first and greatest teacher.

My siblings, whom I own a lot to and adore in this life, and ask Allah,

For their everlasting support and happiness.

My Friends who were assisting and offering help during these times.

***All** who helped and aided in the creations of this project's Prototype.*

Sincerely

BAAMARA Abdallah

Dedication

I dedicate this modest work to:

The source of my life, energy and happiness in this world, my father and My mother, who have provided me with all the support in various ways to reach this point in my life, with all their capabilities and beauty, that is unforgettable.

The secret of my joy and laughter, who are the blessings from Allah.

That I envy myself for. I ask Allah to grant us everlasting love between us and to bless them in their lives.

My Friends who were assisting and offering help during these times.

***All** who has provided me with academic and moral support.*

Sincerely

MOUSSLEMAL Abdesselam

ملخص

تبحث هذه المذكرة في تصنيف إشارات تخطيط القلب تلقائيًا باستخدام الشبكات العصبية الالتفافية عبر تحويل الإشارات أحادية البعد إلى صور طيفية ثنائية الأبعاد. تستجيب هذه الدراسة للحاجة إلى كشف اضطرابات نظم القلب بدقة وقابلية للتوسع باستخدام التعلم العميق. تم تطوير واختبار سلسلة متكاملة تشمل المعالجة المسبقة والتحويل وتدريب الشبكات العصبية وتقييم الأداء. درست التجارب تأثير خيارات المعمارية وعدد الدورات التدريبية ودقة الصور على الأداء. أظهرت النتائج تحقيق الشبكات العصبية لدقة عالية خصوصًا في النبضات الطبيعية. لا تزال تحديات مثل عدم توازن الفئات وفرط التكيف قائمة وتحد من التعميم. تقترح الأعمال المستقبلية استكشاف معماريات متقدمة وتوسيع البيانات.

الكلمات الرئيسية: تخطيط القلب (ECG), الشبكات العصبية الالتفافية (CNN), التصنيف التلقائي, المخططات الطيفية

Abstract

This thesis investigates automatic ECG signal classification using CNNs by transforming 1D signals into 2D spectrogram images. It addresses the need for accurate, scalable arrhythmia detection with deep learning approaches. A complete pipeline, including preprocessing, transformation, CNN training, and evaluation, was developed and tested. Experiments examined architecture choices, training epochs, and input resolution impacts on performance. Results show CNNs achieve high accuracy, particularly on normal beats. Challenges like class imbalance and overfitting remain and limit generalization. Future work includes advanced architectures, data augmentation, and larger dataset validation.

Keywords: ECG, Convolutional Neural Networks (CNN), Automatic Classification, Spectrograms

Résumé

Ce mémoire étudie la classification automatique des signaux ECG en utilisant des CNN en transformant des signaux 1D en images spectrogrammes 2D. Il répond au besoin de détection précise et évolutive des arythmies via l'apprentissage profond. Un pipeline complet, incluant prétraitement, transformation, entraînement et évaluation des CNN, a été développé et testé. Des expériences ont examiné les choix d'architecture, les époques d'entraînement et l'impact de la résolution sur la performance. Les résultats montrent que les CNN atteignent une grande précision, notamment sur les battements normaux. Des défis tels que le déséquilibre des classes et le surapprentissage persistent. Les travaux futurs incluent des architectures avancées et l'augmentation des données.

Mots clés : ECG, Réseaux de neurones convolutifs (CNN), Classification automatique, Spectrogrammes

Table Of Contents

Chapter I

Chapter II

Chapter III

Chapter IV

List Of Figures

Chapter I

Chapter II

Chapter III

Chapter IV

List Of Tables

Chapter I

Chapter II

Chapter III

Chapter IV

List of Abbreviations

ECG: Electrocardiogram

1D: One-Dimensional

2D: Two-Dimensional

CNN: Convolutional Neural Network

AI: Artificial Intelligence

ML: Machine Learning

DL: Deep Learning

TFD: Time-Frequency Domain

DWT: Discrete Wavelet Transform

STFT: Short-Time Fourier Transform

ACC: Accuracy

General Introduction

General Introduction

The (ECG) is one of the most essential tools in cardiology, widely used to monitor the electrical activity of the heart and detect abnormal conditions such as arrhythmias. With cardiovascular diseases remaining a major cause of mortality worldwide, there is a growing need for more accurate, automated, and scalable ECG analysis methods.

Traditional ECG analysis methods rely heavily on manual feature extraction or rule-based algorithms, which may not fully capture the complex temporal and morphological patterns in ECG signals. In contrast, deep learning—particularly (CNNs)—offers a promising solution by automatically learning features from data, especially when ECG signals are transformed into image-like representations.

This thesis addresses the central research question: how can one-dimensional ECG signals be effectively transformed into two-dimensional representations suitable for classification by CNNs? The main objective is to design and evaluate a deep learning system that classifies ECG signals using spectrogram-based visual representations.

To achieve this, we follow a structured methodology:

- **Chapter 1** presents the fundamentals of ECG signals, their components (P wave, QRS complex, T wave), and common noise artifacts.
- **Chapter 2** introduces CNNs, detailing their architecture and role in signal classification, with a focus on biomedical applications.
- **Chapter 3** explores transformation techniques such as STFT, CWT, and DWT, which convert 1D signals into 2D images like spectrograms and scalograms.
- **Chapter 4** describes the complete experimental pipeline—data preprocessing, image generation, CNN implementation, training, and evaluation using metrics like accuracy and confusion matrices.

This study contributes to the ongoing development of AI-based solutions in biomedical signal processing by demonstrating how CNNs can effectively classify ECG signals when paired with appropriate signal-to-image transformations.

Chapter 1

ECG Signal Basics

I. Introduction

An essential tool in cardiology, the (ECG) captures the electrical activity of the heart. It offers vital details about the rhythm and operation of the heart, which are necessary for identifying and keeping track of different cardiac conditions. The P wave, QRS complex, and T wave—the three main parts of the ECG waveform—represent different stages of the cardiac cycle. In addition to discussing common sources of noise and artifacts that can obstruct ECG interpretation, this chapter presents the fundamental concepts of ECG signals, including their physiological origins and key waveform characteristics. As discussed in the following chapters, a thorough comprehension of these ideas is necessary for the efficient analysis and classification of ECG data using contemporary computational techniques.

II. Definition and importance of the ECG signal

A medical test called an (ECG) shows the heart's electrical activity as a waveform over time. Physicians can use this test to assess heart function and identify diseases like arrhythmias or heart damage. An electrocardiograph is a device that records the ECG by measuring tiny electrical signals between skin-placed electrodes. [1]

In electrocardiography, a "lead" refers to the path between two electrode points through which the heart's electrical signals are recorded. The arrangement of these leads greatly affects the accuracy of measuring cardiac activity. In 1902, Willem Einthoven developed a standardized lead system, which greatly improved the diagnostic capabilities of electrocardiography and significantly impacted the field of cardiology, as shown in Figure 1. [1], However, ECG signals may be distorted by various artifacts—unwanted signals caused by movement, muscle activity, or electrical noise—which complicate their interpretation.

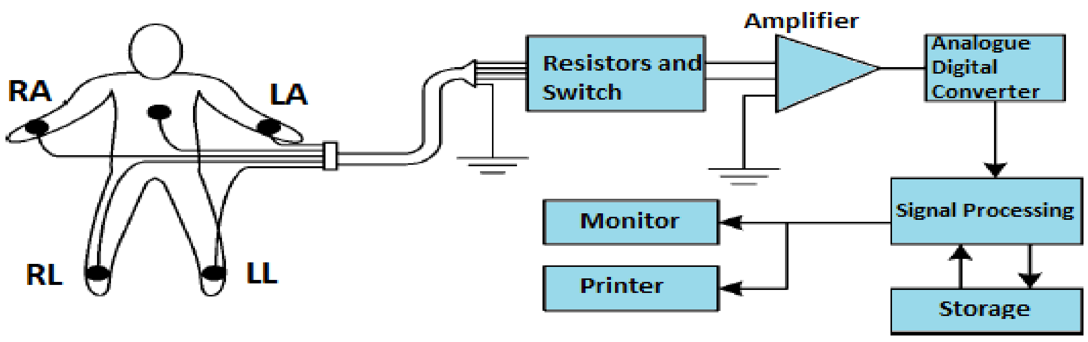


Figure II-1: Block Diagram of a Three Lead Electrocardiograph

III. Signal Processing Techniques for Clean ECG Signal

To obtain clean and reliable ECG signals, several signal processing techniques are employed:

- **High-pass filters:** These filters remove slow transitions in the signal, such as baseline drift.
- **Notch filters:** Remove power line interference (50/60 Hz) .
- **Low-pass filters:** Reduce high-frequency noise generated by muscle activity.
- **Waveform transforms:** remove noise while preserving important signal characteristics.
- **Heartbeat detection algorithms:** Heartbeats are automatically detected in ECG signals using techniques like the Pan-Tompkins algorithm.

When combined, these methods enhance the quality of ECG data and help identify cardiac abnormalities like arrhythmias and myocardial infarctions.

IV. Research and Academic Use of ECG Signals

Electrocardiogram signals are widely used in academic research for a variety of purposes.

- **Benchmarking and algorithm development:** ECG signals are essential in the development and testing of new algorithms in biomedical engineering.
- **Educational applications:** They serve as practical tools in medical and engineering programs for teaching signal interpretation.
- **Cardiovascular simulations:** ECG signals are applied in computational models to study heart behavior and disease mechanisms [2].

V. Signal Processing and Feature Extraction

Various signal processing techniques are employed to extract meaningful features from raw ECG signals. These methods include:

- **Filtering:** To eliminate noise and artifacts such as baseline wander and power-line interference .
- **Wavelet transforms:** For denoising and feature localization.
- **Principal Component Analysis (PCA):** To reduce signal complexity and highlight key features.
- **Machine learning and deep learning models:** To classify ECG patterns and detect abnormalities such as arrhythmias.

These techniques significantly enhance the diagnostic power of ECG analysis and support the development of automated systems for cardiovascular healthcare.

VI. Fundamental Signal Characteristics

The ECG waveform consists of three main components:

- **P wave**: represents atrial depolarization; its amplitude is less than 0.25 mV and it typically lasts 80–120 ms.
- **QRS complex**: represents ventricular depolarization; it typically has an amplitude of 1-2 mV and lasts 60-100 ms..
- **T wave**: represents ventricular repolarization; its amplitude is less than 0.5 mV and it typically lasts 160–200 ms..

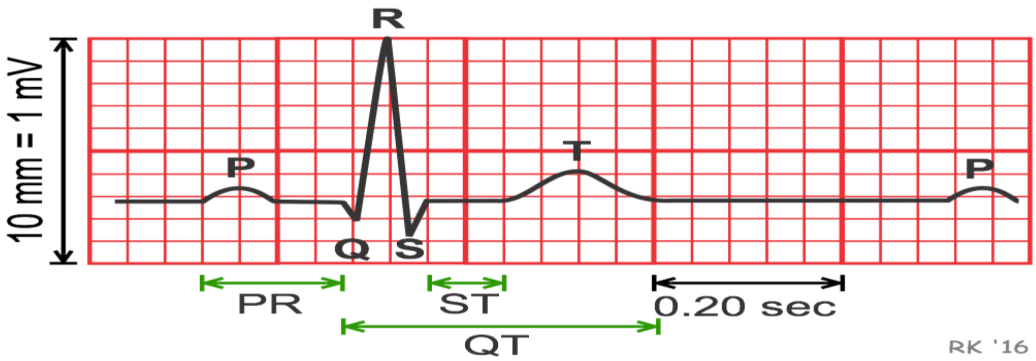


Figure VI-1: ECG signal characteristics [28]

VII. ECG Signal Noise and Artifacts

ECG signals are susceptible to various types of interference, which can degrade their quality. These include:

VII.1 Physiological artifacts

- **Muscle artifacts:** Electromyographic (EMG) noise due to skeletal muscle activation.
- **Respiratory and motion artifacts:** Leading to baseline drift and signal distortion.
- **Electrode-skin impedance variations:** Caused by sweat, temperature changes, or poor electrode contact.

VII.2 Technical artifacts

- **Electrode contact noise**: Poor contact with the skin may cause intermittent signal loss.
- **Power-line interference**: Common 50/60 Hz interference from mains electricity, which is addressed by notch filters [3].

Figure 3 illustrates various types of noise artifacts in an ECG signal, including baseline wander, muscle noise, and power-line interference.

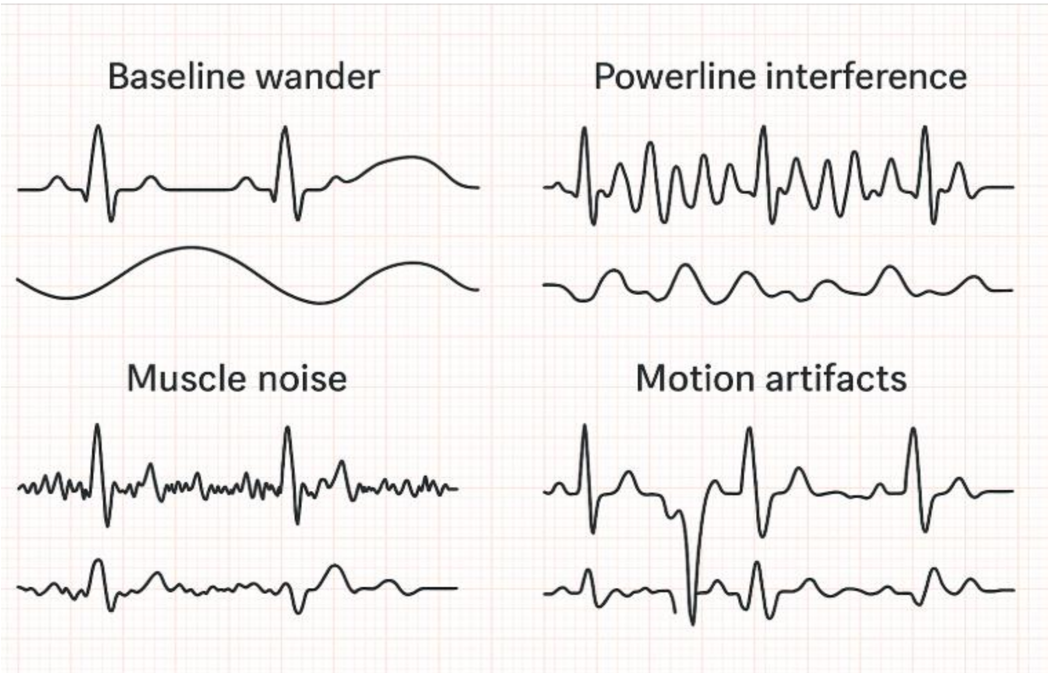


Figure VII-1: Illustration Different Types of Noise Artifacts in ECG Signal [3].

VIII. Main Components of the ECG Signal

The ECG signal consists of several key components that represent different phases of the heart's electrical activity:

1. **P Wave**: Reflects atrial depolarization, initiated at the sinoatrial (SA) node and transmitted through Bachmann's bundle [5].
   - **Morphology and Duration**: The P wave is normally smooth and positive in lead II, with an amplitude of <0.25 mV and a duration of 80–120 ms. Abnormalities may indicate atrial enlargement or ectopic activity.
   - **Frequency Content**: Primarily between 5–30 Hz; advanced detection techniques are needed in noisy environments [4].
2. **QRS Complex**: Represents ventricular depolarization.
   - **Biophysical Origin**: The Q wave originates from the septum, the R wave from the ventricular walls, and the S wave from the basal areas of the ventricles [5] .
   - **Morphology and Duration**: Typically 60–100 ms in duration and 1–2 mV amplitude in limb leads, and up to 3 mV in chest leads [6] [7] , A QRS wider than 120 ms may indicate conduction block or ectopic activity.
   - **Frequency Content:** Concentrated in the range of 8–50 Hz due to rapid voltage changes, requiring high sampling rates and careful filtering [4].
3. **T Wave**: Represents ventricular repolarization.
   - **Biophysical Origin:** The T wave reflects repolarization driven by apicobasal and transmural gradients [8] .
   - **Morphology and Duration:** The amplitude is typically <0.5 mV and the duration ranges from 160–200 ms. Inverted or flattened T waves may signal ischemia or electrolyte imbalance.
   - **Frequency Content:** The frequency content of the T wave is mostly below 10 Hz, and low-pass filtering (<40 Hz) is typically used to reduce noise without distorting the signal.

Table VIII-1: Table of common ECG signal artifacts.

Noise/Artifact Type	Source	Frequency Range	Impact on Signal	Mitigation Techniques
Baseline Wander	Respiration, patient motion, electrode impedance variations	< 0.5 Hz	Shifts the ECG baseline	Hight-pass filtering, polynomial fitting
Power line interference	Mains power (50/60 Hz)	50 or 60 Hz	Adds sinusoidal interference	Notch filters, adaptive filtering
Electromyographic (EMG) noise	Skeletal muscle activity	~20-100 Hz	Distrots high-frequency component	Adaptive filtering, wavelet transform-based techniques
Electrode motion artifacts	Patient movement, electrode displacement	Variable (often < 5 Hz)	Causes transient distortions	Signal averaging mechanical stabilization, adaptive filtering

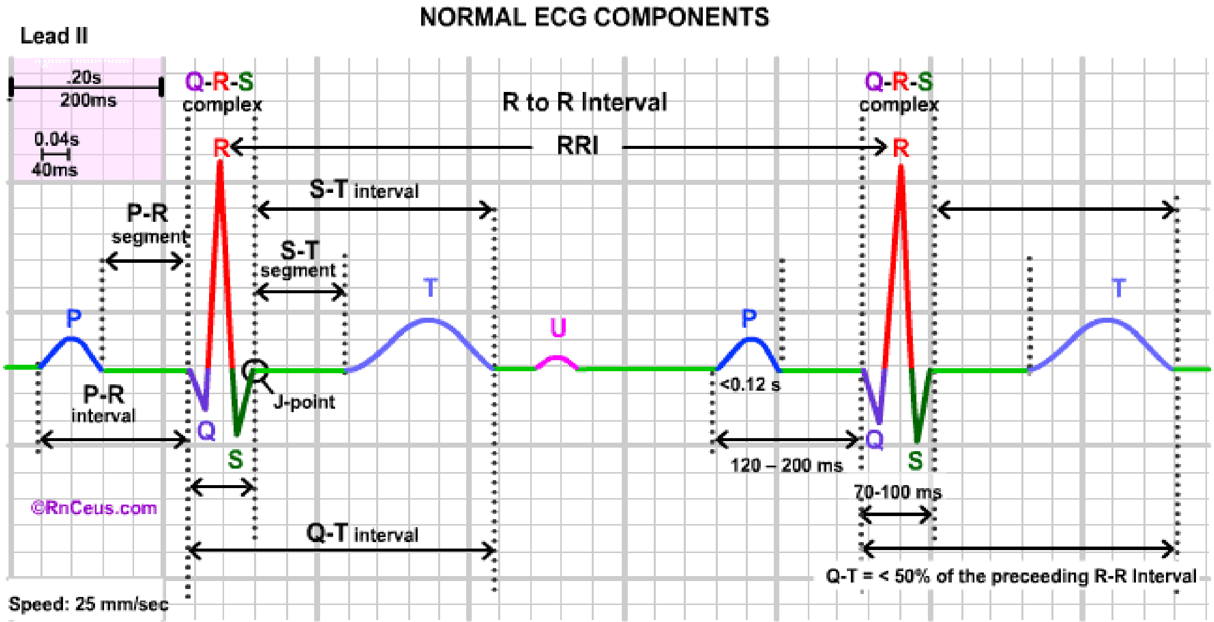


Figure VIII-1: ECG Signal with P wave, QRS complex, and T wave [29]

IX. MIT-BIH Arrhythmia Database Description

A popular benchmark dataset for cardiac arrhythmia classification and ECG signal processing is the MIT-BIH Arrhythmia Database. Part of the PhysioNet project, the dataset was initially made available in 1980 and was created by Beth Israel Hospital and the Massachusetts Institute of Technology. It includes 48 half-hour two-lead ECG recordings from 47 subjects that have been annotated by skilled cardiologists and sampled at 360 Hz.

In addition to rhythm variations and signal artifacts, each recording features beat-by-beat annotations of both normal and abnormal heartbeats. The signals provide a variety of signal morphologies and are obtained from one of the chest leads (V1, V2, V4, or V5) and modified limb lead II. Especially in machine learning and deep learning contexts, the database is regarded as the gold standard for training and validating arrhythmia detection algorithms.

X. Beat Types Considered in This Study

This study focuses on the classification of five representative beat types selected from the MIT-BIH Arrhythmia Database. These include both normal and abnormal cardiac events and are consistent with the AAMI EC57 standard for heartbeat classification [9].

X.1 Normal beat (N)

- Labeled as: N in the database.
- Description: Represents the standard sinus rhythm generated by the sinoatrial (SA) node.
- Clinical significance: Serves as the baseline or reference class in most classification tasks.

X.2 Ventricular ectopic beat (V)

- Labeled as: V or PVC (Premature Ventricular Contraction).
- Description: Occurs when the ventricles contract prematurely due to an ectopic pacemaker.
- Clinical significance: Common in various arrhythmic conditions; may indicate underlying heart disease.

X.3 Fusion beat (F)

- Labeled as: F in the database.
- Description: A hybrid beat formed by the fusion of a normal and a ventricular ectopic beat.
- Clinical significance: May complicate arrhythmia classification due to its mixed morphology.

X.4 Atrial premature beat (S)

- Labeled as: S or APC (Atrial Premature Contraction).
- Description: Caused by early activation of the atria due to an ectopic atrial focus.
- Clinical significance: Often benign but may trigger atrial arrhythmias in some patients.

X.5 Aberrated atrial beat (Q)

- Labeled as: A (Atrial escape or aberrant atrial beat).
- Description: A supraventricular beat with abnormal conduction through the ventricles.
- Clinical significance: Indicates disrupted atrioventricular conduction pathways.

Table X-1: Summary of the five beat types used in this study.

Beat type	Label origin	Description
N	Sinoatrial node	Normal sinus rhythm
V	Ventricular ectopic focus	Premature contraction of ventricles
F	Mixed ventricular/normal	Fusion of normal and ectopic ventricular impulses
S	Atrial ectopic focus	Premature activation from an atrial source
Q	Aberrant supraventricular	Atrial beat with abnormal ventricular conduction

These beat types cover a broad spectrum of clinically significant rhythms, ensuring that the classification model is both robust and generalizable to various cardiac conditions.

XI. Medical Applications and Challenges

ECG signals are critical in diagnosing various cardiac conditions, including arrhythmias, ischemia, and conduction disorders. The increasing use of wearable devices and mobile health platforms enables continuous, ambulatory ECG monitoring, which aids in the early detection of abnormalities in high-risk patients, even in rural or remote areas [10].

However, several challenges remain in ECG signal analysis:

- **Signal preprocessing**: The signal-to-noise ratio can be increased with the use of sophisticated techniques like wavelet denoising, empirical mode decomposition (EMD), and band-pass filtering, but their application must be done carefully, [1].
- **Patient variability**: The generalization of ECG classification models can be greatly impacted by patient differences, both within and between patients .
- **Data limitations**: The effectiveness of machine learning algorithms is limited by the small or unbalanced size of many publicly available datasets.

- **Interpretability**: Interpretability of machine learning models is a persistent problem, as clinicians are frequently reluctant to put their trust in "black-box" models. To address these issues, explainable AI methods like saliency maps are being investigated [11].
- **Ethical considerations**: Clinical adoption of ECG-based AI models requires algorithmic fairness, privacy, and informed consent [12].

XII. Conclusion

The foundation of cardiovascular diagnostics is ECG analysis. The clinical utility of signal processing is being further enhanced by developments in signal processing and the incorporation of artificial intelligence (AI); however, issues like noise, patient variability, data limitations, interpretability, and ethical considerations need to be addressed. Unlocking the full clinical potential of ECG signals requires constant algorithm development, dataset quality enhancements, and the application of explainable AI techniques. Remote heart health monitoring, precise arrhythmia diagnosis, and real-time cardiovascular event prediction will all become more feasible as these obstacles are overcome, particularly with the incorporation of wearable technology and Internet of Things technologies.

Chapter 2
Convolutional neural networks (CNNs) Applied to signal processing

I. Introduction

A class of deep, feed-forward artificial neural networks called (CNNs) uses backpropagation to learn the temporal and spatial hierarchies of features. Convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification or regression tasks make up a typical CNN architecture , In order to train such networks, millions of parameters must be optimized using stochastic gradient descent (SGD) and its variations, as well as regularization strategies like batch normalization and dropout to avoid overfitting, By learning intricate signal morphologies straight from raw data, CNNs have proven to be highly effective in biomedical signal processing tasks like ECG arrhythmia detection [13], EEG pattern recognition, and EMG classification. CNNs may perform poorly on rule-based measurements (such as precise interval calculations) and frequently behave as "black boxes," which makes them difficult to interpret clinically, even though they can gradually get better with more data and find new diagnostic markers .

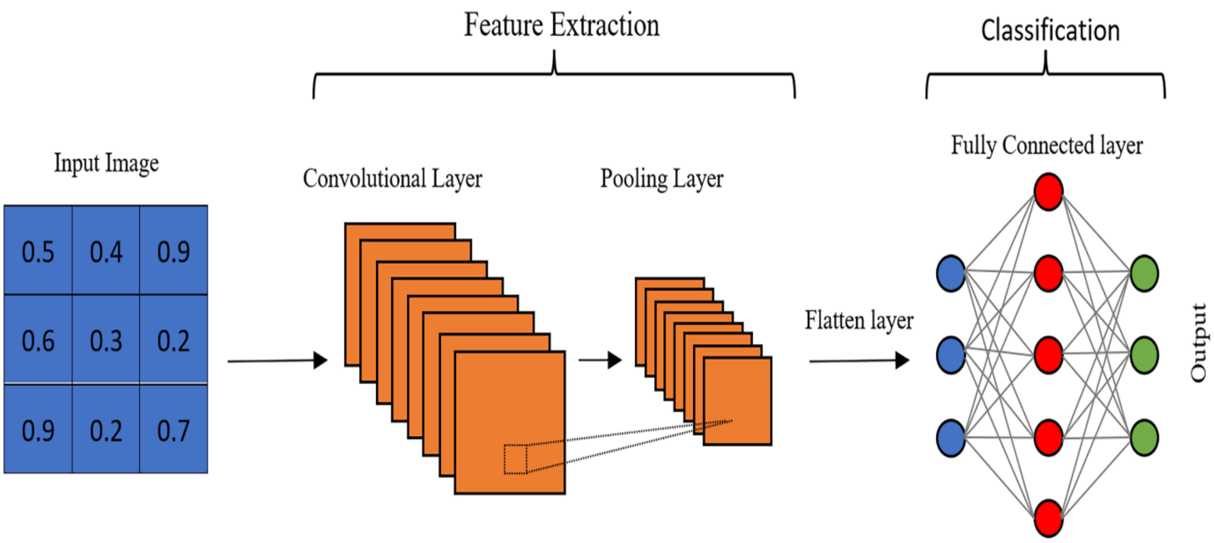


Figure I-1: Architecture of a CNN [18]

II. Architecture of a CNN

Processing network-structured data such as images requires special-purpose models called CNNs. The convolutional layers within CNNs are designed to apprehend spatial hierarchies-from low-level features such as corners and edges to high-level features such as parts and objects-and are suitably structured to do so, depending upon the application, for automatic and adaptive learning of these spatial hierarchies for image processing, e.g., object detection, semantic segmentation, and image classification [13], Eventually CNNs along with all processing and architectural methods, must be trained and optimized effectively for maximal conveyance.

A CNN typically consists of several layers:

- **Input Layer**
- **Convolutional Layer (CONV)**
- **Activation Function Layer**
- **Pooling Layer (POOL)**
- **Fully Connected Layer (FC)**
- **Output Layer**

II.1 Convolutional Layers

Convolutional layers consist of a set of filters (kernels) that slide over the input, computing dot products to produce feature maps that capture local patterns [14], Key hyper parameters include kernel size (commonly 3×3 or 5×5), stride (the step size of the filter), and padding (to control output dimensions).

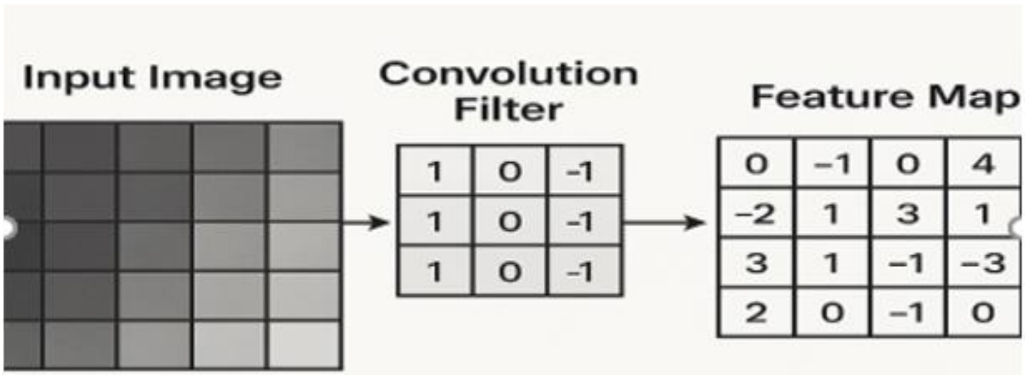


Figure II-1: Example of Convolutional Layer

II.2 Pooling Layers

Pooling layers perform down sampling, typically via max-pooling, which retains the most prominent activation within each receptive field, or average-pooling, which takes the mean response. Pooling helps reduce computational load and introduces spatial invariance [14].

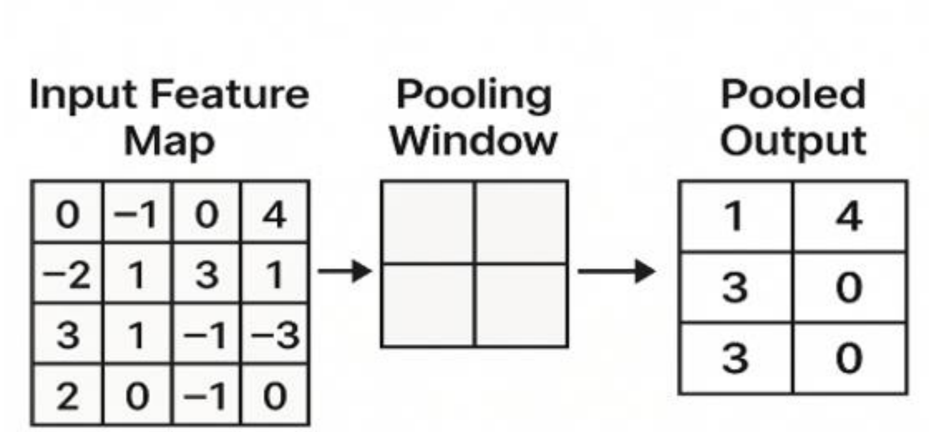


Figure II-2: Example of Pooling Layer

II.3 Fully Connected Layers

After several convolutional and pooling stages, the feature maps are flattened and passed through one or more fully connected (dense) layers, culminating in an output layer (e.g., softmax for classification) [14], These layers integrate the learned features to produce final predictions.

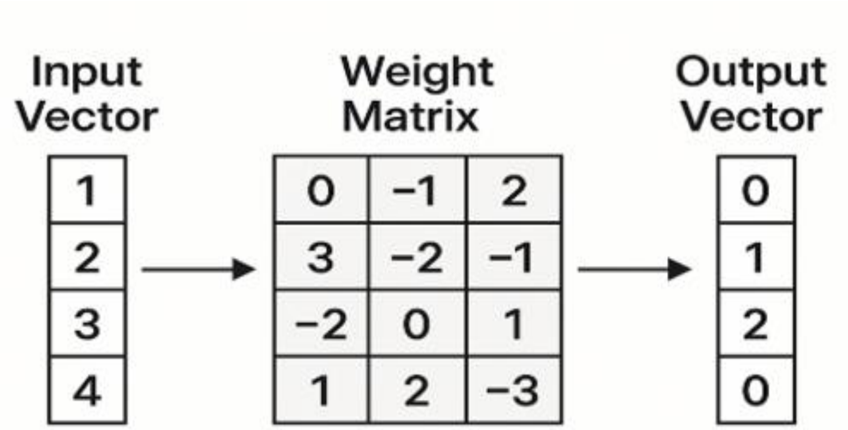


Figure II-3: Fully Connected Layer

III. Training and Optimizing CNNs

Training CNNs entails minimizing a loss function (e.g., cross-entropy) over the network's parameters via backpropagation and optimization algorithms. Stochastic Gradient Descent (SGD) with momentum, Adam, and RMSProp are widely used optimizers offering different trade-offs in convergence speed and stability [15].

III.1 Optimization Algorithms

III.1.1 Stochastic Gradient Descent (SGD)

SGD updates model parameters by computing gradients on mini-batches:

$$\theta t + 1 = \theta t - \eta \nabla \theta L(\theta t)$$

Where η is the learning rate, and L(θt) is the loss function [16].

III.1.2 Adaptive Optimization Methods: Adam

Adam combines ideas from momentum and RMSProp:

$$mt = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta L(\theta_t)$$

Were:

mt: first moment vector at time step t (exponential moving average of gradients)

β: exponential decy rate for the firstmoment (typical around 0.9)

mt−1: previous first moment vector (from step t-1)

∇θL(θt): gradient of the loss fuction L with respect to parameters θ at step t

θt: model parameters at step t

L(θt): loss function evaluated at parameters θt

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)[\nabla_\theta L(\theta_t)]^2$$

Were:

Vt: second moment estimate atstep t (moving average of squared gradient)

β2: exponential decay rate for the second moment (commonly β2=0.999)

vt−1: previous second moment estimate at step t-1

[∇θL(θt)]2: Element wise square of the gradient vector (not a matrix square)

$$\widehat{m}_t = \frac{w_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$\vartheta_{-L+1} = \theta_t - \eta \frac{\widehat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

Were:

m^t: bias- corrected first moment estimate (mean of average)

vt: second moment estimate -running average of squared gradient

v^t : bias-corrected seconde moment of estimate (variance of gradient)

Adam is known for fast convergence and reliable performance across various deep learning tasks.

III.2 Regularization Techniques

III.2.1 Dropout

Dropout randomly disables neurons during training to reduce overfitting:

$$\mathrm{Hi} = \mathrm{Hi} * \mathrm{Zi}, \quad \mathrm{Zi} \sim \mathrm{Bernoulli(P)}$$

Were:

Hi: activation (or output) of the i the neuron/unit before/ after dropout (context-dependent

Zi: random binary variable (mask) for unit I, sampled from a Bernoulli distribution

P: dropout keep probability (ex: p=0.5 keeps half the neurons active)

III.2.2 Batch Normalization

Batch normalization normalizes inputs within a mini-batch to stabilize and accelerate training:

$$\hat{\mathrm{x}} = \frac{\mathrm{x}-\mathrm{u}}{\sqrt{\sigma^2+\epsilon}}, \qquad \mathrm{y} = \gamma\hat{\mathrm{x}} + \beta$$

Were:

X: input activation (before normalization),a scalar or vector from a mini batch

μ or u: mean of the batch

σ2: varienc of the batch

ϵ: small constant for numerical stability (ex:10 power 5)

γ: learnable scale parameter (usedto re-scale the normalized value)

β: learnable output parameter(used to re-center the output)

It reduces internal covariate shift and improves generalization.

III.3 Data Augmentation

Data augmentation increases training data diversity through:

- **Geometric transformations** (rotation, flipping, scaling)
- **Color adjustments** (brightness, contrast)
- **Noise injection**

The data enhances the diversity of the training dataset by applying various transformation to the input data, common techniques include geometric changes like rotation, flipping, and scaling; color adjustments such as altering brightness or contrast and injection noise, these augmentation help prevent overfitting and improve the model's ability to generalize new, unseen data.

Table III-1: Common Data Augmentation Techniques

Technique	Description	Example
Geometric Transformations	Applies changes to the spatial characteristics of the data, such as rotation, scaling, and flipping.	- Rotation (e.g., 90°, 180°) - Flipping (horizontal/vertical) - Scaling
Color Adjustments	Alters the color properties of the data, including brightness, contrast, and saturation.	- Brightness adjustment - Contrast variation - Saturation modification
Noise Injection	Adds random noise to the data, which can improve robustness by simulating real-world variability.	Gaussian noise - Salt-and-pepper noise
Translation	Shifts the data along the X and Y axes to simulate different viewpoints or camera angles.	Horizontal or vertical translation
Shearing	Applies a geometric transformation that shifts pixels in one direction, simulating perspective changes.	Shearing by 10° in horizontal or vertical direction
Zooming	Simulates zoom by randomly scaling the image, helping the model handle different object sizes	Random zoom applied to the input data

A summary of techniques used to increase dataset diversity and improve model generalization.

IV. Learning Rate Scheduling

To avoid local minima and enhance convergence, methods like step decay, cosine annealing, and cyclical learning rates modify the learning rate during training. The optimization process can more successfully traverse the intricate loss landscape with this dynamic approach, possibly discovering flatter, broader minima that more broadly apply to unknown data [17].

V. Architecture Pruning and Distillation

Knowledge distillation into smaller "student" networks and model compression through the removal of unnecessary filters help maximize inference efficiency without sacrificing accuracy. These methods allow for quicker predictions and deployment on devices with limited resources, improving efficiency while maintaining performance [18].

VI. Applications of CNNs in Biomedical Signal Processing

(CNNs), renowned for their effectiveness in image analysis, have shown significant promise in processing and interpreting a wide range of biomedical signals. Biological signals are often complex and noisy, which makes CNNs well-suited for automatically learning hierarchical features from raw data. Several key applications of CNNs in biomedical signal processing are reviewed here.

VI.1 Electrocardiogram ECG Analysis

Electrocardiograms are essential for diagnosing a number of cardiac disorders because they capture the electrical activity of the heart, CNNs have demonstrated promise in automating ECG analysis and increasing its accuracy.

VI.2 Electroencephalogram (EEG) Analysis

EEGs are used to diagnose and track neurological conditions like epilepsy, sleep disorders, and brain-computer interfaces (BCIs), by measuring the electrical activity of the brain.

VI.3 Electromyogram (EMG) Analysis

Electromyograms measure the electrical activity of muscles and are used in the diagnosis of neuromuscular disorders and prosthetic control.

VI.4 Other Biomedical Signals

CNNs are also being applied to other types of biomedical signals, including:

- **Photoplethysmography (PPG):** Used for heart rate variability analysis and blood pressure estimation.
- **Magnetoencephalography (MEG):** Utilized for studying brain activity with high temporal resolution.
- **Respiratory Sounds:** Used for detecting respiratory disorders.

These applications highlight the versatility of CNNs in dealing with diverse biomedical signal types and their potential to revolutionize diagnostic techniques.

VII. Challenges

Despite their remarkable potential, there are several challenges that must be overcome for CNNs to be fully integrated into biomedical signal processing applications:

- **High Dimensionality and Complexity:** Some biomedical signals exhibit high dimensionality, making it difficult to manage and process them efficiently with CNNs [19].
- **Interpretability:** CNNs are often seen as "black boxes," making it difficult for clinicians to trust the models and understand the reasoning behind predictions [19].
- **Scarcity of Labeled Data:** Large, labeled datasets are often necessary for training CNNs effectively, but obtaining these datasets can be difficult and costly [19].

VIII. Conclusion

(CNNs) provide robust, automated methods for ECG analysis, achieving high accuracy in arrhythmia detection and enabling real-time, patient-specific monitoring on lightweight devices. However, several challenges remain, including the need for large labeled datasets, the "black box" nature of CNNs, computational constraints, and sensitivity to noise and signal variability. Future research should focus on enhancing CNN interpretability, improving real-time deployment on embedded systems, and addressing the limitations posed by noisy and low-quality ECG signals.

Chapter 3

Techniques for transforming a 1D signal into an image

I. Introduction

The transformation of one-dimensional (1D) signals into two-dimensional (2D) image representations has emerged as a powerful strategy for leveraging the capabilities of image processing and computer vision techniques, particularly (CNNs), for the analysis of time-series data. This chapter provides a comprehensive overview of various methodologies employed to achieve this transformation. These techniques encompass time-frequency analysis methods such as Short-Time Fourier Transform (STFT) and Wavelet Transform, which map temporal variations into spectral content across time, generating spectrograms and scalograms, respectively, Other approaches involve encoding signal characteristics into spatial patterns, including Gramian Angular Fields (GAF) and Markov Transition Fields (MTF), which represent temporal correlations and transitions as image textures. The rationale behind these transformations lies in exploiting the inherent spatial feature extraction capabilities of CNNs to identify complex temporal patterns and dependencies within the original 1D signal.

II. Introduction to Signal Transformations

Signal transformations are fundamental tools in signal processing used to analyze signal characteristics in alternate domains—mainly time, frequency, and time-frequency domains [20], In engineering, these transformations reveal hidden structures, filter noise, compress information, and detect transient events.

II.1 Fourier Transform (FT)

is a powerful mathematical tool that decomposes a function of time (a signal) into its constituent frequencies. In simpler terms, it reveals the "recipe" of frequencies that make up a particular signal, Instead of viewing a signal as it evolves over time, the Fourier Transform allows us to see how much of each frequency is present within that signal [20].

Any periodic signal can be represented as a sum of sine and cosine waves at different frequencies, amplitudes, and phases, The Fourier Transform extends this concept to non-periodic signals as well, representing them as a continuous spectrum of frequencies.

For a continuous-time function f(t), the Fourier Transform F(ω) is defined as:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(w) e^{jwt}\, dw$$

where:

– F(ω) is the frequency domain representation of the signal.

– f(t) is the time domain representation of the signal.

– ω is the angular frequency (2πf, where f is the linear frequency in Hertz).

– j is the imaginary unit (√−1).

– e−jωt=cos(ωt)−jsin(ωt) (Euler's formula).

The **Inverse Fourier Transform** allows us to reconstruct the original time-domain signal from its frequency domain representation:

$$F(w) = \int_{-\infty}^{\infty} e^{-jwt}\, dt$$

II.2 Types of Fourier Transforms

There isn't just one "Fourier Transform." The specific type used depends on whether the input signal is continuous or discrete, and whether it is periodic or aperiodic [21] :

- **Continuous-Time Fourier Transform (CTFT):** For continuous, aperiodic signals. This is the form defined above.
- **Fourier Series (FS):** For continuous, periodic signals. It represents the signal as a sum of discrete harmonics [28].
- **Discrete-Time Fourier Transform (DTFT):** For discrete, aperiodic signals (sequences of numbers). The output is a continuous, periodic function in the frequency domain.
- **Discrete Fourier Transform (DFT):** For discrete, periodic signals (or a finite-length segment of a signal treated as periodic). Both the input and output are discrete sequences.

- **The Fast Fourier Transform (FFT):** is an efficient algorithm for computing the DFT and is widely used in digital signal processing [28].

III. Short-Time Fourier Transform (STFT)

Analysing how the frequency content of a signal varies over time is accomplished powerfully by the Short-Time Fourier Transform (STFT). For non-stationary signals whose frequency features change, it is very helpful. The STFT offers a time-localized frequency analysis[28], unlike the conventional Fourier Transform, which offers a global frequency spectrum of the whole signal.

For a discrete-time signal x[n] and a window function w[n] of length N, the STFT X[m,k] is defined as:

$$X[n,k] = \sum_{n=-\infty}^{\infty} x[n]\, w[n-mH]\, e^{-j\frac{2\pi}{N}kn}$$

where:

- x[n] is the input signal.
- w[n] is the window function.
- m is the time index of the window (related to the window's position).
- k is the frequency index.
- H is the hop size (the number of samples the window is shifted at each step).
- N is the length of the DFT (which is often equal to or greater than the window length).
- j is the imaginary unit.

To get this algebraic form :

$$STFT\{x\}\,(t,w) = \int_{-\propto}^{\propto} x(T)\, w(T-t)e^{-jwt}\, dt$$

where w(t) is a window function (example: Hamming, Gaussian) centered around t.

III.1 Spectrogram

A spectrogram is a standard approach to view STFT output. The spectrogram shows, for time and frequency, the magnitude squared (or power) of the STFT coefficients. Usually, the horizontal axis represents time; the vertical axis indicates frequency; and at each point the intensity—or color—represents the power of that frequency component at that moment [22]. The **spectrogram** is the squared magnitude of the STFT:

$$Spectrogram(t, f) = |STFT(t, f)|^2$$

Were:

Spectrogram(t,f): power spectral density (energy) at time t and frequency f

T: time index or frame position (centre of the window in the time domain)

F: frequency bin (usually in Hz)

STFT(t,f): short time fourier transform of the signal at time t , frequency f

It provides a visual time-frequency representation of a signal and is widely used in speech recognition, EEG processing, and radar systems.

III.2 Time-Frequency Resolution Trade-off

A fundamental characteristic of the STFT is the trade-off between **time resolution** and **frequency resolution** [22]**.**

III.2.1 Short Window

Better time resolution from a smaller frame enables exact time location of events. But it produces lower frequency resolution, which makes it challenging to tell closely spaced frequencies apart.

III.2.2 Long Window

More frequency resolution from a longer window lets one clearly separate closely spaced frequencies. Still, it results in less time resolution since events are smeared over a longer period.

The choice of window length is critical and relies on the particular application; and the features of the signal under analysis. While for signals with slowly changing frequencies needing exact spectral analysis a longer window would be more appropriate [22], for signals with rapidly changing frequencies a shorter window could be better.

III.3 Applications of STFT

The STFT and its visualization, the spectrogram, are widely used in various fields

- **Audio Processing:** Analyzing music and speech, identifying frequencies of notes, detecting changes in timbre, speech recognition.
- **Speech Analysis:** Formant analysis, speaker identification [22].
- **Telecommunications**: Analyzing signal characteristics in communication channels.
- **Radar and Sonar:** Analyzing echoes to determine the distance and velocity of objects.
- **Biomedical Signal Processing:** Analyzing EEG, ECG, and EMG signals to study physiological activity and detect abnormalities. For example, changes in the frequency content of an EEG signal during a seizure can be clearly visualized in a spectrogram.
- **Seismic Analysis:** Studying earthquake signals and the Earth's structure [22].
- **Machine Condition Monitoring:** Detecting faults in rotating machinery by analyzing vibration signals.

Within the framework of converting a 1D signal into an image for CNN analysis, the spectrogram produced by the STFT provides a useful 2D representation that records the temporal change of the frequency content of the signal. After that, CNNs can be taught to identify patterns in these spectrograms matching to certain events or classes in the original 1D data.

IV. Continuous Wavelet Transform (CWT)

The Continuous Wavelet Transform (CWT) is a mathematical tool used to analyze non-stationary signals by providing a time-frequency representation. It decomposes a signal into wavelets, which are scaled and shifted versions of a mother wavelet function. The CWT of a signal x(t)x(t)x(t) is defined as:

$$X_{\omega}(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\propto}^{\propto} x(t)\psi\left(\frac{t-b}{a}\right)dt$$

where (a) is the scale parameter, (b) is the translation parameter, and $\psi(t)$ is the mother wavelet [23], This transform allows for the analysis of signals at various scales and positions, making it suitable for detecting transient features and localized frequency components.

IV.1 Applications

CWT has been extensively applied in various fields:

- **Biomedical Signal Processing**: For analyzing EEG and ECG signals to detect anomalies such as epileptic seizures and cardiac arrhythmias [23].
- **Geophysics**: In the analysis of seismic data to identify and characterize different waveforms and events.
- **Mechanical Engineering**: For fault detection in machinery by analyzing vibration signals [23].

IV.2 Advantages

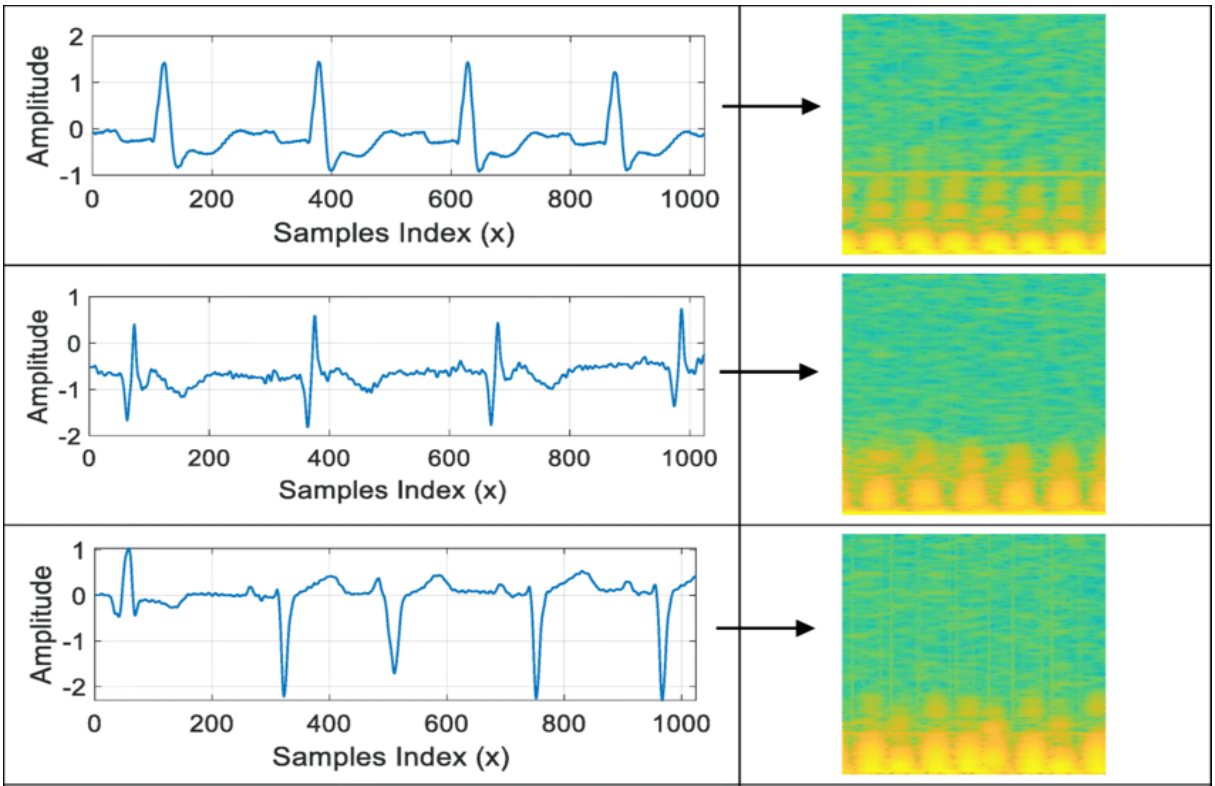
- Provides a detailed time-frequency representation.
- Effective for analyzing non-stationary signals.
- Capable of capturing transient features.

IV.3 Limitations

- Computationally intensive due to redundancy.
- Not suitable for real-time applications without optimization.

IV.4 Illustrative Example

Spectrograms are effective for analyzing signals with stationary or slowly varying frequency content [23], however they have limitation in time frequency resolution due to the fixed window size.



(a) (b)

Figure IV-1: (a) ECG frames, and (b) time-frequency spectrogram images [23].

V. Discrete Wavelet Transform (DWT)

Decomposing signals into approximation and detail coefficients, the Discrete Wavelet Transform (DWT) offers a multi-resolution study of them. It is carried out with low-pass and high-pass filters in filter banks then down sampling [24]. The breakdown is stated recursively.

$$A_{j+1}[n] = \sum_k h[k-2n]A_j[k]$$

$$D_{j+1}[n] = \sum_k g[k-2n]A_j[k]$$

where Aj[n] is the approximation coefficients at level j, Dj[n] is the detail coefficients at level j, and h[n] and g[n] are the low-pass and high-pass filter coefficients, respectively. This hierarchical decomposition allows for efficient representation and analysis of signals at various resolutions.

V.1 Applications

DWT is widely used in:

- **Image Compression**: Employed in standards like JPEG 2000 for efficient image encoding.
- **Denoising**: Removing noise from signals and images by thresholding wavelet coefficients.
- **Feature Extraction**: In machine learning, DWT is used to extract features from signals for classification tasks, such as detecting epileptic seizures from EEG data [24].
- **Biomedical Signal Processing**: For detecting QRS complexes in ECG signals.

V.2 Advantages

- Efficient computation with lower redundancy compared to CWT.
- Suitable for real-time applications.
- Provides both time and frequency localization [24].

V.3 Limitations

- Less precise frequency resolution compared to CWT.
- Choice of wavelet function can significantly affect performance.

V.4 Illustrative Example

Applying DWT to an ECG signal allows for the extraction of detail coefficients that highlight the QRS complexes, which are critical for heart rate analysis.

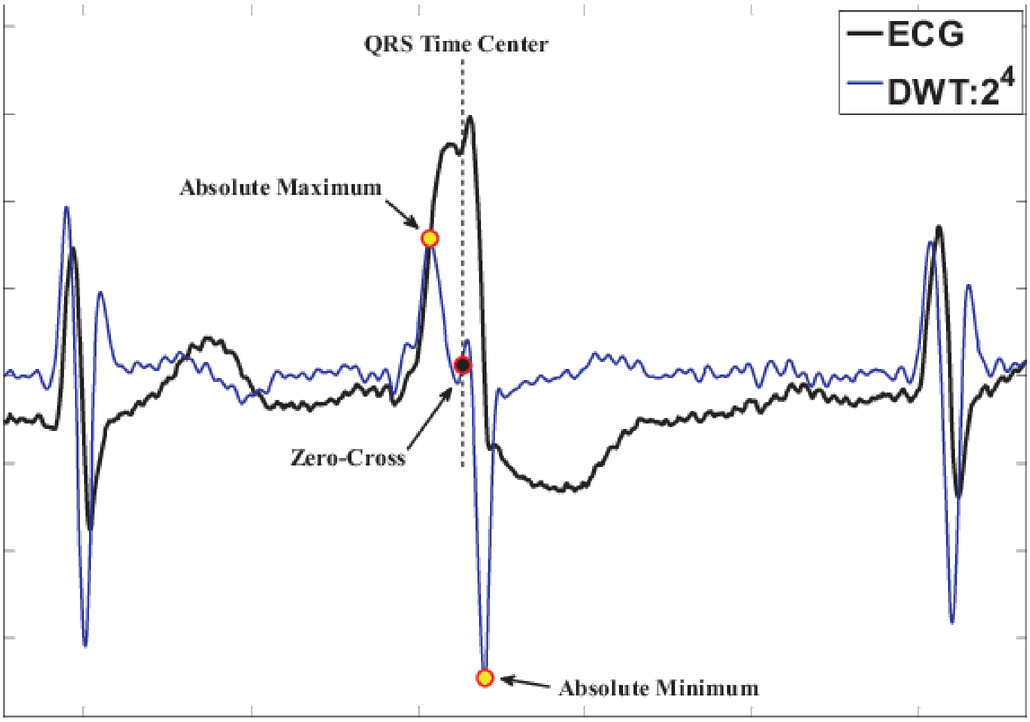


Figure V-1: Determination QRS complex using DWT scale 2^4 [18]

VI. Comparative Analysis of CWT and DWT

Table VI-1: Comparison between CWT and DWT [24].

Feature	CWT	DWT
Signal representation	Continuous scale and translation	Discrete scale and translation
Redundancy	High	Low
Computational load	High	Low
Time-frequency resolution	High	Moderate
Real-time suitability	Limited	Suitable
Application	Detailed signal analysis	Compression, denoising, feature extraction

Essential tools in signal processing, both Continuous and Discrete Wavelet Transforms have special advantages. For research on non-stationary signals, CWT provides thorough time-frequency analysis, which is perfect. By comparison, DWT is well-suited for applications needing real-time processing and data compression and offers effective computation, The particular needs of the application—such as those for resolution, processing resources, and real-time capabilities—will determine whether of CWT and DWT is more appropriate.

VII. Scalogram

A scalogram is derived from the Continuous Wavelet Transform (CWT) and provides a time-scale representation of a signal. Unlike the spectrogram, the scalogram offers multi-resolution analysis, making it suitable for capturing transient features in non-stationary signals like ECGs.

VII.1 Continuous Wavelet Transform (CWT)

$$X_{\omega}(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\propto}^{\propto} x(t)\psi * \left(\frac{t-b}{a}\right) dt$$

Where:

- x(t): Input signal
- ψ(t): Mother wavelet
- a: Scale parameter
- b: Translation parameter
- *: Complex conjugation

The scalogram is the squared magnitude of the CWT coefficients:

$$Scalogram(a,b) = |CWT_x(a,b)|^2$$

Were:

- **Scalogram(a,b):** the energy of the signal x(t) at scale a and time/translation b
- **CWTx(a,b):** the continuous wavelet transform of signal x(t) at scale a and translation b
- **x(t):** inout signal (in the time domain)

Scalograms are particularly useful for analyzing ECG signals, as they can reveal subtle features like P-waves, QRS complexes, and T-waves with high temporal and frequency resolution [24].

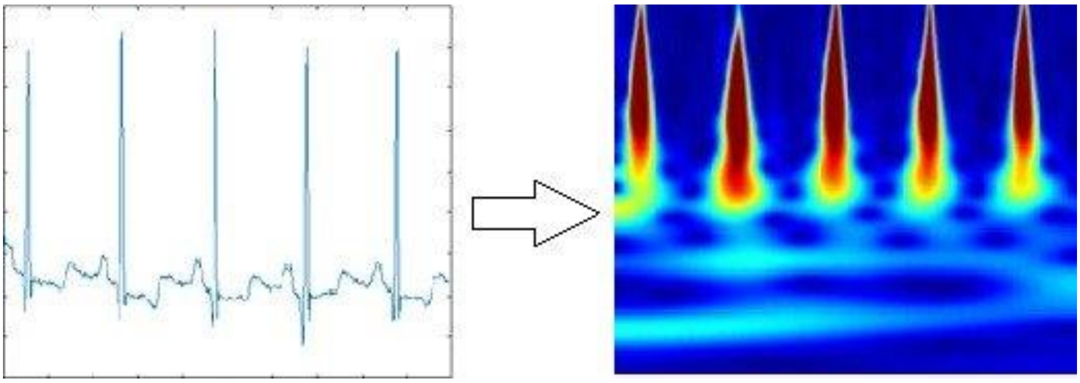


Figure VII-1: 1D ECG into corresponding 2D scalogram [227*227] [24].

VII.2 Illustrative Example

Consider a signal composed of multiple frequency components that vary over time. Applying the CWT to this signal yields a scalogram—a visual representation showing how the frequency content of the signal evolves. Peaks in the scalogram indicate the presence of specific frequency components at particular times.

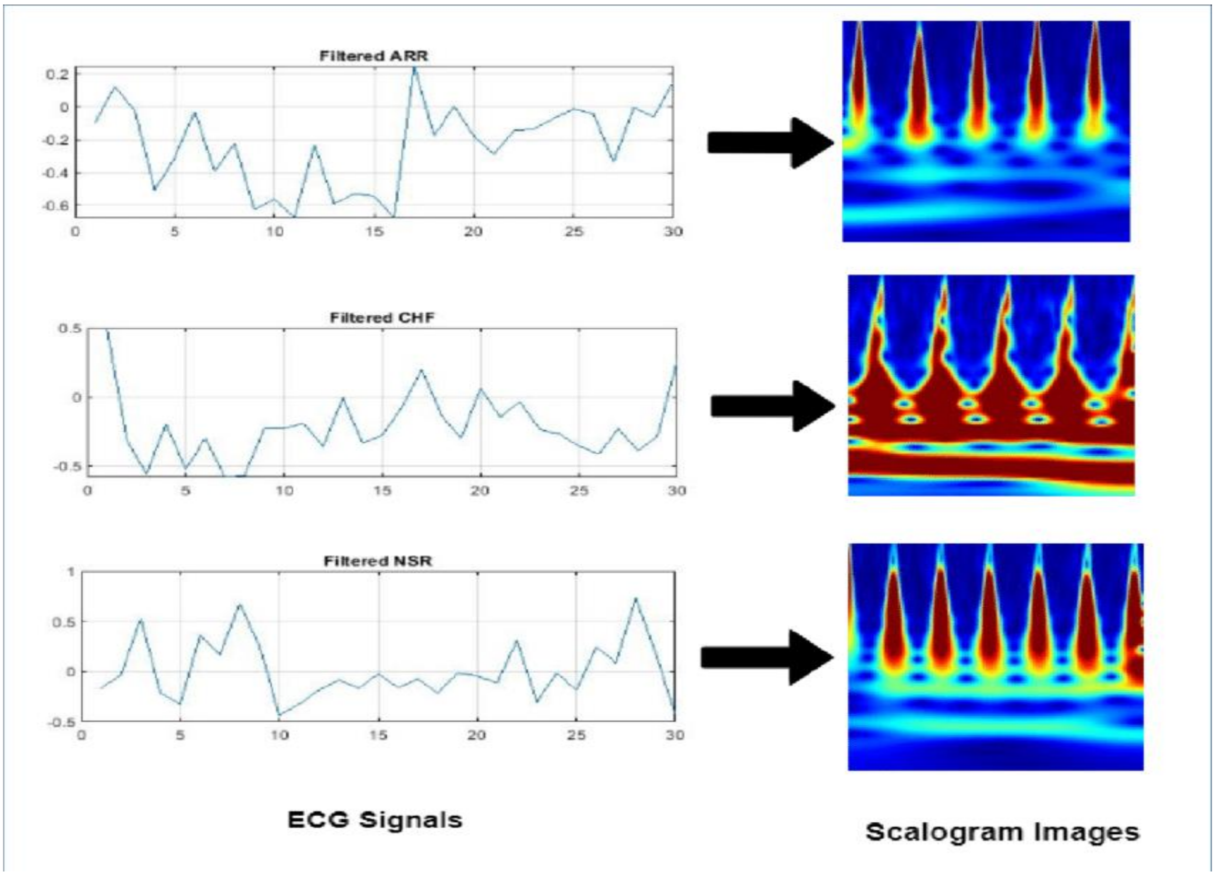


Figure VII-2: 1D ECG signals into Scalogram using CWT [24].

VIII. Comparison of Different Techniques Optimal Choice for ECG

Various techniques have been developed for ECG signal analysis, each with its advantages and limitations; This section compares several methods to determine the optimal choice for specific applications.

VIII.1 Time-Frequency Analysis Techniques

VIII.1.1 Short-Time Fourier Transform (STFT)

STFT provides a fixed-resolution time-frequency representation, suitable for analyzing signals with stationary frequency content. However, its fixed window size limits its ability to capture transient features in ECG signals.

VIII.1.2 Wavelet Transform

Wavelet Transform offers multi-resolution analysis, making it effective for detecting transient features in ECG signals, such as QRS complexes, Discrete Wavelet Transform (DWT) is commonly used for ECG denoising and feature extraction.

VIII.1.3 Hilbert-Huang Transform (HHT)

HHT is an adaptive method that decomposes signals into intrinsic mode functions (IMFs) using Empirical Mode Decomposition (EMD) and then applies the Hilbert Transform to obtain instantaneous frequency data, HHT is effective for analyzing non-linear and non-stationary signals like ECGs.

VIII.2 Machine Learning and Deep Learning Approaches

Recent advancements have seen the integration of machine learning and deep learning techniques for ECG classification and arrhythmia detection. (CNNs) have been employed to automatically extract features from time-frequency representations like spectrograms and scalograms [25].

Table VIII-1: Performance Comparison of Different Techniques.

Technique	Accuracy (%)	Application
STFT + CNN	94.4	Arrhythmia detection
Scalogram + CNN	98.3	ECG classification
HHT	Variable	Feature extraction
DWT	98.1	QRS detection

IX. Conclusion

Especially in biological applications like (ECG), signal transformation methods are basic in evaluating and understanding complicated data. Less effective for signals with fast changing frequencies, the Short-Time Fourier Transform (STFT) offers a time-frequency representation but is limited by a set resolution. Effective capture of both temporal and frequency information, Wavelet Transforms—including Continuous (CWT) and Discrete (DWT)—offer multi-resolution analysis. With spectrograms (from STFT) and scalograms (from CWT), visual aids help one grasp signal properties; scalograms provide superior fit for non-stationary signals. Choosing the suitable technique—be it STFT, CWT, DWT, or their visual representations—in ECG signal processing depends on the particular analysis requirements, including noise reduction, feature extraction [26], or classification. Combining several techniques with cutting-edge algorithms improves diagnosis accuracy and efficiency.

Chapter4 :

Methodology, Results and Discussion

I. ECG Signal Database and Pre-processing

I.1 Introduction

This chapter outlines the processing steps involved in the proposed method for (ECG) signal classification. The approach consists of three main stages: preprocessing, signal-to-image transformation, and classification.

This work builds upon the methodology proposed by Amin Ullah et al, [27] in their study *"Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation"*, where ECG signals are converted into two-dimensional spectrogram images and classified using deep learning techniques. Inspired by their framework, we adopt and adapt a similar pipeline tailored to our specific dataset and experimental goals.

In the first stage, a one-dimensional ECG signal obtained from the MLII lead of the MIT-BIH Arrhythmia Database is preprocessed. The signal is filtered to remove noise and normalized to the range [0, 1] to ensure consistency and suitability for subsequent processing.

The second stage involves transforming the one-dimensional ECG signal into a two-dimensional image using a spectrogram transformer. This transformation facilitates the creation of a labeled dataset comprising five classes: normal beat, premature ventricular contraction (PVC), supraventricular ectopic, fusion beat, and unknown types of arrhythmic beats. These image representations form the input for classification.

The third and most critical stage is the classification of the spectrogram images using a (CNN). The model is designed to learn and identify discriminative visual patterns corresponding to different ECG classes. Its performance is evaluated using metrics such as classification accuracy and the confusion matrix, which reflect both overall and class-wise prediction quality.

It is important to note that MATLAB is used for the first two stages—signal preprocessing and spectrogram generation—while Google Colab is employed for the training and evaluation of the CNN model using Python and deep learning libraries.

The following sections detail each component of the pipeline and present experimental results, offering insights into the strengths, limitations, and potential of the proposed classification approach.

I.2 Image Generation and Dataset Preparation

To develop our CNN-based system for ECG signal classification, a critical step involves transforming the ECG signals into images. This transformation was carried out in the MATLAB environment.

The process begins by loading the prefiltered ECG signals from the MIT-BIH Arrhythmia Database. A single heartbeat is then extracted using a fixed-width rectangular window. After extraction, the signal is normalized to the range [0, 1] to standardize the input and improve consistency across samples. Finally, the spectrogram transformation is applied to convert the one-dimensional signal into a two-dimensional image that represents the time-frequency distribution of the signal's energy.

To train our (CNN) model, we used a selected dataset of spectrogram images generated from ECG signals in MATLAB. These images were saved in PNG format, visually capturing the temporal and frequency characteristics of each heartbeat. The images were organized into separate folders, each corresponding to a specific class or cardiac condition, thereby facilitating the construction of a well-labeled dataset.

This visual representation allowed the CNN to learn essential morphological features of the ECG waveform, such as the QRS complex and the P and T waves, enhancing its ability to distinguish between different arrhythmic and normal conditions.

After generating the spectrogram images, the dataset was transferred to the Google Colab environment, where it was used to train the CNN model. As part of the preprocessing pipeline, all images were resized to a uniform dimension of 224×224 pixels, and their pixel values were normalized to ensure consistency and improve the training performance.

The data was also split into training, validation, and test sets at a ratio of 80-10-10 to ensure objective evaluation of the model.

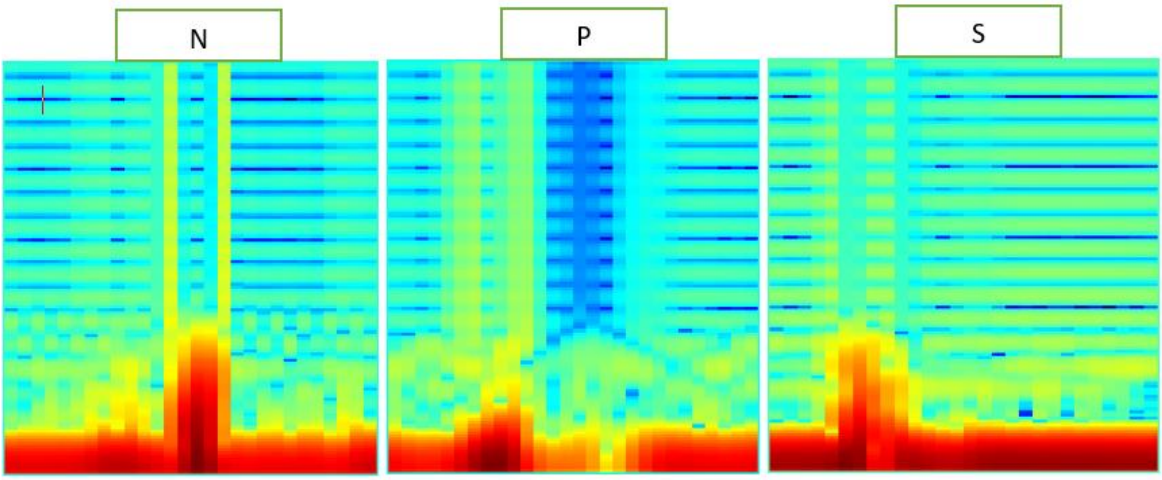


Figure I-1 : Three images representing N class, P class and S class

Figure I-1 shows a preview of three images representing, Normal, PVC and Supraventricular ectopic beats, selected from the ECG spectrogram database generated using MATLAB. As can be seen, each image contains a visual representation of the temporal and frequency characteristics of the cardiac signal, showing different spectral intensities that indicate changes in cardiac activity.

- The contrast between dark and light colors within the images indicates the intensity of the frequency energy at specific times.
- The recurring patterns or peaks evident in these images often correspond to the QRS complex or P and T waves, the essential elements of ECG analysis.
- These images are ideal for feeding a CNN model, which can learn the distinctive visual patterns associated with each disease category.

Table I-1 presents the selected arrhythmia types, and the number of beats used to validate the performance of the proposed classification method. The dataset includes five classes, following the standard classification scheme of the MIT-BIH Arrhythmia Database.

Table I-1 : Distribution ECG's signals types

Class	Likely Description	Number of Images
N	**Normal** (Normal sinus rhythm)	11,508
V	Ventricular ectopic beat	948
S	Supraventricular ectopic beat	226
F	Fusion beat	102
Q	Unknown/Noise	2

- The vast majority of data (~89%) belongs to class "N", indicating a significant class imbalance.
- **Classes "V" and "S"** represent types of arrhythmias and may be clinically relevant for detecting ventricular fibrillation or premature atrial contractions.
- **Class "Q" contains only two samples,** which means it should be treated cautiously or excluded from training due to its extremely limited representation.

I.2.1 The Flowchart

The following flowchart summarizes the main stages of our ECG signal classification pipeline. The process begins with signal preprocessing, followed by transformation into spectrogram images, and concludes with the construction, training, and evaluation of a (CNN) for arrhythmia classification.

Step 1: Preprocessing

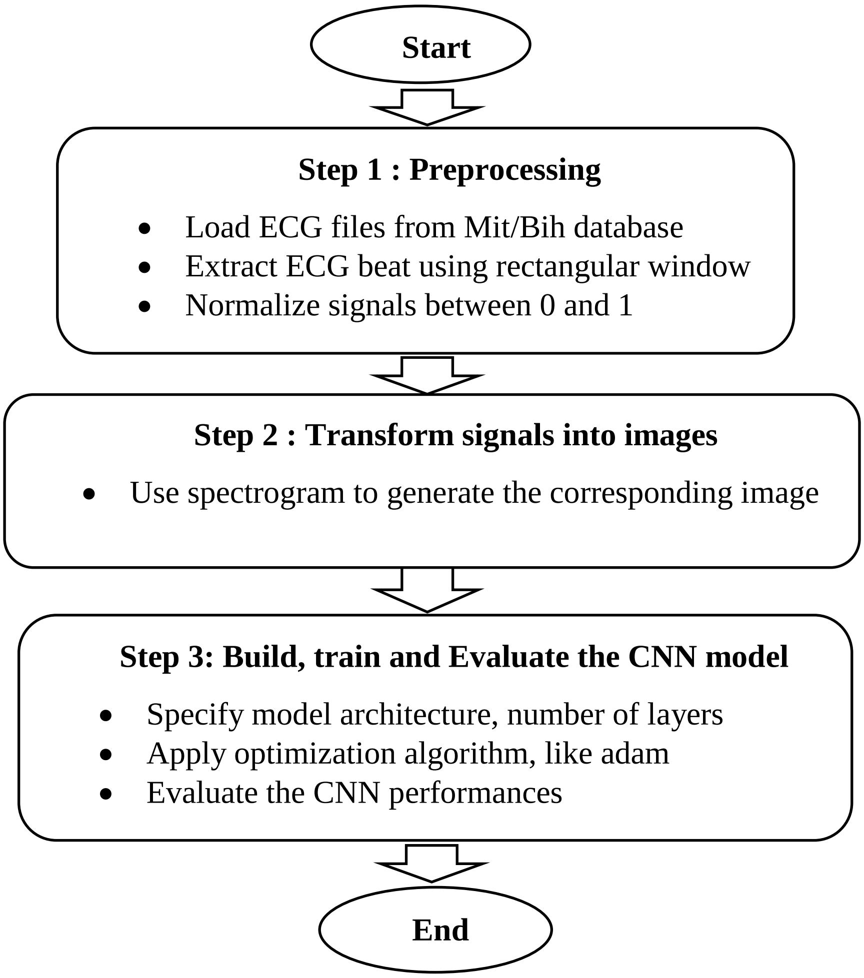
Filter and normalize ECG signals from the MIT-BIH database using MATLAB.

Step 2: Signal-to-Image Transformation

Convert preprocessed signals into spectrogram images labeled into five classes.

Step 3: Build, Train, and Evaluate CNN

Use Google Colab to classify the images with a CNN and evaluate performance.



II. Software and libraries used in the implementation

II.1 Software

In our project we use Google Colab as software tool as it offers the ability of using free GPU, here is a brief overview of Google Colab.

Google Colab (short for Collaboratory) is a Google cloud-based service that allows users to execute Python scripts and conduct machine learning tasks on CPUs, GPUs, and TPUs. It employs the following techniques:

- **Virtual machines:** They are used by Google Colab to execute scripts and carry out calculations on the cloud. These virtual machines come preconfigured with a number of machine learning libraries and frameworks, including TensorFlow and PyTorch.
- **Jupyter Notebooks:** Google Colab is built on Jupyter notebooks, which offer an interactive environment for executing code, viewing data, and creating documentation. Users have the ability to make, modify, and share notebooks.
- **GPU and TPU Acceleration:** Tensor Processing Units (TPUs) and GPU support are available in Google Colab to enhance the speed of machine learning operations. As a result, it becomes possible to efficiently handle larger datasets and train models more rapidly.
- **Integration with Google Drive:** Google Colab is connected with Google Drive, allowing users to save and retrieve data and notes from any location. Users may also access their data by mounting their Google Drive in Colab.
- **Collaboration Features:** Google Colab's collaboration features allow users to collaborate with others by sharing their notebooks and code. Users may also provide comments on code and notebooks and collaborate in real time.
- **Code Snippets:** Google Colab has a large library of code snippets and examples for easily implementing basic machine learning tasks and approaches.

Overall, Google Colab is a robust and adaptable platform for machine learning and data science, with a variety of features and tools to support a wide range of workflows and projects.

II.2 Programming language

Python was the programming language of choice for our project, particularly in Google Colab. Python is widely used in Google Colab and is recognized as a popular high-level programming language. It is extensively employed in various domains, including web development, data analysis, machine learning, artificial intelligence, and more.

II.2.1 Frameworks and libraries

- **Python Standard Library (os) :** Built-in Python module used for interacting with the operating system. It allows file and directory manipulation, path management, and environment variables. Essential for handling file structures programmatically.
- **Pillow (PIL) :** A Python library for image processing, based on the original PIL. It supports opening, editing, and saving many image formats. Often used in machine learning pipelines for preprocessing images.
- **NumPy :** The foundational package for numerical computation in Python. It provides multi-dimensional array objects and tools to work with them efficiently. Crucial for data manipulation and scientific computing.
- **PyTorch :** An open-source machine learning library developed by Facebook AI. It offers dynamic computation graphs, GPU acceleration, and extensive tools for building and training deep learning models.
- **TorchVision :** A utility library built on top of PyTorch for computer vision tasks. It includes common datasets, model architectures, and image transformation tools. Speeds up development in vision-related projects.
- **Scikit-learn :** A robust library for classical machine learning. It provides tools for data splitting, feature selection, classification, regression, and model evaluation. Useful for tasks like splitting datasets into train/test.
- **Matplolib :** Matplotlib is a Python data visualization package that may be used to generate static, animated, and interactive displays.

III. Implementation of CNNs for ECG image classification

III.1 Architecture of the model used

The CNN model we have used, contains different important layers, each with a specific function related to feature extraction and decision making. The model typically starts with convolutional layers that search for local patterns and textures in 2D images. This is followed by ReLU activation functions, which add nonlinearity to the model and enable it to learn more complex representations. Pooling layers, (MaxPooling), are used to reduce the space of feature maps. This helps reduce costs and prevent overfitting. The outputs of the convolutional blocks are then flattened and fed through fully connected layers, which use the found features to classify the data. A dropout layer is included to randomly turn off neurons during training, improving the model's generalization ability. The final layer typically uses SoftMax activation (via CrossEntropyLoss in PyTorch) to generate a probability distribution over the output classes.

Figure V-1 shows the proposed CNN Architecture for Spectrogram Image Classification. This framework aims to achieve efficiency in processing spectral electrocardiogram (ECG) signals by capturing the spatial and frequency components important for accurate classification.

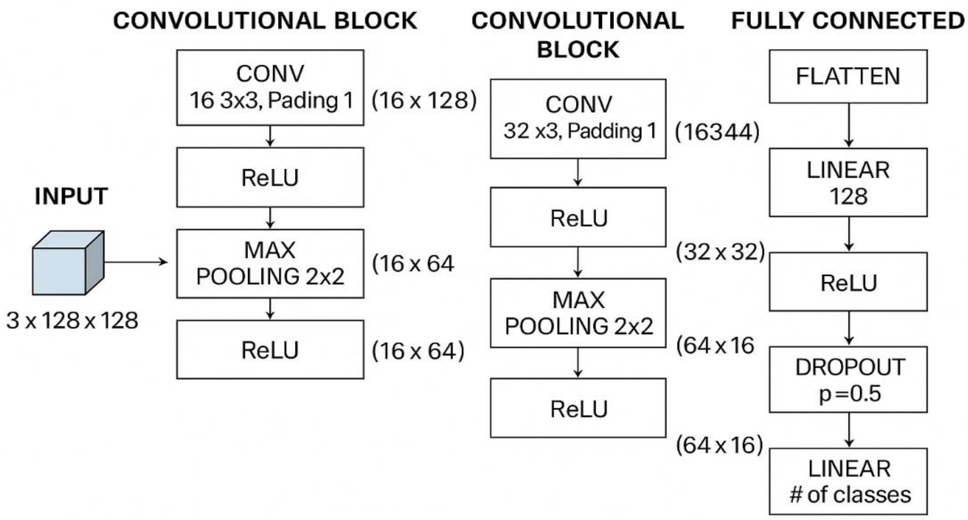


Figure III-1: (CNN) Architecture for Spectrogram Image Classification

III.2 Training Configuration and Optimization Strategy

This section outlines the main training parameters and optimization techniques used to develop the CNN model for ECG signal classification.

Selecting an appropriate loss function is essential for effective model training. In this work, CrossEntropyLoss is used, as it is well-suited for multi-class classification problems. To update the model's weights, the Adam optimizer is employed. Adam offers adaptive learning rates and efficient convergence, making it particularly effective for deep neural networks and dynamic, non-stationary data such as ECG signals.

The learning rate is set to a relatively low value of 0.001 to strike a balance between convergence speed and training stability. The model is trained using mini-batches of size 32, which helps in stabilizing updates and accelerating training.

To study the impact of training duration on model performance, we used a variable number of epochs, allowing us to analyze how different epoch counts influence classification accuracy. This approach provided insight into the optimal number of training cycles needed for effective learning without overfitting.

To further reduce the risk of overfitting, Dropout regularization is applied during training. This technique improves the model's generalization by preventing it from becoming overly reliant on specific neurons.

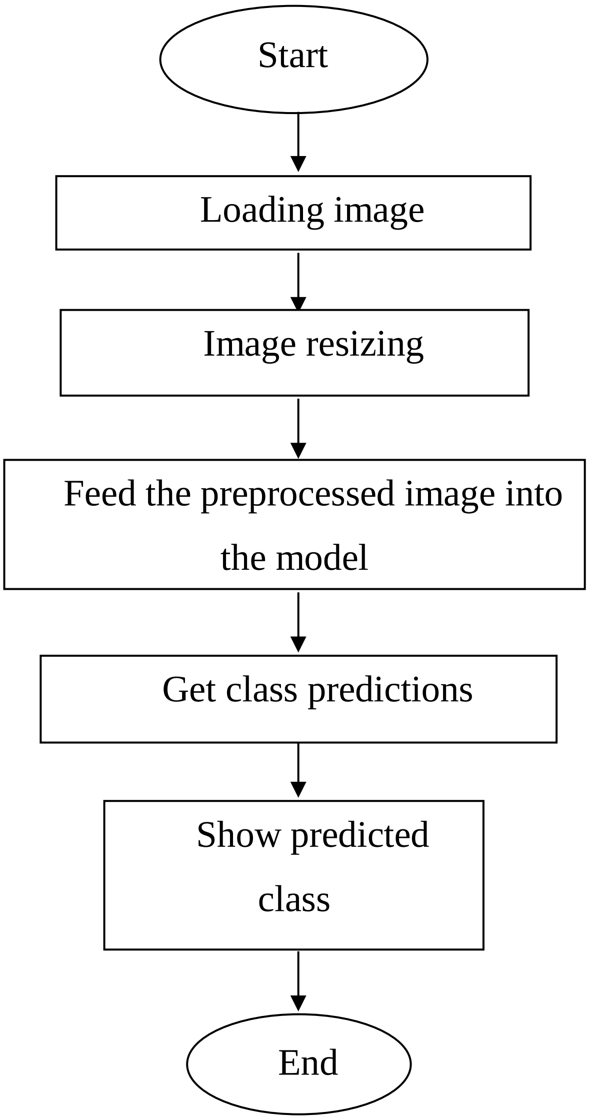
III.3 Training Setups

- **Model**: CNN with 3 convolutional layers + 2 fully connected layers
- **Input**: 128x128 RGB spectrograms (converted from ECG signals)
- **Loss function**: CrossEntropyLoss
- **Optimizer**: Adam (learning rate = 0.001)
- **Dataset split**: 80% train, 10% validation, 10% test

III.4 Model training

The model is trained over several epochs using the 'Adam' optimizer and the 'categorical_crossentropy' loss function. After each epoch, the model is evaluated on the validation data to calculate the loss and accuracy, Accuracy represents the proportion of correct predictions made by the model. These steps allow us to measure the model's performance and adjust its weights to improve results.

III.5 Flowchart of Prediction



After determining the model architecture, we are going to use, we will now conduct several experiments to optimize this architecture. The goal is to explore different configurations, adjust hyperparameters, and test different techniques to improve the model's performance.

IV. Experiment 1: The proposed Baseline Model Evaluation

This experiment aims to establish a baseline model to serve as a reference for subsequent improvements. A simple CNN is trained on 128×128 ECG spectral images for 15 epochs using the default training setup described in Section III.3. No regularization techniques or architectural optimizations (ex: dropout, batch normalization, or data augmentation) are applied.

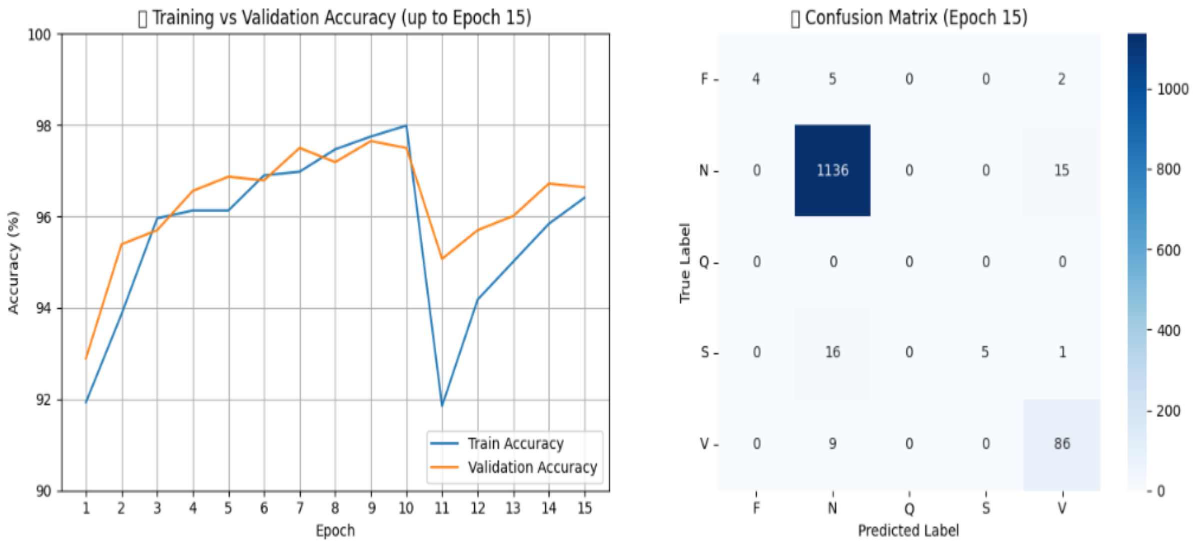


Figure IV-1: training vs validation accuracy and confusion matrix (Epoch 15)

Figure V-1 summarizes the model's performance. The left panel shows the evolution of training and validation accuracy, which reach approximately 96.41% and 96.64%, respectively, by the final epoch. The close alignment of the accuracy curves indicates effective learning and strong generalization, with minimal overfitting. The right panel presents the confusion matrix, used to assess class-level prediction quality on the test set.

The model performs particularly well on the dominant N (normal) class, while showing moderate success on minority classes such as F (fusion beats), S (supraventricular ectopic beats), and V (ventricular ectopic beats). Misclassifications in these classes are mainly attributed to dataset imbalance, which limits the model's exposure to less frequent patterns.

Overall, this baseline experiment confirms that a simple CNN can achieve high accuracy and generalization under default conditions. However, it also highlights the need for targeted improvements—such as regularization and data balancing—to enhance performance on underrepresented classes. These strategies will be explored in the next phase of the study.

V. Experiment 2: Improve the Baseline model

This experiment investigates whether architectural and training enhancements can improve the CNN model's generalization and classification performance—especially for underrepresented ECG classes—while keeping the training duration fixed at 15 epochs, as in the baseline (***Figure V-1***). Enhancements include data augmentation, regularization techniques (Dropout, Batch Normalization), and the use of Focal Loss to mitigate class imbalance. The goal is to evaluate the trade-off between increased model complexity and actual gains in predictive accuracy, particularly for minority classes.

V.1 Training Setup and Modifications

Several key modifications were made to improve the model:

- **Architecture**: Batch Normalization was added after convolutional layers to stabilize learning, and a Dropout layer (rate 0.5) was introduced before the fully connected layers to reduce overfitting.
- **Data Augmentation**: Applied transformations included horizontal flips, ±10° rotations, and brightness/contrast adjustments to increase data diversity and robustness.
- **Loss Function & Optimization**: Focal Loss was used to prioritize harder-to-classify samples and mitigate class imbalance. Adam optimizer (learning rate 0.001) was coupled with a ReduceLROnPlateau scheduler to improve convergence.
- **Dataset**: The same 128×128 spectrogram dataset was used, split into 80% training, 10% validation, and 10% testing. Accuracy was tracked across 15 epochs; performance was evaluated using the test set confusion matrix.

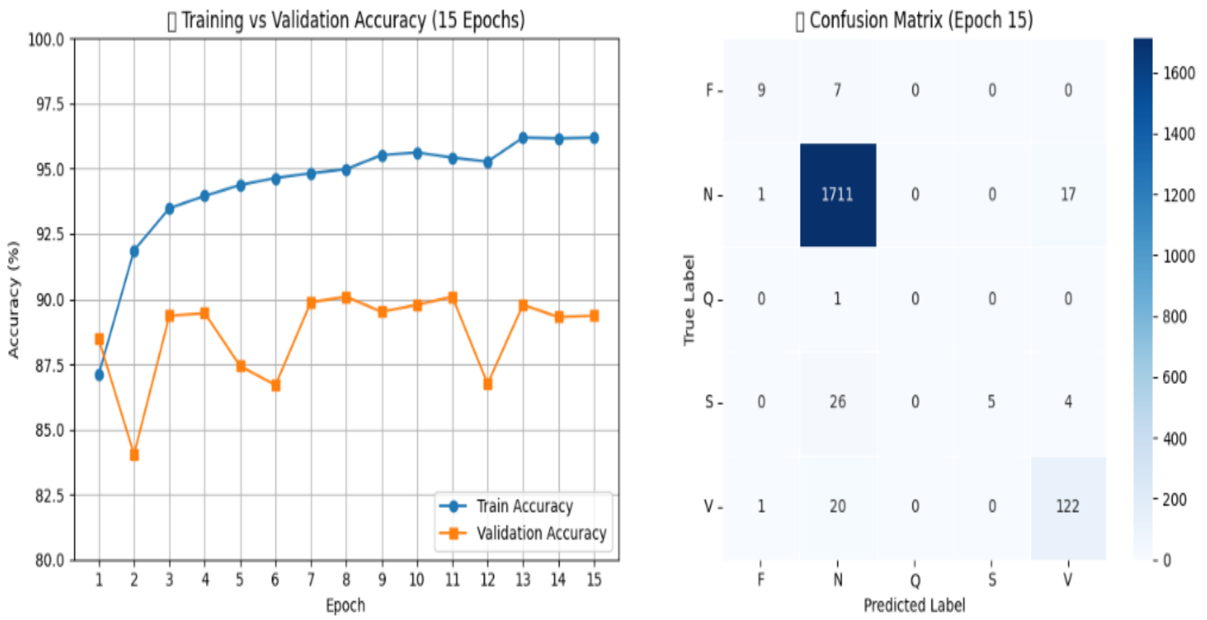


Figure V-1 : after improvement (Epoch 15)

Figure V-2 presents the training and validation accuracy curves along with the confusion matrix:

- **Accuracy**: The model achieved ~96.20% training accuracy, but validation accuracy fluctuated significantly and settled around 89.36%. This divergence signals overfitting, where the model fits the training data well but fails to generalize effectively.
- **Confusion Matrix**: While performance remained strong for the dominant class N (Normal), misclassification increased for minority classes V (ventricular ectopic beats) and S (supraventricular ectopic beats), often being predicted as N. This highlights persistent bias toward the majority class, despite the use of Focal Loss and data augmentation.

The implemented modifications enhanced model complexity and training accuracy but failed to improve validation performance consistently. Validation instability and weak minority class detection suggest overfitting with limited generalization capability.

Despite Dropout and Batch Normalization providing some regularization, the model remained sensitive to class imbalance. Focal Loss and data augmentation proved insufficient to address N class dominance, highlighting the model's inability to effectively learn minority class features.These results demonstrate that increasing complexity does not guarantee better performance, particularly in the presence of class imbalance. Effective generalization requires carefully tuned strategies that align model architecture with dataset characteristics.

VI. Experiment 3: Effect of Training Duration on CNN Performance

This experiment evaluates how increasing the number of training epochs(15, 20, and 25) affects CNN performance in ECG spectrogram classification.

VI.1 Training and Validation Accuracy

As shown in Figures V-3 and V-4, increasing the number of epochs led to higher training and validation accuracy, peaking at approximately 98%. However, the improvement beyond 15 epochs was marginal, with performance plateauing around epoch 20. This suggests that further training offered minimal additional learning benefit.

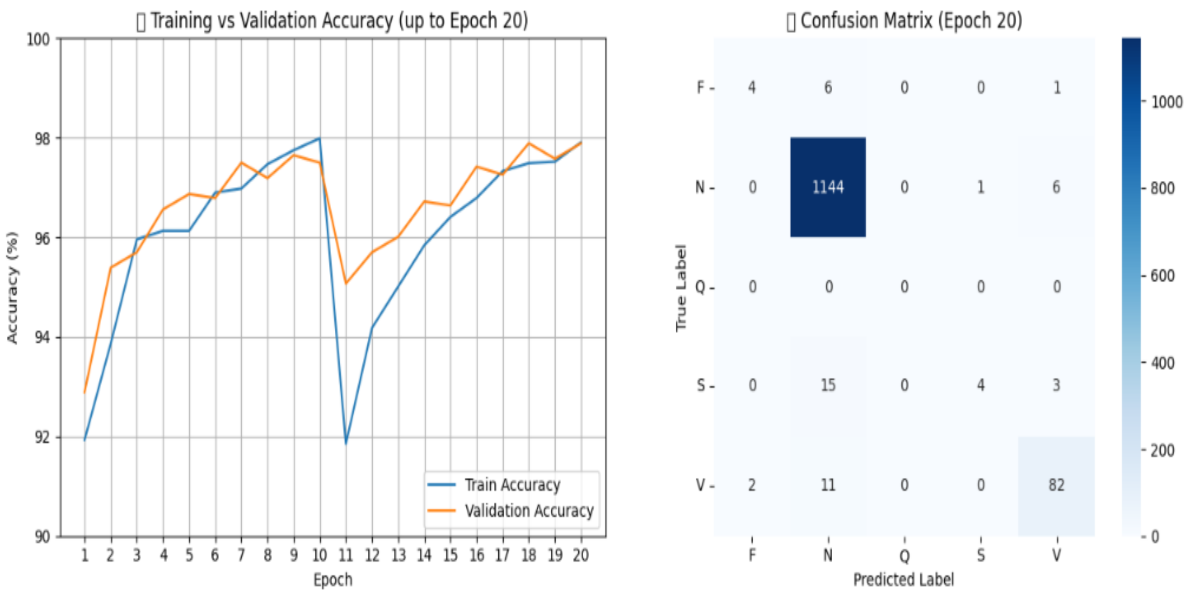


Figure VI-1: training vs validation accuracy and confusion matrix (Epoch 20)

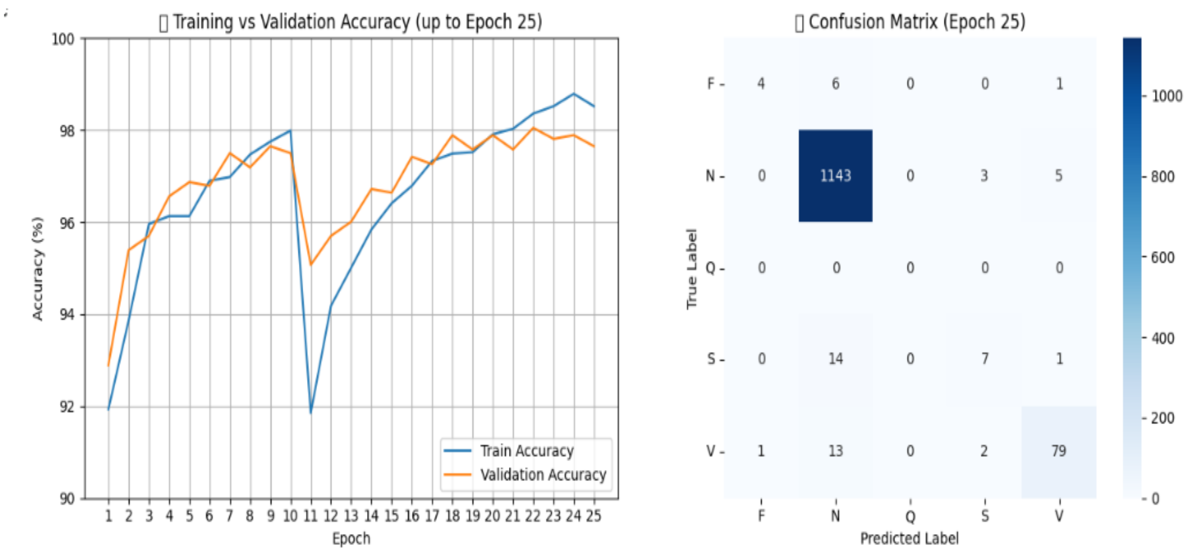


Figure VI-2: training vs validation accuracy and confusion matrix (Epoch 25)

VI.2 Overfitting and Validation Stability

Validation accuracy curves became smoother with longer training, indicating improved convergence. Nevertheless, minor overfitting was observed: validation performance for minority classes (ex: V and S) slightly declined despite continued gains in training accuracy.

VI.3 Class-wise Performance

Confusion matrices revealed persistent misclassification of minority classes, especially V and S, across all epoch settings. In contrast, the majority class N consistently showed high accuracy. This confirms that simply increasing training duration does not resolve class imbalance.

VI.4 Computational Cost

Training time increased significantly with more epochs (from ~418 s (15 epochs) to ~2178 s (25 epochs)) without proportional accuracy gains. This indicates a poor trade-off between time and performance at higher epochs.

Training for more epochs improves model accuracy and stability but with diminishing returns beyond 15–20 epochs. For practical applications, especially in resource-constrained environments, 15 to 20 epochs strike the best balance between efficiency and performance. Further improvements should focus on regularization, model design, and class imbalance handling.

Table VI-1: Accuracy and Execution Time Across Epoch Settings

Number of Epochs	Training Accuracy (%)	Execution Time (s)	Validation Accuracy (%)	Notes on Minority Classes (V, S, F, Q)
15	~96.41	~418.55	~96.64	Moderate performance, slight misclassifications
(Improved CNN) 15	~96.20	~450	~89.36	Overfitting, higher confusion in V and S
20	~98.00	~1400.9	~98.00	Slight improvement, still class imbalance
25	~98.00	~2177.82	~98.00	No significant gain, minor drops in V and N

VII. Experiment 4: Influence of Image Resolution (64×64 vs 128×128)

This experiment investigates how input image resolution affects CNN-based ECG spectrogram classification. Two models were trained using the same CNN architecture, training parameters, and dataset split, differing only in input resolution: 64×64 and 128×128 pixels. The aim is to determine whether higher resolution improves overall accuracy, generalization, and detection of underrepresented ECG classes.

VII.1 Model Trained on 64×64 Images

The first configuration used spectrograms resized to 64×64 pixels. Training was performed using the Adam optimizer (learning rate = 0.001), CrossEntropyLoss, and an 80/10/10 training-validation-test split. No regularization or data augmentation was used.

The model achieved a validation accuracy of ~97%, with stable and smooth training curves. As shown in ***Figure V-7***, the accuracy trend demonstrated good convergence. However, the confusion matrix reveals persistent misclassifications between V (ventricular ectopic beats) and S (supraventricular ectopic beats). The reduced spatial resolution likely led to a loss of critical spectral details needed to distinguish subtle arrhythmic patterns.

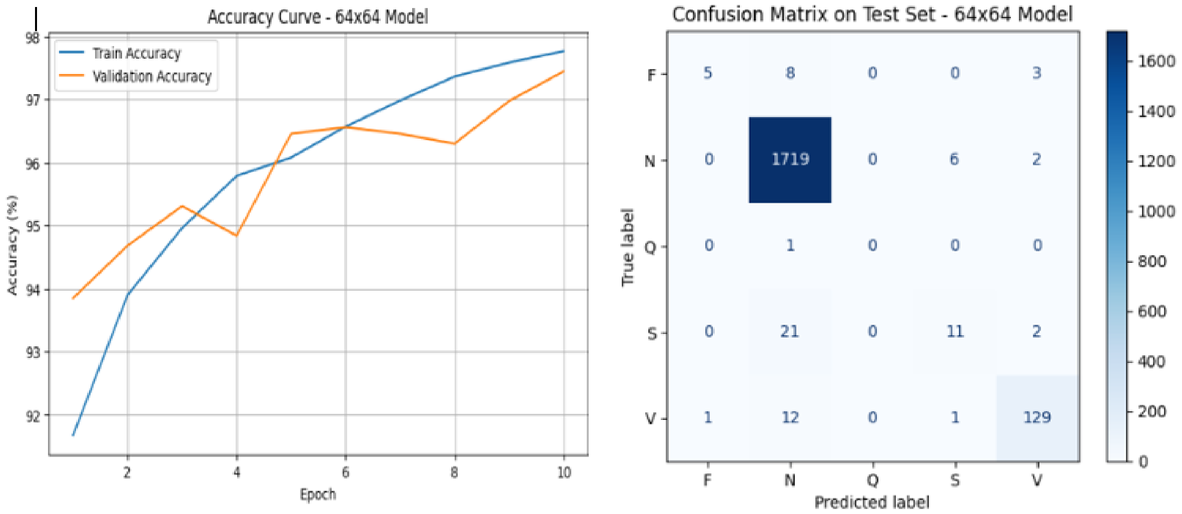


Figure VII-7: Accuracy of number epochs and confusion matrix size 64*64

Additionally, training was efficient, completing in approximately 420 seconds, making this resolution favorable for fast experimentation or use in computationally constrained environments.

VII.2 Model Trained on 128×128 Images

In the second configuration, input spectrograms were resized to 128×128 pixels, with all other training conditions held constant.

This model achieved a slightly higher validation accuracy of ~98%, with smoother training and validation curves and no significant signs of overfitting. As shown in Figure V-8, the confusion matrix indicates reduced misclassifications in minority classes like V and S, confirming the benefit of higher spatial resolution in capturing finer spectral distinctions.

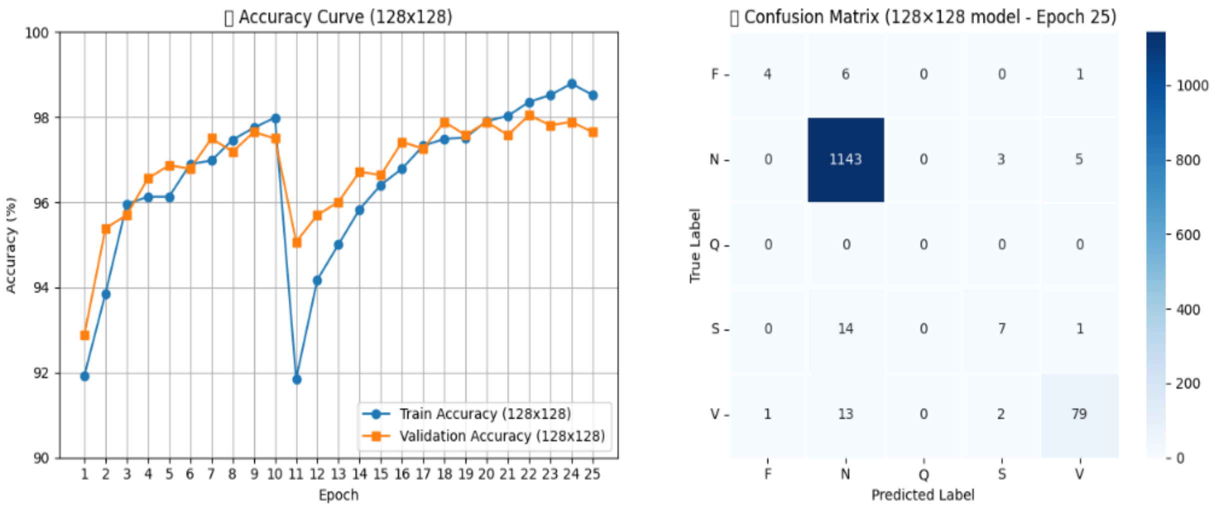


Figure VII-8: Accuracy of number epochs and confusion matrix size 128*128

However, this improvement came at a computational cost: training time increased to ~1755 seconds, more than four times longer than the 64×64 model. This highlights a trade-off between performance and computational demand.

VII.3 Comparative Summary and Recommendation

A comparative summary of the two setups is provided in ***Table VII-1***. The results show that while 128×128 images yield marginally better performance, the computational cost is significantly higher. For resource-limited scenarios, 64×64 remains a practical choice with acceptable accuracy. For diagnostic-critical applications, where small improvements in minority class detection are valuable, the 128×128 model is preferable.

Table VII-1: CNN Performance Comparison Across Image Resolutions

Image Size	Validation Accuracy (%)	Training Time (s)	Notes on Class Performance
64×64	~97	~420.11	Good accuracy, minor confusion in V and S classes
128×128	~98	~1755.40	Improved accuracy, fewer misclassifications in minority classes

VIII. Experiment 5: Impact of CNN Depth on ECG Spectrogram Classification

This experiment evaluates how CNN architectural depth influences the classification of ECG spectrogram images. Specifically, it compares a full CNN model with two convolutional layers to a simplified model with a single convolutional layer. The objective is to assess the trade-offs between model complexity, accuracy, training efficiency, and generalization, especially in the context of limited resources.

VIII.1 Model Architectures

The two CNN variants share the same fully connected layers and training setup but differ in convolutional depth and parameter count.

Table VIII-1: comparison of CNN Architecture

Feature	Full Model	Simplified Model
Convolutional Layers	2 (ex: 16 & 32 filters)	1 (ex: 16 filters)
Depth & Abstraction	Deeper and more abstract	Shallow and basic
Dropout / Regularization	Possible (in full model)	Typically not used
Fully Connected Layers	Present in both	Present in both
Trainable Parameters	Higher	Lower

VIII.2 Performance Comparison

The full model achieved superior performance, especially in complex and minority classes such as F (fusion) and S (supraventricular ectopic beats). Its deeper structure enabled more effective hierarchical feature extraction, resulting in validation accuracy of ~97–98%.

In contrast, the simplified model, while significantly faster and more memory-efficient, struggled to capture subtle ECG variations. It reached validation accuracy of ~92–94%, with higher misclassification rates, particularly in underrepresented classes.

VIII.3 Training Time and Efficiency

- **Simplified Model**: Required substantially less training time (~40–50% faster) and used fewer computational resources, making it well-suited for real-time or embedded applications.
- **Full Model**: Demanded longer training but delivered more reliable performance, especially in imbalanced datasets where robust feature learning is critical.

VIII.4 Summary and Practical Recommendations

This experiment highlights the classic performance–efficiency trade-off:

- Use the Full Model for applications where accuracy, robustness, and minority class performance are critical (ex: diagnostic systems or clinical decision support).
- Use the Simplified Model in scenarios where speed, low power, or limited memory are more important (ex: wearable devices or mobile health monitoring).

Table VIII-2: Summary (Full vs. Simplified CNN Models)

Feature	Full Model (Two Conv Layers)	Simplified Model (One Conv Layer)
Convolutional Layers	2 (16 filters + 32 filters)	1 (16 filters)
Feature Extraction	Deeper and more abstract	Shallow and basic
Model Complexity	Higher	Lower
Training Speed	Slower	Faster
Risk of Overfitting	Higher (especially on small datasets)	Lower
Classification Accuracy	Typically higher (if data is sufficient)	Typically lower
Suitability for Complex Tasks	Better	Less suitable
Memory Usage	Higher	Lower

IX. Conclusion

This chapter presented a comprehensive experimental study on the classification of ECG spectrogram images using convolutional neural networks (CNNs). Through a series of controlled experiments, we investigated the impact of key parameters (namely training duration, image resolution, architectural enhancements, and model depth) on classification performance, generalization, and computational cost.

The baseline model demonstrated that even a simple CNN, trained for 15 epochs, can achieve strong performance with good generalization, particularly for the majority class. However, attempts to improve this baseline using architectural enhancements such as Batch Normalization, Dropout, and Focal Loss showed mixed results. While training accuracy increased, validation accuracy fluctuated, highlighting the risk of overfitting when added complexity is not carefully managed.

Increasing the number of training epochs beyond 15 yielded only marginal improvements in accuracy while significantly increasing training time. Results suggest that training for 15 to 20 epochs offers an optimal trade-off between performance and efficiency.

In terms of input resolution, the use of 128×128 spectrograms led to improved classification accuracy and better handling of minority classes compared to 64×64 inputs. However, this came at the cost of substantially longer training times, emphasizing the importance of aligning model input resolution with the constraints of the deployment environment.

Finally, comparing full and simplified CNN architectures revealed that deeper models achieved better overall accuracy and class-wise performance, particularly on challenging cases. However, simplified models trained faster and required fewer resources, making them better suited for real-time or low-power applications.

In conclusion, the experiments underscore the importance of balancing accuracy, generalization, and resource efficiency when designing CNN-based models for ECG classification. Effective performance depends not only on architectural depth and input quality but also on thoughtful tuning of training strategy, especially in the presence of class imbalance and computational constraints.

General Conclusion

General conclusion

This thesis has demonstrated the feasibility and effectiveness of using (CNNs) for the automatic classification of (ECG) signals by converting one-dimensional time-series data into two-dimensional spectrogram images. The study provides a complete pipeline (from signal preprocessing and transformation to CNN training and evaluation) establishing a practical and scalable approach to cardiac arrhythmia detection.

The work began with a theoretical foundation, covering the clinical importance of ECG signals, the characteristics of arrhythmic beats, and the potential of deep learning (particularly CNNs) in extracting relevant patterns from image-like representations of biomedical data.

A series of experiments were conducted to examine how various factors influence classification performance:

- The baseline CNN, trained on 128×128 spectrograms for 15 epochs, achieved strong overall accuracy, particularly for the majority class (normal beats).
- Architectural enhancements such as Dropout, Batch Normalization, and Focal Loss improved training performance but introduced instability in validation accuracy, especially in the presence of class imbalance.
- Increasing the number of training epochs to 20 and 25 resulted in minimal accuracy gains but significantly increased computational cost, suggesting that 15–20 epochs provide an optimal balance between performance and efficiency.
- Comparing input resolutions revealed that higher-resolution spectrograms (128×128) improved classification of minority classes but at the cost of increased training time.
- Lastly, a comparison between deep and shallow CNN architectures highlighted the trade-off between accuracy and computational efficiency. While the full model achieved better generalization and class-wise accuracy, the simplified model offered faster training and lower memory usage, making it suitable for real-time or embedded applications.

Despite these encouraging results, the study also uncovered persistent challenges (most notably, class imbalance and overfitting) which particularly affected underrepresented arrhythmia types such as fusion and supraventricular beats. These issues limit the model's generalization capability and underscore the need for additional refinements.

To address current limitations and enhance model robustness, future work could explore:

- The integration of deeper or more advanced CNN architectures such as ResNet, DenseNet, or EfficientNet.
- Data augmentation and balancing strategies tailored to rare arrhythmic classes.
- Multi-modal approaches that combine clinical metadata with spectrogram features.
- The use of transfer learning to leverage pre-trained models on larger-scale ECG datasets.
- Validation on larger, more diverse, and real-world datasets to ensure generalizability.

References

References

[1] J. M. Burkitt, "Principles of Electrocardiography," *Principles of Electrocardiography,* vol. 33, no. 2, p. 77, Feb. 2023.

[2] N. E. (. Intervals, "Medscape Reference," Medscape Reference, 6 jan 2023. [Online]. Available: https://emedicine.medscape.com/article/2172196-overview.

[3] R. H. A. a. S. Mori, "Wilhelm His Junior and his bundle," *Journal of Electrocardiology,* vol. 49, no. 4, p. 637–643, 21 jun 2016.

[4] Z. Z. S. W. F. S. T. X. S. S. Y. H. C. L. Y. L. X. S. a. J. H. hi Su, "An ECG Signal Acquisition and Analysis System Based on Machine Learning with Model Fusion," *Sensors,* vol. 23, no. 17, p. 7, 9 sep 2023.

[5] M. A. a. M. S. S. Karpagachelvi, "ECG Feature Extraction Techniques—A Survey Approach," vol. 42, no. 3, p. 5, 9 May May 2010.

[6] J. Mulindi, ",The Instrumentation for Recording ECG Signals,," 08 Jun 2023. [Online]. Available: www.biomedicalinstrumentationsystems.com. [Accessed 7 mai 2022].

[7] AJ, "Ventricular repolarization components on the electrocardiogram: cellular basis and clinical significance,," *Journal of the American College of Cardiology,* vol. 42, no. 4, p. 777–785, 6 Aug 2003.

[8] X. W. a. J. Lian, "Motion artifact reduction in ECG signal processing," *n Proc. IEEE Eng. Med. Biol. Soc,* vol. 45, no. 2, p. 156–162, 8 3 2018.

[9] A. Y. H. e. al, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine,* vol. 25, no. 1, p. 65–69, 12 jan 2019.

[10] G. R. A. e. al, "Frequency content and characteristics of ventricular conduction," *Journal of Electrocardiology,* vol. 5, no. 3, pp. 33-36, 2015.

[11] G. S. W. a. H. J. L. Marriott, "Marriott’s Practical Electrocardiography," vol. 54, no. 9, pp. 45-50, 6 5 2020.

[12] A. P. و. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. Syst. Man Cybern. C,* vol. 40, no. 1, p. 1–12, 10 avril 2010.

[13] J. Escobar-Ordoñez et al., "Comparison of 1D Convolutional Neural Networks for Arrhythmias Classification Using ECG Signals," *Comparison of 1D Convolutional Neural Networks for Arrhythmias Classification Using ECG Signals,* vol. 45, p. 862–876, jan 2024 (20).

[14] B. e. a. Stevens, "Convolutional neural networks: an overview and application in radiology," *What are Convolutional Neural Networks,* vol. 2, no. 42, p. 50, 2004.

[15] K. Lakhdari and N. Saeed, "A new vision of a simple 1D Convolutional Neural Networks (1D-CNN)," *Intelligence-based medicine,* vol. 6, p. 100080, nov 2022(21).

[16] Y. B. a. A. C. I. Goodfellow, Deep Learning. Cambridge, vol. 9, MA,: MIT Press, 2016.

[17] X. e. a. Huang, "Optimization and acceleration of CNNs," *Optimization and acceleration of CNNs,* vol. 7, no. 42, p. 41, may 2011.

[18] D. P. K. a. J. Ba, "Adam: A Method for Stochastic Optimization," *Adam: A Method for Stochastic Optimization,* vol. 6, no. 53-59, p. 41, may 2008.

[19] D. Poulikakos and M. Malik, " Challenges of ECG monitoring and ECG interpretation in dialysis units," *Journal of Electrocardiology,* vol. 49, p. 855–859, 2016 (23).

[20] N. S. A. H. a. S. B. M. Meshkian, "Analysis of 1D NMR spectra with 2D image processing techniques," *Physica Scripta,* vol. 4, pp. 45-55, jan 2025.

[21] M. A. K. K. M. A. H. Z. Klai, "Comparative Analysis of Fourier Transform Variants: Performance, Applications, and Efficiency," *Journal of Computational Analysis and Applications,* vol. 33, pp. 841-852, jan 2024.

[22] L. Stankovic, "On the STFT Inversion Redundancy," 7 may 2016. [Online]. Available: https://doi.org/10.1109/TCSII.2015.2482438.

[23] A. &. K. M. Gautam, "ECG Analysis using Continuous Wavelet Transform (CWT)," *IOSR Journal of Engineering,,* vol. 6, pp. 44-60, may 2012.

[24] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 11, p. 674–693, Jul 1989.

[25] A. F. e. al, "Deep learning for health informatics," *IEEE J. Biomed. Health Inform*, vol. 21, no. 1, p. 4–21, 2018.

[26] I. D. a. L. S. E. Sejdic, "Quantitative Performance Analysis of Scalogram as Instantaneous Frequency Estimator," *IEEE Transactions on Signal Processing,* vol. 56, no. 8, p. 3837–3845, Aug 2008.

[27] A. Ullah, S. Anwar, M. Bilal and R. Mehmood, "Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image," *Representation. Remote Sens*, vol. 4, no. 1685, p. 12, may 2020.

[28] R. Klabunde, "CV Physiology | Electrocardiogram (EKG, ECG)," Cvphysiology, 7 jan 2012. [Online]. Available: https://cvphysiology.com/Arrhythmias/A009.

[29] T. Rechard, "Normal Components of the EKG Waveform," Normal Components of the EKG Waveform, 5 sep 2021. [Online]. Available: www.rnceus.com.

[30] J. P. M. &. K. Y. Harrison, "Characterization of Electromyographic Interference in ECG Analysis," *EEE Transactions on Biomedical Engineering,* vol. 67, no. 9, pp. 1231-1240, 9 4 2020.

[31] "PLUX Biosignals," 2 avril 2010. [Online]. Available: https://www.pluxbiosignals.com.