CLASSIFICATION OF ARRHYTHMIA BY USING DEEP WITH 2D ECG SPECTRAL IMAGE REPRESENTATION

Team ID: PNT2022TMID16055

INTRODUCTION

1.1 Project Overview

This project is the design and implementation of the detection of arrhythmia detection using deep learning models Electrocardiogram (ECG) is a simple non-invasive measure to identify heart-related issues such as irregular heartbeats known as arrhythmias Deep CNN-based algorithm is implemented for training the model, artificial intelligence and machine learning are being utilized in a wide range of healthcare-related applications and datasets, many arrhythmia classifiers using deep learning methods have been proposed in recent years. However, the sizes of the available datasets from which to build and assess machine learning models are often very small and the lack of well-annotated public ECG datasets is evident. In this paper, we propose a deep transfer learning framework that is aimed toto perform classification on a small. The proposed method is to fine-tune general-purpose image classifier ResNet-18 with the MIT-BIH arrhythmia dataset in an accurate MIT-BIH arrhythmia dataset per further investigation many existing deep learning modes failed to avoid data leakage against AAMI recommendations

1.2 Purpose

The purpose of the project is to design and implement of deep learning model deployed for the detection of heart disease and the prediction

2. LITERATURE SURVEY

Amin Ullah Syed Muhammad Anwar, Muhammad Bilal, and Raja Majid Mehmood(2020) The electrocardiogram (ECG) is one of the most extensively employed signals used in the diagnosis and prediction of cardiovascular diseases (CVDs). The ECG signals can capture the heart's rhythmic irregularities, commonly known as arrhythmias. A careful study of ECG signals is crucial for diagnosing patients' acute and chronic heart conditions. In this study, we propose a two-dimensional (2-D) convolutional neural network (CNN) model for the classification of ECG signals into eight classes; namely, normal beat, premature ventricular contraction beat, paced beat, right bundle branch block beat, left bundle branch block beat, atrial premature contraction beat, ventricular flutter wave beat, and ventricular escape beat. The one-dimensional ECG time series signals are transformed into 2-D spectrograms through short-time Fourier transform. The 2-D CNN model consisting of four convolutional layers and four pooling layers is designed for extracting robust features from the

input spectrograms. Our proposed methodology is evaluated on a publicly available MIT-BIH arrhythmia dataset. We achieved a state-of-the-art average classification accuracy of 99.11%, which is better than those of recently reported results in classifying similar types of arrhythmias. The performance is also significant in other indices, including sensitivity and specificity, which indicates the success of the proposed method. Jagdeep Rahul L

Jagdeep Rahul Lakhan Devi Sharma (2022) Atrial fibrillation (Afib) is a heart arrhythmia that is linked to a number of other cardiac-related issues. The incidence of Afib increases with age, causing high risks of stroke. Accurate and reliable detection of Afib remains a challenge and is valuable for clinical diagnosis. This work presents a novel approach for the detection of Afib using both a 1-Delectrocardiogram signal and its time-frequency representation as an image (2D). The signal was preprocessed uilizing a 2-stage median filter and least-square filter followed by normalization before applying it to artificial intelligence-based models for classification. A bi-directional long short-term memory network was trained and tuned to attain high accuracy. Our proposed method shows favorable performances applying ECG to perform classification on a small of 98.85% in the 2-D time-frequency representation. Better classification accuracy and use of short-duration ECG signal compared to existing state-of-art methods make this method suitable for an automated, reliable, and timely detection of Afib.

Fatma Murat, Ozal Yildirim, Muhammed Talo, Ulas Baran Baloglu, Yakup Demir, URajendra Acharya (2020) Deep learning models have become a popular mode linked for several (ECG)data. Investigators have used a variety of deep-learning techniques for this application. Here in, a detailed examination of deep learning methods for ECG arrhythmia detection is provided. Approaches used by investigators are examined, and their contributions to the field are detailed. For this purpose, journal papers have been surveyed according to the methods used. In addition, various deep learning models and experimental studies are described and discussed. A five-class ECG dataset containing 100,022 beats was then utilized for further analysis of deep learning techniques. The constructed models were examined with this dataset, and the results are presented. This, therefore provides information concerning deep learning approaches used for arrhythmia classification and suggestions for further research in this area.

Kurniawan, I Ketut Eddy Purnama, Mpu Hambyah Syah Bagaskara Aji (2021) Cardiovascular disease is part of global death main cause. It is the term for all types of diseases that affect the heart or blood vessels. Heart disease is a type of cardiovascular disease. It can be detected early by examining the arrhythmia presence. Arrhythmia is an abnormal heart rhythm that is commonly diagnosed and evaluated by analyzing electrocardiogram (ECG) signals. In classical techniques, a cardiologist/ clinician used an electrocardiogram (ECG) to monitor the heart rate and rhythm of patients then read the journal activity of patients to diagnose the presence of arrhythmias and to develop appropriate treatment plans. However, The classical techniques take time and effort. The development of arrhythmias diagnosis, toward computational processes, such as arrhythmias detection and classification by using machine learning and deep learning. A convolutional neural network (CNN) is a popular method used to classify arrhythmia. Dataset pre-processing was also considered to achieve the best performance models. MIT-BIH Arrhythmia Database was used as our dataset. Our study used the Efficient N et- which is a type of convolutional neural network to perform the classification of five types of arrhythmias. In pre-processing, the ECG signal was cut each 1 second (360 data), signaling mentation is applied to balance the amount of data in each class, and then the Continues Wavelet Transform (CWT) is employed to transform the ECG signal into a scalogram. The dataset is then distributed into subsets by using the modulo operation to get variants of data in each subset. The colormap is applied to convert scalograms into RGB images. By this scheme, our study achieved superior accuracy than the existing method, with an accuracy rate of 99.97%.

Rui Hu, Jie Chen, Li Zhou (2022) Recently, much effort has been put into solving arrhythmia classification problems with machine learning-based methods. However, inter-heartbeat dependencies have been ignored by many researchers which possess the potential to boost arrhythmia classification performance. To address this problem, this paper proposes a novel transformer-based deep learning neural network, ECG DETR, which performs arrhythmia detection on continuous single-lead ECG segments. The proposed model simultaneously predicts the positions and categories of all the heartbeats within an ECG segment. Therefore, the proposed method is a more compact end-to-end arrhythmia detection algorithm compared with beat-by-beat classification methods as explicit heartbeat segmentation is not required. The performance and generalizability of our proposed scheme are verified on the MIT-BIH arrhythmia database and MIT-BIH atrial fibrillation database. Experiments are carried out on three different arrhythmia detection tasks including 8, 4, and 2 distinct labels respectively using10-fold cross-validation. According to the results, the

suggested method yields comparable performance in contrast with previous works considering both heartbeat segmentation and classification, which achieved an overall accuracy of 99.12%, 99.49%, and 99.23% on the three aforementioned tasks.

Faezeh Nejati Hatamian, Nishant Ravikumar, Sulaiman Vesal(2020) Cardiovascular diseases are the most common cause of mortality worldwide. Detection of atrial fibrillation (AF) in the asymptomatic stage can help prevent strokes. It also improves clinical decision-making through the delivery of suitable treatment such as anticoagulant therapy, in a timely manner. The clinical significance of such early detection of AF in electrocardiogram (ECG) signals has inspired numerous studies in recent years, of which many aims to solve this task by leveraging machine learning algorithms. ECG datasets containing AF samples, however, usually suffer from severe class imbalance, which in fun accounted for, affects the performance of classification algorithms. Data augmentation is a popular solution to tackling this problem. In this study, we investigate the impact of various data augmentation algorithms, e.g., oversampling, Gaussian Mixture Models (GMMs), and Generative Adversarial Networks (GANs), on solving the class imbalance problem. These algorithms are quantitatively and qualitatively evaluated, compared, and discussed in detail. The results show that deep learning-based AF signal classification methods benefit more from data augmentation using GANs and GMMs, than oversampling. Furthermore, the GAN results in circa 3% better AF classification accuracy on average while performing comparably to the GMM in terms of f1-score.

Rashidah Funke Olanrewaju, S. Noorjannah Ibrahim, Ani Liza Asnawi, Hunain Altaf(2021) According to World Health Organization (WHO) report an estimated 17.9 million lives are being lost each year due to cardiovascular diseases (CVDs) and is the top contributor to the death causes. 80% of cardiovascular cases include heart attacks and strokes. This work is an effort to accurately predict common heart diseases such as arrhythmia (ARR) and congestive heart failure (CHF) along with the normal sinus rhythm (NSR) based on the integrated model developed using continuous wavelet transform (CWT) and deep neural networks. The proposed method used in this research analyses the time-frequency features of an electrocardiogram (ECG) signal by first converting the 1D ECG signals to the 2DScalogram images and subsequently the 2D images are being used as an input to the 2Ddeep neural network model-Alex Net. The reason behind converting the ECG signals to 2Dimages is that it is easier to extract deep features from images rather than from the raw data for training purposes in Alex Net. The dataset used for this research was obtained from Massachusetts Institute of Technology-Boston's Beth Israel Hospital (MIT-BIH) arrhythmia database, MIT-BIH normal sinus rhythm database, and Beth Israel Deaconess Medical Center(BIDMC)

congestive heart failure database. In this work, we have identified the best-fit parameters for the Alex Net model that could successfully predict common heart diseases with an accuracy of 98.7%. This work is also being compared with the recent research done in the field of ECG Classification for the detection of heart conditions and proves to be an effective technique for the classification. Ozal Yildirima, MuhammedTaloa, BetulAybUlas BaranBalogluc, GalipAydinbU, Rajendra Acharya (2020) In this study, a deep-transfer learning approach is proposed for the automated diagnosis of diabetes mellitus (DM), using heart rate (HR) signals obtained from electrocardiogram (ECG)data. Recent progress in deep learning has contributed significantly to improvement in the quality of healthcare. In order for deep learning models to perform well, large datasets are required for training. However, a difficulty in the biomedical field is the lack of clinical data with expert annotation. A recent, commonly implemented technique to train deep learning models using small datasets is to transfer the weighting, developed from a large dataset, to the current model. This deep learning transfer strategy is generally employed for two-dimensional signals. Herein, the weighting of models pre-trained using two-dimensional large image data was applied to one-dimensional HR signals. The one-dimensional HR signals were then converted into frequency spectrum images, which were utilized for application to well-known pre-trained models, specifically: AlexNet, VggNet, ResNet, and DenseNet. The DenseNet pre-trained model yielded the highest classification average accuracy of 97.62%, and sensitivity of 100%, to detect DM subjects via HR signal recordings. In the future, we intend to further test this developed model by utilizing additional data along with cloud-based storage to diagnose DM via heart signal analysis. Ahmed S. Eltrass, Mazhar B.Tayel, Abeer I.Ammar (2022) Electrocardiogram (ECG) is an important noninvasive diagnostic method for the interpretation and identification of various kinds of heart diseases. In this work, a new Deep Learning (DL)approach is proposed the for automated identification of Congestive Heart Failure (CHF) and Arrhythmia (ARR) with high accuracy and low computational requirements. This study introduces, for the first time, a new ECG diagnosis algorithm that combines the Convolutional Neural Network (CNN) with the Constant-Q Non-Stationary Gabor Transform (CQ-NSGT). The CQ-NSGT algorithm is investigated to transform the 1-D ECG signal into a 2-D time-frequency representation that will be fed to a pre-trained CNN model, called AlexNet. Extracted features with the AlexNet architecture are used as relevant features to be discriminated by a Multi-Layer Perceptron (MLP) technique into three different cases, namely CHF, ARR, and Normal Sinus Rhythm (NSR). The performance of the proposed CNN with CQ-NSGT is compared versus CNN with Continuous Wavelet Transform (CWT), revealing the effectiveness of the CQ-NSGT

algorithm. The proposed approach is examined with real ECG records, and the experimental results show the superior performance of the proposed approach over other existing techniques in terms of accuracy 98.82%, sensitivity 98.87%, specificity 99.21%, and precision 99.20%. This demonstrates the effectiveness of the proposed system in enhancing ECG diagnosis accuracy.

Bazi, Haikel AlHichri, Naif Alajlan, Farid Melgani, Ronald R Yager (2022) Atrial fibrillation (AF) is the most common heart rhythm disturbance in clinical practice. It often starts with asymptomatic and very short episodes, which are extremely difficult to detect without long-term monitoring of the patient's electrocardiogram (ECG). Although recent portable and wearable devices may become very useful in this context, they often record ECG signals strongly corrupted with noise and artifacts. This impairs automatized ulterior analyses that could only be conducted reliably through a previous stage of automatic identification of high-quality ECG intervals. So far, a variety of techniques for ECG quality assessment have been proposed, but poor performances have been reported on recordings from patients with AF. This work introduces a novel deep learning-based algorithm to robustly identify high-quality ECG segments within the challenging environment of single-lead recordings alternating sinus rhythm, AF episodes, and other rhythms. The method is based on the high learning capability of a convolutional neural network, which has been trained with 2-D images obtained when turning ECG signals into wavelet scalograms. For its validation, almost 100,000 ECG segments from three different databases have been analyzed during 500 learning-testing iterations, thus involving more than 320,000ECGs analyzed in total. The obtained results have revealed a discriminant ability to detect high-quality and discard low-quality ECG excerpts of about 93%, only misclassifying around5% of clean AF segments as noisy ones. In addition, the method has also been able to deal with raw ECG recordings, without requiring signal preprocessing or feature extraction as in previous stages. Consequently, it is particularly suitable for portable and wearable device embedding, facilitating early detection of AF as well as other automatized diagnostic facilities by reliably providing high-quality ECG excerpts to

further processing stages. Han Lia, Xinpei Wanga, Changchun Liu, Peng Lib Yu Jiao (2021) Electrocardiogram (ECG) and phonocardiogram (PCG) are both non-invasive and convenient tools that can capture abnormal heart states caused by coronary artery disease (CAD).However, it is very challenging to detect CAD relying on ECG or PCG alone due to low diagnostic sensitivity. Recently, several studies have attempted to combine ECG and PCG signals for diagnosing heart abnormalities, but only conventional manual features have been used. Considering the strong feature extraction capabilities of deep learning, this paper develops a multi-input convolutional neural network (CNN) framework that integrates time, frequency, and time-frequency domain deep features of ECG and PCG for CAD detection. Simultaneously recorded ECG and PCG signals from 195 subjects are used. The proposed framework consists of 1-D and 2-D CNN models and uses signals, spectrum images, and time-frequency images of ECG and PCG as inputs. The framework combining multi-domain deep features of two-modal signals is very effective in classifying non-CAD and subjects, achieving an accuracy, sensitivity, and specificity of 96.51%, 99.37%, and90.08%, respectively. The comparison with existing studies demonstrates that our method is very competitive in CAD detection. The proposed approach is very promising in assisting real-world CAD diagnosis, especially under general medical conditions. Mehmet Akif Ozdemir, Gizem Dilara Ozdemir & Onan Guren (2021) Two different classification scenarios are conducted on a publicly available paper-based ECG image dataset to reveal the diagnostic capability and performance of the proposed approach. In the first scenario, ECG data labeled as COVID-19 and No-Findings (normal) are classified to evaluate COVID-19 classification ability. According to the results, the proposed approach provides encouraging COVID-19 detection performance with an accuracy of96.20% and F1-Score of 96.30%. In the second scenario, ECG data labeled as Negative(normal, abnormal, and myocardial infarction) and Positive (COVID-19) are classified to evaluate COVID-19 diagnostic ability. The experimental results demonstrated that the proposed

approach provides satisfactory COVID-19 prediction performance with an accuracy of 93.00% and F1-Score of 93.20%. Furthermore, different experimental studies are conducted to evaluate the robustness of the proposed approach. Yong Xiaa, Naren Wulana, Kuanquan Wanga, Henggui Zhangab (2020) Atrial fibrillation (AF) is the most common cardiac arrhythmia. The incidence of AF increases with age, causing high risks of stroke and increased morbidity and mortality. Efficient and accurate diagnosis of AF based on the ECG is valuable in clinical settings and remains challenging. In this paper, we proposed a novel method with high reliability and accuracy for AF detection via deep learning. The proposed method using deep convolutional neural networks shows high sensitivity, specificity, and accuracy, and, therefore, is a valuable tool for AF detection. Jagdeep Rahula, Marpe Sorab, Lakhan Dev, Sharma Vijay, KumarBohatd (2021) Accurate and early detection of cardiac arrhythmia present in an electrocardiogram (ECG)can prevent many premature deaths. Cardiac arrhythmia arises due to the improper conduction of electrical impulses throughout the heart. In this paper, we propose an improved RR interval-based cardiac arrhythmia classification approach. The Discrete Wavelet Transform (DWT) and median filters were used to remove high-frequency noise and baseline wander from the raw ECG. Next, the processed ECG was segmented after the determination of the QRS region. We extracted the primary feature RR interval and other statistical features from the beats to classify the Normal, Premature Ventricular Contraction(PVC), and Premature Atrial Contraction (PAC). The K-Nearest Neighbour (k-NN), Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB), and Random Forest (RF)classifier were utilized for classification. Overall performance of SVM with Gaussian kernel achieved Se % = 99.28, Sp % = 99.63, +P % = 99.28, and Acc % = 99.51, which is better than the other classifiers used in this method. The obtained results of the proposed method are significantly better and more accurate. Xiuzhu Yanga, Xinyue Zhanga, Mengyao Yanga Lin Zhang (2021) Owing to widely available digital ECG data and recent

advances in deep learning techniques, automatic ECG arrhythmia classification based on deep neural networks has gained growing attention. However, existing neural networks are mainly validated on single-lead ECG, not involving the correlation and difference between multiple leads, while multiple leads ECG provides a more complete description of the cardiac activity in different directions. This paper proposes a 12-lead ECG arrhythmia classification method using a cascaded convolutional neural network (CCNN) and expert features. The one-dimensional (1-D) CNN is firstly designed to extract features from each single-lead signal. Subsequently, considering the temporal correlation and spatial variability between multiple leads, features are cascaded as input to two-dimensional (2-D) densely connected ResNet blocks to classify the arrhythmia. Furthermore, features based on expert knowledge are extracted and a random forest is applied to get a classification probability. Results from both CCNN and expert features are combined using the stacking technique as the final classification result. The method has been validated against the first China ECG Intelligence Challenge, obtaining a final score of86.5% for classifying 12-lead ECG data with multiple labels into 9 categories. Kuldeep Singh Chouhan, Jyoti Gajrani, Bhavna Sharma, and Satya Narayan Tazi (2021) As cardiovascular diseases (CVDs) are a serious concern to modern medical science to diagnose at an early stage, it is vital to building a classification model that can effectively reduce mortality rates by treating millions of people in a timely manner. An electrocardiogram (ECG)is a specialized instrument that measures the heart's physiological responses. To accurately diagnose a patient's acute and chronic heart problems, an in-depth examination of these ECG signals is essential. The proposed model consists of a convolutional neural network having three convolutional, two pooling, and two dense layers. The proposed model is trained and evaluated on the MIT-BIH arrhythmia and PTB diagnostic datasets. The classification accuracy is 99.16%, which is higher than state-of-the-art studies on similar arrhythmias. Recall, precision, and F1 score of the proposed model are 96.53%, 95.15%, and99.17%, respectively.

The proposed model can aid doctors explicitly in the detection and classification of arrhythmias. Ahmed S. Eltrass, Mazhar B. Tayel & Abeer I. Ammar (2020) Electrocardiogram (ECG) serves as the gold standard for non-invasive diagnosis of several types of heart disorders. In this study, a novel hybrid approach of the deep neural network combined with linear and nonlinear features extracted from ECG and heart rate variability(HRV) is proposed for ECG multi-class classification. The proposed system enhances the ECG diagnosis performance by combining optimized deep learning features with an effective aggregation of ECG features and HRV measures using chaos theory and fragmentation analysis. The constant-Q non-stationary Gabor transform technique is employed to convert the 1-D ECG signal into a 2-D image which is sent to a pre-trained convolutional neural network structure, called AlexNet. The pair-wise feature proximity algorithm is employed to select the optimal features from the AlexNet output feature vector to be concatenated with the ECG and HRV measures. The concatenated features are sent to different types of classifiers to distinguish three distinct subjects, namely congestive heart failure, arrhythmia, and normal sinus rhythm (NSR). The results reveal that the linear discriminant analysis classifier has the highest accuracy compared to the other classifiers. The proposed system is investigated with real ECG data taken from well-known databases, and the experimental results show that the proposed diagnosis system outperforms other recent state-of-the-art systems in terms of accuracy of 98.75%, specificity of 99.00%, a sensitivity of 98.18%, and computational time 0.15 s. This demonstrates that the proposed system cato assist cardiologists in enhancing the accuracy of ECG diagnosis in real-time clinical settings. HassanSerhal, NassibAbdallah, Jean-Marie MarionaPierre, Chauvet MohamadOueidatb Anne Humeau-Heurtiera (2021) Atrial fibrillation (AF) is the most common supraventricular cardiac arrhythmia, resulting in high mortality rates among affected patients. AF occurs as episodes coming from irregular excitations of the ventricles that affect the functionality of the heart and can increase the risk of stroke and heart attack. Early and

automatic prediction, detection, and classification of AF are important steps for effective treatment. For this reason, it is the subject of intensive research in both medicine and engineering fields. The latter research focuses on three axes: prediction, classification, and detection. Knowing that AF is often asymptomatic and that its episodes are often very short, its automatic early detection is a very complicated but clinically important task to improve AF treatment and reduce the risks for the patients. This article is a review of publications from the past decade, focusing on AF episode prediction, detection, and classification using wavelets and artificial intelligence (AI).Forty-five articles were selected of which five are about AF in general, four articles compare accuracy, recall, and precision between Fourier transform (FT) and wavelets transform (WT), and thirty-six are about detection, classification, and prediction of AF with WT: 15 are based on deep learning (DL) and 21 on conventional machine learning (ML). Of the thirty-six studies, thirty were published after 2015, confirming that this particular research area is very important and has great potential for future research. Parul Madan, Vijay Singh, Devesh Pratap Singh, Manoj Diwakar, Bhaskar Pant (2022) Arrhythmias are defined as irregularities in the heartbeat rhythm, which may infrequently occur in a human's life. These arrhythmias may cause potentially fatal complications, which may lead to an immediate risk to life. Thus, the detection and classification of arrhythmias is a pertinent issue for cardiac diagnosis. (1) Background: To capture these sporadic events, an electrocardiogram (ECG), a register containing the heart's electrical function, is considered the gold standard. However, since ECG carries a vast amount of information, it becomes very complex and challenging to extract the relevant information from visual analysis. As a result, designing an efficient (automated) system to analyze the enormous quantity of data possessed by ECG is critical. (2) Method: This paper proposes a hybrid deep learning-based approach to automate the detection and classification process. This paper makes two-fold contributions. First, 1D ECG signals are translated into 2D Scalogram images to automate noise filtering and feature extraction. Then,

based on experimental evidence, by combining two learning models, namely 2D convolutional neural network (CNN) and the Long Short-Term Memory (LSTM) network, a hybrid model called 2D-CNN-LSTM is proposed.(3) Result: To evaluate the efficacy of the proposed 2D-CNN-LSTM approach, we conducted a rigorous experimental study using the widely adopted MIT–BIH arrhythmia database. The obtained results show that the proposed approach provides ≈98.7%, 99%, and 99% accuracy for Cardiac Arrhythmias (ARR), Congestive Heart Failure (CHF), and Normal Sinus Rhythm(NSR), respectively. Moreover, it provides an average sensitivity of the proposed model of98.33% and a specificity value of 98.35%, for all three arrhythmias. (4) Conclusions: For the classification of arrhythmias, a robust approach has been introduced where 2D scalogram images of ECG signals are trained over the CNN-LSTM model. The results obtained are better as compared to the other existing techniques and will greatly reduce the amount of intervention required by doctors. For future work, the proposed method can be applied over some live ECG signals and Bi-LSTM can be applied instead of LSTM. Mahwish Naz, Jamal Hussain Shah, Muhammad Attique Khan, Muhammad Sharif, Mudassar Raza1, Robertas Damaševičius (2021) Provocative heart disease is related to ventricular arrhythmias (VA). Ventricular tachyarrhythmia is an irregular and fast heart rhythm that emerges from inappropriate electrical impulses in the ventricles of the heart. Different types of arrhythmias are associated with different patterns, which can be identified. An electrocardiogram (ECG) is the major analytical tool used to interpret and record ECG signals. ECG signals are non-linear and difficult to interpret and analyze. We propose a new deep-learning approach for the detection of VA. Initially, the ECG signals are transformed into images that have not been done before. Later, these images are normalized and utilized to train the AlexNet, VGG-16and Inception-v3 deep learning models. Transfer learning is performed to train a model and extract the deep features from different output layers. After that, the features are fused by a concatenation approach, and the best features are selected using a

heuristic entropy calculation approach. Finally, supervised learning classifiers are utilized for final feature classification. The results are evaluated on the MIT-BIH dataset and achieved an accuracy of97.6% (using Cubic Support Vector Machine as a final stage classifier).

2.1 Existing Problem

In existing system machine learning based algorithm has been implemented in existing system machine learning algorithm has been implemented in the existing system In the literature, the ECG analysis generally consists of the following steps: 1) ECG signal preprocessing and noise attenuation, 2) heartbeat segmentation, 3) feature extraction, and 4) learning/classification

Machine learning models are widely used for arrhythmia classification in the literature [2,3,5,7,8,13,14,15]. Mi Hye Song et al. proposed a support vector machine-based classifier with reduced features derived by linear discriminant analysis [5]. Inspired by the success of the Hidden Markov Model (HMM) in modeling speech waveforms for automatic speech recognition, D A Coast et al. applied HMM method in ECG arrhythmia analysis. The model can combine the temporal information and statistical knowledge of the ECG signal in one single parametric model [15]. Awni Y. Hannun et al. proposed an end-to-end deep learning approach that directly takes raw ECG signal as input and produces classifications without feature engineering or feature selection [8]. Mousavi, Sajad et al. proposed an automatic ECG-based heartbeat classification approach by utilizing a sequence-to-sequence deep learning method to automatically extract temporal and statistical features of the ECG signals

Our work differs from the studies in 2-fold: 1) it leverages the Short-term Fourier Transform (STFT) to convert 1D ECG signal into 2D time-frequency domain data. Therefore, it is feasible to apply a pre-trained 2D Convolution Neural Network in arrhythmia analysis; 2) it is evaluated using MIT-BIH dataset with “inter-patient” training/testing split paradigm detailed

2.2 References

1. Kachuee, Mohammad, Shayan Fazeli, and Majid Sarrafzadeh. "Ecg heartbeat classification: A deep transferable representation." 2018 IEEE international conference on healthcare informatics (ICHI). IEEE, 2018

2.S. Zhang, W. Wang, J. Ford, and F. Makedon, “Learning from incomplete ratings using non-negative matrix factorization,” in Proc. 6th SIAM Int. Conf. Data Mining, 2006, pp. 549–553.

3.T. Hofmann and J. Puzicha, “Latent class models for collaborative filtering,” in Proc. 6th Int. Joint Conf. Artif. Intell., 1999, pp. 688–693.

4.B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in Proc. 10th Int. World Wide Web Conf., 2001, pp. 285–295

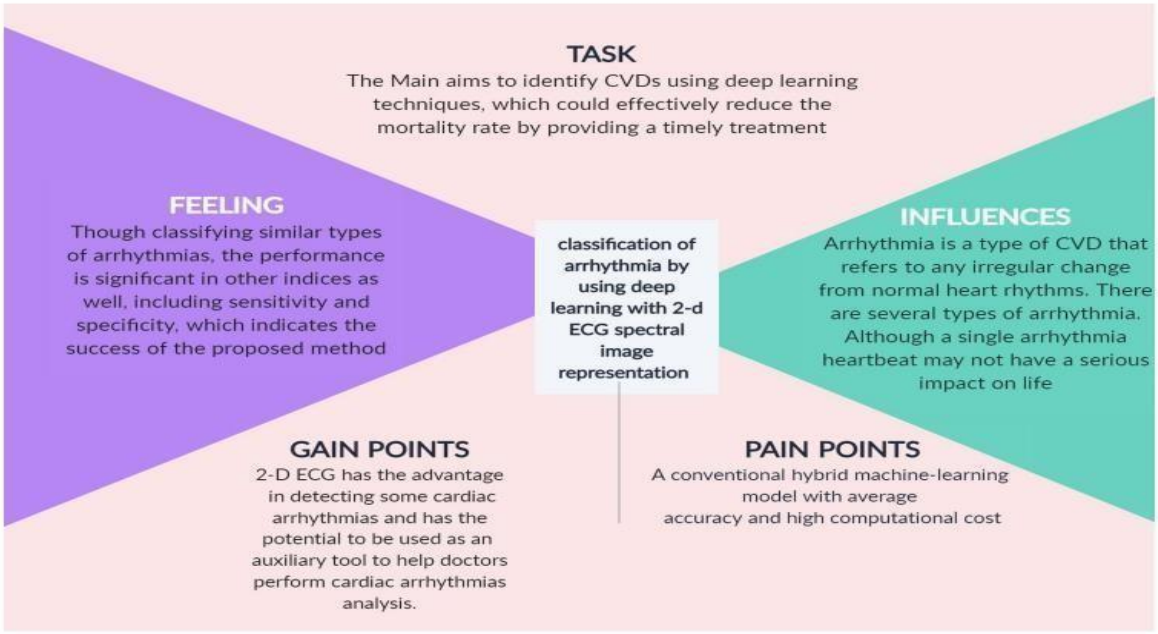
5.T. George and S. Merugu, “A scalable collaborative filtering framework based on co-clustering,” in Proc. 5th IEEE Int. Conf. Data Mining, 2005, pp. 625–628.

2.3 Problem Statement Definition

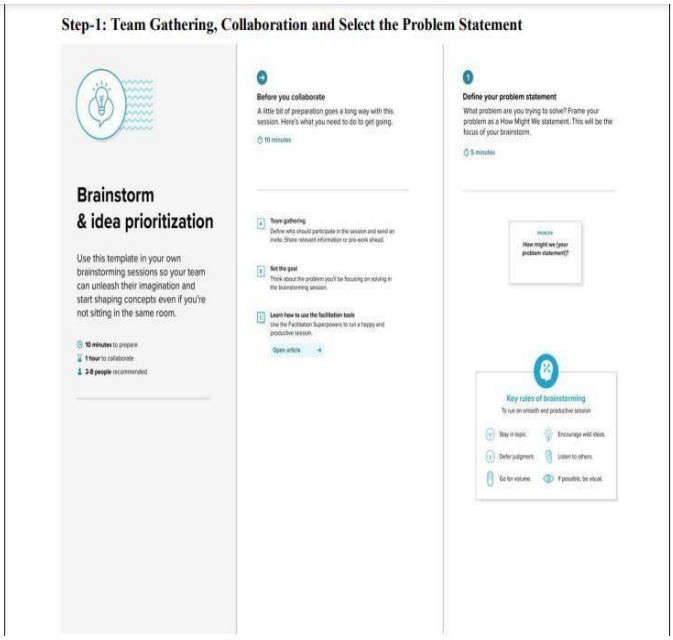
- Cardiologists by using various values which occurred during the ECG recording can decide whether the heartbeat is normal or not. Since observation of these values is not always clear, the existence of an automatic ECG detection system is required
- Luz, Eduardo José da S., et al. "ECG-based heartbeat classification for arrhythmia detection: A survey." Computer methods and programs in biomedicine 127 (2016): 144-164
- Romdhane, Taissir Fekih, and Mohamed Atri Pr. "Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss." Computers in Biology and Medicine 123 (2020): 103866.

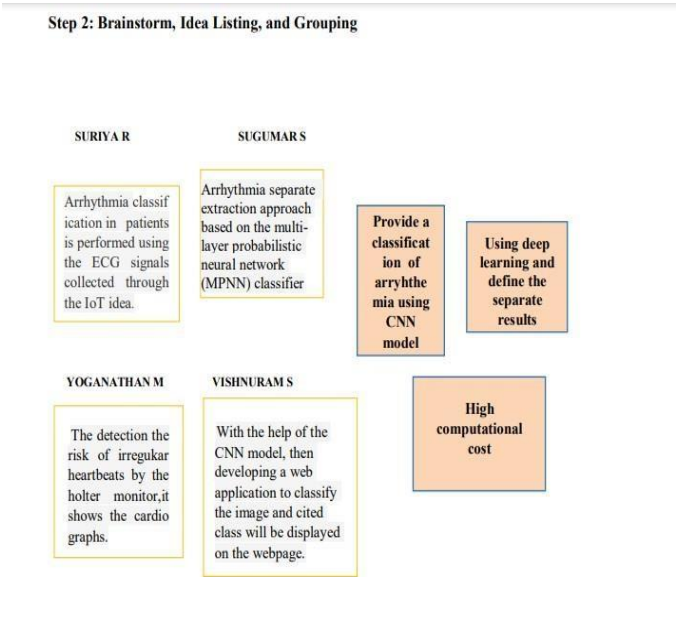
3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

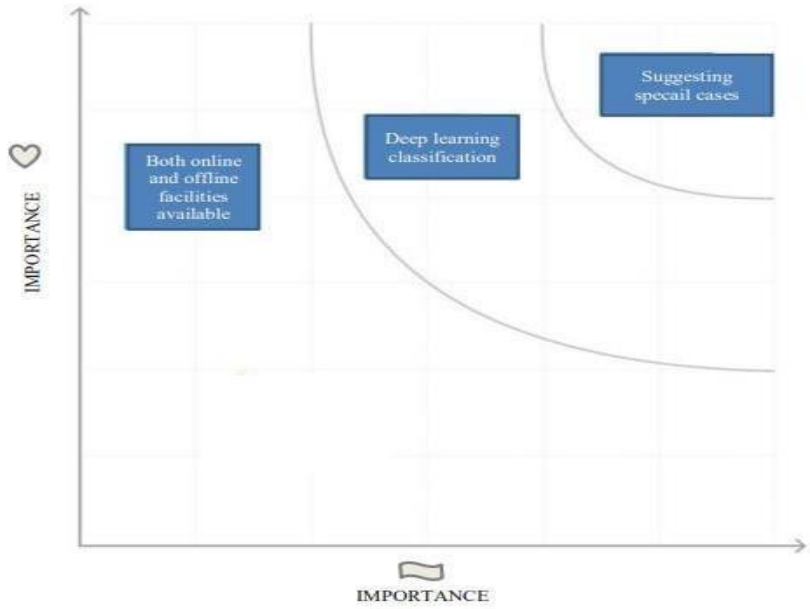


3.2 Ideation and Brainstorming





Step 3: Idea Prioritization

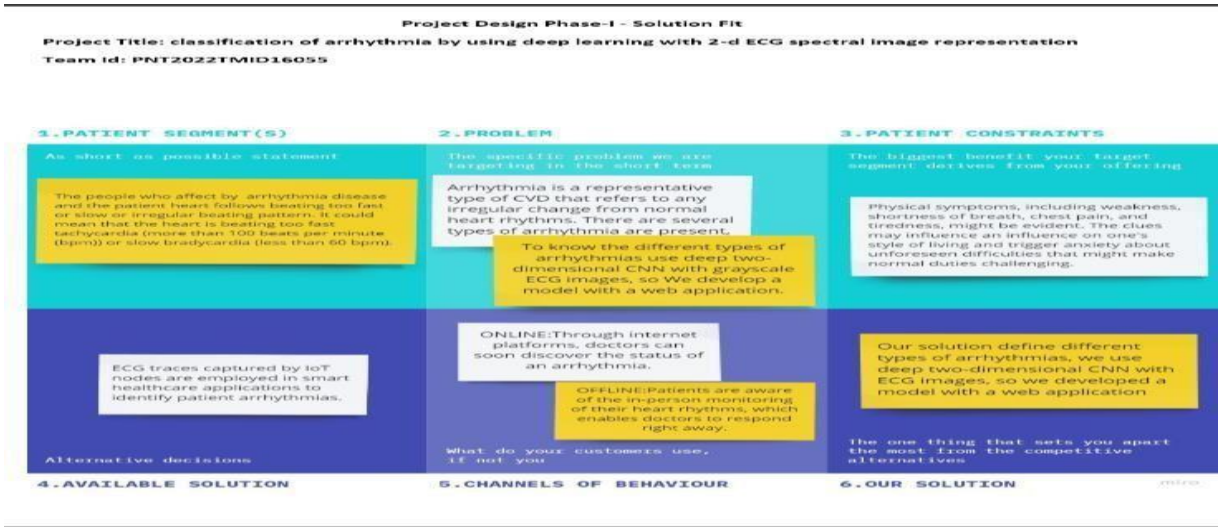


3.3 Proposed Solutions

Deep learning based using of train the image The MIT-BIH database, an ECG database provided by the Massachusetts Institute of Technology and based on international standards and annotated information by multiple experts (Moody and Mark, 2001) is used in this study. The MIT-BIH database has been frequently used by the academic community in research for the detection and classification of arrhythmic heartbeats. The MIT-BIH database contains 48 ECG recordings, each recording time is 30 min, the sampling frequency is 360 Hz, and each ECG record is composed of two leads. MIT-BIH database can make adjustments and corrections based on the information annotated by experts and optimization algorithms. Furthermore, it learns from existing solutions for self-optimization.

This paper proposes a novel deep-learning approach to identify arrhythmias in ECG signals. The proposed approach identifies arrhythmia classes using a Convolutional Neural Network (CNN) trained by two-dimensional (2D) ECG beat images. Firstly, ECG signals, which consist of 5 different arrhythmias, are segmented into heartbeats which are transformed into 2D grayscale images. Afterward, the images are used as input for training a new CNN architecture to classify heartbeats.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form , Registration through Gmail
FR-2	User Confirmation	Confirmation via Email , Confirmation via OTP
FR-3	Get User Input	Upload image as jpeg , Upload image as png
FR-4	Save Image	Images are saved in the uploads folder
FR-5	Chat with Doctor	Consult with Doctor
FR-6	Report Generation	Get complete Report

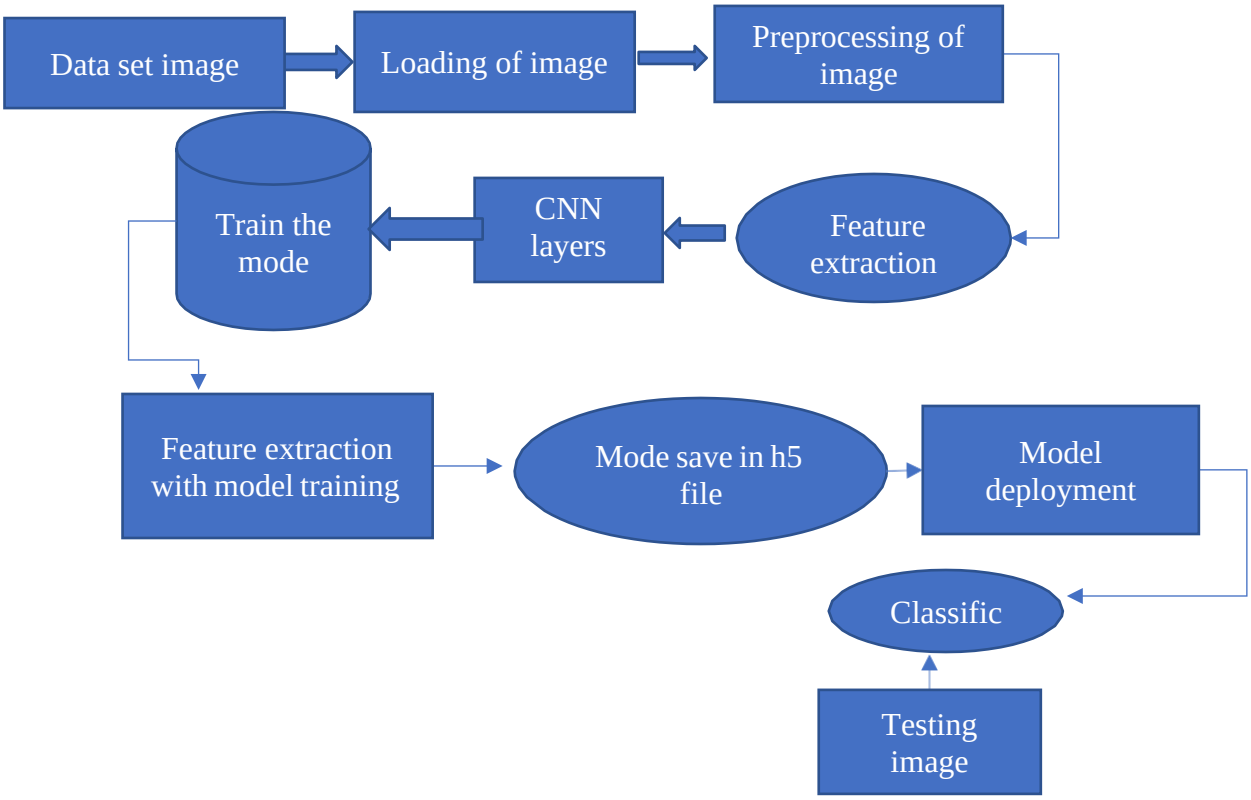
4.2 Non -functional requirements

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

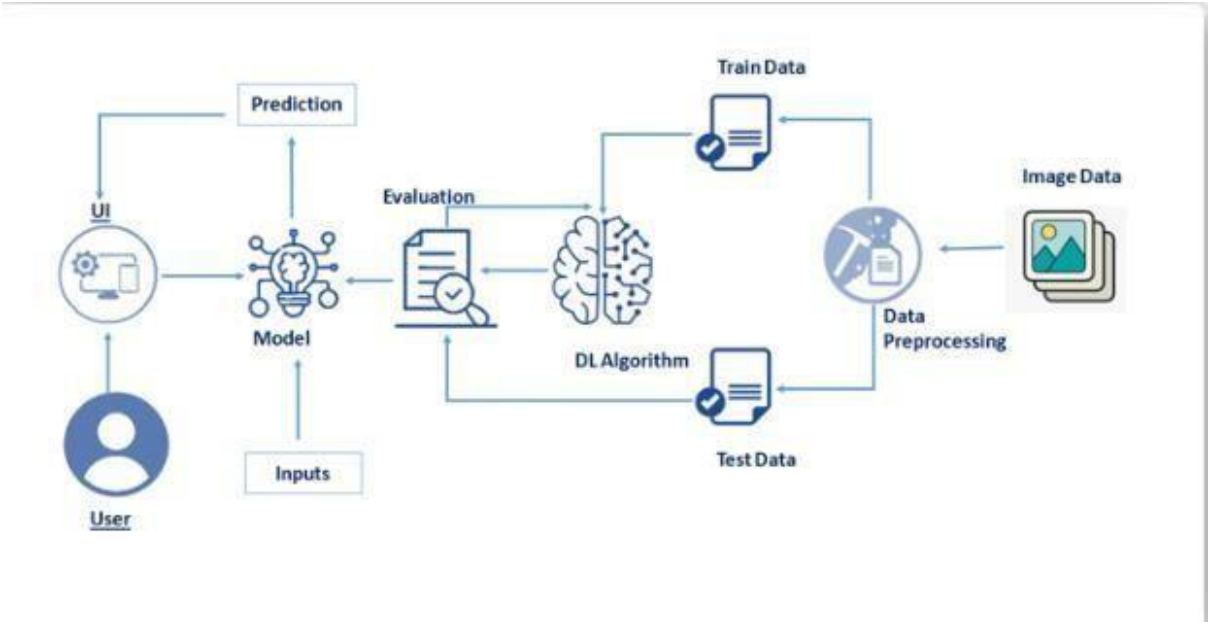
FRNo.	Non-Functional Requirement	Description
NFR-1	**Usability**	Classification of Arrhythmia with the help of AI.
NFR-2	**Security**	User's data cannot be accessed by unauthorized people.
NFR-3	**Reliability**	The system performs without failure.
NFR 4	**Performance**	High accuracy.
NFR 5	**Availability**	Anyone who is authorized.
NFR-6	**Scalability**	Does not affect the performance even though.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solutions and Technical Architecture



6.PROJECT PLANNING AND SCHEDULING

Milestones	Activities
Project development phase	Delivery of sprint- 1,2,3,4
Create and configure and IBM cloud services	Create IBM Watson
Create and access deep learning	Create v1 to interact with app deploy
	Create IBM and connect with python
Create & database in cloud and DB	Launch the cloudant DB and Create database
Develop the python flask	Install the python software
	Develop python code
Create the web application	Develop the web application

Milestones	Activities	Description
Ideation phase	literature	Literature survey on the selected project & information gathering
	Empathy Map	Prepare empathy map to capture the user pains & gains, prepare list of problem statement
	Ideation	Organizing the brainstorming session and priorities the top 3 ideas based on feasibility & importance

7.CODING

```
import numpy as np
import pandas  as  pd
from pathlib import Path
import os.path

import tensorflow as tf



dir = Path('../input/ecg-image-data/ECG_Image_data/train')

filepaths = list(dir.glob(r'**/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], file
paths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

dataframe = pd.concat([filepaths , labels] , axis=1)
dataframe
dataframe['Label'].value_counts()
dataframe['Label'].unique()
```

```
samples = []
for category in ['N', 'M', 'Q', 'S', 'V']:
    category_slice = dataframe.query("Label == @category")
    samples.append(category_slice.sample(2223, random_state=1))

dataframe_train = pd.concat(samples, axis=0).sample(frac=1.0, random_st
ate=1).reset_index(drop=True)
dataframe_train['Label'].value_counts()
dir = Path('../input/ecg-image-data/ECG_Image_data/train')

filepaths = list(dir.glob(r'F/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], file
paths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

F = pd.concat([filepaths , labels] , axis=1)
F
%%time
dir = Path('../input/ecg-image-data/ECG_Image_data/test')

filepaths = list(dir.glob(r'**/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], file
paths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

dataframe_test = pd.concat([filepaths , labels] , axis=1)
dataframe_test
```

Preprocessing of code

```
size=64
color_mode='grayscale'
batch_size=32
train_images = train_generator.flow_from_dataframe(
    dataframe=dataframe_train,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=True,
    seed=42,
    subset='training'
)
```

```
val_images = train_generator.flow_from_dataframe(
    dataframe=dataframe_train,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=True,
    seed=42,
    subset='validation'
)

test_images = test_generator.flow_from_dataframe(
    dataframe=dataframe_test,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=False
)
```

Model summary

```
size=64
color_mode='grayscale'
batch_size=32
train_images = train_generator.flow_from_dataframe(
    dataframe=dataframe_train,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=True,
    seed=42,
    subset='training'
)

val_images = train_generator.flow_from_dataframe(
    dataframe=dataframe_train,
    x_col='Filepath',
    y_col='Label',
```

```
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=True,
    seed=42,
    subset='validation'
)

test_images = test_generator.flow_from_dataframe(
    dataframe=dataframe_test,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=False
)

import keras
checkpoint = keras.callbacks.ModelCheckpoint(
    filepath='best_model.h5',
    save_weights_only=False,
    monitor='val_accuracy',
    mode='max',
    save_best_only=True,
    verbose=1)


model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)


import matplotlib.pyplot as plt

plt.plot(result.history['loss'])
plt.plot(result.history['val_loss'])
plt.legend(['Training', 'Validation'])
plt.title('Training and Validation losses')
plt.xlabel('epoch')
```

Html coding

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Aroma Shop - Home</title>
        <link rel="icon" href="static/img/Fevicon.png" type="image/png">
  <link rel="stylesheet" href="static/vendors/bootstrap/bootstrap.min.css">
  <link rel="stylesheet" href="static/vendors/fontawesome/css/all.min.css">
        <link rel="stylesheet" href="static/vendors/themify-icons/themify-icons.css">
  <link rel="stylesheet" href="static/vendors/nice-select/nice-select.css">
  <link rel="stylesheet" href="static/vendors/owl-carousel/owl.theme.default.min.css">
  <link rel="stylesheet" href="static/vendors/owl-carousel/owl.carousel.min.css">

  <link rel="stylesheet" href="static/css/style.css">
</head>
<body>
  <!--================ Start Header Menu Area ================-->
        <header class="header_area">
    <div class="main_menu">
      <nav class="navbar navbar-expand-lg navbar-light">
        <div class="container">
          <a class="navbar-brand logo_h" href="index.html"><img src="static/img/logo.png"
alt=""></a>
         <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
          aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
         </button>
         <div class="collapse navbar-collapse offset" id="navbarSupportedContent">
          <ul class="nav navbar-nav menu_nav ml-auto mr-auto">
           <li class="nav-item active"><a class="nav-link" href="index.html">Home</a></li>
           <li class="nav-item submenu dropdown">
            <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
              aria-expanded="false">Shop</a>
             <ul class="dropdown-menu">
              <li class="nav-item"><a class="nav-link" href="category.html">Shop
Category</a></li>
              <li class="nav-item"><a class="nav-link" href="single-product.html">Product
Details</a></li>
              <li class="nav-item"><a class="nav-link" href="checkout.html">Product
Checkout</a></li>
              <li class="nav-item"><a class="nav-link"
href="confirmation.html">Confirmation</a></li>
              <li class="nav-item"><a class="nav-link" href="cart.html">Shopping
Cart</a></li>
```

```
            </ul>
                                                    </li>
            <li class="nav-item submenu dropdown">
              <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
                aria-expanded="false">Blog</a>
              <ul class="dropdown-menu">
                <li class="nav-item"><a class="nav-link" href="blog.html">Blog</a></li>
                <li class="nav-item"><a class="nav-link" href="single-blog.html">Blog
Details</a></li>
              </ul>
                                                    </li>
                                                    <li class="nav-item submenu
dropdown">
              <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
                aria-expanded="false">Pages</a>
              <ul class="dropdown-menu">
                <li class="nav-item"><a class="nav-link" href="sign.html">Login</a></li>
                <li class="nav-item"><a class="nav-link" href="/register">Register</a></li>
                <li class="nav-item"><a class="nav-link" href="tracking-
order.html">Tracking</a></li>
              </ul>
            </li>
            <li class="nav-item"><a class="nav-link" href="contact.html">Contact</a></li>
          </ul>

          <ul class="nav-shop">
            <li class="nav-item"><button><i class="ti-search"></i></button></li>
            <li class="nav-item"><button><i class="ti-shopping-cart"></i><span class="nav-
shop_circle">3</span></button> </li>
            <li class="nav-item"><a class="button button-header" href="#">Buy Now</a></li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
 </header>
        <!--================ End Header Menu Area =================-->

 <main class="site-main">

  <!--================ Hero banner start =================-->
  <section class="hero-banner">
   <div class="container">
    <div class="row no-gutters align-items-center pt-60px">
     <div class="col-5 d-none d-sm-block">
      <div class="hero-banner_img">
       <img class="img-fluid" src="static/img/home/hero-banner.png" alt="">
      </div>
```

```
        </div>
        <div class="col-sm-7 col-lg-6 offset-lg-1 pl-4 pl-md-5 pl-lg-0">
          <div class="hero-banner_content">
            <h4>Shop is fun</h4>
            <h1>Browse Our Premium Product</h1>
            <p>Us which over of signs divide dominion deep fill bring they're meat beho upon own earth without morning over third. Their male dry. They are great appear whose land fly grass.</p>
            <a class="button button-hero" href="#">Browse Now</a>
          </div>
        </div>
      </div>
    </div>
  </section>
  <!--================ Hero banner start =================-->

  <!--================ Hero Carousel start =================-->
  <section class="section-margin mt-0">
    <div class="owl-carousel owl-theme hero-carousel">
      <div class="hero-carousel_slide">
        <img src="img/home/hero-slide1.png" alt="" class="img-fluid">
        <a href="#" class="hero-carousel_slideOverlay">
          <h3>Wireless Headphone</h3>
          <p>Accessories Item</p>
        </a>
      </div>
      <div class="hero-carousel_slide">
        <img src="static/img/home/hero-slide2.png" alt="" class="img-fluid">
        <a href="#" class="hero-carousel_slideOverlay">
          <h3>Wireless Headphone</h3>
          <p>Accessories Item</p>
        </a>
      </div>
      <div class="hero-carousel_slide">
        <img src="static/img/home/hero-slide3.png" alt="" class="img-fluid">
        <a href="#" class="hero-carousel_slideOverlay">
          <h3>Wireless Headphone</h3>
          <p>Accessories Item</p>
        </a>
      </div>
    </div>
  </section>
  <!--================ Hero Carousel end =================-->

  <!-- ================ trending product section start ================ -->
  <section class="section-margin calc-60px">
    <div class="container">
      <div class="section-intro pb-60px">
        <p>Popular Item in the market</p>
        <h2>Trending <span class="section-intro_style">Product</span></h2>
```

```
        </div>
        <div class="row">
          <div class="col-md-6 col-lg-4 col-xl-3">
            <div class="card text-center card-product">
              <div class="card-product_img">
                <img class="card-img" src="static/img/product/product1.png" alt="">
                <ul class="card-product_imgOverlay">
                  <li><button><i class="ti-search"></i></button></li>
                  <li><button><i class="ti-shopping-cart"></i></button></li>
                  <li><button><i class="ti-heart"></i></button></li>
                </ul>
              </div>
              <div class="card-body">
                <p>Accessories</p>
                <h4 class="card-product_title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
                <p class="card-product_price">$150.00</p>
              </div>
            </div>
          </div>
          <div class="col-md-6 col-lg-4 col-xl-3">
            <div class="card text-center card-product">
              <div class="card-product_img">
                <img class="card-img" src="static/img/product/product2.png" alt="">
                <ul class="card-product_imgOverlay">
                  <li><button><i class="ti-search"></i></button></li>
                  <li><button><i class="ti-shopping-cart"></i></button></li>
                  <li><button><i class="ti-heart"></i></button></li>
                </ul>
              </div>
              <div class="card-body">
                <p>Beauty</p>
                <h4 class="card-product_title"><a href="single-product.html">Women
Freshwash</a></h4>
                <p class="card-product_price">$150.00</p>
              </div>
            </div>
          </div>
          <div class="col-md-6 col-lg-4 col-xl-3">
            <div class="card text-center card-product">
              <div class="card-product_img">
                <img class="card-img" src="static/img/product/product3.png" alt="">
                <ul class="card-product_imgOverlay">
                  <li><button><i class="ti-search"></i></button></li>
                  <li><button><i class="ti-shopping-cart"></i></button></li>
                  <li><button><i class="ti-heart"></i></button></li>
                </ul>
              </div>
              <div class="card-body">
                <p>Decor</p>
```

```
            <h4 class="card-product_title"><a href="single-product.html">Room Flash
Light</a></h4>
            <p class="card-product_price">$150.00</p>
          </div>
        </div>
      </div>
      <div class="col-md-6 col-lg-4 col-xl-3">
        <div class="card text-center card-product">
          <div class="card-product_img">
            <img class="card-img" src="static/img/product/product4.png" alt="">
            <ul class="card-product_imgOverlay">
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
            </ul>
          </div>
          <div class="card-body">
            <p>Decor</p>
            <h4 class="card-product_title"><a href="single-product.html">Room Flash
Light</a></h4>
            <p class="card-product_price">$150.00</p>
          </div>
        </div>
      </div>
      <div class="col-md-6 col-lg-4 col-xl-3">
        <div class="card text-center card-product">
          <div class="card-product_img">
            <img class="card-img" src="static/img/product/product5.png" alt="">
            <ul class="card-product_imgOverlay">
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
            </ul>
          </div>
          <div class="card-body">
            <p>Accessories</p>
            <h4 class="card-product_title"><a href="single-product.html">Man Office
Bag</a></h4>
            <p class="card-product_price">$150.00</p>
          </div>
        </div>
      </div>
      <div class="col-md-6 col-lg-4 col-xl-3">
        <div class="card text-center card-product">
          <div class="card-product_img">
            <img class="card-img" src="static/img/product/product6.png" alt="">
            <ul class="card-product_imgOverlay">
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
```

```
            </ul>
          </div>
          <div class="card-body">
            <p>Kids Toy</p>
            <h4 class="card-product_title"><a href="single-product.html">Charging
Car</a></h4>
            <p class="card-product_price">$150.00</p>
          </div>
        </div>
      </div>
      <div class="col-md-6 col-lg-4 col-xl-3">
        <div class="card text-center card-product">
          <div class="card-product_img">
            <img class="card-img" src="static/img/product/product7.png" alt="">
            <ul class="card-product_imgOverlay">
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
            </ul>
          </div>
          <div class="card-body">
            <p>Accessories</p>
            <h4 class="card-product_title"><a href="single-product.html">Blutooth
Speaker</a></h4>
            <p class="card-product_price">$150.00</p>
          </div>
        </div>
      </div>
      <div class="col-md-6 col-lg-4 col-xl-3">
        <div class="card text-center card-product">
          <div class="card-product_img">
            <img class="card-img" src="static/img/product/product8.png" alt="">
            <ul class="card-product_imgOverlay">
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
            </ul>
          </div>
          <div class="card-body">
            <p>Kids Toy</p>
            <h4 class="card-product_title"><a href="#">Charging Car</a></h4>
            <p class="card-product_price">$150.00</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
<!-- ================ trending product section end ================ -->
```

```
    <!-- ================ offer section start ================= -->
    <section class="offer" id="parallax-1" data-anchor-target="#parallax-1" data-300-
top="background-position: 20px 30px" data-top-bottom="background-position: 0 20px">
      <div class="container">
        <div class="row">
          <div class="col-xl-5">
            <div class="offer_content text-center">
              <h3>Up To 50% Off</h3>
              <h4>Winter Sale</h4>
              <p>Him she'd let them sixth saw light</p>
              <a class="button button--active mt-3 mt-xl-4" href="#">Shop Now</a>
            </div>
          </div>
        </div>
      </div>
    </section>
    <!-- ================ offer section end ================= -->

    <!-- ================ Best Selling item  carousel ================= -->
    <section class="section-margin calc-60px">
      <div class="container">
        <div class="section-intro pb-60px">
          <p>Popular Item in the market</p>
          <h2>Best <span class="section-intro_style">Sellers</span></h2>
        </div>
        <div class="owl-carousel owl-theme" id="bestSellerCarousel">
          <div class="card text-center card-product">
            <div class="card-product_img">
              <img class="img-fluid" src="static/img/product/product1.png" alt="">
              <ul class="card-product_imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
              </ul>
            </div>
            <div class="card-body">
              <p>Accessories</p>
              <h4 class="card-product_title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
              <p class="card-product_price">$150.00</p>
            </div>
          </div>

          <div class="card text-center card-product">
            <div class="card-product_img">
              <img class="img-fluid" src="static/img/product/product2.png" alt="">
              <ul class="card-product_imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
```

```
            <li><button><i class="ti-heart"></i></button></li>
           </ul>
          </div>
          <div class="card-body">
           <p>Beauty</p>
           <h4 class="card-product_title"><a href="single-product.html">Women
Freshwash</a></h4>
           <p class="card-product_price">$150.00</p>
          </div>
         </div>

         <div class="card text-center card-product">
          <div class="card-product_img">
           <img class="img-fluid" src="static/img/product/product3.png" alt="">
           <ul class="card-product_imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
           </ul>
          </div>
          <div class="card-body">
           <p>Decor</p>
           <h4 class="card-product_title"><a href="single-product.html">Room Flash
Light</a></h4>
           <p class="card-product_price">$150.00</p>
          </div>
         </div>

         <div class="card text-center card-product">
          <div class="card-product_img">
           <img class="img-fluid" src="static/img/product/product4.png" alt="">
           <ul class="card-product_imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
           </ul>
          </div>
          <div class="card-body">
           <p>Decor</p>
           <h4 class="card-product_title"><a href="single-product.html">Room Flash
Light</a></h4>
           <p class="card-product_price">$150.00</p>
          </div>
         </div>

         <div class="card text-center card-product">
          <div class="card-product_img">
           <img class="img-fluid" src="static/img/product/product1.png" alt="">
           <ul class="card-product_imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
```

```
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
          </ul>
        </div>
        <div class="card-body">
          <p>Accessories</p>
          <h4 class="card-product_title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
          <p class="card-product_price">$150.00</p>
        </div>
      </div>

      <div class="card text-center card-product">
        <div class="card-product_img">
          <img class="img-fluid" src="static/img/product/product2.png" alt="">
          <ul class="card-product_imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
          </ul>
        </div>
        <div class="card-body">
          <p>Beauty</p>
          <h4 class="card-product_title"><a href="single-product.html">Women
Freshwash</a></h4>
          <p class="card-product_price">$150.00</p>
        </div>
      </div>

      <div class="card text-center card-product">
        <div class="card-product_img">
          <img class="img-fluid" src="static/img/product/product3.png" alt="">
          <ul class="card-product_imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
          </ul>
        </div>
        <div class="card-body">
          <p>Decor</p>
          <h4 class="card-product_title"><a href="single-product.html">Room Flash
Light</a></h4>
          <p class="card-product_price">$150.00</p>
        </div>
      </div>

      <div class="card text-center card-product">
        <div class="card-product_img">
          <img class="img-fluid" src="static/img/product/product4.png" alt="">
          <ul class="card-product_imgOverlay">
```

```
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
            </ul>
          </div>
          <div class="card-body">
            <p>Decor</p>
            <h4 class="card-product_title"><a href="single-product.html">Room Flash
Light</a></h4>
            <p class="card-product_price">$150.00</p>
          </div>
        </div>
      </div>
    </div>
  </section>
  <!-- ================ Best Selling item  carousel end ================ -->

  <!-- ================ Blog section start ================ -->
  <section class="blog">
    <div class="container">
      <div class="section-intro pb-60px">
        <p>Popular Item in the market</p>
        <h2>Latest <span class="section-intro_style">News</span></h2>
      </div>

      <div class="row">
        <div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
          <div class="card card-blog">
            <div class="card-blog_img">
              <img class="card-img rounded-0" src="static/img/blog/blog1.png" alt="">
            </div>
            <div class="card-body">
              <ul class="card-blog_info">
                <li><a href="#">By Admin</a></li>
                <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
              </ul>
              <h4 class="card-blog_title"><a href="single-blog.html">The Richland Center
Shooping News and weekly shooper</a></h4>
              <p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged
divided light Forth.</p>
              <a class="card-blog_link" href="#">Read More <i class="ti-arrow-
right"></i></a>
            </div>
          </div>
        </div>

        <div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
          <div class="card card-blog">
            <div class="card-blog_img">
              <img class="card-img rounded-0" src="static/img/blog/blog2.png" alt="">
```

```
            </div>
            <div class="card-body">
              <ul class="card-blog_info">
                <li><a href="#">By Admin</a></li>
                <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
              </ul>
              <h4 class="card-blog_title"><a href="single-blog.html">The Shopping News also offers top-quality printing services</a></h4>
              <p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged divided light Forth.</p>
              <a class="card-blog_link" href="#">Read More <i class="ti-arrow-right"></i></a>
            </div>
          </div>
        </div>

        <div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
          <div class="card card-blog">
            <div class="card-blog_img">
              <img class="card-img rounded-0" src="static/img/blog/blog3.png" alt="">
            </div>
            <div class="card-body">
              <ul class="card-blog_info">
                <li><a href="#">By Admin</a></li>
                <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
              </ul>
              <h4 class="card-blog_title"><a href="single-blog.html">Professional design staff and efficient equipment you'll find we offer</a></h4>
              <p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged divided light Forth.</p>
              <a class="card-blog_link" href="#">Read More <i class="ti-arrow-right"></i></a>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>
  <!-- ================ Blog section end ================= -->

  <!-- ================ Subscribe section start ================= -->
  <section class="subscribe-position">
    <div class="container">
      <div class="subscribe text-center">
        <h3 class="subscribe_title">Get Update From Anywhere</h3>
        <p>Bearing Void gathering light light his eavening unto dont afraid</p>
        <div id="mc_embed_signup">
          <form target="_blank" action="https://spondonit.us12.list-manage.com/subscribe/post?u=1462626880ade1ac87bd9c93a&amp;id=92a4423d01" method="get" class="subscribe-form form-inline mt-5 pt-1">
```

```
            <div class="form-group ml-sm-auto">
              <input class="form-control mb-1" type="email" name="EMAIL"
placeholder="Enter your email" onfocus="this.placeholder = "" onblur="this.placeholder =
'Your Email Address '" >
              <div class="info"></div>
            </div>
            <button class="button button-subscribe mr-auto mb-1" type="submit">Subscribe
Now</button>
            <div style="position: absolute; left: -5000px;">
              <input name="b_36c4fd991d266f23781ded980_aefe40901a" tabindex="-1"
value="" type="text">
            </div>

          </form>
        </div>

      </div>
    </div>
  </section>
  <!-- ================ Subscribe section end ================ -->



 </main>


 <!--================ Start footer Area  ================-->
        <footer class="footer">
                <div class="footer-area">
                        <div class="container">
                                <div class="row section_gap">
                                        <div class="col-lg-3 col-md-6 col-sm-6">
                                                <div class="single-footer-widget tp_widgets">
                                                        <h4 class="footer_title large_title">Our
Mission</h4>
                                                        <p>
                                                                So seed seed green that winged
cattle in. Gathering thing made fly you're no
                                                                divided deep moved us lan
Gathering thing us land years living.
                                                        </p>
                                                        <p>
                                                                So seed seed green that winged
cattle in. Gathering thing made fly you're no divided deep moved
                                                        </p>
                                                </div>
                                        </div>
                                        <div class="offset-lg-1 col-lg-2 col-md-6 col-sm-6">
                                                <div class="single-footer-widget tp_widgets">
```

```
                                        <h4 class="footer_title">Quick
Links</h4>
                                        <ul class="list">
                                            <li><a href="#">Home</a></li>
                                            <li><a href="#">Shop</a></li>
                                            <li><a href="#">Blog</a></li>
                                            <li><a
href="#">Product</a></li>
                                            <li><a href="#">Brand</a></li>
                                            <li><a
href="#">Contact</a></li>
                                        </ul>
                                    </div>
                                </div>
                                <div class="col-lg-2 col-md-6 col-sm-6">
                                    <div class="single-footer-widget instafeed">
                                        <h4 class="footer_title">Gallery</h4>
                                        <ul class="list instafeed d-flex flex-
wrap">
                                            <li><img
src="static/img/gallery/r1.jpg" alt=""></li>
                                            <li><img
src="static/img/gallery/r2.jpg" alt=""></li>
                                            <li><img
src="static/img/gallery/r3.jpg" alt=""></li>
                                            <li><img
src="static/img/gallery/r5.jpg" alt=""></li>
                                            <li><img
src="static/img/gallery/r7.jpg" alt=""></li>
                                            <li><img
src="static/mg/gallery/r8.jpg" alt=""></li>
                                        </ul>
                                    </div>
                                </div>
                                <div class="offset-lg-1 col-lg-3 col-md-6 col-sm-6">
                                    <div class="single-footer-widget tp_widgets">
                                        <h4 class="footer_title">Contact
Us</h4>
                                        <div class="ml-40">
                                            <p class="sm-head">
                                                <span class="fa fa-
location-arrow"></span>
                                                Head Office
                                            </p>
                                            <p>123, Main Street, Your
City</p>

                                            <p class="sm-head">
                                                <span class="fa fa-
phone"></span>
```

```
                                                    Phone Number
                                                </p>
                                                <p>
                                                    +123 456 7890 <br>
                                                    +123 456 7890
                                                </p>

                                                <p class="sm-head">
                                                    <span class="fa fa-
envelope"></span>
                                                    Email
                                                </p>
                                                <p>
                                                    free@infoexample.com
<br>
                                                    www.infoexample.com
                                                </p>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            </div>

                            <div class="footer-bottom">
                                <div class="container">
                                    <div class="row d-flex">
                                        <p class="col-lg-12 footer-text text-center">
                                            <!-- Link back to Colorlib can't be removed.
Template is licensed under CC BY 3.0. -->
Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights
reserved | This template is made with <i class="fa fa-heart" aria-hidden="true"></i> by <a
href="https://colorlib.com" target="_blank">Colorlib</a>
<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. --></p>
                                    </div>
                                </div>
                            </div>
                        </footer>
                        <!--================ End footer Area  =================-->



    <script src="static/vendors/jquery/jquery-3.2.1.min.js"></script>
    <script src="static/vendors/bootstrap/bootstrap.bundle.min.js"></script>
    <script src="static/vendors/skrollr.min.js"></script>
    <script src="static/vendors/owl-carousel/owl.carousel.min.js"></script>
    <script src="static/vendors/nice-select/jquery.nice-select.min.js"></script>
    <script src="static/vendors/jquery.ajaxchimp.min.js"></script>
    <script src="static/vendors/mail-script.js"></script>
    <script src="static/js/main.js"></script>
```

</body>
</html>

8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

8.1 TYPES OF TESTS

8.1.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.1.3 Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.1.5 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure, or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

8.2 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

8.2.1 Test Strategy And Approach

Field testing will be performed manually and functional tests will be written in detail.

8.2.2 Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages, and responses must not be delayed.

8.2.3 Features To Be Tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

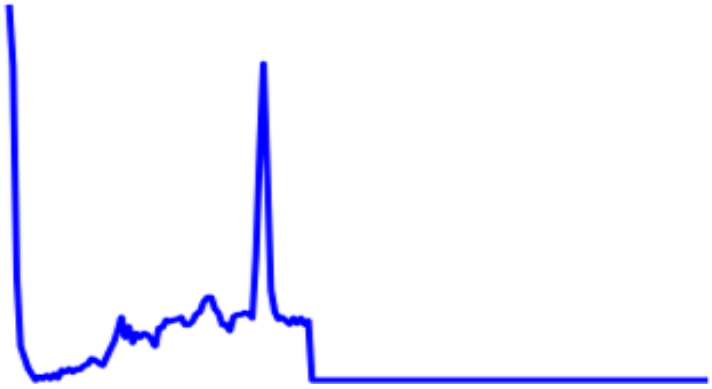
8.4 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

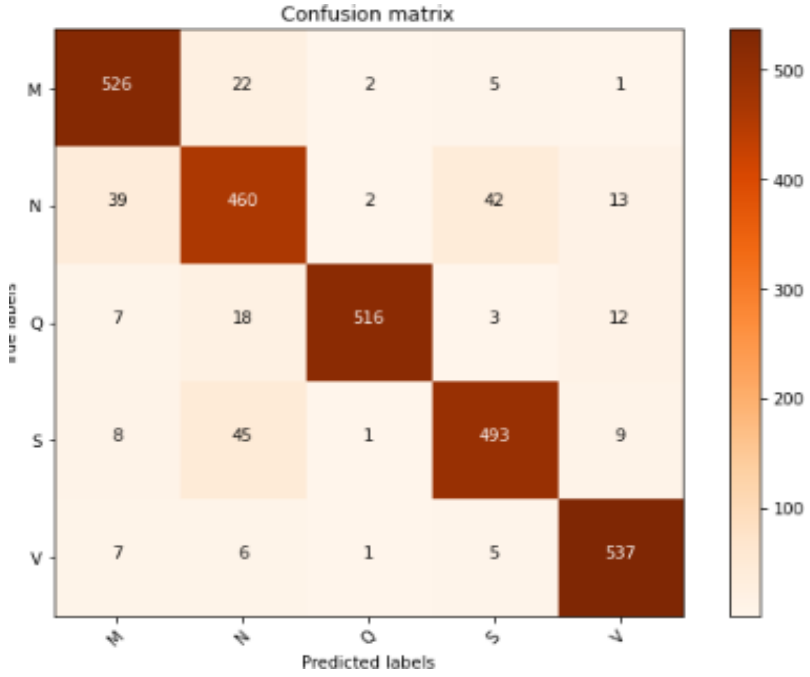
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.RESULTS

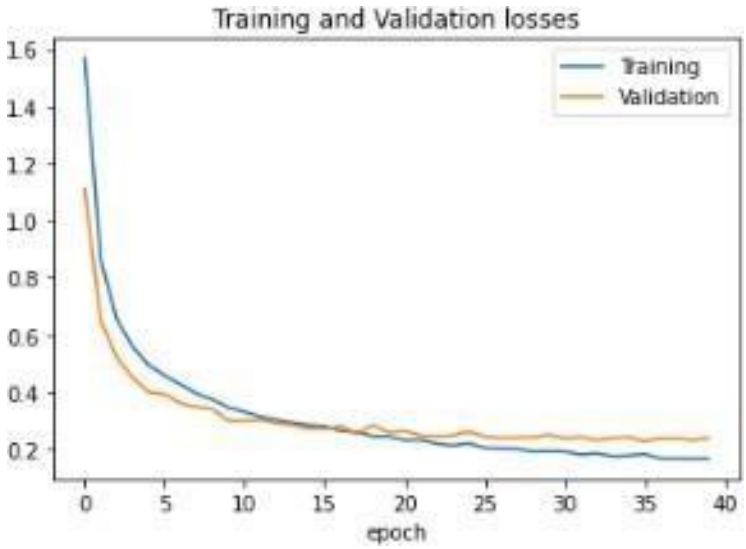
Testing Image



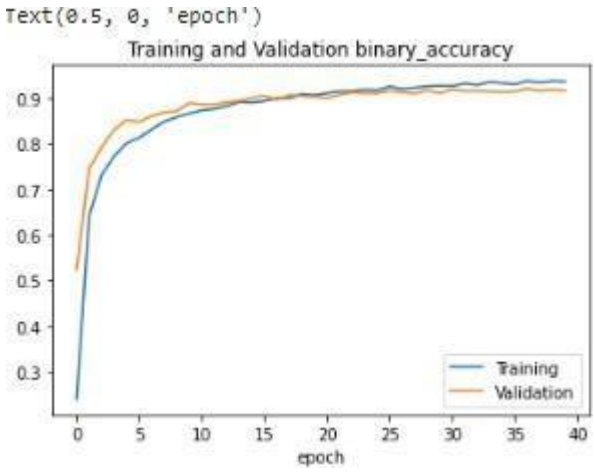
Confusion Matrix



Training And Validation Losses



Training And Testing Accuracy



10.Advantage and Disadvantages

1. High accuracy
2. High sensitivity
3. High reliability
4. Reduced loss

11. Conclusions

This project is designed the In using the MIT-BIH arrhythmia database, we have proposed a system for the automatic processing of the ECG for the classification of arrhythmia images. The database of MIT-BIH is processed visually and a waveform detection method is proposed for detecting the QRS waveform. A CNN model was built to train and classify the ECG images. Experimental results show that according to the ANSI/AAMI EC57 evaluation criteria, The accuracy rate of a ventricular ectopic beat can reach 95.9% and the sensitivity evaluation is 93.0%. For the supraventricular ectopic beat class, the accuracy rate is 93.2% and the sensitivity evaluation is 81.3%

12. Future Scope

In future work, we designed real-time implementation of processor design of arrhythmia classification.

13.APPENDIX

Source Code

In [16]:

```
import numpy as np
import pandas as pd
from pathlib import Path
import os.path

import tensorflow as tf
```

In [2]:

```
import cv2
path =r'../input/ecg-image-data/ECG_Image_data/test/F/F113.png'
x=cv2.imread(path)
x.shape
```

Out[2]:

```
(288, 432, 3)
```

N: Normal beat

S: Supraventricular premature beat

V: Premature ventricular contraction

F: Fusion of ventricular and normal beat

Q: Unclassifiable beat

train set

In [3]:

```
%%time
dir = Path('../input/ecg-image-data/ECG_Image_data/train')

filepaths = list(dir.glob(r'**/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], filepaths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

dataframe = pd.concat([filepaths , labels] , axis=1)
dataframe
```

```
CPU times: user 2.49 s, sys: 2.43 s, total: 4.92 s
Wall time: 3min 25s
```

Out[3]:

	Filepath	Label
0	../input/ecg-image-data/ECG_Image_data/train/N...	N
1	../input/ecg-image-data/ECG_Image_data/train/N...	N
2	../input/ecg-image-data/ECG_Image_data/train/N...	N
3	../input/ecg-image-data/ECG_Image_data/train/N...	N
4	../input/ecg-image-data/ECG_Image_data/train/N...	N
...	...	...
99194	../input/ecg-image-data/ECG_Image_data/train/V...	V
99195	../input/ecg-image-data/ECG_Image_data/train/V...	V
99196	../input/ecg-image-data/ECG_Image_data/train/V...	V
99197	../input/ecg-image-data/ECG_Image_data/train/V...	V
99198	../input/ecg-image-data/ECG_Image_data/train/V...	V

99199 rows × 2 columns

In [4]:

```
dataframe['Label'].value_counts()
```

Out[4]:

```
N    75709
M     8405
Q     6431
V     5789
S     2223
F      642
Name: Label, dtype: int64
```

In [5]:

```
dataframe['Label'].unique()
```

Out[5]:

```
array(['N', 'F', 'M', 'Q', 'S', 'V'], dtype=object)
```

In [9]:

```
samples = []
for category in ['N', 'M', 'Q', 'S', 'V']:
    category_slice = dataframe.query("Label == @category")
    samples.append(category_slice.sample(2223, random_state=1))

dataframe_train = pd.concat(samples, axis=0).sample(frac=1.0, random_state=1).reset_index(d
dataframe_train['Label'].value_counts()
```

Out[9]:

```
N    2223
Q    2223
M    2223
S    2223
V    2223
Name: Label, dtype: int64
```

In [ ]:

```
dir = Path('../input/ecg-image-data/ECG_Image_data/train')

filepaths = list(dir.glob(r'F/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], filepaths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

F = pd.concat([filepaths , labels] , axis=1)
F
```

In [ ]:

```
dataframe_train = pd.concat([F , dataframe1] , axis=0)
dataframe_train['Label'].value_counts()
```

In [11]:

```
dataframe_train
```

Out[11]:

	Filepath	Label
0	../input/ecg-image-data/ECG_Image_data/train/N...	N
1	../input/ecg-image-data/ECG_Image_data/train/Q...	Q
2	../input/ecg-image-data/ECG_Image_data/train/N...	N
3	../input/ecg-image-data/ECG_Image_data/train/M...	M
4	../input/ecg-image-data/ECG_Image_data/train/S...	S
...	...	...
11110	../input/ecg-image-data/ECG_Image_data/train/S...	S
11111	../input/ecg-image-data/ECG_Image_data/train/V...	V
11112	../input/ecg-image-data/ECG_Image_data/train/N...	N
11113	../input/ecg-image-data/ECG_Image_data/train/Q...	Q
11114	../input/ecg-image-data/ECG_Image_data/train/N...	N

11115 rows × 2 columns

test set

In [10]:

```
%%time
dir = Path('../input/ecg-image-data/ECG_Image_data/test')

filepaths = list(dir.glob(r'**/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], filepaths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

dataframe_test = pd.concat([filepaths , labels] , axis=1)
dataframe_test
```

```
CPU times: user 469 ms, sys: 478 ms, total: 947 ms
Wall time: 44.6 s
```

Out[10]:

	Filepath	Label
0	../input/ecg-image-data/ECG_Image_data/test/N/...	N
1	../input/ecg-image-data/ECG_Image_data/test/N/...	N
2	../input/ecg-image-data/ECG_Image_data/test/N/...	N
3	../input/ecg-image-data/ECG_Image_data/test/N/...	N
4	../input/ecg-image-data/ECG_Image_data/test/N/...	N
...	...	...
24794	../input/ecg-image-data/ECG_Image_data/test/V/...	V
24795	../input/ecg-image-data/ECG_Image_data/test/V/...	V
24796	../input/ecg-image-data/ECG_Image_data/test/V/...	V
24797	../input/ecg-image-data/ECG_Image_data/test/V/...	V
24798	../input/ecg-image-data/ECG_Image_data/test/V/...	V

24799 rows × 2 columns

In [12]:

```
dataframe_test['Label'].value_counts()
```

Out[12]:

```
N    18926
M     2101
Q     1608
V     1447
S      556
F      161
Name: Label, dtype: int64
```

In [13]:

```
samples = []
for category in ['N', 'M', 'Q', 'S', 'V']:
    category_slice = dataframe_test.query("Label == @category")
    samples.append(category_slice.sample(556, random_state=1))

dataframe_test = pd.concat(samples, axis=0).sample(frac=1.0, random_state=1).reset_index(dr
dataframe_test['Label'].value_counts()
```

Out[13]:

```
V    556
S    556
M    556
N    556
Q    556
Name: Label, dtype: int64
```

In [ ]:

```
dir = Path('../input/ecg-image-data/ECG_Image_data/test')

filepaths = list(dir.glob(r'F/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], filepaths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

F1 = pd.concat([filepaths , labels] , axis=1)
F1
```

In [ ]:

```
dataframe_test = pd.concat([F1 , dataframe_test] , axis=0)
dataframe_test['Label'].value_counts()
```

In [14]:

```
dataframe_test
```

Out[14]:

	Filepath	Label
0	../input/ecg-image-data/ECG_Image_data/test/V/...	V
1	../input/ecg-image-data/ECG_Image_data/test/V/...	V
2	../input/ecg-image-data/ECG_Image_data/test/S/...	S
3	../input/ecg-image-data/ECG_Image_data/test/M/...	M
4	../input/ecg-image-data/ECG_Image_data/test/N/...	N
...	...	...
2775	../input/ecg-image-data/ECG_Image_data/test/V/...	V
2776	../input/ecg-image-data/ECG_Image_data/test/M/...	M
2777	../input/ecg-image-data/ECG_Image_data/test/M/...	M
2778	../input/ecg-image-data/ECG_Image_data/test/N/...	N
2779	../input/ecg-image-data/ECG_Image_data/test/M/...	M

2780 rows × 2 columns

Image generators

In [17]:

```
train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    brightness_range=[0.2,1.0],
    validation_split=0.2

)

test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
)
```

In [18]:

```
size=64
color_mode='grayscale'
batch_size=32
train_images = train_generator.flow_from_dataframe(
    dataframe=dataframe_train,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=True,
    seed=42,
    subset='training'
)

val_images = train_generator.flow_from_dataframe(
    dataframe=dataframe_train,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=True,
    seed=42,
    subset='validation'
)

test_images = test_generator.flow_from_dataframe(
    dataframe=dataframe_test,
    x_col='Filepath',
    y_col='Label',
    target_size=(size, size),
    color_mode=color_mode,
    class_mode='categorical',
    batch_size=batch_size,
    shuffle=False
)
```

```
Found 8892 validated image filenames belonging to 5 classes.
Found 2223 validated image filenames belonging to 5 classes.
Found 2780 validated image filenames belonging to 5 classes.
```

In [19]:

```
test_images.class_indices
```

Out[19]:

```
{'M': 0, 'N': 1, 'Q': 2, 'S': 3, 'V': 4}
```

Modeling

In [ ]:

```
"""
from tensorflow.keras.applications.vgg16 import VGG16

pretrained_model = VGG16(
                                input_shape=(224,224,3),
                                include_top= False,
                                weights='imagenet',
                                pooling='avg'
                                )

pretrained_model.trainable = False
"""""
```

In [25]:

```
from tensorflow.keras.layers import *
from tensorflow.keras.models import *



model = Sequential ()
model.add(Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=(size,size,1)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))


model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(5, activation='softmax'))

model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_12 (Conv2D)           (None, 64, 64, 32)        320
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 62, 62, 32)        9248
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
dropout_8 (Dropout)          (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 31, 31, 64)        18496
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 29, 29, 64)        36928
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
dropout_9 (Dropout)          (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 14, 14, 64)        36928
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 12, 12, 64)        36928
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
dropout_10 (Dropout)         (None, 6, 6, 64)          0
_________________________________________________________________
flatten_2 (Flatten)          (None, 2304)              0
_________________________________________________________________
```

```
dense_4 (Dense)              (None, 512)               1180160
_________________________________________________________________
dropout_11 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 2565
=================================================================
Total params: 1,321,573
Trainable params: 1,321,573
Non-trainable params: 0
_________________________________________________________________
```

In [26]:

```
import keras
checkpoint = keras.callbacks.ModelCheckpoint(
    filepath='best_model.h5',
    save_weights_only=False,
    monitor='val_accuracy',
    mode='max',
    save_best_only=True,
    verbose=1)
```

In [28]:

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

In [29]:

```
%%time
result=model.fit(
    train_images,
    validation_data=val_images,
    epochs=40,
    callbacks=[checkpoint]
)
```

```
Epoch 1/40
278/278 [==============================] - 51s 178ms/step - loss: 1.5666 -
accuracy: 0.2410 - val_loss: 1.1112 - val_accuracy: 0.5250

Epoch 00001: val_accuracy improved from -inf to 0.52497, saving model to b
est_model.h5
Epoch 2/40
278/278 [==============================] - 48s 174ms/step - loss: 0.8634 -
accuracy: 0.6422 - val_loss: 0.6503 - val_accuracy: 0.7458

Epoch 00002: val_accuracy improved from 0.52497 to 0.74584, saving model t
o best_model.h5
Epoch 3/40
278/278 [==============================] - 49s 175ms/step - loss: 0.6559 -
accuracy: 0.7322 - val_loss: 0.5214 - val_accuracy: 0.7926

Epoch 00003: val_accuracy improved from 0.74584 to 0.79262, saving model t
o best_model.h5
Epoch 4/40
```

Result

In [30]:

```
from keras.models import load_model
best_model=load_model('./best_model.h5')
results = best_model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.26851
Test Accuracy: 91.08%
```

In [ ]:

```
model.save('ECG-rgb,version9,train99,val100,test100.h5')
```

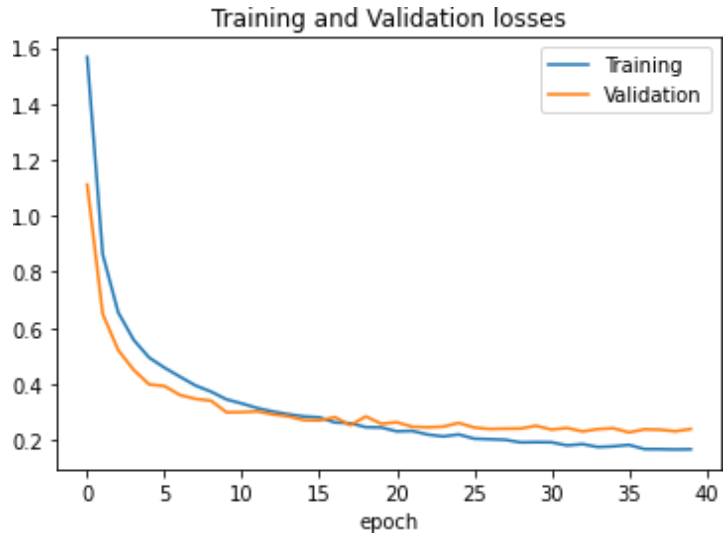
download (./best_model.h5)

In [31]:

```
import matplotlib.pyplot as plt

plt.plot(result.history['loss'])
plt.plot(result.history['val_loss'])
plt.legend(['Training', 'Validation'])
plt.title('Training and Validation losses')
plt.xlabel('epoch')
```

Out[31]:

```
Text(0.5, 0, 'epoch')
```

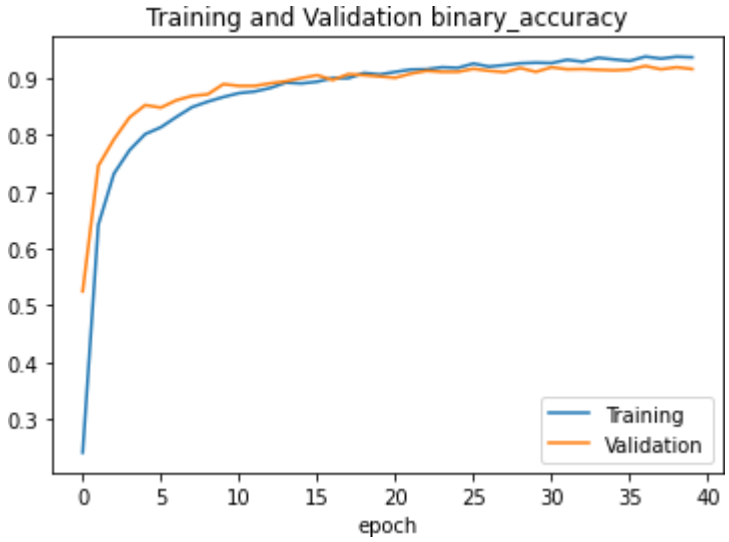


In [32]:

```
plt.plot(result.history['accuracy'])
plt.plot(result.history['val_accuracy'])
plt.legend(['Training', 'Validation'])
plt.title('Training and Validation binary_accuracy')
plt.xlabel('epoch')
```

Out[32]:

```
Text(0.5, 0, 'epoch')
```



In [33]:

```
y_pred = best_model.predict(test_images)
y_pred
```

Out[33]:

```
array([[3.04751597e-06, 3.38918483e-07, 6.03136874e-09, 2.45154924e-10,
        9.99996662e-01],
       [3.57728140e-06, 8.67695417e-06, 1.77200627e-06, 1.69542341e-06,
        9.99984264e-01],
       [3.76527325e-08, 1.53904466e-07, 7.08798797e-13, 9.99999762e-01,
        7.15602057e-12],
       ...,
       [9.84780669e-01, 1.34125147e-02, 1.11143360e-11, 1.80682982e-03,
        1.94587138e-10],
       [1.00830775e-02, 9.09655750e-01, 3.71530838e-02, 1.72309689e-02,
        2.58769598e-02],
       [9.99119580e-01, 8.13806662e-04, 3.37510193e-11, 6.65202970e-05,
        7.39629746e-10]], dtype=float32)
```

In [34]:

```
y_pred = np.argmax(y_pred, axis=1)
y_pred
```

Out[34]:

```
array([4, 4, 3, ..., 0, 1, 0])
```

classification report & confusion matrix

In [35]:

```
from sklearn.metrics import confusion_matrix, classification_report


cm = confusion_matrix(test_images.labels, y_pred)
print(cm)
print('classification report')
print( classification_report(test_images.labels, y_pred))
```

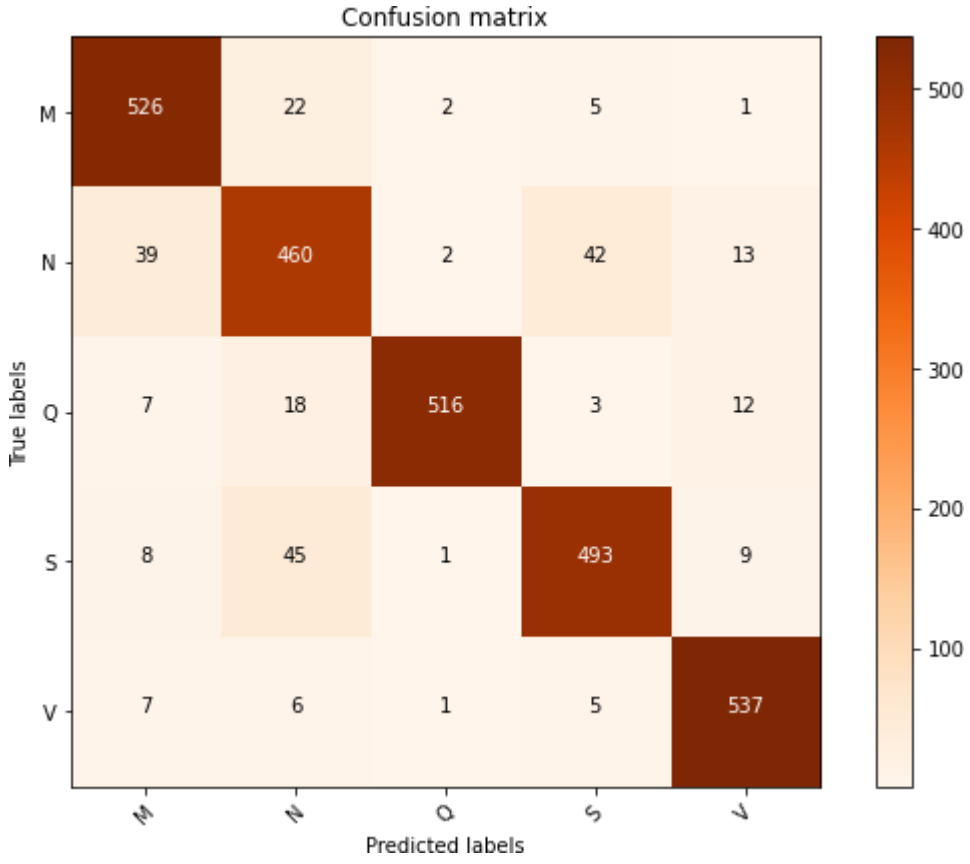
```
[[526  22   2   5   1]
 [ 39 460   2  42  13]
 [  7  18 516   3  12]
 [  8  45   1 493   9]
 [  7   6   1   5 537]]
classification report
              precision    recall  f1-score   support

           0       0.90      0.95      0.92       556
           1       0.83      0.83      0.83       556
           2       0.99      0.93      0.96       556
           3       0.90      0.89      0.89       556
           4       0.94      0.97      0.95       556

    accuracy                           0.91      2780
   macro avg       0.91      0.91      0.91      2780
weighted avg       0.91      0.91      0.91      2780
```

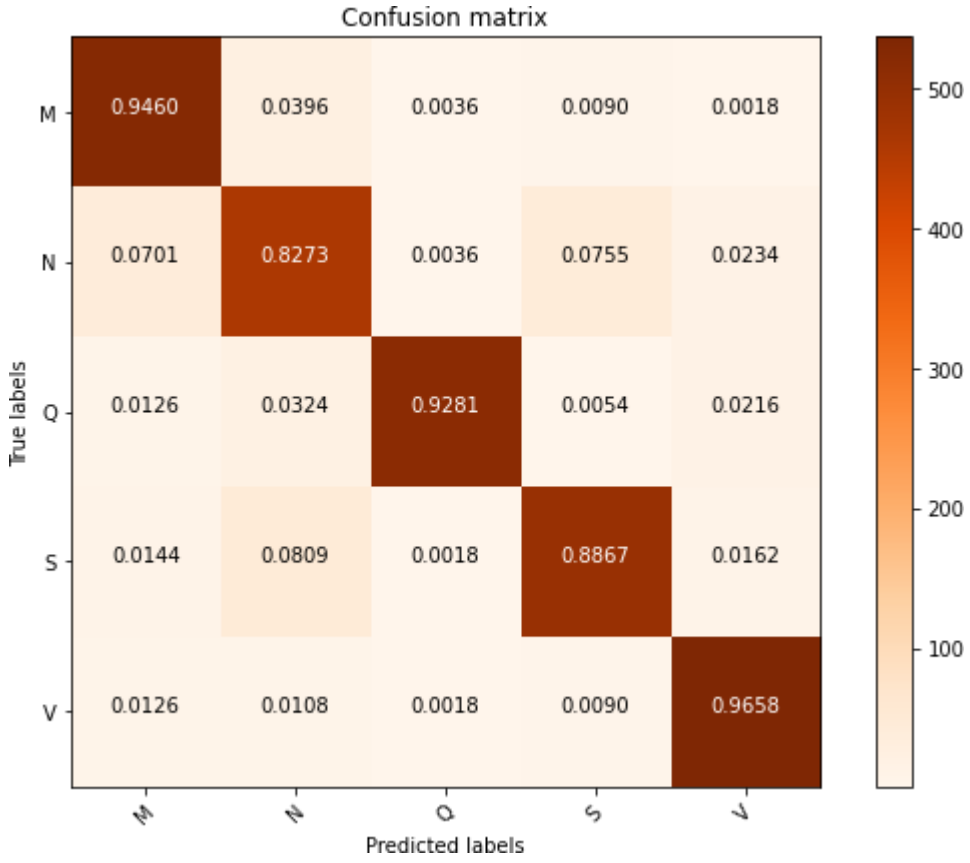
ploting the confusion matrix

In [38]:

```
import itertools

def plot_confusion_matrix(cm, target_names, title='Confusion matrix', cmap=None, normalize=
    """
    arguments
    ----------
    cm:           confusion matrix from sklearn.metrics.confusion_matrix

    target_names: given classification classes such as [0, 1, 2]
                  the class names, for example: ['high', 'medium', 'low']

    title:        the text to display at the top of the matrix

    cmap:         the gradient of the values displayed from matplotlib.pyplot.cm
                  see http://matplotlib.org/examples/color/colormaps_reference.html

    normalize:    If False, plot the raw numbers
                  If True, plot the proportions
    """

    if cmap is None:
        cmap = plt.get_cmap('Oranges')

    plt.figure(figsize=(8, 6))
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()

    if target_names is not None:
        tick_marks = np.arange(len(target_names))
        plt.xticks(tick_marks, target_names, rotation=45)
        plt.yticks(tick_marks, target_names)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]


    thresh = cm.max() / 1.5 if normalize else cm.max() / 2
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        if normalize:
            plt.text(j, i, "{:0.4f}".format(cm[i, j]),
                     horizontalalignment="center",
                     color="white" if cm[i, j] > thresh else "black")
        else:
            plt.text(j, i, "{:,}".format(cm[i, j]),
                     horizontalalignment="center",
                     color="white" if cm[i, j] > thresh else "black")


    plt.tight_layout()
    plt.ylim(len(target_names)-0.5, -0.5)
    plt.ylabel('True labels')
    plt.xlabel('Predicted labels')
    plt.savefig(title + '.png', dpi=500, bbox_inches = 'tight')
    plt.show()



# a tuple for all the class names
```

```
target_names = ('M', 'N', 'Q', 'S', 'V' )
plot_confusion_matrix(cm, target_names)
plot_confusion_matrix(cm, target_names,normalize=True)
```





GitHub link:

https://github.com/IBM-EPBL/IBM-Project-14740-1659589389

DEMO VIDEO LINK:

https://drive.google.com/drive/folders/1Ufva6RVA3nhm269ks9zNXiryN5KFcQDA