



# Medical Diagnosis System (MDS)

Computer aided AI based system for medical diagnosis

## Supervised by:

Dr. Hany Aly Elghaish

## Team members

محمد عبد الجليل محمد الدهروري  
محمد صابر محمد رزق  
محمد ابراهيم السيد المكاوي  
هشام محمد عوض اسماعيل

احمد علي محمد عبد الجوارد  
اسماء قدرى علي الشامي  
رحاب محمد عبدالله داود  
كريم محمد احمد الشعيرى

2022-2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
مَا نَحْنُ بِحَالٍ إِلَّا مَا سَعَىٰ

## **ACKNOWLEDGEMENT**

First and foremost, praises and thanks to the God, the Almighty, for his showers of blessings throughout our graduation project work to complete the mission successfully.

We would like to thank our supervisor Dr. Hany Aly Elghaish for bringing the weight of his considerable experience and knowledge to this project and providing invaluable guidance throughout this project, his dynamism, vision, sincerity and motivation have deeply inspired us, he has taught us the methodology to carry out the project and to present the project works as clearly as possible, it was a great privilege and honor to work under his guidance, we are extremely grateful for what he has offered us.

We are extremely grateful to our parents for their love, prayers, caring and sacrifices for educating and preparing us for our future.

We would like to say thanks to our friends and colleagues, for their constant encouragement.

Finally, we thank all people who have supported us to complete the project directly or indirectly.

## **ABSTRACT**

Artificial intelligence (AI) in medicine refers to the use of machine learning algorithms to explore medical data and reveal insights to assist enhance patient experiences and health outcomes.

Artificial intelligence (AI) is increasingly becoming a crucial component of modern healthcare because to recent advancements in computer science and informatics, medical professionals are supported by AI algorithms and other apps that are driven by AI in clinical settings and ongoing research.

Clinical decision support and image analysis are two of the most frequent uses of AI in medical applications right now, AI tools are being used in medical imaging to analyze CT scans, x-rays, MRIs, and other images for lesions or other findings that a human radiologist might miss, clinical decision support tools assist healthcare providers in making decisions about treatments, medications, mental health, and other patient needs by giving them quick access to information that's relevant to their patient.

# Table of Contents

<b>ACKNOWLEDGEMENT.....</b>	<b>3</b>
<b>ABSTRACT.....</b>	<b>4</b>
<b>Table of Contents .....</b>	<b>5</b>
<b>Chapter 1 INTRODUCTION .....</b>	<b>9</b>
1.1    Preamble .....	10
1.2    Background of the Problem .....	10
1.3    Statement of the problem .....	11
1.4    Objectives of the project.....	12
1.5    Scope of the project.....	12
1.6    Significance of the project .....	12
1.7    Organization of the project report .....	13
<b>Chapter 2 LITERATURE REVIEW AND BACKGROUND .....</b>	<b>14</b>
2.1    Introduction.....	15
2.2    COVID-19 detection state-of-the-art.....	15
2.3    Heartbeat abnormalities categorization state-of-the-art.....	18
2.4    Chest x-ray abnormalities localization state-of-the-art .....	21
2.5    Brain tumor state-of-the-art .....	23
2.5.1    Brain tumor categorization .....	23
2.5.2    Brain tumor segmentation from MRI images .....	26
2.6    Skin cancer detection state-of-the-art .....	28
2.7    Background .....	31
2.7.1    Convolution neural networks (CNN) .....	31
2.7.2    Long-short time memory (LSTM).....	32
2.7.3    Multi-head attention (MHA).....	33
2.7.4    Adaptive feature recalibration (AFR).....	34
2.7.5    Adaptive kernel size convolution (AKSC) .....	35
2.7.6    Vision transformer model (ViT) .....	35
2.7.7    U-Net segmentation model .....	36
2.7.8    YOLO v5 object detection model.....	37
2.7.9    PyQt5 GUI module and PyQt5 Designer .....	37
2.7.10    HTML, CSS and JavaScript .....	38
2.7.11    Flask module.....	39
<b>Chapter 3 METHODOLOGY AND PROPOSED FRAMEWORK .....</b>	<b>40</b>
3.1    Introduction.....	41
3.2    Over view of the MDS framework.....	41

<b>3.3 Development process .....</b>	43
<b>3.4 Deployment and GUI.....</b>	44
<b>Chapter 4 EXPERIMENTAL RESULTS.....</b>	<b>47</b>
<b>4.1 Introduction.....</b>	48
<b>4.2 Development of AI models .....</b>	48
<b>4.2.1 Heartbeat abnormalities detection from electrocardiogram (ECG) .....</b>	<b>48</b>
<b>4.2.1.1 Motivation.....</b>	48
<b>4.2.1.2 Dataset description.....</b>	49
<b>4.2.1.3 Reference paper for result comparison.....</b>	52
<b>4.2.1.4 Development methodology .....</b>	52
<b>4.2.1.5 2-D convolution methodology .....</b>	53
<b>4.2.1.6 1-D convolution methodology .....</b>	65
<b>Tuning the number of heads of the MHA layer .....</b>	96
<b>Tuning the reduction ratio of the AFR layer.....</b>	105
<b>Experiments summary with over-sampling.....</b>	114
<b>Weighted cross entropy vs focal loss .....</b>	119
<b>Experiments summary with weighted cross entropy .....</b>	141
<b>GPU specs .....</b>	142
<b>Training time and number of parameters .....</b>	143
<b>4.2.1.7 Final proposed framework.....</b>	144
<b>4.2.1.8 Method validation .....</b>	145
<b>4.2.1.9 Testing with real world cases (online testing).....</b>	146
<b>4.2.1.10 Hardware system for real time diagnosis of ECG signal.....</b>	151
<b>Part 1: The Hardware.....</b>	152
<b>Part 2: The Broker (server).....</b>	156
<b>Part 3: The GUI .....</b>	158
<b>Summary.....</b>	159
<b>4.2.2 COVID-19 detection from chest x-ray images.....</b>	<b>160</b>
<b>4.2.2.1 Motivation.....</b>	160
<b>4.2.2.2 Dataset description.....</b>	160
<b>4.2.2.3 Reference papers for result comparison .....</b>	162
<b>4.2.2.4 Development methodology .....</b>	162
<b>4.2.2.5 2-D convolution methodology .....</b>	162
<b>Experiments summary with different pre-trained models.....</b>	175
<b>Tuning the number of heads of the MHA layer .....</b>	182
<b>Tuning the reduction ratio of the AFR layer .....</b>	188
<b>Tuning the patch size of the vision transformer.....</b>	196

<b>Experiments summary.....</b>	197
<b>Training time and number of parameters .....</b>	198
<b>4.2.2.6 Final proposed framework.....</b>	199
<b>4.2.2.7 Method validation .....</b>	200
<b>4.2.2.8 Testing with real world cases (online testing).....</b>	201
<b>4.2.3 Brain tumor categorization from MRI images.....</b>	209
<b>    4.2.3.1 Motivation.....</b>	209
<b>    4.2.3.2 Dataset description.....</b>	209
<b>    4.2.3.3 Development methodology .....</b>	211
<b>    4.2.3.4 2-D convolution .....</b>	212
<b>Tuning the number of heads of the MHA layer .....</b>	225
<b>Tuning the reduction ratio of the AFR layer.....</b>	232
<b>Experiments summary.....</b>	242
<b>Training time and number of parameters .....</b>	243
<b>    4.2.3.5 Final proposed framework.....</b>	244
<b>    4.2.3.6 Method validation .....</b>	244
<b>    4.2.3.7 Testing with real world cases (online testing).....</b>	245
<b>4.2.4 Brain tumor segmentation from MRI images .....</b>	256
<b>    4.2.4.1 Motivation.....</b>	256
<b>    4.2.4.2 Dataset description.....</b>	256
<b>    4.2.4.3 Development methodology .....</b>	259
<b>    4.2.4.4 Reference papers for result comparison .....</b>	261
<b>    4.2.4.5 Experiments with U-Net architecture .....</b>	264
<b>4.2.5 Chest x-ray abnormalities localization and categorization .....</b>	268
<b>    4.2.5.1 Motivation.....</b>	268
<b>    4.2.5.2 Dataset description.....</b>	268
<b>    4.2.5.3 Development methodology .....</b>	269
<b>    4.2.5.4 Reference papers for result comparison .....</b>	271
<b>    4.2.5.5 Experiments with yolo object detection model.....</b>	271
<b>    4.2.5.6 Testing with real world cases (online testing).....</b>	274
<b>4.2.6 Skin cancer detection from images.....</b>	275
<b>    4.2.6.1 Motivation.....</b>	275
<b>    4.2.6.2 Dataset description.....</b>	275
<b>    4.2.6.3 Development methodology .....</b>	278
<b>    4.2.6.4 Reference papers for result comparison .....</b>	278
<b>    4.2.6.5 Experiments.....</b>	278
<b>    4.2.6.6 Testing with real world cases (online testing).....</b>	281
<b>4.2.7 AI models results summary and conclusion .....</b>	287

<b>4.3 GUI and deployment.....</b>	<b>289</b>
<b>4.3.1 Home page .....</b>	<b>289</b>
<b>4.3.2 Brain tumor service page and model deployment.....</b>	<b>291</b>
<b>4.3.3 Covid-19 detection service page and model deployment .....</b>	<b>295</b>
<b>4.3.4 Heartbeat abnormalities service page and model deployment .....</b>	<b>297</b>
<b>4.3.5 Chest abnormalities service page and model deployment.....</b>	<b>299</b>
<b>4.4 conclusion and future work.....</b>	<b>301</b>
<b>Appendix: links .....</b>	<b>303</b>
<b>References.....</b>	<b>306</b>

# **Chapter 1**

# **INTRODUCTION**

## **1.1 Preamble**

This chapter presents the background of the problem in Sec. 1.2, while Sec. 1.3 and 1.4 show, the problem statement and objectives, respectively, finally, the scope of the project, significance and the organization of the project report are demonstrated in Sec 1.5, 1.6 and 1.7.

## **1.2 Background of the Problem**

There is enormous hope that the application of artificial intelligence (AI) will result in significant advancements in all areas of healthcare, from diagnosis to therapy. There is already significant proof that AI systems perform as well as or better than humans in a variety of medical tasks, for example the examination of medical pictures or the correlation of symptoms and biomarkers from electronic medical records (EMRs) with the description and prognosis of the condition, smartphones and wireless technology advancements have made it possible to get on-demand healthcare services through search engines and applications for health tracking and easy diagnosis, they have also made it possible to offer healthcare through remote interactions, available anytime and anywhere, such services help save expenses and minimize unnecessarily exposing patients to infectious diseases at the clinic, which is important for underserved areas and places lacking experts, Telehealth technology is especially important in developing nations where the healthcare infrastructure can be developed to suit the requirements of the present advances, even though the idea is straightforward, these solutions still require extensive testing validation in the real world by experts to guarantee patient safety and efficacy.

It is believed that AI may enhance any procedure used in the operation and delivery of healthcare, for instance, the ability of AI to reduce costs for the healthcare system is a key factor in the adoption of AI medical applications, the estimations says that AI applications might reduce US healthcare expenses by \$150 billion annually by 2026, through constant monitoring and counseling, AI-based technology will play a major role in assisting individuals in maintaining their health by ensuring earlier diagnosis, customized treatments, and more effective follow-ups, it is generally believed that AI technologies will support and improve human work rather than overtly replace the work of doctors and other healthcare professionals, AI is capable of assisting medical staff with a range of tasks, including administrative workflow, clinical documentation, patient outreach, and specialist support for image analysis, medical device automation, and patient monitoring, in other words, it help support the diagnosis and medical treatment. [1]

### **1.3 Statement of the problem**

#### **Developing AI model that can differentiate between COVID-19, pneumonia and normal cases from chest x-ray images:**

Polymerase chain reactions can be used to identify COVID-19 by spotting the virus genetic material, however the molecular test findings can take many hours or even days to return, chest radiographs on the other hand may be obtained quickly, AI classification models can speed up the diagnosis process by using deep learning classification models.

#### **Developing AI model that can differentiate between some types of brain tumors from MRI images:**

The best method to detect brain tumors is magnetic resonance imaging (MRI) which produces images that are examined by a radiologist, a manual examination can be error-prone due to the level of complexity involved in brain tumors and their properties, it can be extremely difficult and time-consuming to create results from an MRI in poor nations due to a lack of skilled medical professionals and a lack of understanding of malignancies, AI classification models can help with this.

#### **Developing AI model that can segment a brain tumor in MRI images:**

Brain tumor segmentation is a critical task in the diagnosis and treatment of brain cancer, accurate segmentation of the tumor from MRI images is essential for treatment planning and monitoring disease progression, manual segmentation is time-consuming, therefore, there is a need for automated segmentation methods, deep learning-based approaches have shown promising results in brain tumor segmentation, however, challenges such as variability in imaging protocols, lack of standardization, and limited clinical validation need to be addressed to improve the clinical utility of these methods, overcoming these challenges could lead to more accurate and efficient segmentation, which in turn could improve patient outcomes and reduce the need for multiple surgeries.

#### **Developing AI model that can differentiate between some cardiac abnormalities from the electrocardiogram (ECG):**

Heart disease is one of the major causes of death worldwide, an estimated number of 12 million people dies from heart disease every year, prediction of heart disease is a critical challenge in the area of clinical data analysis, machine learning (ML) has been shown to be effective in assisting with making decisions and predictions from the large quantity of data produced by the healthcare industry, the electrocardiogram (ECG) is a crucial tool for determining the structure and

function of the heart, automated classification of the ECG data aids for early diagnosis in real time before the patient could be seen physically by the doctor.

### **Developing AI model that can localize and categorize chest abnormalities from x-ray images:**

The chest radiograph is possibly one of the most challenging problem that radiologists and clinicians face on a daily basis, even a skilled doctor may misdiagnose a patient as a result of how a chest x-ray is interpreted, Computer-aided detection and diagnosis would assist reduce the load on doctors and improve the diagnostic quality.

## **1.4 Objectives of the project**

1. Dealing with data imbalance problems
2. Utilizing suitable data augmentation methods
3. Understanding and utilization of different evaluation metrics
4. Utilization of pre-trained models
5. Developing AI models with a good performance measures compared to the state-of-the-art
6. Deploying the developed models in a desktop application and a web application

## **1.5 Scope of the project**

1. The proposed system is a prototype as a proof of concept and can't be used in real world cases
2. The proposed system based on AI models will be deployed in a desktop application and a web application
3. The integration with management systems and databases is out of scope
4. Security issues are out of scope

## **1.6 Significance of the project**

The main advantages of using the proposed medical diagnosis system (MDA) can be briefed as follows:

**Lowering the cost of the healthcare process:** As it reduces the number of surgeries required in the case of the diagnosis of the brain tumor and the detection of COVID-19 from chest x-ray images without the need for Polymerase chain reactions.

**Reduce diagnostics time and effort:** The Polymerase chain reactions for COVID-19 may take hours and even days to return its result, the proposed system can help speed up this process and ensure that the patient receives the proper medical treatment as soon as possible

**Help patients receive a better treatment:** The surgery required to obtaining a tissue sample for genetic study of brain tumors can take several weeks to determine the genetic characterization of the tumor, the proposed system can differentiate between some types of brain tumors from the MRI image refining the diagnosis and treatment process.

**Help fill the shortage for medical experts:** the proposed system can give an initial diagnosis in almost no time with low cost in case there wasn't an available doctor to examine the patient at the moment.

**Help medicine students reinforce their diagnosis skills**

## **1.7 Organization of the project report**

**Remaining chapters will present the following**

**Chapter 2:** presents a literature review of the problem

**Chapter 3:** presents the phases of the project development, software and hardware requirements and the basic elements of the UI of the proposed system

**Chapter 4:** presents the development process

# **Chapter 2**

## **LITERATURE REVIEW AND BACKGROUND**

## **2.1 Introduction**

This chapter discusses the previous work done in each topic of the proposed system services which includes COVID-19 detection in sec. 2.2 cardiac abnormalities in sec. 2.3, chest abnormalities localization and categorization in sec.2.4, brain tumor detection and segmentation in sec.2.5, the skin cancer detection service in sec.2.6, and the background in sec. 2.7

## **2.2 COVID-19 detection state-of-the-art**

### **2.2.1 Survey**

Reverse transcription polymerase chain reaction (RT-PCR) is the standard technique for detecting COVID-19 infection, and results can be obtained in a few hours or two days, however, this test is time-consuming and expensive, and it also requires close contact between the doctor and the patient being tested, deep learning and other modern computer science approaches are being investigated by several researchers as means of detecting COVID-19 from lung scans, as in [2] the author combined conventional data augmentation methods with generative adversarial networks (GANs) to solve the issue of data limitation, while in [3] the author concluded that the frequency of CT findings was correlated with the progression of infection, in [4] features were extracted from the penultimate layer of an EfficientNet-based pre-trained model, and their dimensionality were decreased using kernel principal component analysis (PCA), random forest and support vector machine (SVM) were then used for prediction, the author of [6] utilized x-ray pictures of patients with lungs that were infected with COVID-19 and those with lungs that weren't to produce a data collection of x-ray images that was then used to predict COVID-19 automatically in patients, the COVID-19-infected x-ray pictures are distinguished from the non-infected x-ray images using a ResNet-50 pre-trained CNN model which achieved results of 98% accuracy, the author of [7] used deep learning methods on a set of x-ray pictures of 70 individuals who had COVID-19 verified. From a public chest x-ray image data collection, additional pictures of pneumonia patients have been added, the model is used to spot differences in x-ray pictures between COVID-19-infected individuals and patients with pneumonia, the suggested deep-learning model was able to detect COVID-19 instances with a 90% sensitivity and non-COVID-19 cases with an 87.24% specificity, in [8] the author made use of a custom deep-learning model named DarkNet to perform binary classification to distinguish between COVID-19 from normal cases and multi-class classifications to distinguish between COVID-19, pneumonia and normal cases, the experiments used a publicly available data set of COVID-19 x-ray images and another publicly available data set for non-infected and pneumonia x-ray images, the complete data set included 127 COVID-19 x-ray images and 500 pneumonia x-ray images

and 500 non-infected x-ray images, the results produced an accuracy score of 98% for binary classification and an accuracy score of 87.02% for multi-class classification.

### **2.2.2 Objective**

According to [46, 47], the main objectives of COVID-19 detection from chest x-ray images are:

1. Early detection: To develop a system that can accurately detect COVID-19 from chest x-ray images at an early stage, before symptoms become severe.
2. Improve patient management: To aid in the triage and management of COVID-19 patients, by providing a fast and reliable diagnostic tool to identify those who are likely to have the disease.
3. Reduce spread of disease: To help prevent the spread of COVID-19 by quickly identifying infected individuals and isolating them from the general population.
4. Aid in resource allocation: To assist healthcare professionals in allocating resources, such as hospital beds and ventilators, to patients who require urgent care.
5. Improve accuracy of diagnosis: To develop a system that can accurately distinguish between COVID-19 and other respiratory diseases, such as pneumonia, to avoid misdiagnosis and ensure appropriate treatment is provided.
6. Enhance scalability: To develop a system that is scalable, such that it can be deployed in various healthcare settings, including hospitals, clinics, and remote locations, to improve access to COVID-19 testing.

### **2.2.3 Limitations**

There are several limitations of COVID-19 detection from chest X-ray images according to [48, 49], some of which include:

1. Lack of specificity: Chest X-ray images can show abnormalities that are indicative of COVID-19, but these abnormalities can also be caused by other respiratory illnesses, such as pneumonia or bronchitis. This can lead to false positives and misdiagnosis.
2. Radiation exposure: X-ray imaging involves exposure to ionizing radiation, which can potentially increase the risk of cancer and other health problems, particularly if multiple images are taken over a short period of time.

3. Limited sensitivity: Chest X-ray images may not always be able to detect early stages of COVID-19 infection, especially in cases where symptoms are mild or absent.
4. Variability in imaging quality: The quality of chest X-ray images can vary depending on the imaging equipment used, the patient's positioning during the procedure, and other factors. This can lead to variability in the accuracy of COVID-19 detection.
5. Interpretation challenges: Interpreting chest X-ray images requires a high level of expertise and experience, which may not be readily available in all healthcare settings.
6. Lack of standardized protocols: There is currently no standardized protocol for using chest X-ray images for COVID-19 diagnosis, which can lead to variability in interpretation and diagnosis.
7. Insufficient data: There is still limited data available on the use of chest X-ray images for COVID-19 detection, particularly in comparison to other diagnostic tools such as PCR tests. This can limit the reliability and accuracy of the diagnosis

#### **2.2.4 Challenges and addressing**

According to [50, 51, 52], here are some challenges and how they can be addressed in COVID-19 detection from chest X-ray images:

1. Overlapping features with other respiratory diseases: One of the challenges in COVID-19 detection from chest X-ray images is the overlapping features with other respiratory diseases, such as influenza and pneumonia. This can lead to misdiagnosis and delayed treatment. To address this challenge, machine learning algorithms can be trained on a large dataset of chest X-ray images to accurately differentiate COVID-19 from other respiratory diseases.
2. Variability in imaging protocols: Another challenge is the variability in imaging protocols used in different healthcare facilities, which can affect the quality and interpretation of chest X-ray images. To address this challenge, standardized imaging protocols and reporting guidelines should be developed and implemented to ensure consistent and accurate interpretation of chest X-ray images.
3. Limited sensitivity and specificity: Chest X-ray has limited sensitivity and specificity in detecting COVID-19, especially in asymptomatic or early-stage patients. To address this challenge, chest X-ray can be used in conjunction with other diagnostic tools, such as reverse transcription-polymerase chain reaction (RT-PCR) and computed tomography (CT) scans, to improve the accuracy of COVID-19 detection.

4. Radiation exposure: Chest X-ray involves exposure to ionizing radiation, which can increase the risk of cancer and other health issues. To address this challenge, chest X-ray should only be used when clinically necessary, and the radiation dose should be kept as low as reasonably achievable.
5. Resource limitations: In resource-limited settings, access to chest X-ray machines and trained radiologists may be limited, which can hinder the timely and accurate diagnosis of COVID-19. To address this challenge, portable chest X-ray machines and telemedicine technologies can be used to provide remote access to expert radiologists and facilitate timely diagnosis and treatment of COVID-19.

## **2.3 Heartbeat abnormalities categorization state-of-the-art**

### **2.3.1 Survey**

A heart produces tiny electrical impulses which spread through the heart muscle, small electrical impulses from the heart go throughout the heart muscle, and an ECG machine can detect these impulses, normal, healthy cardiac ECGs have a distinctive shape, any cardiac rhythm irregularity or heart muscle damage can alter the electrical activity of the heart, changing the ECG's structure, the classification of the electrocardiogram (ECG) signal has a vital impact on the identification of heart-related diseases, this can ensure the premature finding of heart disease and the proper selection of the patient's customized treatment [9], in [10] the author combined a general multi-class classifier (incremental SVM) with a specific classifier (two-class SVM) and obtained an accuracy of 86%, the author of [11] performed Classification of ECG waveform using RS along with QNN where an overall accuracy of 91.7% was obtained, in [12] the author developed an inter-patient ECG classifier using 1D-CNN for the efficient finding of arrhythmia from 3-classes of heartbeats, while in [13] AlexNet coupled with a back-propagation neural network was implemented to detect dysrhythmia from the ECG waveform of three different heart conditions and an accuracy of 92% was achieved, Also, an accuracy of 92.70% was obtained by the author of [14] where 1D-CNN was used to categorize 5 classes of heartbeats from ECG signals, the author of [15] used a deep CNN approach for detection of arrhythmia and transfer this function to the detection of myocardial infarction analyzing the ECG heartbeat signals and showed an accuracy of 93.40%, the author of [16] used deep CNN to successfully identify arrhythmic heartbeats from the ECG data of 5 different classes of heartbeats obtaining an accuracy of 94.03%, Implementation of 2-D CNN to detect dysrhythmia by classifying (ECG) signals was done by the author of [17] where both the individual characteristics of the beat along with the beat-to-beat temporal relationship was captured which made the performance far superior than the 1-D approaches, Another 2-D deep CNN classifier was

developed in [18] for the efficient detection of arrhythmia from ECG signals which was further optimized by a few deep learning techniques.

### **2.3.2 Objectives**

Objective: To develop a system for accurate detection of heart beat abnormalities from electrocardiogram (ECG) signals [53]:

1. Preprocessing: a. Normalize the ECG signals to ensure consistency across recordings. b. Filter the signals to remove any noise or artifacts. c. Segment the signals into individual heartbeats.
2. Feature Extraction: a. Extract relevant features from the segmented heartbeats, such as amplitude, duration, and morphology. b. Apply statistical methods to the features to capture overall characteristics of the signals.
3. Classification: a. Train a classification model to distinguish between normal and abnormal heartbeats based on the extracted features. b. Evaluate the model's performance using validation datasets.
4. Interpretation: a. Provide a visual output indicating the location of the detected abnormalities on the ECG signal. b. Present information about the type and severity of the abnormality to medical professionals in a user-friendly format.
5. Optimization: a. Continuously refine the classification model and feature extraction algorithms to improve accuracy and reduce false positives. b. Ensure the system is scalable and adaptable to handle large datasets and a wide range of ECG signal variations.

### **2.3.3 Limitations**

Heart beat abnormalities detection from ECG has some limitations that must be taken into consideration [54]. Here are some of them:

1. Limited Sensitivity: While ECG is a widely used diagnostic tool for detecting heart abnormalities, it may not always capture all cardiac events. Some abnormalities may only occur intermittently, making them difficult to detect.
2. Need for Skilled Interpretation: Interpreting ECG signals requires specialized knowledge and training. Even with automated detection systems, a skilled medical professional is required to review the results and make a diagnosis.
3. False Positives and Negatives: Automated detection systems may produce false positives, i.e., indicating abnormalities where there are none, or false

negatives, i.e., failing to detect an abnormality. This can lead to unnecessary or missed treatments and procedures.

4. Variability in ECG Signals: ECG signals may vary depending on factors such as age, gender, medications, and overall health status. This can make it challenging to establish universal norms for what constitutes a normal or abnormal ECG signal.
5. Cost: ECG machines and other diagnostic equipment can be expensive, making them less accessible in some regions. Additionally, specialized training and equipment are required to interpret and analyze the signals.
6. Ethical Considerations: With the increasing use of artificial intelligence and machine learning in automated ECG analysis, there are ethical considerations around the privacy and security of patient data. There is also a risk of biases and discrimination if the algorithms are not designed and tested appropriately.

Overall, while ECG is a valuable diagnostic tool for detecting heart beat abnormalities, it is important to consider its limitations and address them appropriately to ensure accurate and effective diagnoses.

### **2.3.4 Challenges and addressing**

One of the main challenges in detecting heart beat abnormalities from electrocardiogram (ECG) signals is the presence of noise and artifacts that can interfere with the accurate detection of abnormal beats. This can include muscle artifacts, power line interference, and other electrical noise. Additionally, different types of heart beat abnormalities can have similar ECG waveforms, making it difficult to distinguish between them [55, 56].

To address these challenges, several techniques can be employed, including:

1. Signal pre-processing: This involves filtering the ECG signal to remove noise and artifacts. Different types of filters can be used, including high-pass, low-pass, and band-pass filters. This can help to improve the accuracy of abnormal beat detection.
2. Feature extraction: This involves identifying specific features of the ECG waveform that are characteristic of different types of heart beat abnormalities. For example, the duration of the QRS complex can be used to identify ventricular tachycardia. Machine learning algorithms can be used to automatically extract these features and classify abnormal beats.
3. Ensemble learning: This involves combining multiple machine learning algorithms to improve the accuracy of abnormal beat detection. For example, a decision tree algorithm can be combined with a support vector machine to improve the accuracy of ventricular tachycardia detection.

4. Augmenting the dataset: The accuracy of machine learning algorithms can be improved by increasing the size and diversity of the dataset used for training. This can involve collecting more data from different patients or augmenting the existing data through techniques such as data synthesis and data augmentation.
5. Clinical validation: It is important to validate the accuracy of abnormal beat detection algorithms in clinical settings. This can involve comparing the algorithm's performance to that of expert cardiologists or evaluating its ability to predict patient outcomes. This can help to identify areas where the algorithm can be improved and increase confidence in its accuracy.

## **2.4 Chest x-ray abnormalities localization state-of-the-art**

### **2.4.1 Survey**

Performing object detection for various conditions in chest x-rays is one of the most often observed uses of deep learning, automated detection can improve diagnosis of imaging, allowing for the early diagnosis of serious illness and having a big impact on patient outcomes, since the dataset considering the localization problem is recently introduced to the research community, this topic has not received significant investigation, the author of [19] used an object detection model called Detectron-2 and achieved a mean average precision (map) score of 0.235, the author also experienced the effect of some augmentation methods which are horizontal flip, random crop, random perspective transformation, random affine transformation and random color jitter with brightness = (0.8, 1.2), meanwhile, the author of [20] performed a three stage process of detecting whether the image in a posterior-anterior view or not then classifying it into normal and abnormal and finally localizing all findings on the abnormal images with bounding boxes and classify them, a mean average precision score (map) of 0.365 was achieved, also, in[25] the author presented the SVM algorithm for the categorization of cardiomegaly and atelectasis, after segmenting chest x-ray (CXR) images to identify the region of interest (ROI) then gray-level transformation techniques were used to enhance them.

Bone suppression is a technique that can be applied on CXR images, several researches explored bone elimination such as in [21] the author introduced a convolutional neural filter (CNF) that eliminates all bone components while maintaining tissue information, meanwhile, the author of [22] introduced a bone suppression algorithm based on CXR image gradient discrepancies, also, the author of [23] proposed a CNN model utilizing a dataset of CXR radiographs with pulmonary nodules to separate bone structures and soft tissues in various lung areas, furthermore, in [24] the author proposed a DCNN model called DeBoNet that removes bones in CXR images.

## **2.4.2 Objectives**

According to [57], objectives of chest x-ray abnormalities localization and classification include:

1. To improve the accuracy and speed of chest x-ray interpretation.
2. To aid in the early detection and diagnosis of pulmonary diseases and abnormalities.
3. To provide a basis for treatment planning and monitoring.

## **2.4.3 Limitations**

According to [58, 59, 60], some of the limitations of chest x-ray abnormalities localization and classification include:

1. Limited accuracy: While deep learning models have shown promising results in classifying chest x-ray abnormalities, their accuracy is not yet comparable to that of human radiologists.
2. Limited interpretability: Deep learning models can be difficult to interpret, which can limit their usefulness in clinical settings where physicians need to understand the reasoning behind a diagnosis.
3. Limited generalizability: Deep learning models trained on one dataset may not generalize well to other datasets or patient populations, which can limit their usefulness in real-world clinical settings.
4. Limited sensitivity and specificity: While deep learning models can be highly sensitive in detecting abnormalities, they can also produce false positives or miss subtle abnormalities, leading to limited specificity.
5. Limited availability of labeled data: Deep learning models require large amounts of labeled data to train, and there may not be enough annotated data available for certain rare or complex abnormalities.
6. Ethical concerns: The use of AI models in medical diagnosis raises ethical concerns around issues such as bias, accountability, and transparency.

## **2.4.4 Challenges and addressing**

According to [61, 62, 63], here are some challenges associated with chest x-ray abnormalities localization and classification:

Limited availability of labeled data: One major challenge in developing accurate deep learning models for chest x-ray abnormalities localization and classification is the limited availability of labeled data. One potential solution to this challenge is the use of transfer learning, which allows models trained on large datasets to

be fine-tuned on smaller, more specific datasets, transfer learning has been shown to improve the performance of deep learning models for chest x-ray analysis

1. Limited interpretability: Deep learning models can be difficult to interpret, which can limit their usefulness in clinical settings where physicians need to understand the reasoning behind a diagnosis. One potential solution to this challenge is the use of explainable AI techniques, which aim to provide insights into how deep learning models arrive at their decisions. Recent studies have explored the use of explainable AI in chest x-ray analysis (Wang et al., 2021; Zhang et al., 2021).
2. Limited generalizability: Deep learning models trained on one dataset may not generalize well to other datasets or patient populations, which can limit their usefulness in real-world clinical settings. One potential solution to this challenge is the use of multi-center datasets that are more representative of diverse patient populations (Huang et al., 2021).
3. Limited sensitivity and specificity: While deep learning models can be highly sensitive in detecting abnormalities, they can also produce false positives or miss subtle abnormalities, leading to limited specificity. One potential solution to this challenge is the use of ensemble models that combine multiple deep learning models to improve both sensitivity and specificity (Wu et al., 2021).
4. Ethical concerns: The use of AI models in medical diagnosis raises ethical concerns around issues such as bias, accountability, and transparency. Recent studies have explored these ethical concerns in the context of chest x-ray analysis and proposed solutions to address them (Cramer et al., 2021; Shickel et al., 2021).

## 2.5 Brain tumor state-of-the-art

One of the most difficult and time-consuming tasks in medical image processing is the identification and segmentation of brain tumors, magnetic resonance imaging (MRI) is a medical procedure that allows radiologists to visualize the human body without having to do any surgery.

### 2.5.1 Brain tumor categorization

#### 2.5.1.1 Survey

Several classification method was introduced to detect and analyze brain tumors from MRI images, in [27] a method to extract features from the relationship

between tumor and brain lateral ventricular (LaV) deformation has been proposed, and the features are then used for tumor segmentation in (MRI) images, while in [28] features (WST) and wavelet co-occurrence texture feature (WCT) obtained from two level discrete wavelet transform is used for the classification of abnormal brain tissues using a SVM classifier which achieved an accuracy of 97.5%, also, in [29] the author introduced a method for the classification of MRI images in to normal and abnormal one using KNN and SVM classifiers and the accuracy was 97.62% and 98.21% respectively, furthermore, the author of [30] proposed a method to improve the classification of brain tumor from MRI images based on spectral angle based feature extraction and spectral clustering independent component analysis (SC-ICA) along with the SVM classifier achieving an accuracy of 98%, meanwhile, in [31] the author present a neural network technique for the classification of magnetic resonance brain images achieving an accuracy of 73%, also, in [32] the author proposed an automatic method that make use of the capability of back propagation (BP) and Radial Basis Function (RBF) neural network function to classify brain image in to cancerous or non-cancerous one, furthermore, the author of [33] proposed a hybrid approach for the classification of brain tissue in MRI images based on genetic algorithm and SVM, the accuracy of the proposed system varies from 96.37 to 98.99%, meanwhile, in [34] the author introduced a type II fuzzy expert system for T1 weighted MR image-based diagnosis of human brain tumors, furthermore, the author of [35] presented a hybrid approach based on SVM and genetic algorithms for the categorization of brain tissue in MRI images, the accuracy of the proposed system varies from 96.37 to 98.99%, also, in [36] the author proposed a method of intelligent categorization for distinguishing between normal and abnormal brain slices, the least square support vector machine (LS-SVM) was proposed, the classification features are obtained from the slices, this classifier utilized both linear and nonlinear radial basis function (RBF) kernels, the achieved accuracy was about 98.64%.

### **2.5.1.2 Objectives**

Objectives of Brain Tumor Classification from MRI Images [64, 65]:

1. Accurate classification of brain tumors: The primary objective of brain tumor classification from MRI images is to accurately classify brain tumors into different categories based on their characteristics, such as size, shape, location, and intensity. This can aid in the selection of appropriate treatment options and improve patient outcomes.
2. Early detection of brain tumors: Another objective is to detect brain tumors at an early stage when they are more treatable. MRI is a sensitive imaging modality for detecting brain tumors, and machine learning algorithms can

be trained on a large dataset of MRI images to identify early signs of brain tumors.

3. Reduction of diagnostic errors: Brain tumor classification from MRI images can help reduce diagnostic errors and improve the accuracy of brain tumor diagnosis. Machine learning algorithms can learn from a large number of MRI images and identify patterns that may be missed by human radiologists.
4. Personalized treatment planning: Accurate classification of brain tumors from MRI images can aid in personalized treatment planning for patients. Different types of brain tumors require different treatment strategies, and accurate classification can help select the most effective treatment option for each patient.

### **2.5.1.3 Limitations**

Limitations of Brain Tumor Classification from MRI Images [66]:

1. Limited dataset size: One major limitation of brain tumor classification from MRI images is the limited size of available datasets. This can affect the performance of machine learning algorithms and limit their accuracy.
2. Lack of standardization in MRI protocols: MRI protocols can vary between institutions, which can affect the quality and consistency of the images used for brain tumor classification. This can also affect the accuracy of machine learning algorithms trained on these images.
3. Complexity of brain tumor classification: Brain tumor classification from MRI images is a complex task that involves identifying subtle differences in tumor characteristics. Machine learning algorithms may struggle to accurately classify brain tumors when there is significant overlap between different tumor types.
4. Interpretability of machine learning models: Machine learning algorithms can be highly accurate at classifying brain tumors, but their outputs may not be easily interpretable. This can make it difficult for radiologists to understand how the algorithm arrived at its classification decision.

### **2.5.1.4 Challenges and addressing**

Challenges and Solutions for Brain Tumor Classification from MRI Images according to [67]:

1. Variability in image acquisition and preprocessing techniques: MRI images can be acquired using different protocols and machines, leading to variability in the images. Preprocessing techniques can also differ, which

can impact the accuracy of classification models. Solution: Standardization of image acquisition protocols and preprocessing techniques can help to reduce variability.

2. Class imbalance: Brain tumors are relatively rare, which can result in class imbalance in the dataset. Solution: Data augmentation techniques such as oversampling of the minority class can be used to balance the dataset.
3. Interpretability of deep learning models: Deep learning models can be highly accurate, but they are often considered as "black boxes" and their decision-making process is not transparent. Solution: Techniques such as gradient-weighted class activation mapping (Grad-CAM) can be used to visualize the regions of the MRI images that contribute most to the model's classification decision.
4. Limited availability of annotated data: Annotated MRI datasets are relatively scarce, which can limit the development and validation of classification models. Solution: Transfer learning techniques can be used to leverage pre-trained models and reduce the amount of annotated data required.

## **2.5.2 Brain tumor segmentation from MRI images**

### **2.5.2.1 Survey**

Brain tumor segmentation is a critical task in medical image analysis and plays a significant role in diagnosis, treatment planning, and monitoring the progression of brain tumors, the segmentation process involves the identification and delineation of the boundaries of the tumor from the surrounding healthy tissues in the MRI images, in [37] the author designed a model by embedding densely connected dilated convolutional blocks in a shallow 2D U-net-style structure and achieved mean dice score of 85%, in [38] the author proposed an asymmetric encoder-decoder architecture with a larger encoder part for extracting more image features and achieved a dice score of 86%, in [39] the author proposed a slightly modified version of

a U-Net based model using leaky ReLUs instead of traditional ReLUs and in order to produce more consistent results, instance normalization was also applied achieving a dice score of 85%, in [40] the author proposed an ensemble of cascaded CNNs achieving a dice score of 83%, in [41] the author proposed a U-Net architecture very similar to a generic U-Net achieving a dice score of 86%, in [42] the author proposed a residual cyclic unpaired encoder-decoder network (RescueNet) achieving a dice score of 85%, in [43] the author proposed a modified version of U-Net context modules consisted of a pre-activation residual block achieving a dice score of 79%.

### **2.5.2.2 Objectives**

Objectives of Brain Tumor Segmentation from MRI Images include [68]:

1. To develop accurate and reliable methods for segmenting brain tumors from MRI images.
2. To improve the accuracy of diagnosis and treatment planning for brain tumor patients through segmentation of MRI images.
3. To facilitate the development of computer-aided diagnosis (CAD) systems for brain tumor detection and classification.

### **2.5.2.3 Limitations**

Limitations of Brain Tumor Segmentation from MRI Images include [69]:

1. Variability in imaging protocols: There is variability in imaging protocols across different studies, which can affect the comparability and reproducibility of the results.
2. Complexity of tumor boundaries: Tumor boundaries in MRI images can be complex and difficult to accurately delineate, which can affect the accuracy of segmentation methods.
3. Limited validation: Some studies lack external validation of their segmentation methods, which can make it difficult to assess the accuracy and reproducibility of the methods used.
4. Limited focus on non-glioblastoma tumors: Most studies have focused on glioblastoma tumors, with limited research on other types of brain tumors.
5. Limited clinical translation: Despite promising results from some studies, there is limited clinical translation of brain tumor segmentation from MRI images for use in clinical practice.

It is important to acknowledge these limitations when interpreting the findings of recent studies on brain tumor segmentation from MRI images. Further research is needed to address these limitations and improve the clinical utility of this approach for brain tumor patients.

### **2.5.2.4 Challenges and addressing**

Challenges and how they can be addressed in Brain Tumor Segmentation from MRI Images include [70]:

1. Standardization of imaging protocols: The variability in imaging protocols across different studies can affect the comparability and

reproducibility of the results. Standardization of imaging protocols could help overcome this challenge.

2. Improved segmentation methods: The complexity of tumor boundaries in MRI images can be a challenge for accurate segmentation. Development of improved segmentation methods, such as machine learning-based approaches, could help overcome this challenge.
3. Validation: External validation of segmentation methods by independent research groups could help ensure the accuracy and reproducibility of the results.
4. Large-scale validation studies: Large-scale validation studies with diverse patient populations could help overcome the limitations of small sample sizes and limited generalizability of results.
5. Integration into clinical practice: Integration of brain tumor segmentation from MRI images into clinical practice requires demonstration of its clinical utility and practical implementation. Collaboration between researchers, clinicians, and industry could help address this challenge.

## 2.6 Skin cancer detection state-of-the-art

### 2.6.1 Survey

Skin cancer detection is of paramount importance in the realm of healthcare, the early identification of skin cancer plays a crucial role in ensuring timely intervention and improving patient outcomes, in [85] the author proposed a NN ensemble model which is a combination of backpropagation and fuzzy neural networks achieving an accuracy of 91.11%, while in [86] the author proposed an ANN model associated with Levenberg–Marquardt (LM) learning algorithm which achieved achieving a recall of 92.6%, furthermore, in [87] the author utilized genetic algorithms and ANN with hair removal preprocessing achieving an accuracy of 88%, also, in [88] the author proposed a 16 residual block CNN combined with the average of SVM and softmax classifier achieving 85.5% accuracy, in [89] the author combined AlexNet, ResNet-18 and VGG-19 pre-trained models as a feature extractor and used the SVM classifier to classify the extracted features achieving an accuracy of 83.83%, in [90] the author utilized the AlexNet pre-trained model with error-correcting output coding SVM classifier achieving an accuracy of 94.17%, in [91] the author proposed a hybrid of fully CNN with auto encoder and decoder and RNN achieving 98% accuracy.

## **2.6.2 Objectives**

Objectives of skin cancer detection include [92]:

1. To develop accurate and reliable methods for detecting and diagnosing skin cancer from various imaging modalities, such as dermoscopy, reflectance confocal microscopy, and multispectral imaging.
2. To improve the accuracy of diagnosis and treatment planning for skin cancer patients through early detection and precise identification of malignant lesions.
3. To facilitate the development and implementation of computer-aided diagnosis (CAD) systems for automated skin cancer detection and classification.

## **2.6.3 Limitations**

Limitations of Brain Tumor Segmentation from MRI Images include [93]:

- 3 Variability in imaging techniques: Different imaging techniques and protocols are employed across studies, which can impact the comparability and reproducibility of results.
- 4 Complexity of lesion analysis: Skin cancer lesions can exhibit diverse morphological and textural features, making accurate analysis and segmentation challenging.
- 5 Limited validation: Some studies may lack external validation of their detection methods, which is crucial for assessing their accuracy and generalizability.
- 6 Focus on specific skin cancer types: Research efforts have predominantly focused on melanoma, with limited attention given to other types of skin cancer, such as basal cell carcinoma and squamous cell carcinoma.
- 7 Limited clinical translation: Despite promising results in research settings, the translation of skin cancer detection methods into clinical practice remains limited.

## **2.6.4 Challenges and addressing**

Challenges and how they can be addressed in Brain Tumor Segmentation from MRI Images include [94]:

1. Standardization of imaging techniques: Standardization of imaging techniques, acquisition parameters, and imaging protocols can enhance comparability and reproducibility of results across different studies.

2. Advanced image analysis algorithms: Developing advanced image analysis algorithms, including machine learning and deep learning approaches, can help improve the accuracy and efficiency of skin cancer detection.
3. External validation: External validation of detection methods using independent datasets and collaborations with multiple research groups can strengthen the reliability and generalizability of results.
4. Large-scale validation studies: Conducting large-scale validation studies involving diverse patient populations and clinical settings can help overcome the limitations of small sample sizes and improve the clinical applicability of skin cancer detection methods.
5. Integration into clinical practice: Collaboration between researchers, clinicians, and industry stakeholders is essential for the integration of skin cancer detection methods into routine clinical practice. This includes demonstrating the clinical utility and feasibility of the methods in real-world settings.

In the wake of recent advancements in computer science and informatics, the utilization of artificial intelligence (AI) algorithms and other AI-powered applications has experienced a remarkable upsurge, establishing itself as an integral component of contemporary healthcare. These AI-based applications are being extensively explored in research endeavors aimed at augmenting and empowering medical professionals, the presented project endeavors to make a meaningful contribution to this rapidly expanding field through a series of experiments conducted on topics that have previously demonstrated.

## 2.7 Background

### 2.7.1 Convolution neural networks (CNN)

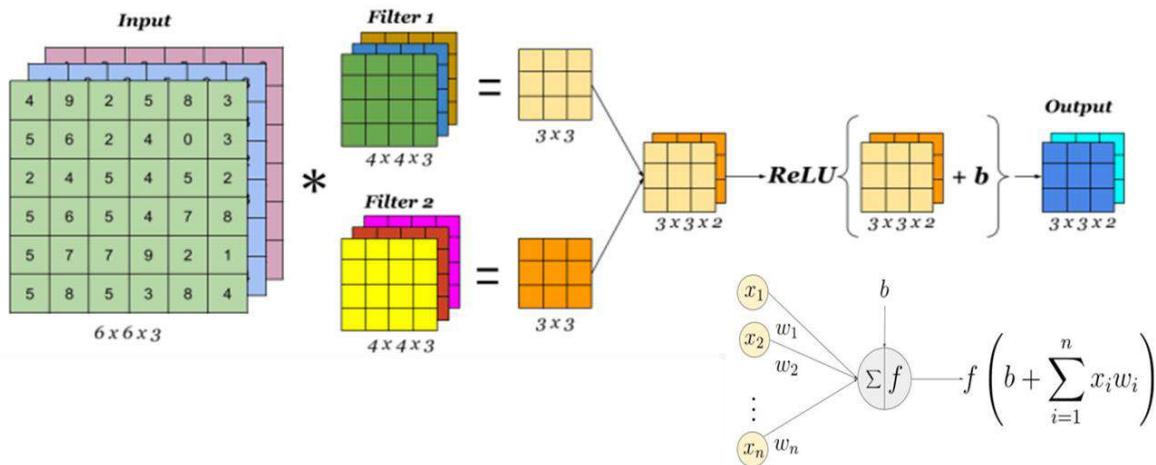


Figure 2- 1 Convolution layer

Convolutional Neural Network (CNN), is a deep learning algorithm that is widely used for image classification and signal classification tasks, it is particularly effective in handling data with a grid-like structure, such as images or time series data. CNNs have revolutionized the field of computer vision and have achieved state-of-the-art performance in various image-related tasks.

At a high level, CNNs consist of multiple interconnected layers, including convolutional layers, pooling layers, and fully connected layers, these layers work together to extract and learn meaningful features from the input data, allowing the network to make accurate predictions.

The main idea behind CNNs is to use convolutional layers to apply a set of filters to the input image or signal. Each filter represents a small receptive field and slides over the input, performing element-wise multiplication and summation (convolution operation) to produce a feature map, the filters capture different patterns or features present in the data, such as edges, textures, or shapes.

The pooling layers, typically applied after convolutional layers, reduce the spatial dimensions of the feature maps while retaining the most salient information, common pooling operations include max pooling or average pooling, which downsample the feature maps by taking the maximum or average value within a defined neighborhood.

After several convolutional and pooling layers, the resulting feature maps are flattened and fed into fully connected layers, these layers connect every neuron

to every neuron in the previous and subsequent layers, they act as a traditional artificial neural network and learn to classify the extracted features into different classes.

During the training process, CNNs learn the optimal values of the filters and weights in the network by minimizing a loss function, typically through backpropagation and gradient descent optimization, this process allows the network to adjust its parameters and improve its ability to recognize and classify patterns in the data.

For image classification, CNNs have achieved remarkable performance on benchmark datasets such as ImageNet, which contains millions of labeled images from thousands of classes, by learning hierarchical representations of features, CNNs can distinguish objects and recognize complex patterns with high accuracy. They have been applied to various image-related tasks, including object detection, image segmentation, and image generation.

Similarly, CNNs can be used for signal classification tasks, such as speech recognition or biomedical signal analysis. In this context, the input data is often represented as a time series or a spectrogram, and the CNN architecture is adapted accordingly to handle the specific characteristics of the signals. By leveraging the hierarchical feature extraction capability of CNNs, they can effectively capture temporal or spectral patterns in the signals and make accurate predictions.

### 2.7.2 Long-short time memory (LSTM)

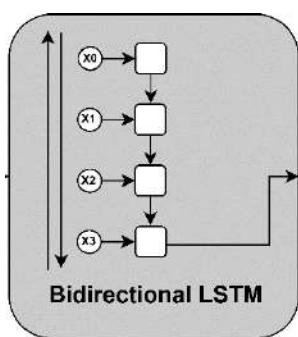


Figure 2- 2 Long-short time memory

Long Short-Term Memory, is a type of recurrent neural network (RNN) architecture that is specifically designed to model and capture long-term dependencies in sequential data, it is widely used in tasks involving temporal data, such as natural language processing, speech recognition, time series analysis, and more.

The key component of an LSTM unit is the memory cell, which allows the network to store and access information over long sequences, the memory cell has a self-loop connection, enabling it to retain information over time. It can selectively update and erase its content, controlled by gates that regulate the flow of information.

An LSTM unit typically consists of three main gates: the input gate, forget gate, and output gate, these gates are responsible for controlling the information flow in and out of the memory cell. Here's a brief description of each gate:

**Input Gate:** The input gate determines how much new information is added to the memory cell, it takes the current input and the previous hidden state as input, applies a sigmoid activation function to transform the values between 0 and 1, and then applies a hyperbolic tangent activation function to create the new candidate values.

**Forget Gate:** The forget gate decides what information should be discarded from the memory cell, it takes the current input and the previous hidden state as input, applies a sigmoid activation function, and multiplies it with the previous cell state to determine which information to retain and which to forget.

**Output Gate:** The output gate regulates the output of the LSTM unit. It takes the current input and the previous hidden state as input, applies a sigmoid activation function to determine which parts of the memory cell should be outputted, and then applies a hyperbolic tangent activation function to squash the values between -1 and 1.

By using these gates, LSTM networks can selectively update and retain important information over long sequences, effectively capturing long-term dependencies in the data, this makes them well-suited for tasks that involve understanding and generating sequences of variable lengths.

### 2.7.3 Multi-head attention (MHA)

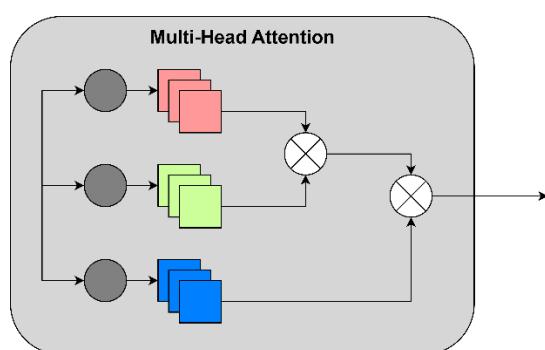


Figure 2- 3 Multi-head attention

Multi-head attention [95] is a component of transformer architectures that helps capture dependencies in sequential data, it differs from LSTM by operating without recurrence and allowing parallel computation, it attends to different parts of the input sequence simultaneously, capturing both local and global dependencies effectively, in time series data, multi-head attention can be used to weigh the importance of different time steps, enabling the model to capture complex patterns and relationships beyond sequential order, this makes it valuable for tasks such as time series forecasting, anomaly detection, and signal processing.

When applied to time series data, multi-head attention allows the model to weigh the importance of different time steps and capture temporal relationships effectively, by attending to various parts of the input sequence simultaneously, multi-head attention can identify and emphasize significant patterns and features at different scales.

For instance, in time series forecasting, the model can utilize multi-head attention to focus on relevant past observations, capturing both short-term and long-term dependencies, this enables the model to make accurate predictions by leveraging the learned patterns and relationships in the data.

#### 2.7.4 Adaptive feature recalibration (AFR)

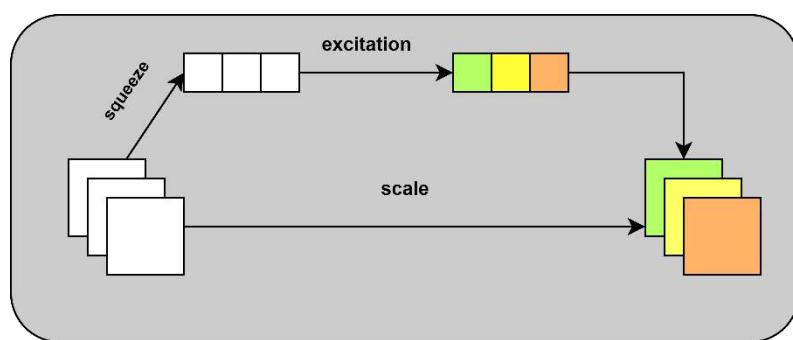


Figure 2- 4 Adaptive feature recalibration (AFR)

Adaptive feature recalibration (AFR) [95] using the Squeeze-and-Excitation (SE) mechanism is a technique employed in deep learning models to dynamically recalibrate the importance of features within a network. It aims to enhance the model's representational power by adaptively reweighting features based on their relevance to the task at hand.

The SE mechanism consists of two main steps: squeezing and exciting. In the squeezing step, the spatial dimensions of the input feature maps are reduced to a channel-wise descriptor by applying global average pooling, this reduces the feature maps into a vector of channel-wise statistics that capture the information across the entire spatial extent.

In the exciting step, the channel-wise descriptor is fed into a small neural network consisting of fully connected layers, these layers learn to model the interdependencies between different channels and capture the importance of each channel, the network usually consists of a bottleneck layer followed by a series of fully connected layers with appropriate activations, the output of the network represents the channel-wise importance scores

### 2.7.5 Adaptive kernel size convolution (AKSC)

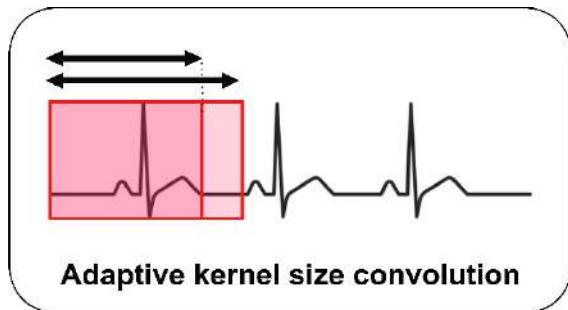


Figure 2- 5 Adaptive kernel size convolution diagram

Adaptive kernel size convolution is a technique in deep learning that allows the kernel size of convolutional layers to be dynamically adjusted during training. In traditional convolutional neural networks (CNNs), the kernel size is fixed and determined prior to training, however, adaptive kernel size convolution introduces the concept of trainable kernel sizes, enabling the network to learn the optimal kernel size for each convolutional operation.

In adaptive kernel size convolution, the kernel size is treated as a learnable parameter rather than a fixed value, during the training process, the network updates and adjusts the kernel size based on the learned representations and the task at hand. This adaptive nature allows the network to adaptively capture spatial dependencies and patterns of different scales within the input data.

### 2.7.6 Vision transformer model (ViT)

The Vision Transformer (ViT) is a transformative architecture that applies the power of self-attention and transformer models to computer vision tasks. Traditionally, convolutional neural networks (CNNs) have dominated the field of computer vision, but ViT offers an alternative approach that has shown impressive results, ViT breaks away from the reliance on convolutional operations and replaces them with self-attention mechanisms.

The core idea behind ViT is to treat images as sequences of patches rather than a grid of pixels, these patches are then linearly embedded to create token representations, similar to the word embeddings used in natural language

processing. The resulting tokenized image sequence is then fed into a transformer encoder, consisting of multiple layers of self-attention and feed-forward neural networks, this allows the model to capture global dependencies and relationships between patches in the image.

### 2.7.7 U-Net segmentation model

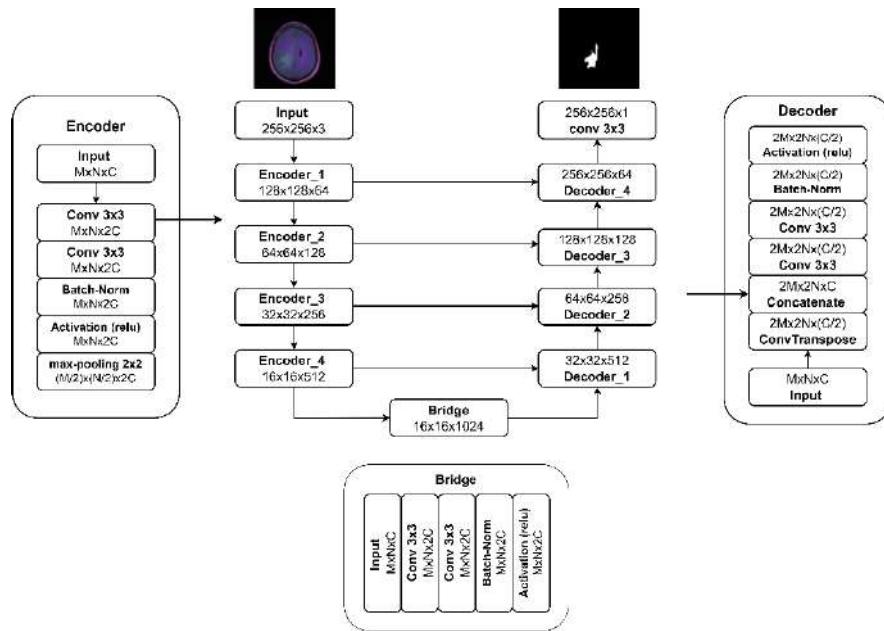


Figure 2- 6 U-Net model architecture example

The U-Net model is a widely used architecture for binary mask segmentation, a task in computer vision where the goal is to classify each pixel in an image as either belonging to the foreground or the background, the U-Net architecture is particularly effective for this task as it combines both localization and contextual information.

The U-Net architecture consists of an encoder-decoder structure. The encoder path, similar to a traditional convolutional neural network, gradually reduces the spatial dimensions of the input image while extracting increasingly abstract features, this encoder path captures the contextual information by analyzing the entire image.

On the other hand, the decoder path takes the learned features from the encoder and gradually upsamples them to the original image size. Each upsampling step is performed using transposed convolutions, which allow the learned features to be projected to higher-resolution feature maps, by combining these high-resolution feature maps with the corresponding low-resolution feature maps from the encoder path, the U-Net model effectively integrates both local and global information.

The skip connections, which directly connect the encoder and decoder at various levels, play a critical role in the U-Net architecture, these skip connections enable the model to recover spatial information and preserve fine-grained details during the up-sampling process. They act as shortcuts for propagating localized information from the encoder to the decoder, improving the model's ability to localize and segment objects accurately.

### **2.7.8 YOLO v5 object detection model**

YOLO (You Only Look Once) v5 is a state-of-the-art object detection model that has gained significant popularity in computer vision tasks. It is known for its speed and accuracy in real-time object detection, YOLO v5 builds upon the success of its predecessors, YOLO v1, v2, v3, and v4, by introducing improvements in architecture and training techniques.

The YOLO v5 architecture follows a single-shot detection approach, meaning it directly predicts bounding boxes and class probabilities for objects in an image in a single pass, this is in contrast to two-stage detectors that employ region proposal methods. YOLO v5 achieves high efficiency by utilizing a convolutional neural network backbone, specifically a modified version of the popular EfficientNet architecture.

The architecture consists of a series of convolutional layers, followed by a feature pyramid network (FPN) that extracts features at different scales to handle objects of various sizes, the FPN merges features from different layers to create a comprehensive representation that captures both local and global contextual information.

YOLO v5 has been widely adopted and applied in various domains, including autonomous driving, surveillance systems, and robotics, its ability to detect and track objects efficiently makes it suitable for real-time applications where fast and accurate object detection is essential.

### **2.7.9 PyQt5 GUI module and PyQt5 Designer**

PyQt5 is a powerful Python binding for the Qt framework, which is a popular cross-platform toolkit for building graphical user interfaces (GUIs), PyQt5 allows developers to create desktop applications with rich user interfaces, making it a versatile choice for GUI development in Python.

PyQt5 provides a comprehensive set of modules and classes that enable developers to create interactive and visually appealing applications, it supports a

wide range of widgets, such as buttons, labels, text boxes, and tables, which can be easily customized and arranged to design intuitive user interfaces.

One of the key features of PyQt5 is the integration with Qt Designer, a visual design tool that simplifies the process of building GUIs. Qt Designer allows developers to create and modify GUI layouts using a drag-and-drop interface, eliminating the need to write extensive code for UI design, with PyQt5, developers can take advantage of Qt Designer to visually create the UI components and then seamlessly integrate them into their Python code.

Using Qt Designer, developers can design complex UIs by placing widgets, setting properties, and defining signal-slot connections. It provides a visual representation of the application's UI, allowing developers to preview and refine the design before implementing the functionality, once the UI design is finalized, the Qt Designer saves the layout in a .ui file, which can be loaded and used in Python code using the PyQt5 library.

### **2.7.10 HTML, CSS and JavaScript**

HTML, CSS, and JavaScript are the fundamental technologies used to create and design websites, they work together to provide structure, style, and interactivity to web pages.

HTML (Hypertext Markup Language) is the markup language used to structure the content of a web page, it consists of a series of tags that define different elements and their relationships on the page, HTML tags define headings, paragraphs, images, links, tables, forms, and more. By using these tags, developers can create the overall structure and layout of the web page, including headings, paragraphs, lists, and other content elements.

CSS (Cascading Style Sheets) is a style sheet language that is used to control the visual presentation of a web page, it provides the styling and formatting instructions to define the colors, fonts, spacing, layout, and other visual aspects of the HTML elements. CSS allows developers to separate the presentation layer from the content layer, making it easier to manage and update the visual design of a website, CSS can be applied directly within HTML using inline styles, embedded within the HTML document using `<style>` tags, or linked externally as a separate CSS file.

JavaScript is a dynamic programming language that enables interactivity and behavior on web pages, it adds functionality to web pages by allowing developers to manipulate the HTML elements and respond to user actions. With JavaScript, developers can create dynamic effects, handle form validation, perform

calculations, fetch and display data from servers, and create interactive features such as sliders, dropdown menus, and image carousels, JavaScript can be embedded within HTML using <script> tags or stored externally in separate .js files.

Together, HTML, CSS, and JavaScript form the backbone of modern web development, HTML provides the structure, CSS handles the visual presentation, and JavaScript adds interactivity and functionality to create a complete web experience, They work in tandem to create responsive, visually appealing, and interactive websites that can adapt to different devices and user interactions

### **2.7.11 Flask module**

Flask is a popular web framework written in Python that allows developers to build web applications quickly and efficiently, it is known for its simplicity, flexibility, and ease of use, making it a great choice for both beginners and experienced developers.

One of the key features of Flask is its lightweight and minimalist design. It provides only the essential tools and features needed for web development, allowing developers to have fine-grained control over the application's structure and behavior, flask follows the "micro" framework approach, which means it provides the core functionality required for web development while leaving additional features and extensions as optional choices.

Flask follows the WSGI (Web Server Gateway Interface) specification, which makes it compatible with a wide range of web servers, including the built-in development server that comes with Flask itself, this flexibility allows developers to choose the web server that best suits their needs.

Flask provides a routing mechanism that maps URLs to specific functions, allowing developers to define routes and handle HTTP requests easily. It supports various HTTP methods such as GET, POST, PUT, and DELETE, enabling the creation of RESTful APIs, with Flask, developers can define routes, extract data from request parameters, and return responses with just a few lines of code.

Flask also supports the use of templates, which are HTML files with placeholders for dynamic content, this allows developers to separate the presentation logic from the application's core logic, flask integrates with Jinja2, a powerful templating engine, which provides features like template inheritance, loops, conditionals, and filters, this makes it easier to generate dynamic HTML pages with data from the application.

# **Chapter 3**

## **METHODOLOGY AND PROPOSED FRAMEWORK**

### 3.1 Introduction

This chapter discusses the development methodology of the project, sec3.2 presents an over view of how the MDS system framework operates, sec. 3.3 presents the development phases of the system, sec. 3.4 presents the GUI and deployment process.

### 3.2 Over view of the MDS framework

The MDS system is a collection of medical services based on AI models that help doctors with some diagnosis issues as discussed before in the significance section of chapter 1 of the book, these AI models will be deployed in a desktop application and as well as a web site, the user would be able to choose between services offered by the system and then upload an input image to the system chest x-ray (CXR) image for example and the system will output the diagnosis of the image based on the output of the AI model of the corresponding service.

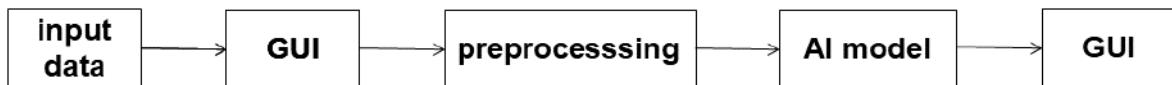


Figure 3-1 General block diagram for the MDS system operation

The skin cancer detection service take an image input and the AI model of the will output a string classifying the image as shown in the previous figure, on the other hand, the heartbeat abnormalities service's input is a time series data either obtained from biomedical signal file (.dat, .mat files) or produced by an ECG sensor and sent to the system via a microcontroller utilizing the MQTT protocol.

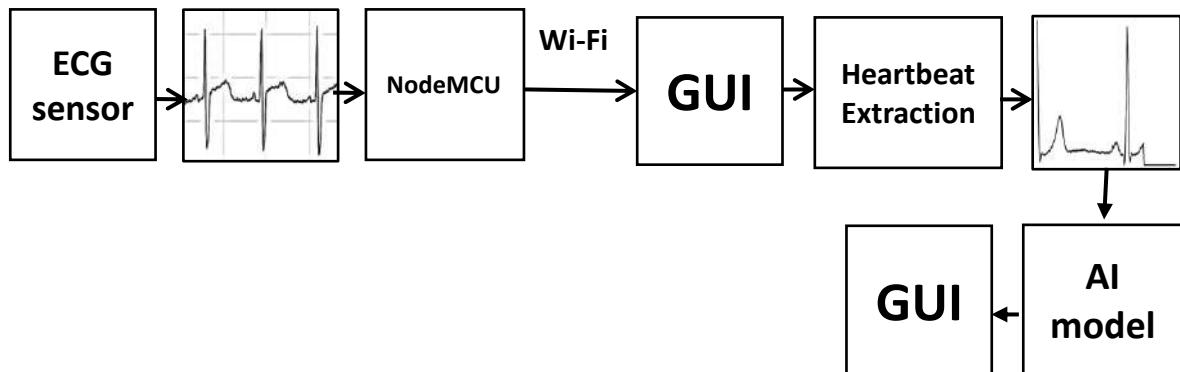


Figure 3-2 Block diagram for the real time heart beat abnormalities service

Also, the chest x-ray abnormalities localization and categorization service will

utilize an object detection model, the input will be a CXR image and the output will be the exact same image with bounding boxes localizing findings in it and a label for each one.

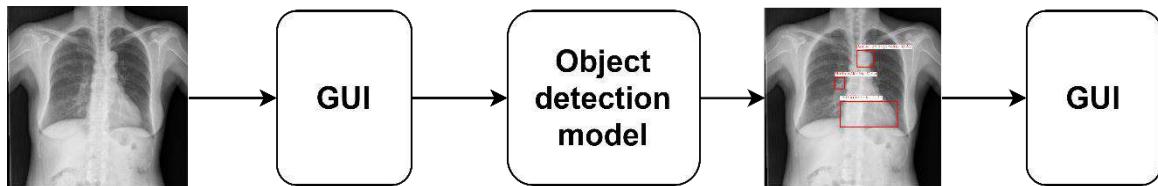


Figure 3-3 Block diagram for chest x-ray abnormalities localization service

The brain tumor detection and segmentation service takes an MRI image as an input and pass it to a classification model which will output a string classifying the image then the image will be passed to a segmentation model if it wasn't diagnosed to be normal, the segmentation model will output the same input image with the tumor region highlighted.

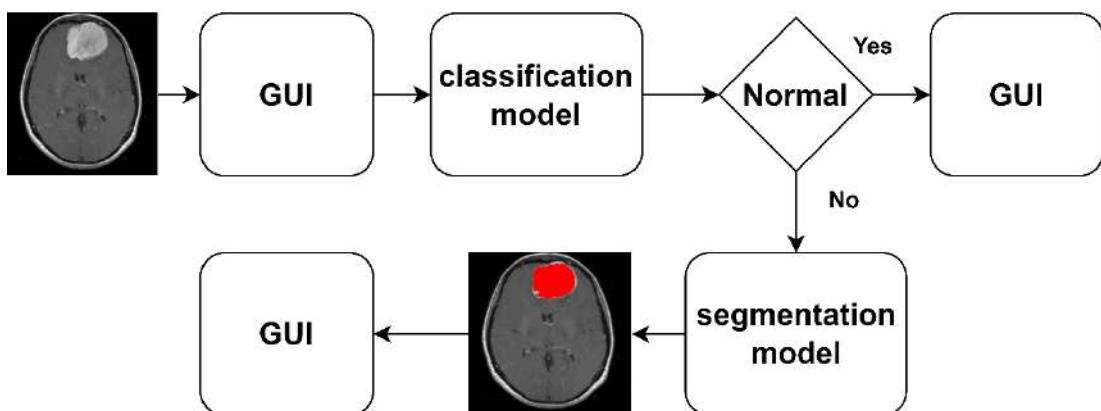


Figure 3-4 block diagram for the brain tumor detection and segmentation service

The covid-19 detection service take a x-ray image of the chest as an input and firstly pass it to a classification model which will output a string classifying the image, if the image wasn't classified to be normal then the GradCam visualization technique will be utilized to visualize the part of the image by which the model managed to classify the image.

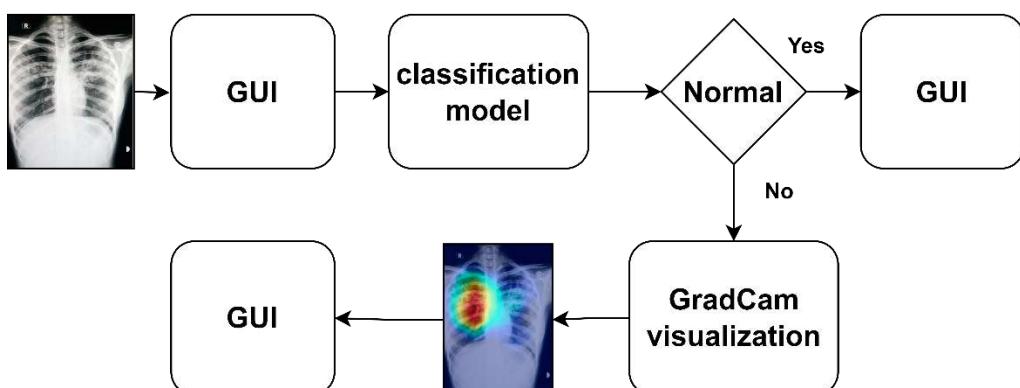


Figure 3-5 block diagram of the covid-19 detection service

### 3.3 Development process

The development process of the project will consist of two phases, in the first phase some AI models corresponding to some proposed services will be developed as well as some parts of the desktop application so that these services could be deployed and to show how the system operates, in the second phase all services will be fully developed achieving a good performance compared to the state-of-the-art, the web application, and the desktop application will be fully designed and developed as well to deploy all of the proposed services in both of them.

AI models will be developed using Python programming language and Keras framework from google since it's the most used framework by AI developers, the development process of an AI model consists of identifying the problem, collecting the dataset and preparing it or utilizing a public dataset, building the model, training, testing and evaluating the model performance, and finally deploying the model.

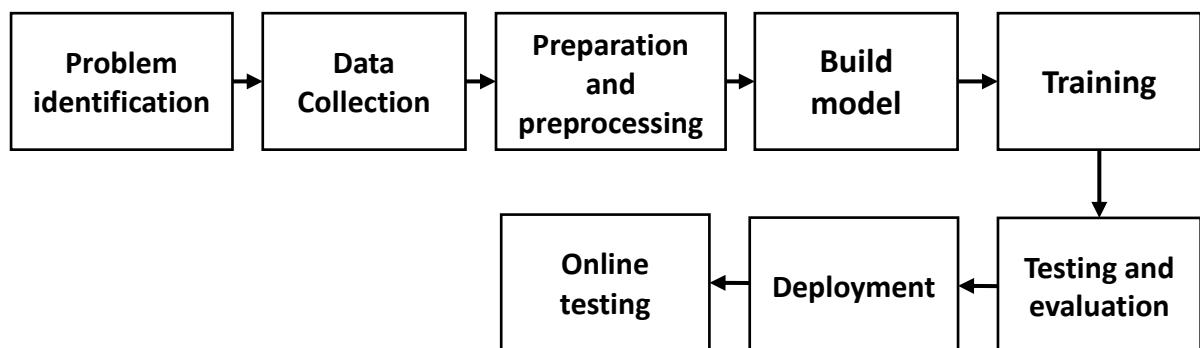


Figure 3-6 AI model development process

#### Problem identification and data collection

The selection of proposed services was based on the availability of a public dataset since it is difficult to collect a good and large enough dataset in the medical field, labeling the data is also a challenge as it would require a large number of medical experts to label such a large amount of data, a text matching method or NLP models could be utilized to check the diagnosis report associated with each sample and extract its label but this is not an efficient way and it will cause miss leading of the learning process of AI models, another reason is that using a public dataset allows for a fair comparison with previous researches and experiments performed on the same dataset and avoiding low-performance experiments.

#### Preparation and preprocessing

Cleaning the data from erroneous and irrelevant data is a very important step to avoid garbage in garbage out situation.

Transforming the data into the required format, for a dataset to be ready for training an AI model, all samples must have the same size (number of rows and columns) and shape (number of channels), resizing and reshaping is a required step as public datasets are not always ready to be used as they are.

Different augmentation methods should be implemented if the quantity of data is low or if the number of samples in each class is not equal which leads to miss training and evaluation of the model.

Dividing the dataset into training and testing sets if not already divided, a common ratio for dataset division is 80% for training and 20% for testing, the training set is often divided into two sets one for training and the other for validation.

### **Building and training models**

Different types of deep neural networks will be utilized such as convolution neural networks (CNN), 2-D convolution and 1-D convolution are examples of which, recurrent neural networks (RNN) such as long-short time memory (LSTM), the architecture of neural networks will be based on trial and error technique.

### **Testing and evaluation**

Testing sets are used to test the performance of models after the training process in which the model never sees the test set

Different evaluation method will be utilized such as accuracy, confusion matrix, recall and f1-score, after analyzing these evaluation metrics, modifications on the model architecture or data preprocessing will be made to improve results.

## **3.4 Deployment and GUI**

The MDS system will be deployed in a desktop application and a web application so that the offered services could be accessed locally or over the internet, the desktop application will be developed using PyQt5 which is a framework for designing and building GUIs in python language, and the web application will be developed using Html, CSS, and JavaScript, basic elements of the home page and service pages are shown in Fig. 10 and Fig. 11 without considering the final designs and icons.

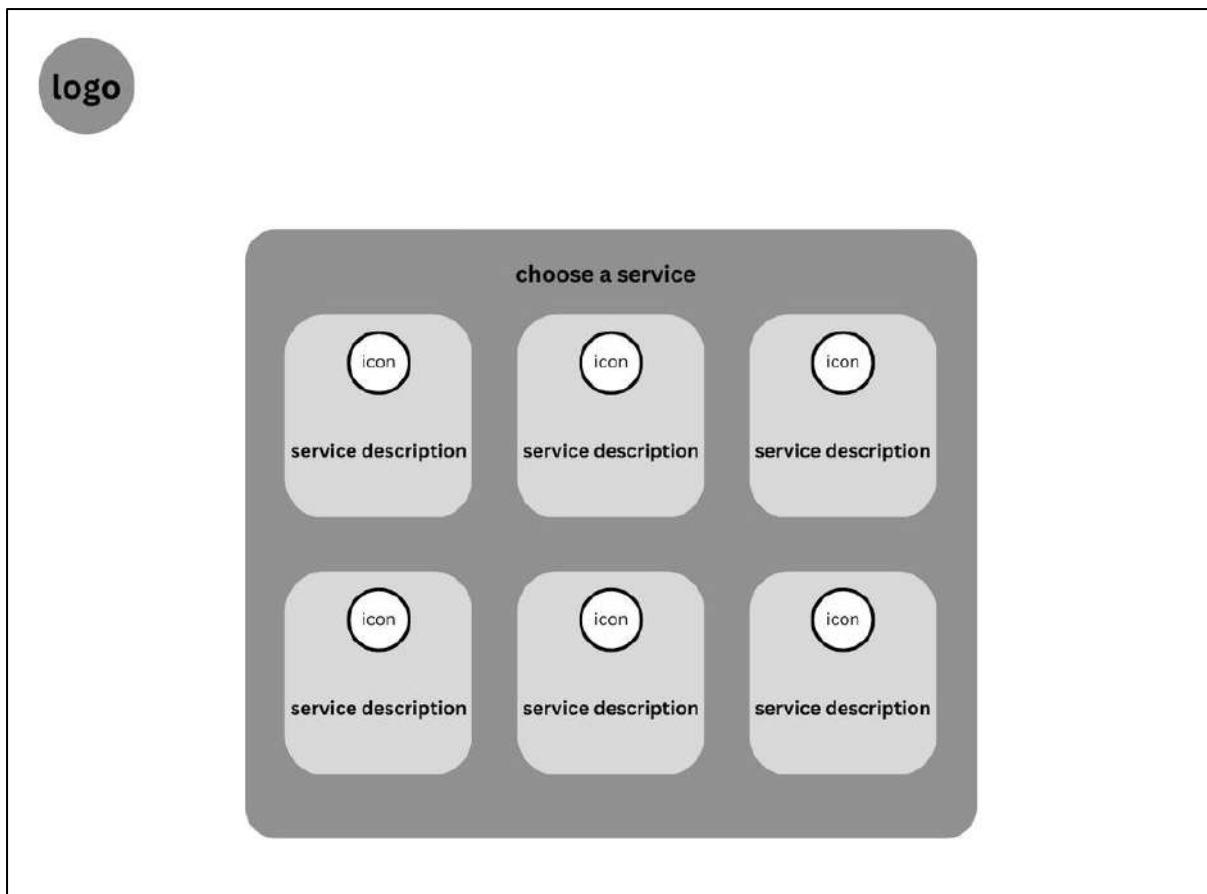


Figure 3-7 Basic elements of the home page

The home page will basically consist of six or five frames enclosed by one big frame, each one of the smaller frames corresponds to one of the proposed services, each frame contains an icon referring to the service and a short description of it, the smaller frames will act as buttons to forward the user to the selected service page.

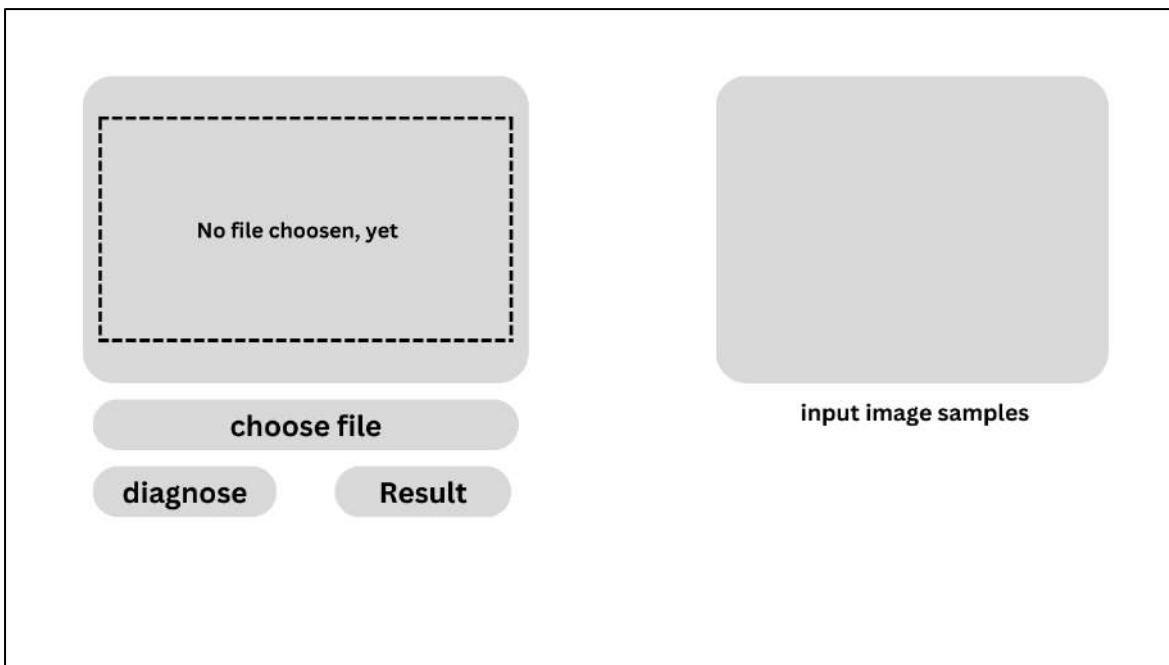


Figure 3-8 Basic elements of a service page

The service page will consist of some samples of the dataset used for training the AI model, a button to open file dialog for the user to select an image to be processed by the system an element to show the selected picture, a button to initialize the diagnosis process in the backend and a text box to display the output result of the model in a formatted way for the user.

# **Chapter 4**

# **EXPERIMENTAL RESULTS**

## **4.1 Introduction**

This chapter presents the development process of the project, sec. 4.2 presents the development process of AI models, sec. 4.3 presents the development process of the hardware system for the heartbeat abnormalities service.

## **4.2 Development of AI models**

AI models will be developed in Python programming language using Keras framework utilizing different evaluation metrics such as accuracy, recall, precision, f1-score and most importantly confusion matrix which allows for the analysis for model performance on each class and also allows for a good comparison with previous researches and experiments done on the same dataset.

### **4.2.1 Heartbeat abnormalities detection from electrocardiogram (ECG)**

The classification of the electrocardiogram (ECG) signal has a vital impact on the identification of heart-related diseases, this can ensure the premature finding of heart disease and the proper selection of the patient's customized treatment, however, the detection of arrhythmia is a challenging task to perform manually, this justifies the necessity of a technique for automatic detection of abnormal heart signals

#### **4.2.1.1 Motivation**

Interpreting ECG signals can be challenging and requires a great deal of expertise and experience as different types of arrhythmias can produce similar patterns on an ECG, making it difficult to make a definitive diagnosis based solely on the ECG tracing, ECG findings may be nonspecific or may overlap with other heart conditions, making it difficult to make a definitive diagnosis based only on the ECG signal.

AI models can help diagnose different types of arrhythmias by analyzing large datasets of ECG recordings to identify unique patterns and features that are indicative of different types of arrhythmias, for example, AI models can learn to recognize specific waveform morphologies, intervals, and amplitudes that are associated with different types of arrhythmias.

#### 4.2.1.2 Dataset description

“ECG Heartbeat Categorization Dataset” is a public dataset available on kaggle published by SHAYAN FAZELI, this dataset is composed of two collections of heartbeat signals derived and preprocessed from two famous databases, the MIT-BIH Arrhythmia Database which is the main dataset used for training the model that will be deployed in the application and The PTB Diagnostic ECG Database which will be used for method validation.

Classes: [N: 0, S: 2, V: 3, F: 4, Q: 5]

#### The MIT-BIH Arrhythmia Database

Table 4-1 class distribution of MIT-BIH Arrhythmia database

	N	S	F	Q	V	total
train	72471	2223	641	6431	5788	87554
test	18118	556	162	1608	1448	21892

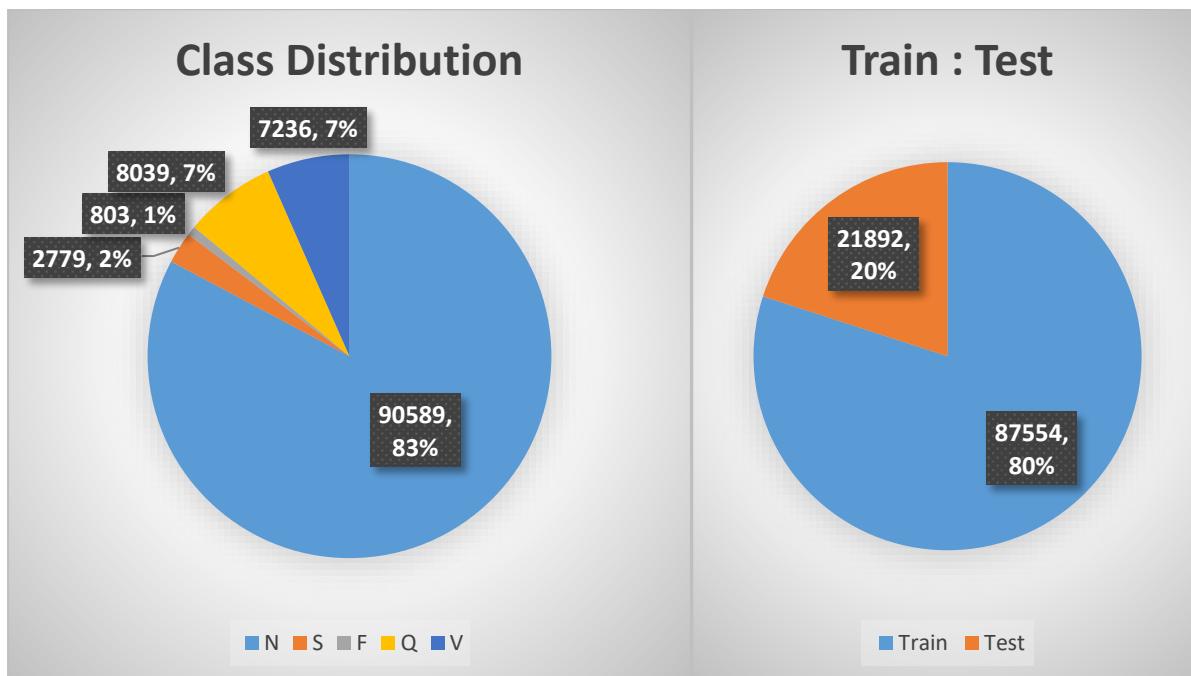


Figure 4-1 class distribution and train-test split of MIT-BIH Arrhythmia database

N: Normal beat

S: Supraventricular premature beat

V: Premature ventricular contraction

F: Fusion of ventricular and normal beat

Q: Unclassifiable beat

## Dataset preprocessing

This dataset contains samples of heartbeats that was extracted from ECG signal using the following steps

- 1) Splitting the continuous ECG signal to 10s windows and select a 10s window from an ECG signal.
- 2) Normalizing the amplitude values to the range of between zero and one.
- 3) Finding the set of all local maximums based on zero crossings of the first derivative.
- 4) Finding the set of ECG R-peak candidates by applying a threshold of 0.9 on the normalized value of the local maximums.
- 5) Finding the median of R-R time intervals as the nominal heartbeat period of that window ( $T$ ).
- 6) For each R-peak, selecting a signal part with the length equal to  $1.2T$ .
- 7) Padding each selected part with zeros to make its length equal to a predefined fixed length.

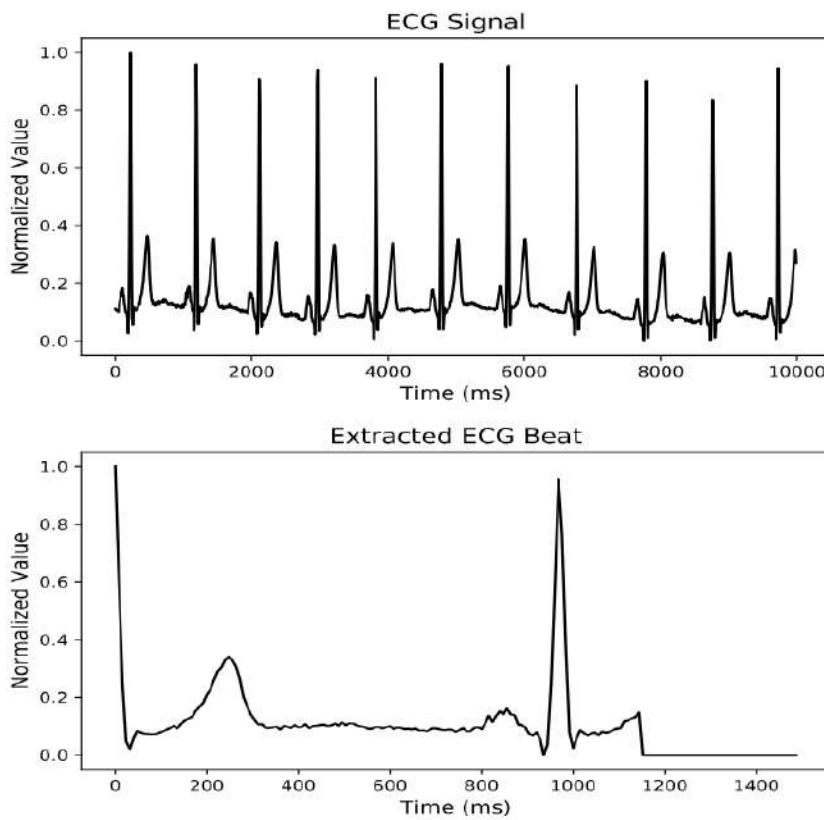


Figure 4-2 Heartbeat extraction result

After these 7 steps the result is a csv file containing 187 value for each sample, these values represent a time series sequence of the extracted heartbeat

## Data source

This dataset was developed and compiled by the **BIH Arrhythmia Laboratory** which is a laboratory in the **Deaconess Medical Center** which is a private, non-profit teaching hospital located in Boston, Massachusetts, USA, it is the third-largest hospital in Boston and is affiliated with Harvard Medical School, the BIH Arrhythmia Laboratory provided the facilities and resources for the collection of the data in association with **Massachusetts Institute of Technology (MIT)** which provided the expertise and technical support for the collection of the data, the data collection process was funded by the **National Institutes of Health (NIH)** which is the primary agency of the US government responsible for biomedical and public health research, this dataset was diagnosed and labeled by the medical expertise provided by the **American Heart Association (AHA)** which is non-profit organization that is dedicated to saving people's lives from heart disease and stroke, the database was made available at **PhysioNet** which is a freely available online repository of biomedical research data as part of the NIH's effort to make biomedical research data more accessible to the public.

- **Deaconess Medical Center:** This hospital provided the facilities and resources for the collection of the data.
- **Massachusetts Institute of Technology:** This University provided the expertise and technical support for the collection of the data.
- **National Institutes of Health:** This agency provided the funding for the collection of the data.
- **American Heart Association:** This non-profit organization provided the medical expertise and oversight for the collection of the data.

The presence of high-profile organizations in the collection and compilation of the MIT-BIH database indicates the importance of AI in this field

#### 4.2.1.3 Reference paper for result comparison

Results of experiments which will be performed in the project will be compared with [15] in which the author proposed a 13 layer residual neural network utilizing 1-D convolution to differentiate between 5 classes of the MIT-BIH dataset achieving an accuracy score of 93.4%

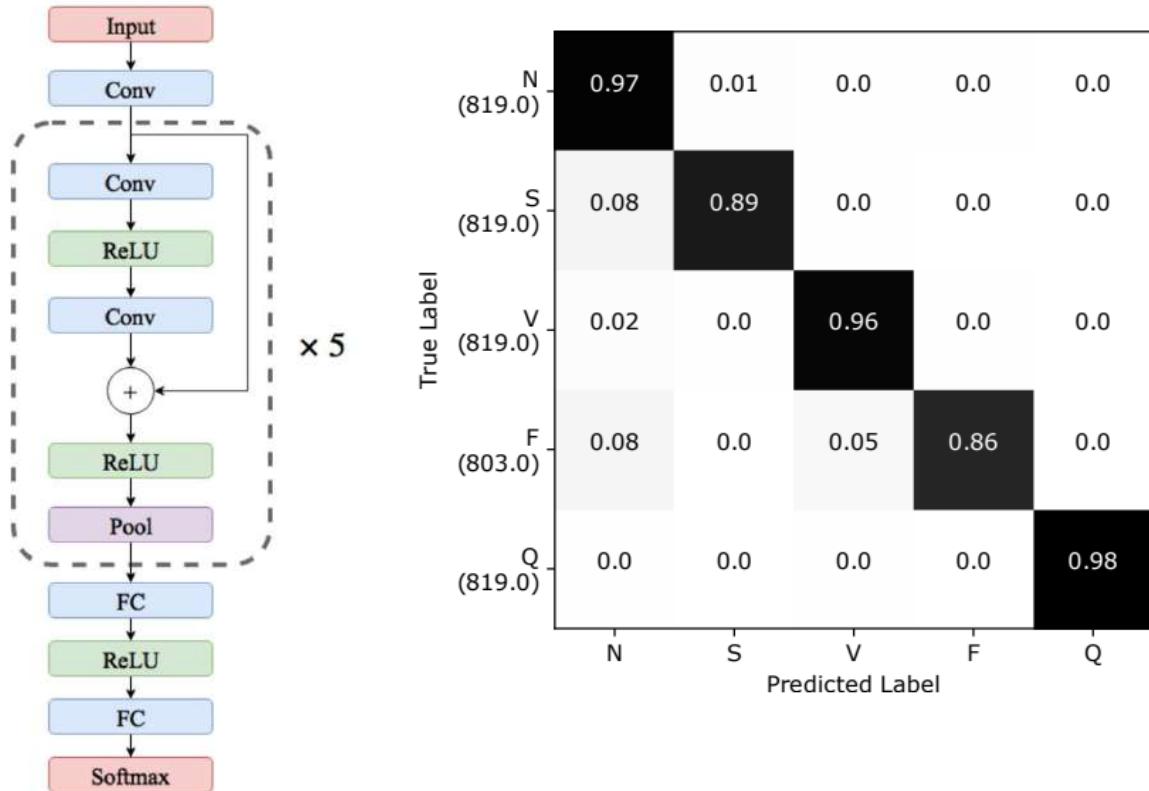


Figure 4-3 [15] proposed network architecture and confusion matrix

#### 4.2.1.4 Development methodology

Since this dataset is in a csv format and image format, it is suitable for training multiple types of AI models such as 1-D convolution and 2-D convolution, these two types of models will be tested on this dataset and the one which achieves best results will be used for deployment.

The following part consists of versions, each version presents experiments with a different model

#### 4.2.1.5 2-D convolution methodology

An image version of the proposed dataset was published on kaggle by HARI MOHAN RAI by the name of ECG\_Image\_Data and it will be used in this methodology.

##### • Version 1

As mentioned before, any DL model will easily identify classes based on color, to proof our theory we trained the Vgg-16 model on this dataset and it achieved an inconvenient high accuracy

#### Data preprocessing

As shown above the dataset is imbalanced so 640 images only were sampled from each class for the training set and 160 image for the testing set, a preprocessing function made specially for the vgg-16 model to preprocess the input images was used with a batch size of 32, all images were normalized.

#### Proposed framework

Vgg-16 pre-trained model was used with a customized classification top to match the problem of the ECG\_image \_data as shown in Fig. 9, relu activation function is used in all the hidden Dense layers and softmax is used in the output layer

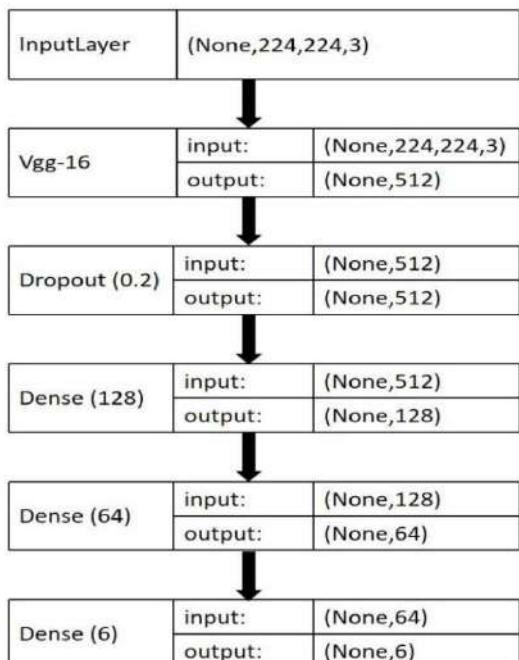
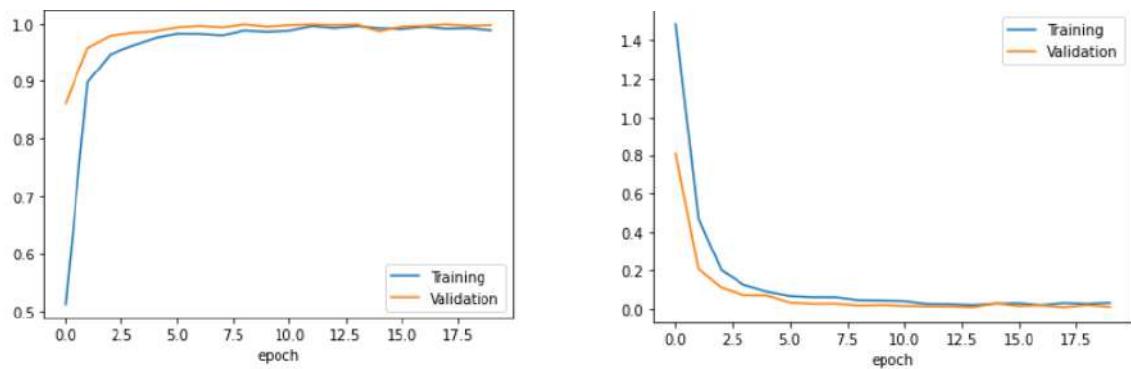


Figure 4-4 version 1 proposed network architecture

#### Result

The training accuracy and the testing accuracy both were almost 100% which proofs that the model classifies the input based on color though we are using only 640 image for each class in the training set which is the number of images in the smallest class which is considered low compared to the number of images in other classes



*Figure 4-5 Training accuracy and loss curves for version 1 model training*

- **Version 2**

Grayscale images were used in this version

### **Data preprocessing**

In this version 640 images were sampled from each class for the training set with a validation split of (0.2) and 160 images were sampled for the testing set, all resized to (64,64) and converted to grayscale with a batch size of 32, all images were normalized.

### **Proposed framework**

A custom CNN architecture as shown was used with 2 similar convolution blocks each containing a 32 filters of size (3,3) with relu activation, a Maxpooling of size (2,2) and a dropout of (0.2) followed by a flatten layer then a Dense top for the 6 class classification (relu+softmax).

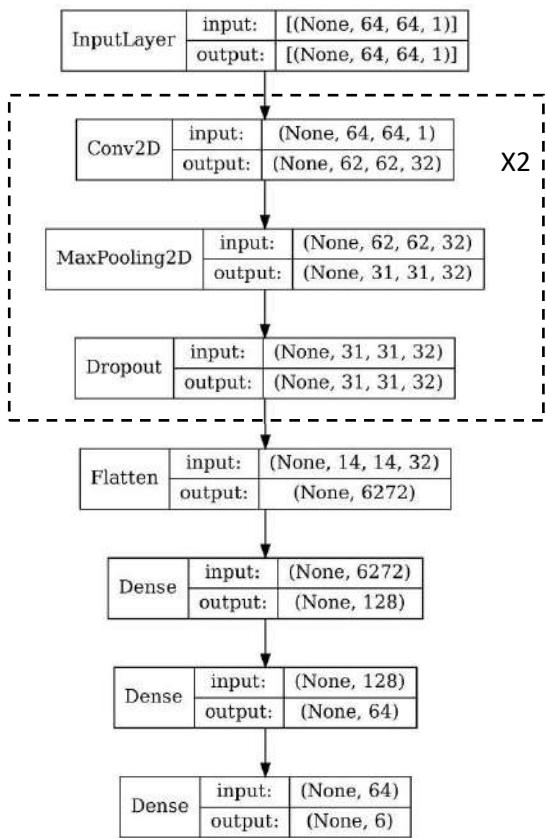


Figure 4-6 version 2 proposed network architecture

## Result

Table 4-2 Results of version 2

	Train acc	Val_acc	Test acc
Proposed	96%	81.51%	81.25%
[15]			93.4%

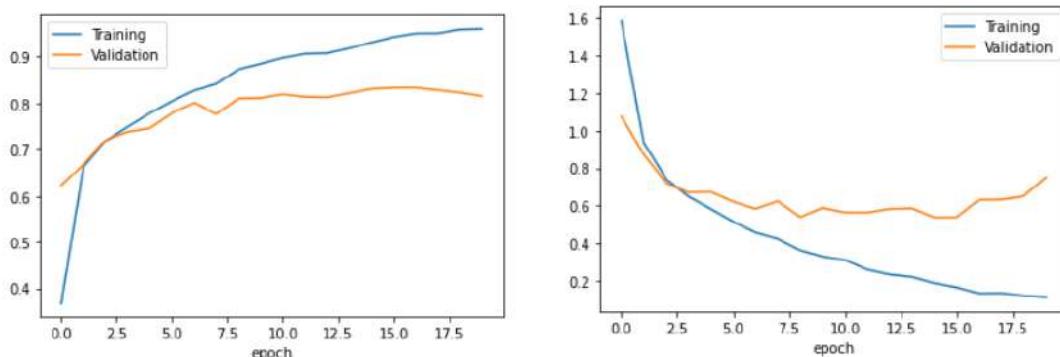


Figure 4-7 training accuracy and loss curves for version 2

This model is over fitted.

## Experiments

### EXP. (1)

A similar architecture without the first Dense layer.

Table 4-3 Results of version 2 EXP. (1)

	Train acc	Val_acc	Test acc
Proposed	94.01%	81.64%	80.52%
[15]			93.4%

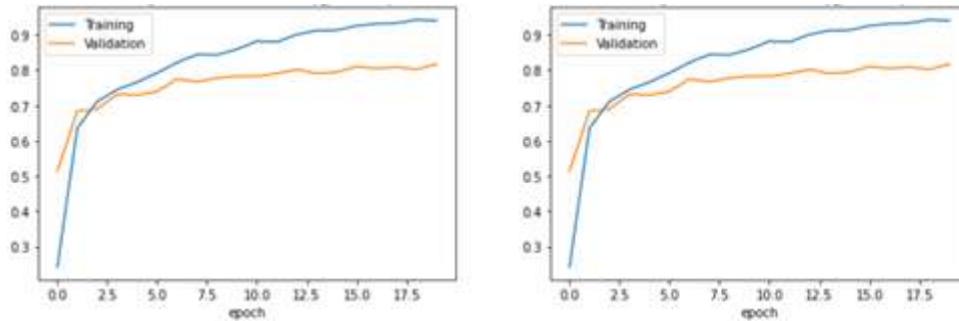


Figure 4-8 training accuracy and loss curves for version 2 EXP. (1)

Still over fitted.

### EXP. (2)

Same as EXP. (1) with a dropout layer of (0.2) between the two remaining Dense layers.

Table 4-4 Results of version 2 EXP. (2)

	Train acc	Val_acc	Test acc
Proposed	93.13%	0.8112%	80.73%
[15]			93.4%

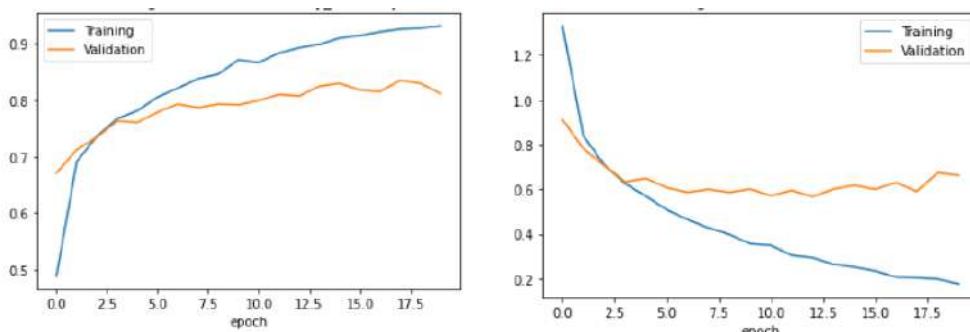


Figure 4-9 training accuracy and loss curves for version 2 EXP. (2)

## Conclusion of version 2

Reducing the number of Dense layers and having a dropout in between them doesn't help with the over fitting.

## • Version 3

Another CNN architecture of a higher depth than the previous one is used with grayscale images

## Data preprocessing

In this version 640 images were sampled from each class for the training set with a validation split of (0.2) and 160 images were sampled for the testing set, all resized to (64,64) and converted to grayscale with a batch size of 32, all images were normalized.

## Proposed framework

This model consists of three convolution blocks each has two Conv2D layers the first has a padding same property, followed by a maxpooling and a dropout of 0.25, the first block has 32 filters of size (3, 3), the other two has 64 filters of the same size, all maxpooling layers has a size of (2, 2) all relu activated, then a flatten layer then a classification top consisting of two Dense layers both relu activated with a dropout of 0.5 in between.

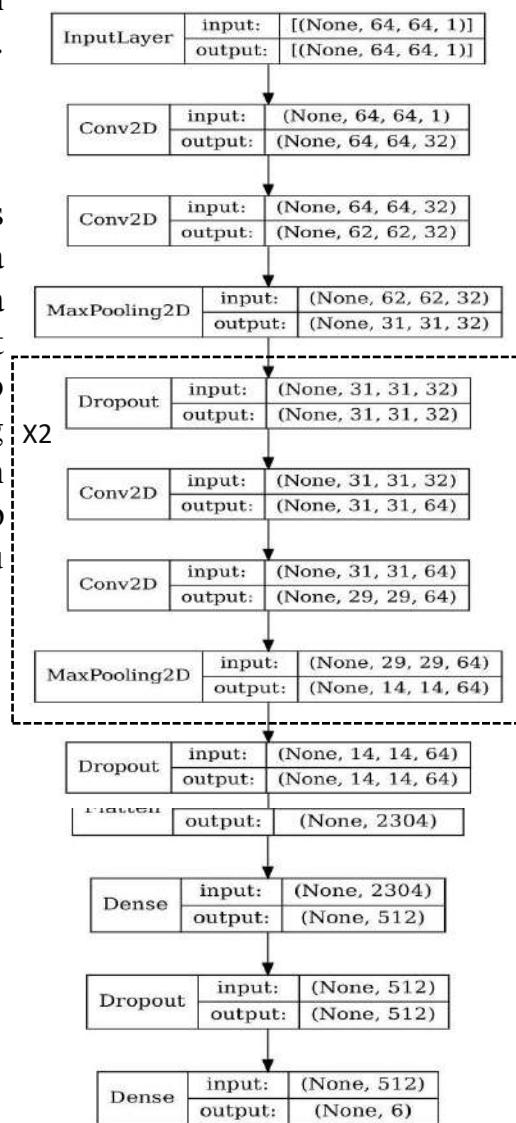


Figure 4-10 version 3 proposed network architecture

## Result

Table 4-5 Results of version 3

	Train acc	Val_acc	Test acc
Proposed	94.92%	85.68%	84.58%
[15]			93.3%

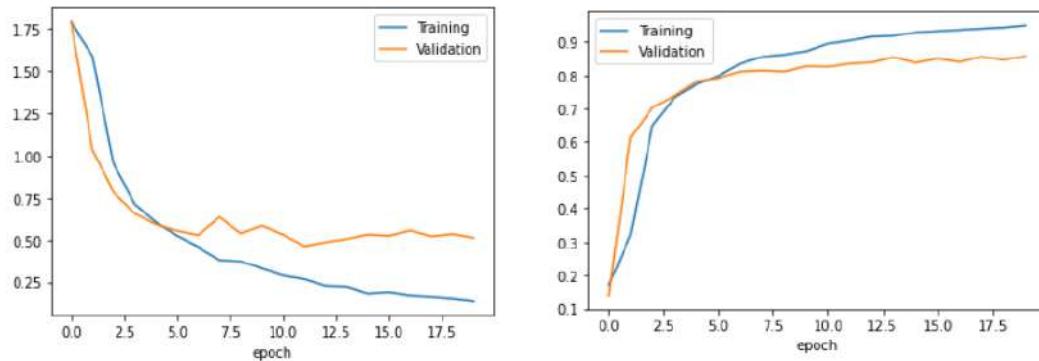


Figure 4-11 training accuracy and loss curves for version 3

This model is over fitted.

## Experiments

### EXP. (1)

Increasing the dropout between the convolution blocks to 0.5

Table 4-6 Results of version 3 EXP. (1)

	Train acc	Val_acc	Test acc
Proposed	94.92%	87.11%	86.15%
[15]			93.4%

	precision	recall	f1-score	support
0	0.88	0.93	0.90	160
1	0.84	0.83	0.84	160
2	0.74	0.76	0.75	160
3	0.96	0.95	0.95	160
4	0.88	0.84	0.86	160
5	0.89	0.86	0.88	160
accuracy			0.86	960
macro avg	0.86	0.86	0.86	960
weighted avg	0.86	0.86	0.86	960

Figure 4-12 classification report of version 3 EXP. (1)

Still over fitted.

## EXP. (2)

Same as EXP. (1) but the class with the smallest number of samples was dropped and a 2223 images were sampled for the training set and 556 for the test set

Table 4-7 Results of version 3 EXP. (2)

	Train acc	Val_acc	Test acc
proposed	93.83%	92.17%	91.08%
[15]			93.4%

```
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      2780
weighted avg                          0.91      0.91      2780
```

Figure 4-13 classification report of version 3 EXP. (2)

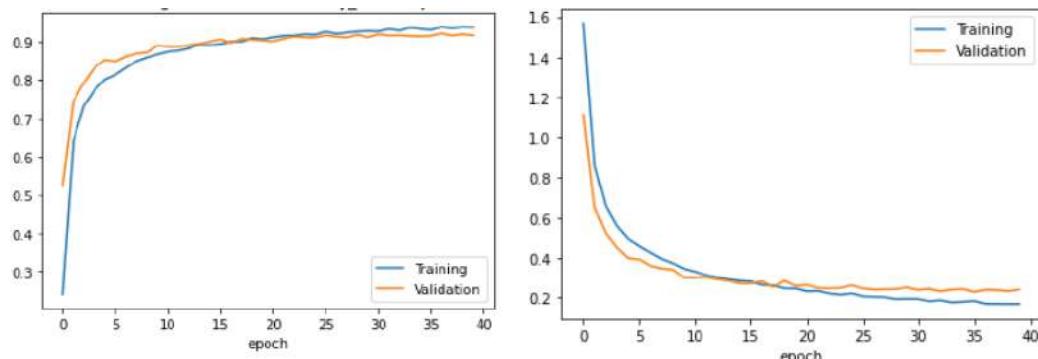


Figure 4-14 training accuracy and loss curves for version 3 EXP. (2)

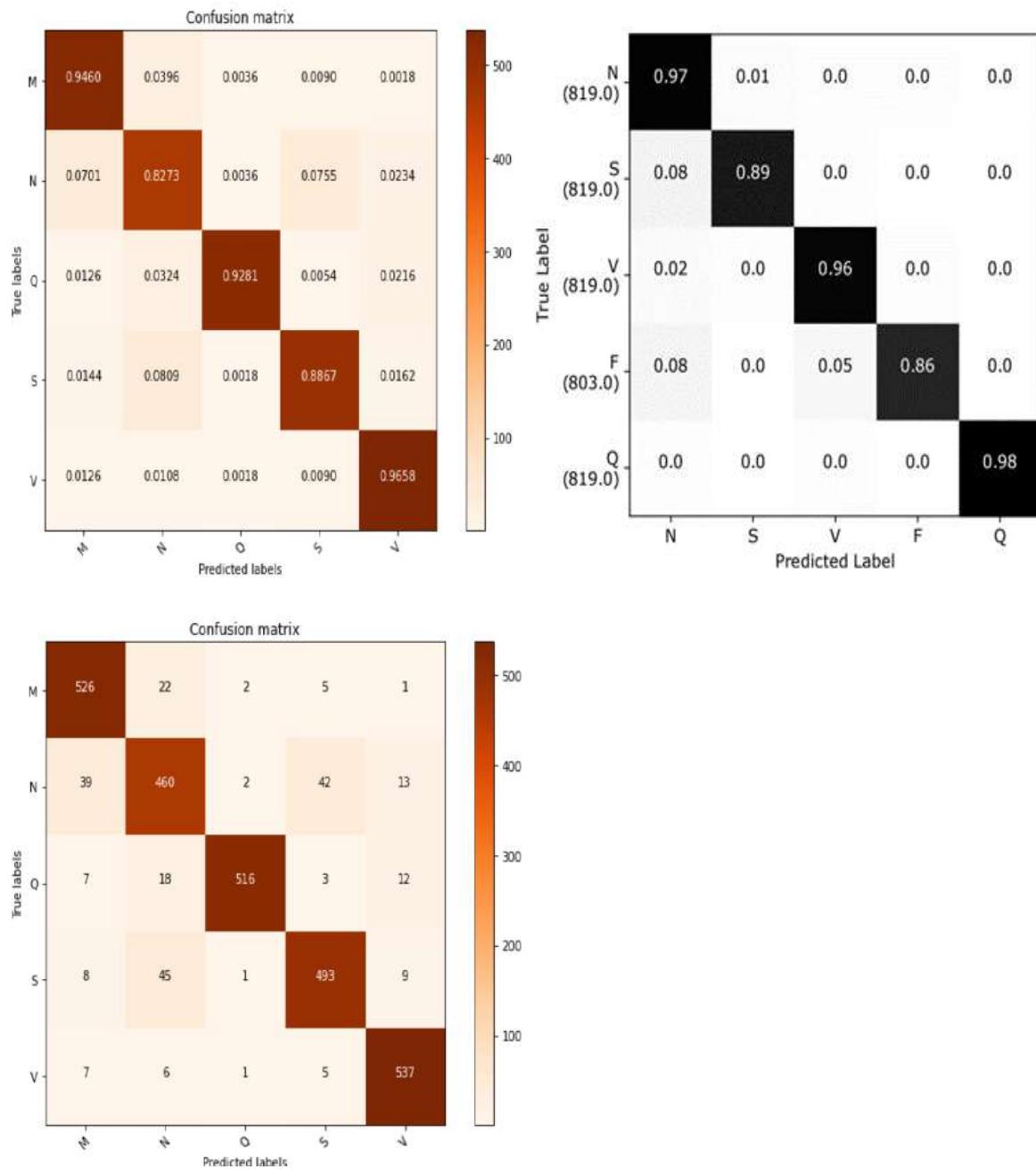


Figure 4-15 version 3 EXP (2) confusion matrices in orange VS [15] confusion matrix in black

Not over fitted, the imbalanced class may be the cause for over fitting, increasing the number of epochs does not help as well.

### Version 3 Conclusion

The performance is highly affected by the imbalance of the number of samples for each class.

## • Version 4

An idea is proposed which is converting the dataset to grayscale and reconverting it back to RGB to get rid of the colors and yet have a 3 channel images which allows for the use of pre-trained models which the low number of samples of the smallest class is suitable for training these pre-trained models as the convolution part of them is already well trained and saved, a desktop tool for images processing called image magic is used to convert the dataset to grayscale.

As a test for the performance of the image magic tool, the proposed model in version 3 EXP. (3) is trained on the new grayscale dataset and the results are mostly the same as it was.

## Data preprocessing

640 images were sampled for the training set with a validation split of 0.2 and 160 images for the test set all resized to (224,224) and converted to 3 channels, a preprocessing function for the vgg16 model was used as well, all images were normalized.

## Proposed framework

VGG-16 pre-trained model with pooling layers set to average pooling followed by a 0.2 dropout then a customized classification top to match the problem, relu activation function is used in all the hidden Dense layers and softmax is used in the output layer

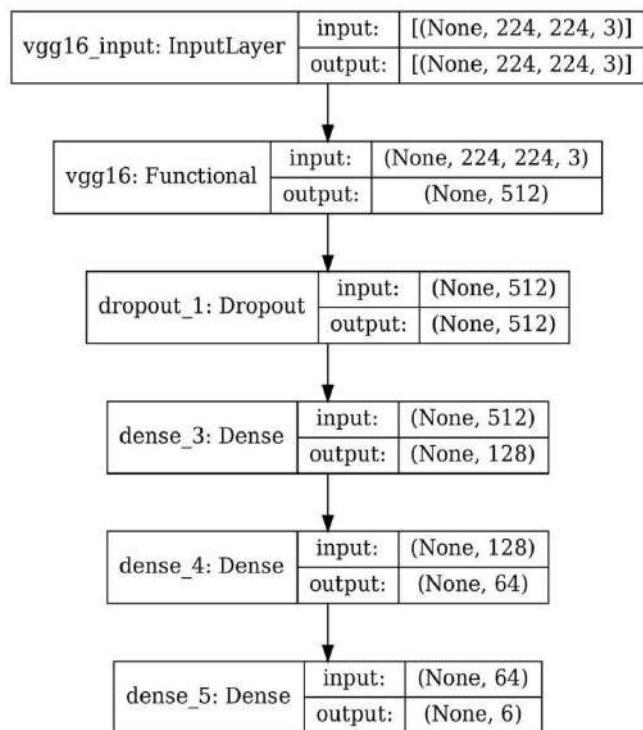


Figure 4-16 version 4 proposed network architecture

## Result

Table 4-8 Results of version 4

	Train acc	Val_acc	Test acc
proposed	74.87%	77.47%	76.56%
[15]			93.4%

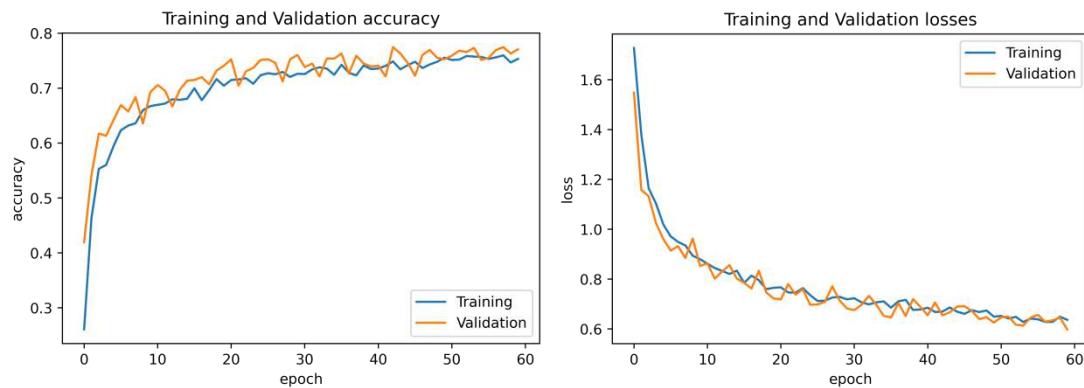


Figure 4-17 training accuracy and loss curves for version 4

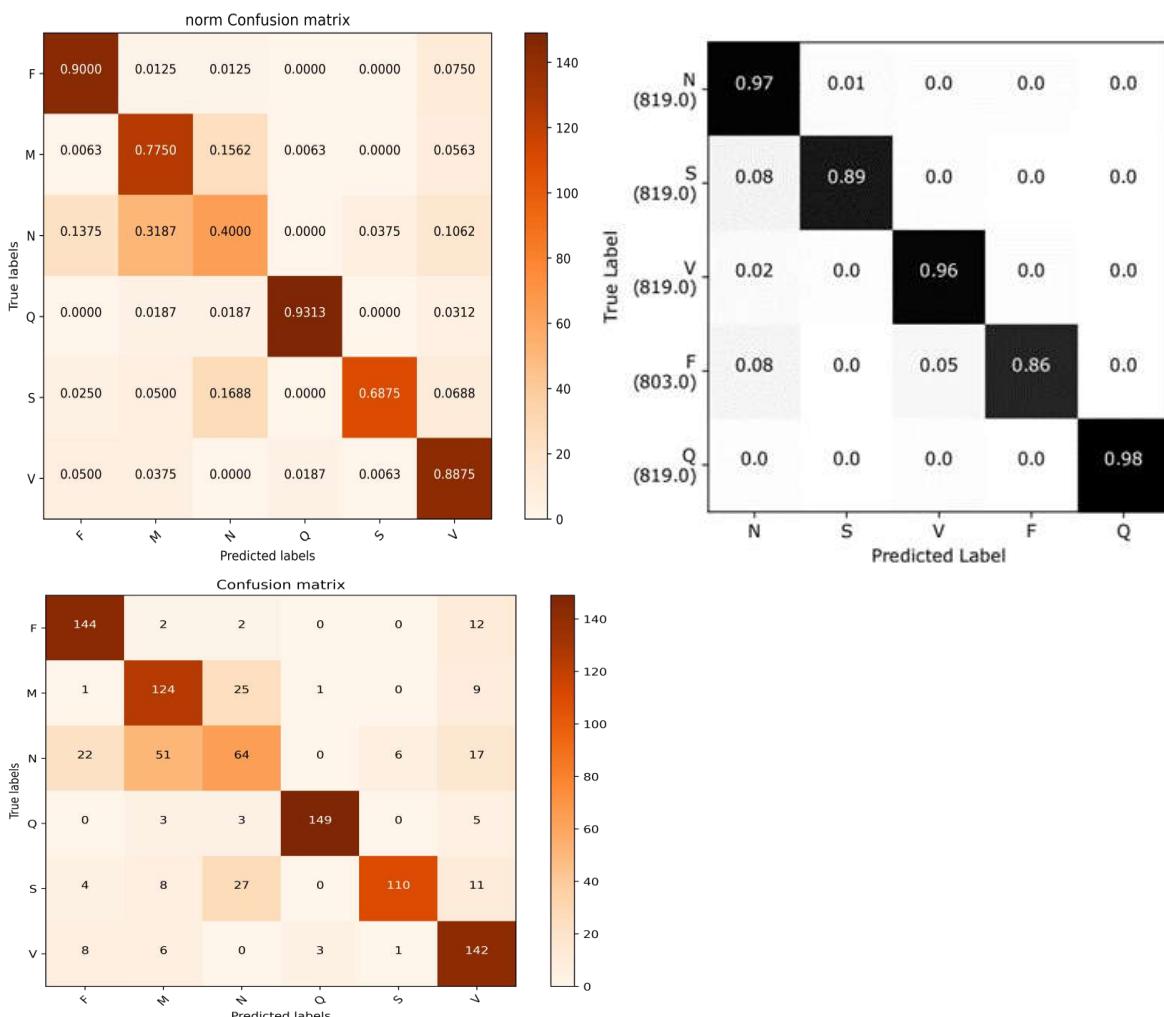


Figure 4-18 version 4 confusion matrices in orange VS [15] confusion matrix in black

classification report				
	precision	recall	f1-score	support
0	0.80	0.90	0.85	160
1	0.64	0.78	0.70	160
2	0.53	0.40	0.46	160
3	0.97	0.93	0.95	160
4	0.94	0.69	0.79	160
5	0.72	0.89	0.80	160
accuracy			0.76	960
macro avg	0.77	0.76	0.76	960
weighted avg	0.77	0.76	0.76	960

Figure 4-19 classification report for version 4

## Version 4 conclusion

VGG-16 model is not suitable for this problem

- **Version 5**

The same idea as version 4 is proposed again with a different pre-trained model

### Data preprocessing

640 images were sampled for the training set with a validation split of 0.2 and 160 images for the test set, all images were resized to (224,224) and converted to 3 channels, a preprocessing function for the ResNet50 model was used as well, all images were normalized.

### Proposed framework

The ResNet50 pre-trained model with pooling layers set to average pooling followed by the same classification top as the previous version, the network architecture is similar to the one in version 4 but with the ResNet50 model instead of VGG-16

### Result

Table 4-9 Result for version 5

	Test acc	Test recall	Test precision	Test AUC
Proposed	96.56%	96.56%	96.56%	0.9986
[15]	93.4%			

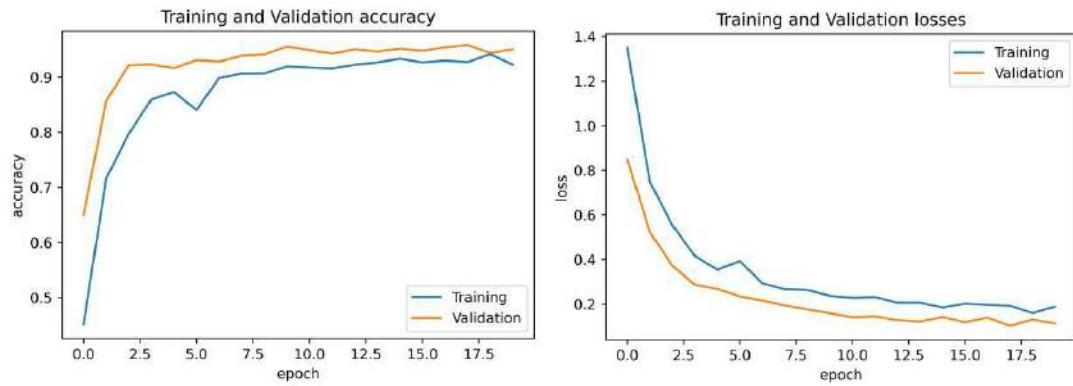


Figure 4-20 training accuracy and loss curves for version 5

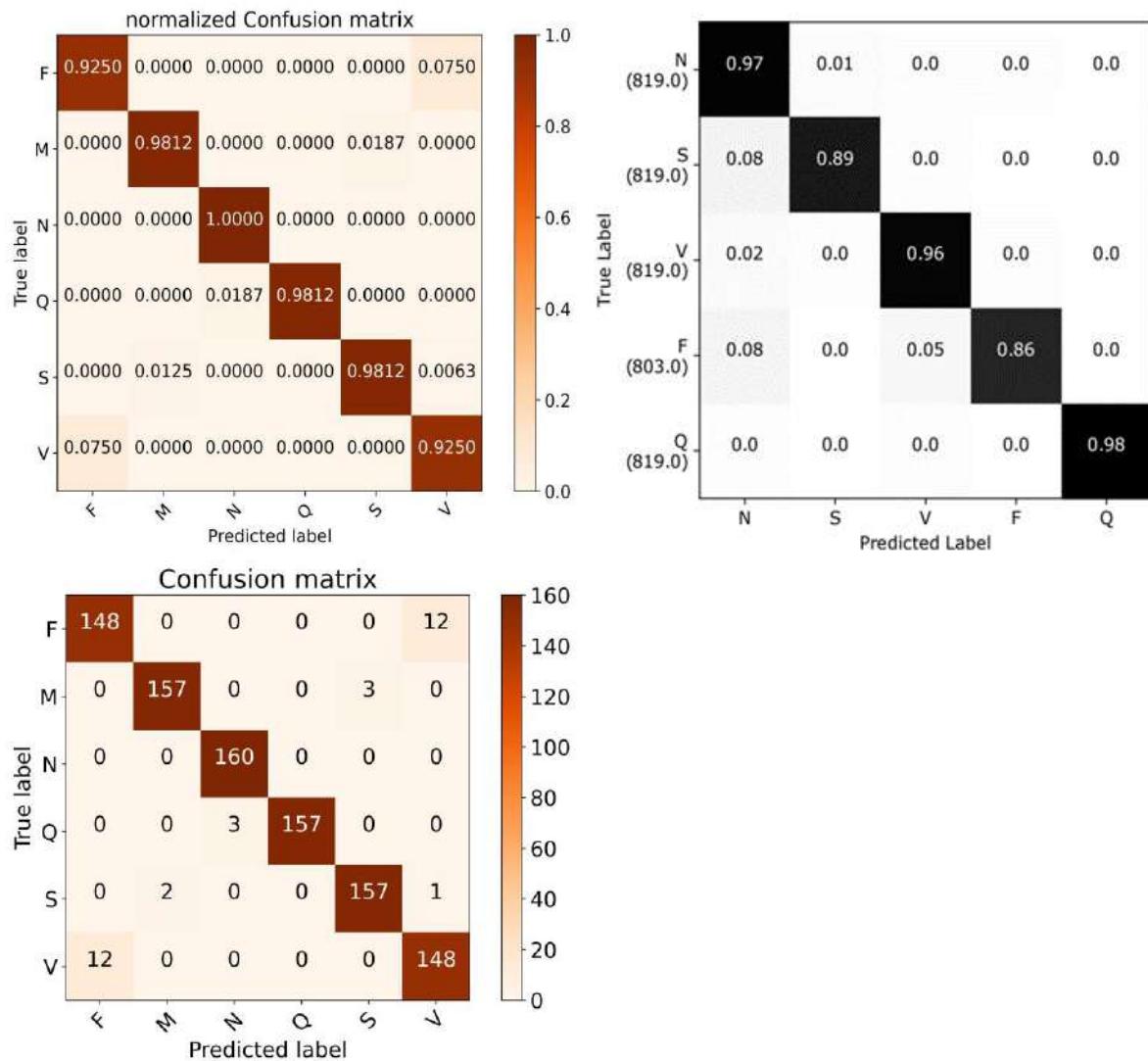


Figure 4-21 version 5 confusion matrices in orange VS [15] confusion matrix in black

## Version 5 conclusion

The ResNet50 architecture is suitable for this problem as it is build based on the idea of residual networks which is known for its good performance as the output of each activated convolution block is added with the input of the activation of

the following block which solve the problem of feature loss when using a large depth CNN models

#### 4.2.1.6 1-D convolution methodology

This methodology will be applied on the ECG Heartbeat Categorization Dataset itself that was described in sec. 4.2.1.1.

- **Version 1**

In this version a 1-D CNN model was introduced

#### Data preprocessing

The training and testing sets were used as they are

#### Proposed framework

As shown in Fig. 34, this CNN architecture consists of three blocks each with a 1D convolution, Batch Normalization and a maxpooling layer of size 2, the first block contains a 128 filters of size 3 while the other two contains 64 of size 3 for the second block and size 2 for the third, then these three block are followed by a 1-D convolution layer similar to the one in the third block, all relu activated, then a Dense top of three layers with a dropout of 0.2 after the first one, the first two are relu activated and the output layer is softmax activated, adam optimizer was used with an adaptive learning rate, the test set was used as validation data.

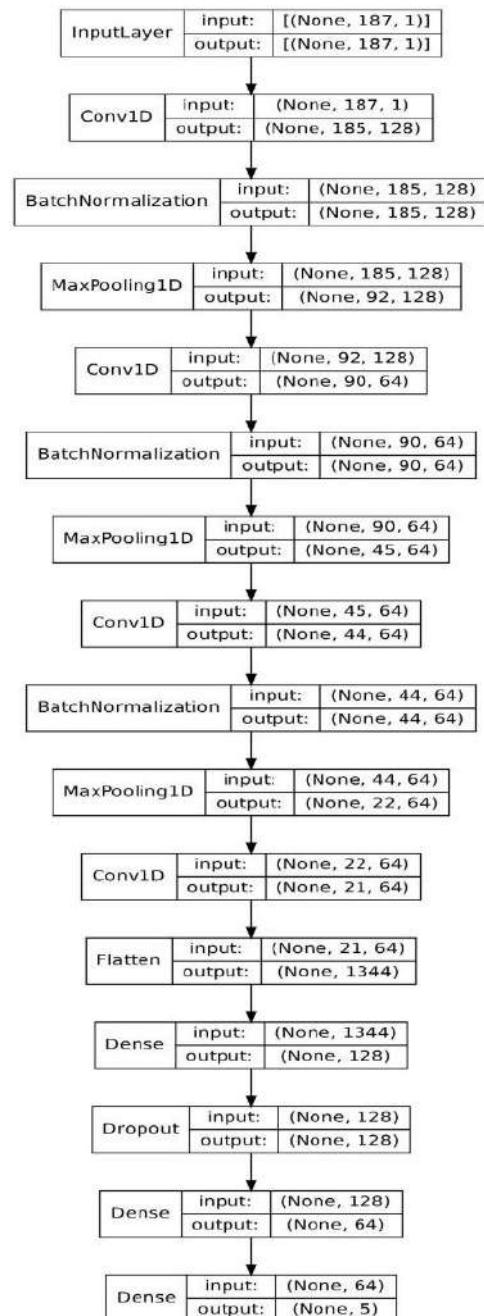


Figure 4-22 version 1 proposed network architecture

## Result

Table 4-10 Results for version 1

	Train acc	Val_acc	Test acc
Proposed	99.04%	98.63%	98.63%
[15]			93.4%

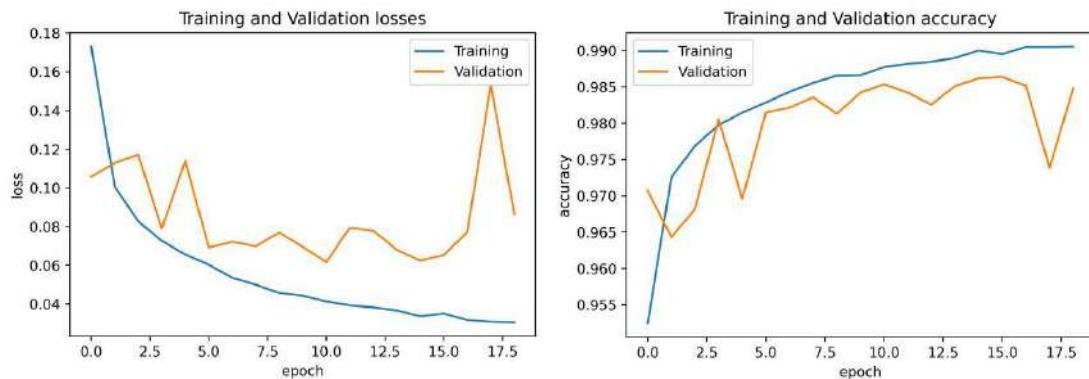


Figure 4-23 training accuracy and loss curves for version 1

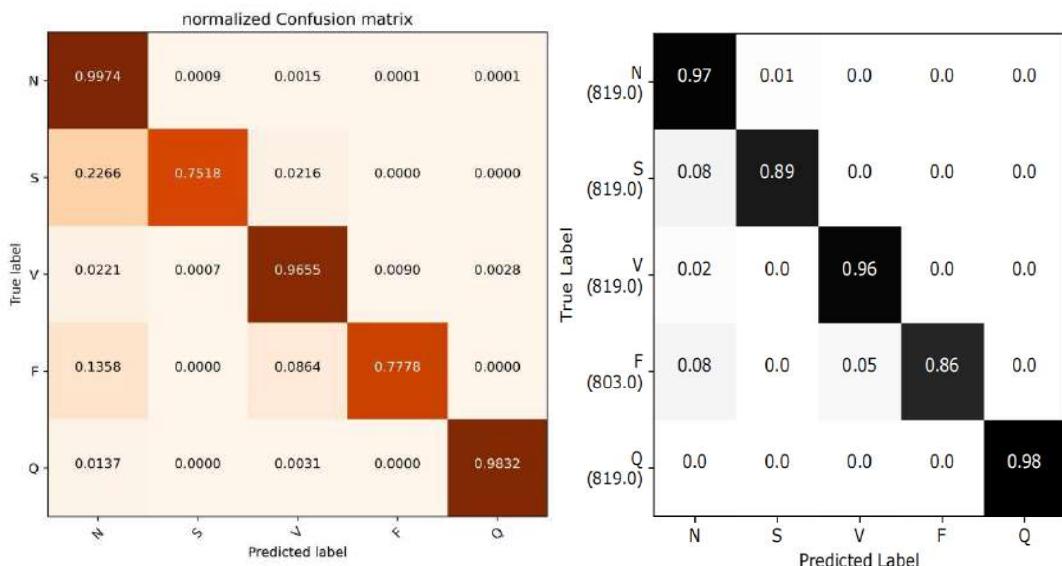


Figure 4-24 version 1 confusion matrix in orange VS [15] confusion matrix in black

Table 4-11 classification report for version 1

index	Class name	Precision	Recall	F1-score	Support
0	N	0.99	1.00	0.99	18118
1	S	0.96	0.75	0.84	556
2	V	0.96	0.97	0.96	1448
3	F	0.89	0.78	0.83	162
4	Q	0.98	0.98	0.99	1608
accuracy				0.986	21892
macro avg		0.96	0.90	0.92	21892
weighted avg		0.99	0.99	0.99	21892

As seen from the confusion matrix and the recall, the model performance on the classes F and S is not good as they have a small number of samples compared to the class N yet the model did good on the classes V and Q as they have more samples than F and S which means that the small number of samples caused the model to predict these class as N as the model is well trained on that class, the accuracy is very high despite the bad performance on some classes because the number of samples in the N class is very large compared to the other classes.

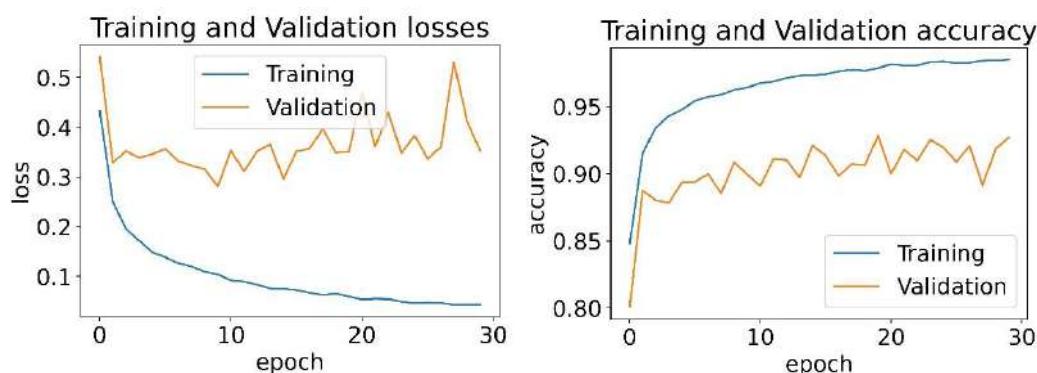
## Experiments

### EXP. (1)

The large number of samples in the class N is probably what is causing the model to have a bad performance on the two classes with the small number of samples as the model is miss predicting them as N, so the number of samples in the class N of the training set was reduced to 6000 sample as the model had a good performance with other class with a similar number of samples and the number of samples in the test set was resampled to 800 for the comparison with [15] testing results

*Table 4-12 Results for version 1 EXP. (1)*

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	92.83%	93.15%	92.77%	0.98
[15]	93.4%			



*Figure 4-25 training accuracy and loss curves for version 1 EXP. (1)*

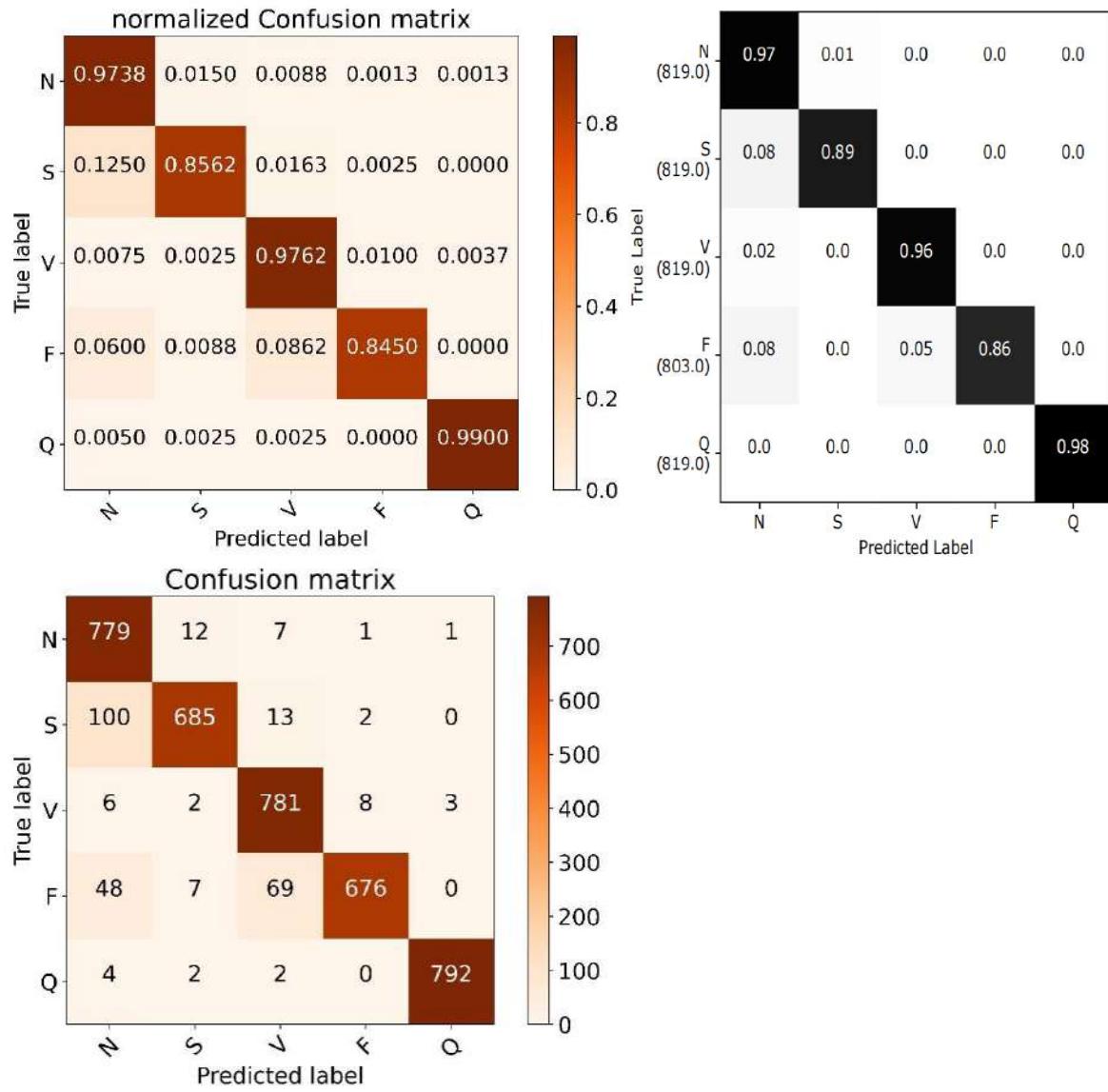


Figure 4-26 version 1 EXP. (1) confusion matrices in orange VS [15] confusion matrix in black

Table 4-13 classification report for version 1 EXP. (1)

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8314	0.9738	0.8969	800
1	S	0.9675	0.8562	0.9085	800
2	V	0.8956	0.9762	0.9342	800
3	F	0.984	0.845	0.9092	800
4	Q	0.995	0.99	0.9925	800
accuracy				0.9283	4000
macro avg		0.9347	0.9283	0.9283	4000
weighted avg		0.9347	0.9283	0.9283	4000

Reducing the number of sampled in the N class caused the performance on the two smallest classes to get better as the recall of the class F increased from 0.78 to 0.845 and from 0.75 to 0.8562 for the class S

## EXP. (2)

The number of samples in the class N of the training set was reduced to 20000 sample and other classes were up-sampled to 20000 sample, the number of samples in the test set was resampled to 800, the dataset was up-sampled to 20000 according to [15]

Table 4-14 Results for version 1 EXP. (2)

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	93.34%	94.30%	93.20%	0.98
[15]	93.4%			

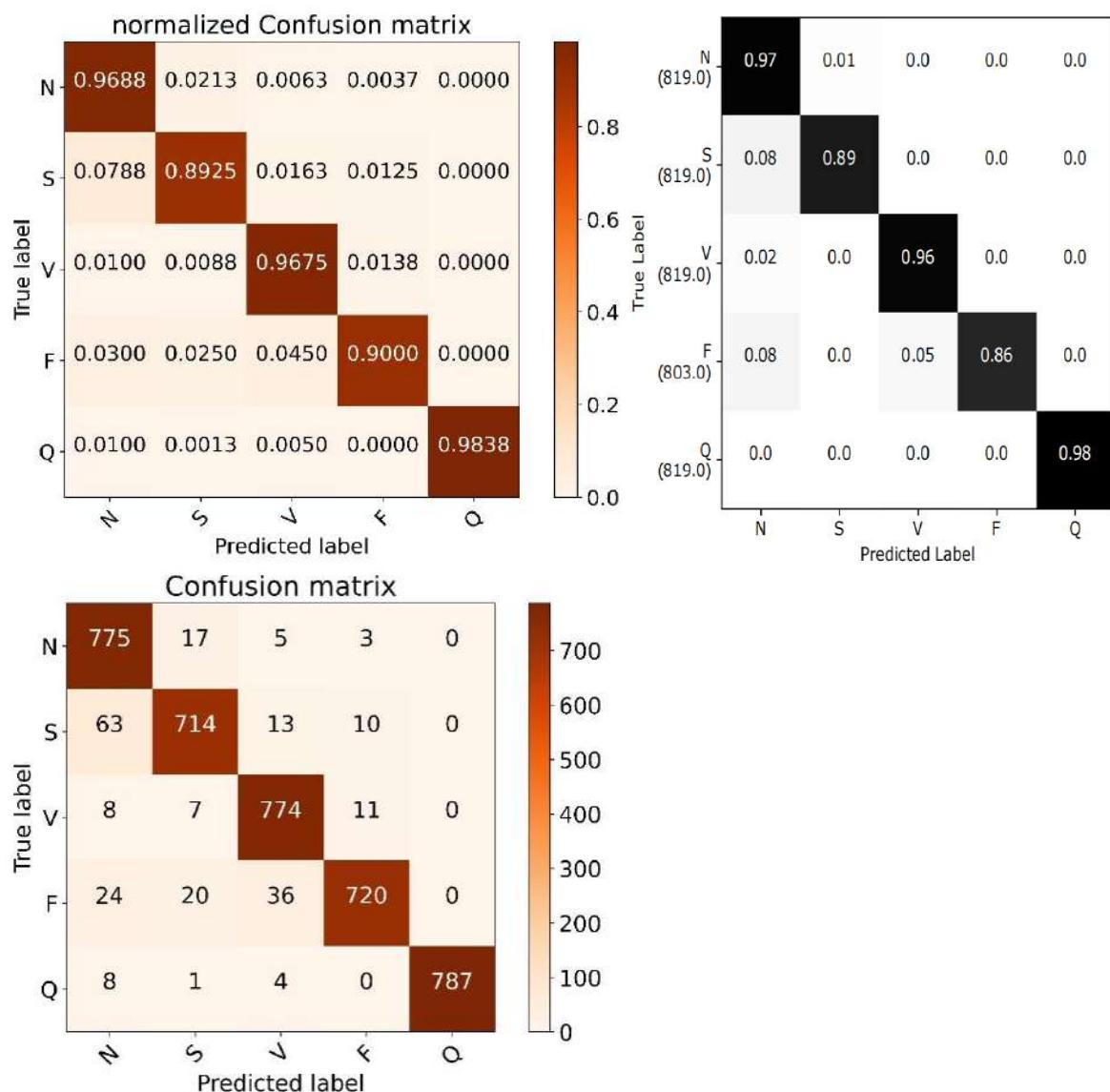


Figure 4-27 version 1 EXP. (2) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-15 classification report for version 1 EXP. (2)*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8827	0.9688	0.9237	800
1	S	0.9407	0.8925	0.916	800
2	V	0.9303	0.9675	0.9485	800
3	F	0.9677	0.9	0.9326	800
4	Q	1	0.9838	0.9918	800
accuracy				0.9334	4000
macro avg		0.9343	0.9425	0.9325	4000
weighted avg		0.9343	0.9425	0.9325	4000

The recall improved on the S and F classes

## Version 1 conclusion

The imbalance between the number of samples in each class has a big effect on the performance of the model on each class as reducing the number of samples in the N class to reduce the imbalance caused the performance to get better on the smallest classes and re-sampling the training set to 20000 results in a much further improvement.

## • Version 2

Another 1-D CNN model was introduced

### Data preprocessing

The training set was re-sampled to 20000 and the test set was re-sampled to 800.

### Proposed framework

This model consists of 3 1-D convolution layers each with a padding same property and relu activation function, the first two have a filter size of 3 and the third has a filter size of 5, the first layer has 32 filters, the second has 64 filters and the third has 128 filters, these three layers were then followed by a 1-D max-pooling layer of size 3 and a drop-out layer of 0.5 followed by a flatten layer and three Dense layers with relu activation function.

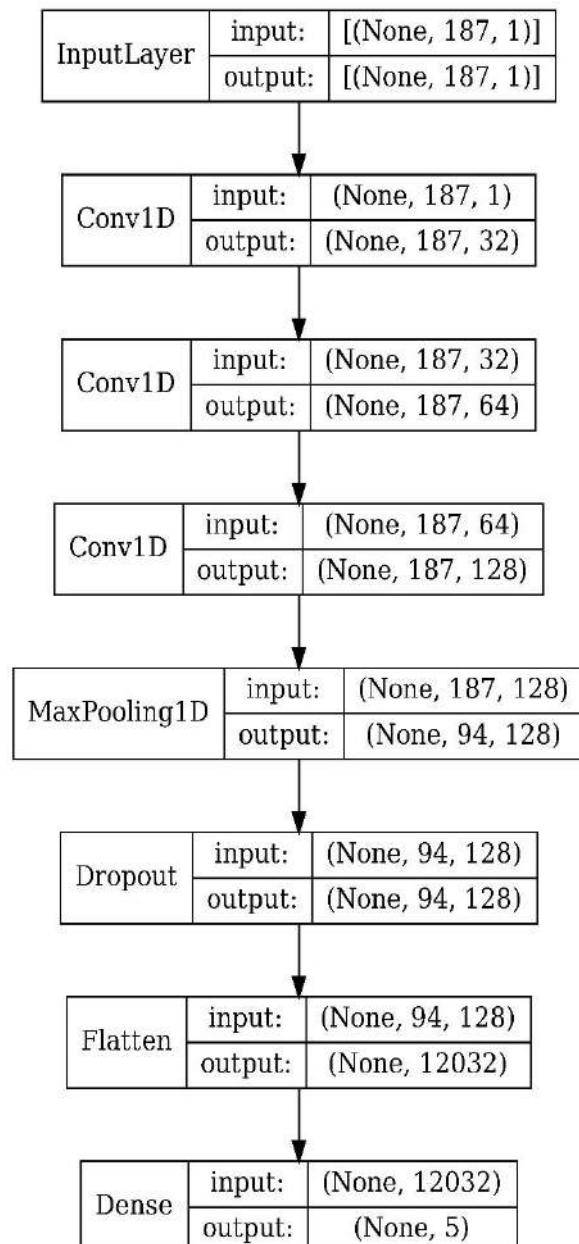


Figure 4-28 version 2 proposed network architecture

## Result

Table 4-16 Results for version 2

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	93.35%	94.33%	93.23%	0.9885
[15]	93.4%			

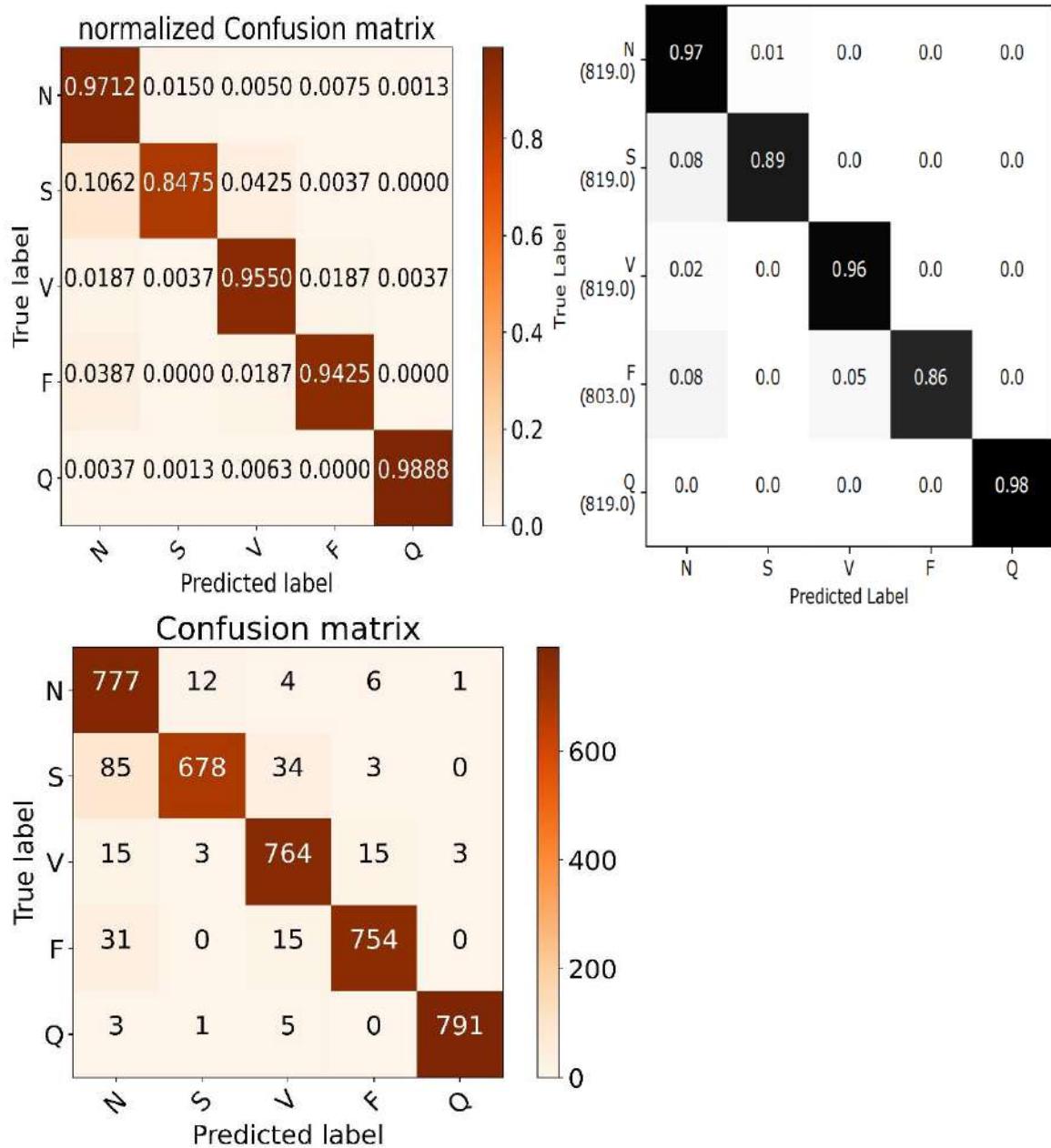


Figure 4-29 version 2 confusion matrices in orange VS [15] confusion matrix in black

*Table 4-17 version 2 classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8529	0.9712	0.9082	800
1	S	0.9769	0.8475	0.9076	800
2	V	0.9294	0.955	0.942	800
3	F	0.9692	0.9425	0.9556	800
4	Q	0.995	0.9888	0.9918	800
accuracy				0.9410	4000
macro avg		0.9447	0.9410	0.9411	4000
weighted avg		0.9447	0.9410	0.9411	4000

## Experiments

### EXP. (1)

A Bidirectional LSTM was added after the max-pooling layer as the LSTM scans the input time series data for dependencies and making it bidirectional makes it scans the data from the start to the end and from the end to the start as if there was a doctor scanning the signal with his eyes.

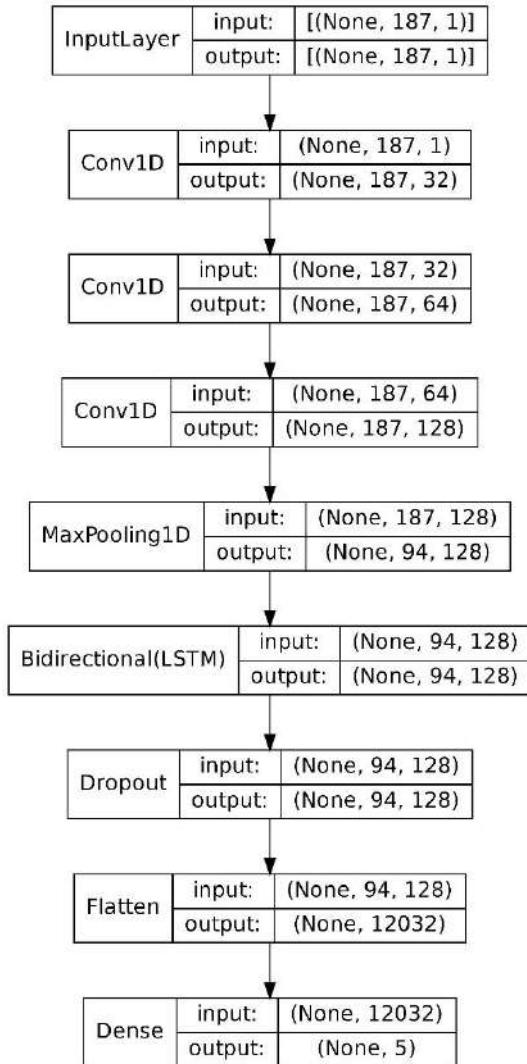


Figure 4-30 version2 EXP. (2) proposed network architecture

Table 4-18 results of version 2 EXP. (2)

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	94.77%	95.08%	94.75%	0.9876
[15]	93.4%			

Table 4-19 version 2 EXP. (2) classification report

index	Class name	Precision	Recall	F1-score	Support
0	N	0.9006	0.9625	0.9305	800
1	S	0.9612	0.8975	0.9282	800
2	V	0.9427	0.9663	0.9543	800
3	F	0.954	0.9325	0.9431	800
4	Q	0.995	0.99	0.9925	800
accuracy					0.9477
macro avg					4000
weighted avg					4000

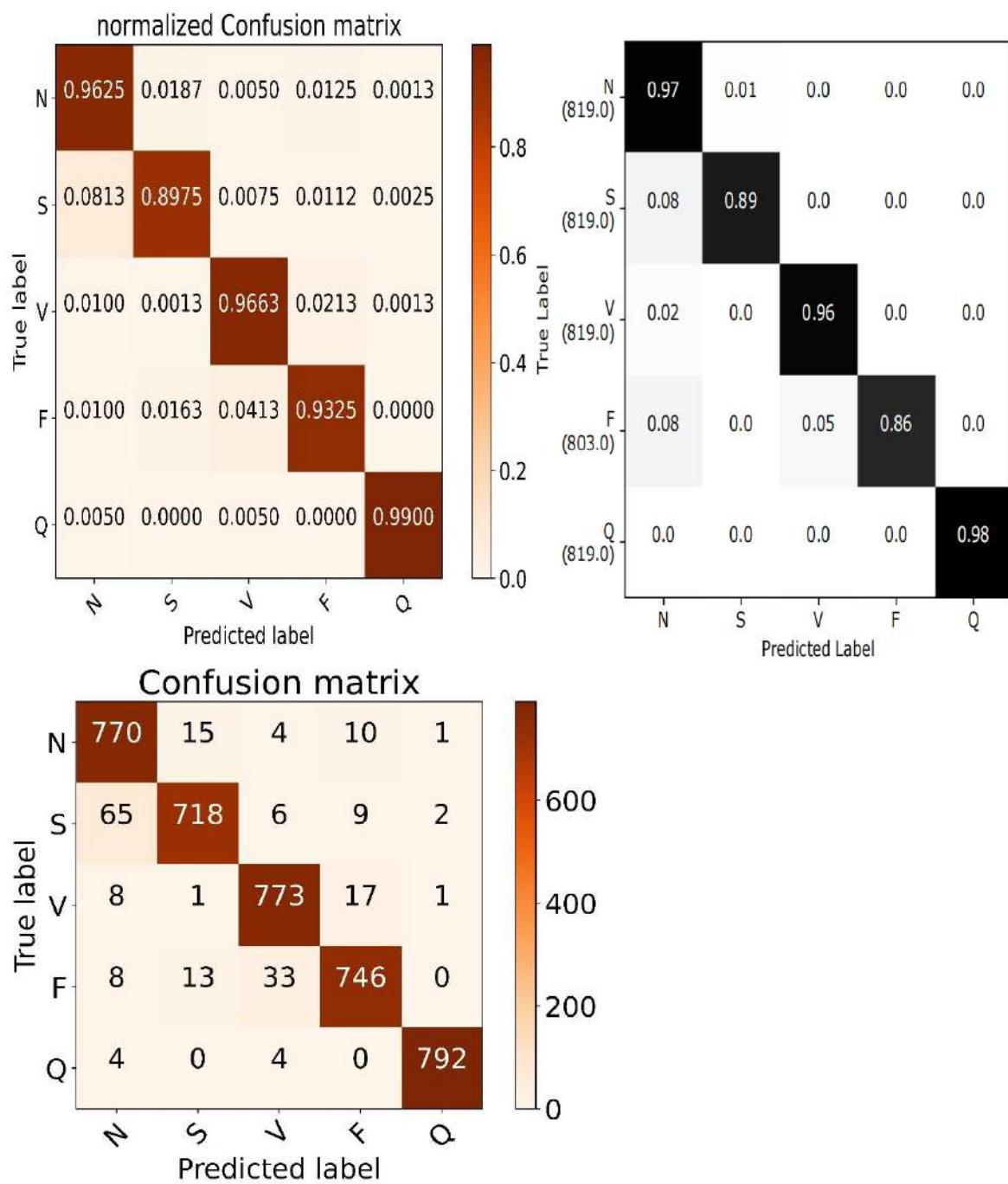


Figure 4-31 version 2 EXP. (2) confusion matrices in orange VS [15] confusion matrix in black

### Version 3

The same model as version 2 was used but the LSTM was replaced with a Multi-Head Attention (MHA) layer, according to [95] the MHA layer is based on the idea of having multiple units or heads operating simultaneously instead of sequentially like the LSTM which reduces the training time, each of these heads takes a part of the input time series data with positional encoding to keep track of which part was given to which head and find out the temporal dependencies between each time step in that part then the outputs of all heads are concatenated with each other to form the final output of the multi-head attention layer which means that the temporal relations between time steps on the short term is considered in the final output which is another advantage of the MHA over the LSTM as its output is based on the final unit only.

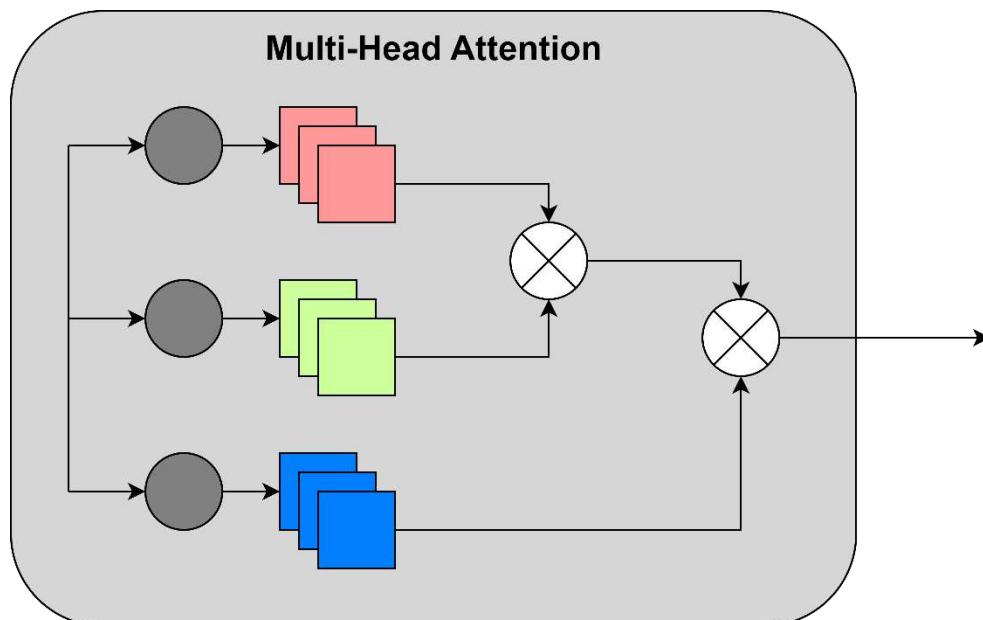


Figure 4-32 Multi-head attention layer block diagram

### Data preprocessing

The training set was re-sampled to 20000 and the test set was re-sampled to 800.

### Proposed framework

The same as version 2 but with a MHA layer instead of the LSTM layer, max-pooling layer set to pooling size of 8 and stride of 2, the MHA layer has a value\_dim = 128 which is equal to the number of channels of its input and it has a key\_dim of 64.

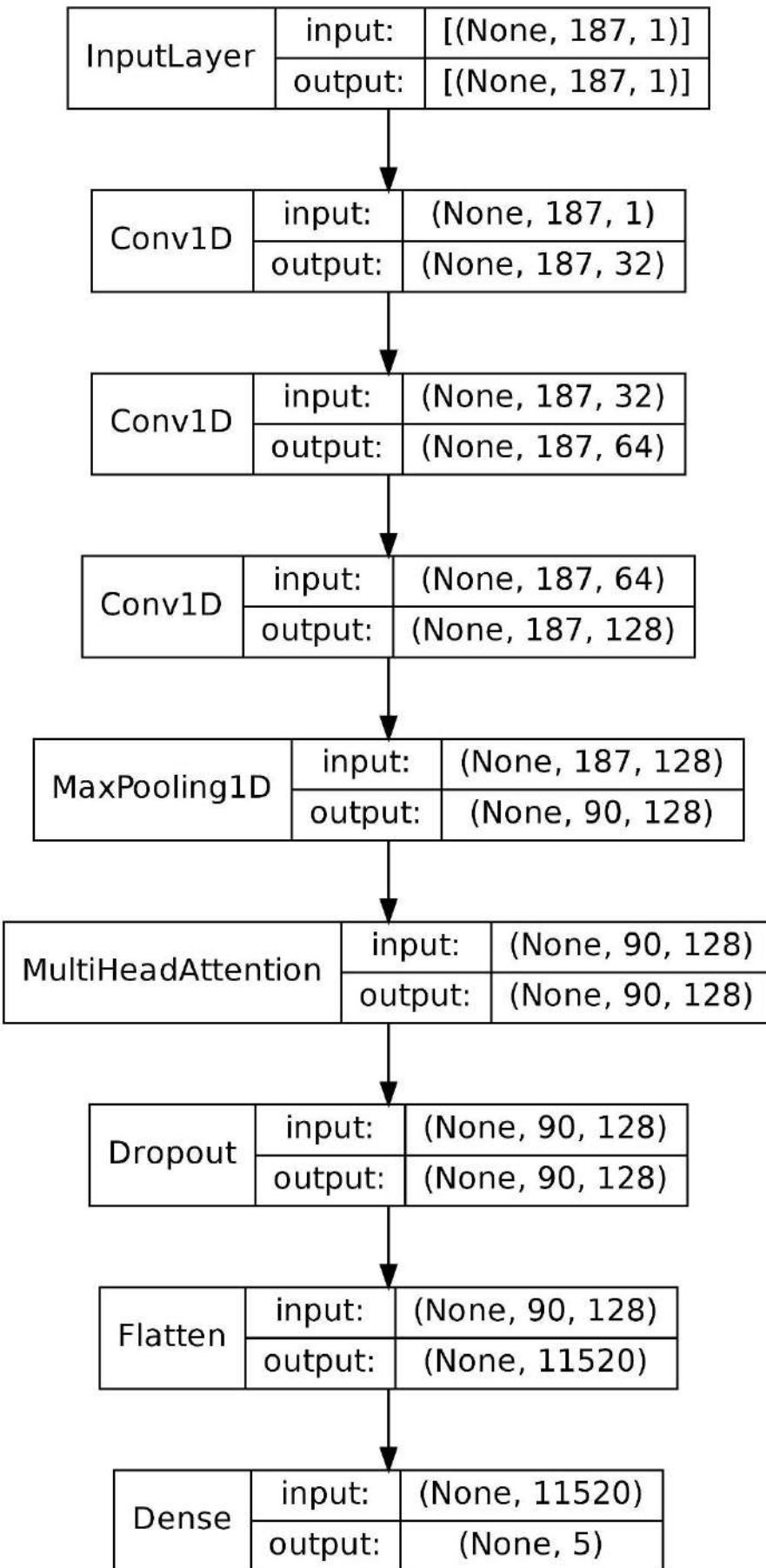


Figure 4-33 version 3 proposed network architecture

The following will demonstrate the tuning process of the number of heads for the MHA layer, note that the number of heads should be dividable by the length of features, in this case the number of feature coming out of the max-pooling layer is 90 so we run our model using 1, 2, 4, 3, 5, 6, 9 and 10 heads.

## EXP. (1)

Number of heads = 1

Table 4-20 testing results for version 3 EXP. (1)

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	93.48%	93.64%	93.45%	0.9806	93.38	91.84
[15]	93.4%					

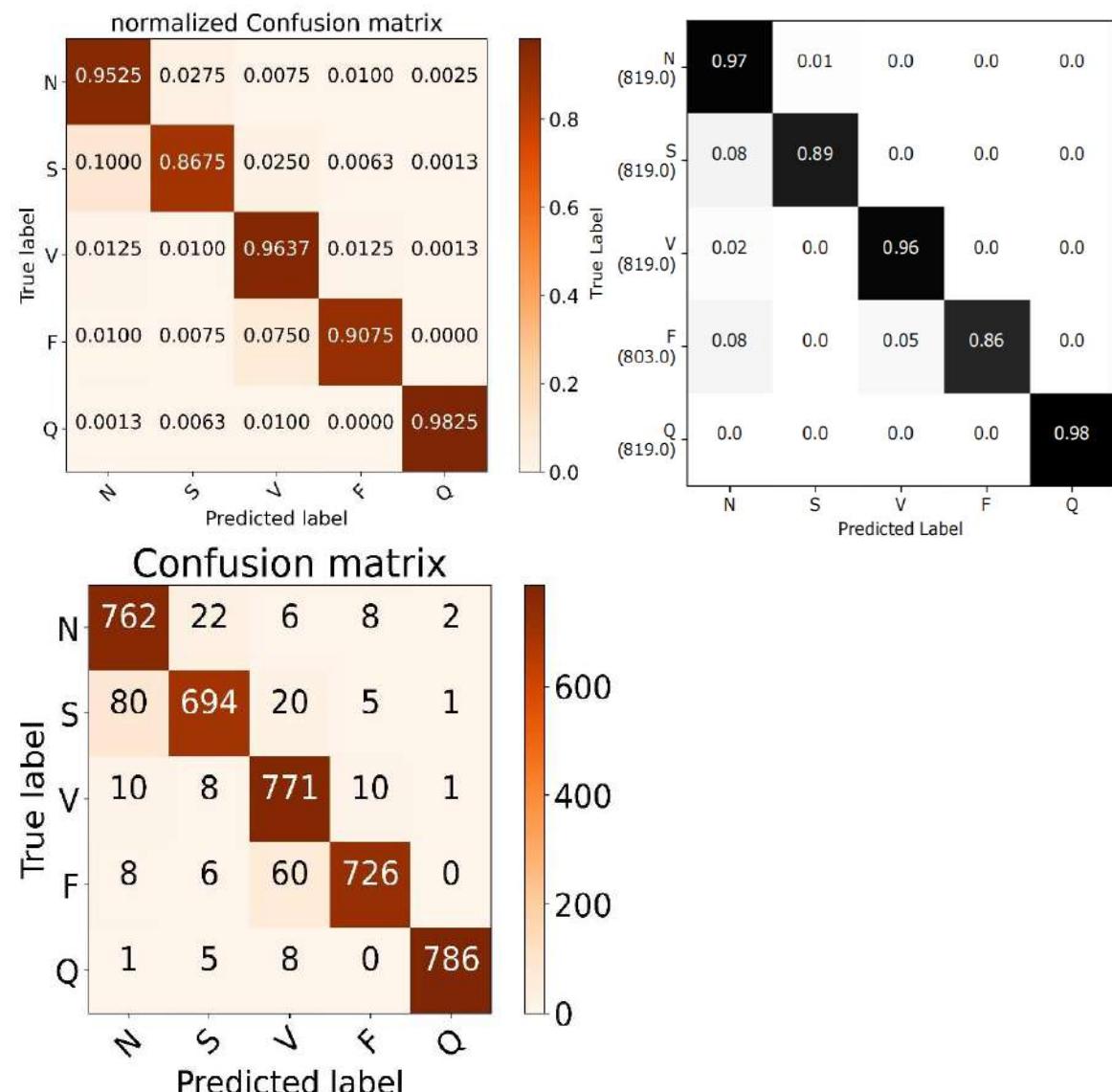


Figure 4-34 version 3 EXP. (1) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-21 version 3 EXP. (1) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.885	0.9525	0.9175	800
1	S	0.9442	0.8675	0.9042	800
2	V	0.8913	0.9637	0.9261	800
3	F	0.9693	0.9075	0.9374	800
4	Q	0.9949	0.9825	0.9887	800
accuracy				0.9347	4000
macro avg		0.9370	0.9347	0.9348	4000
weighted avg		0.9370	0.9347	0.9348	4000

## **EXP. (2)**

Number of heads = 2

*Table 4-22 testing results for version 3 EXP. (2)*

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	94.05%	94.14%	94.02%	0.9838	93.93%	92.56%
[15]	93.4%					

*Table 4-23 version 3 EXP. (2) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.845	0.9675	0.9021	800
1	S	0.9687	0.8512	0.9062	800
2	V	0.9503	0.9563	0.9533	800
3	F	0.9691	0.94	0.9543	800
4	Q	0.9875	0.9875	0.9875	800
accuracy				0.9405	4000
macro avg		0.9441	0.9405	0.9407	4000
weighted avg		0.9441	0.9405	0.9407	4000

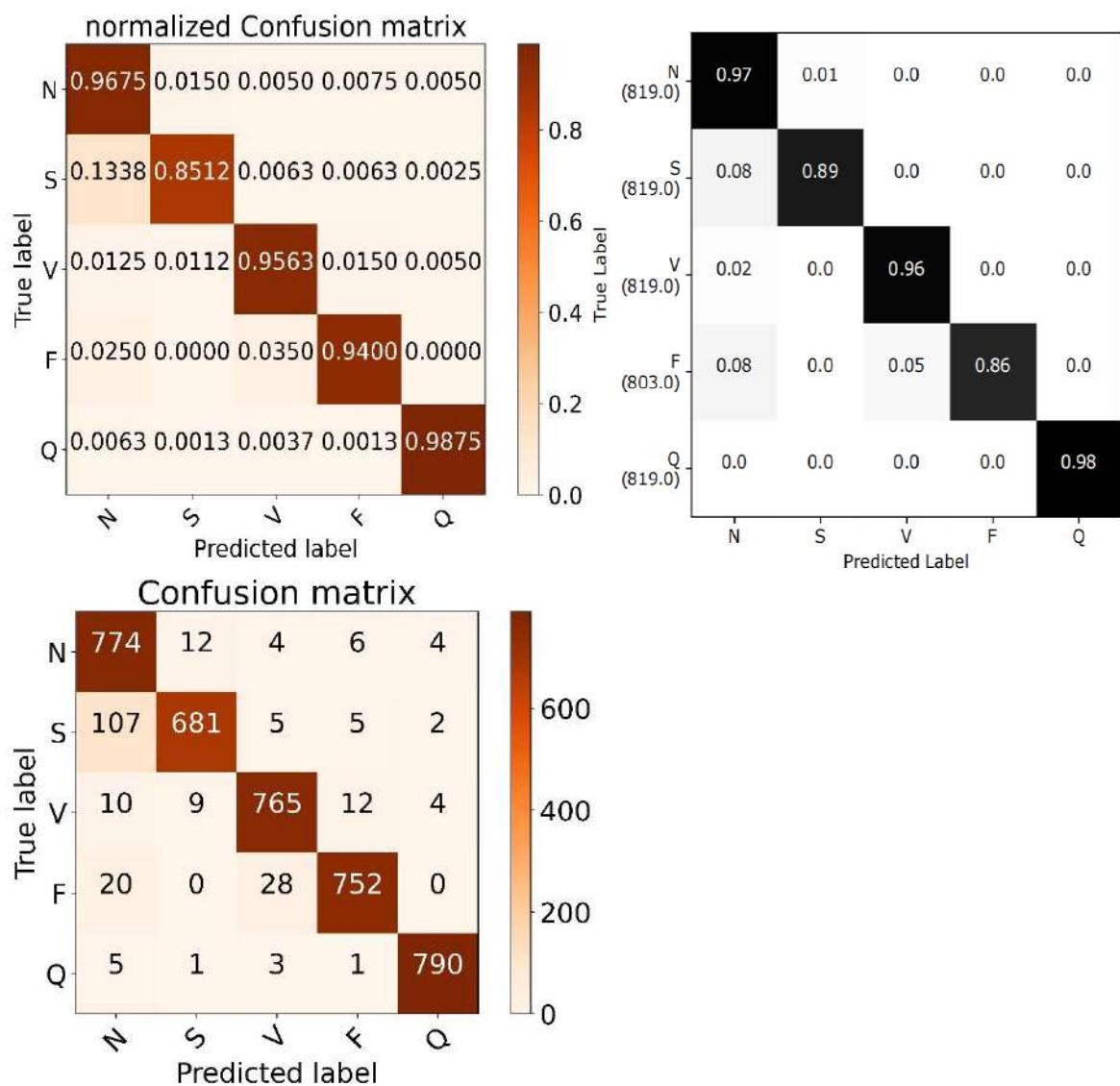


Figure 4-35 version 3 EXP. (2) confusion matrices in orange VS [15] confusion matrix in black

### EXP. (3)

Number of heads = 3

Table 4-24 testing results for version 3 EXP. (3)

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	93.70%	93.77%	93.65%	0.9811	93.62%	92.12%
[15]	93.4%					

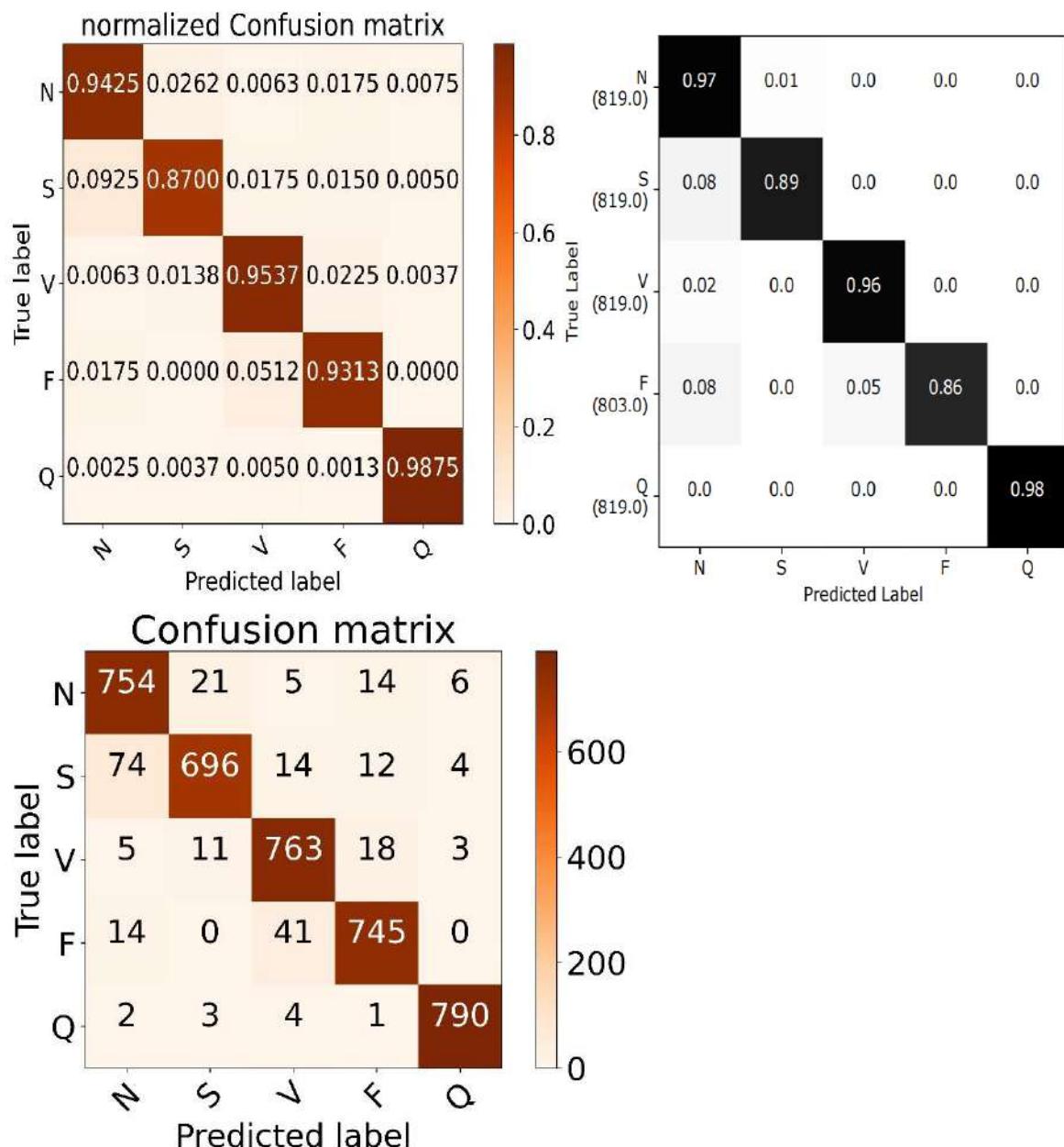


Figure 4-36 version 3 EXP. (3) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-25 version 3 EXP. (3) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8881	0.9425	0.9145	800
1	S	0.9521	0.87	0.9092	800
2	V	0.9226	0.9537	0.9379	800
3	F	0.943	0.9313	0.9371	800
4	Q	0.9838	0.9875	0.9857	800
accuracy				0.9370	4000
macro avg		0.9379	0.9370	0.9369	4000
weighted avg		0.9379	0.9370	0.9369	4000

## **EXP. (4)**

Number of heads = 5

*Table 4-26 testing results for version 3 EXP. (4)*

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	94.15%	94.20%	94.15%	0.9782	94.07%	92.69%
[15]	93.4%					

*Table 4-27 version 3 EXP. (4) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8582	0.9762	0.9135	800
1	S	0.9446	0.895	0.9191	800
2	V	0.9477	0.9513	0.9495	800
3	F	0.9742	0.8962	0.9336	800
4	Q	0.9975	0.9888	0.9931	800
accuracy				0.9415	4000
macro avg		0.9444	0.9415	0.9417	4000
weighted avg		0.9444	0.9415	0.9417	4000

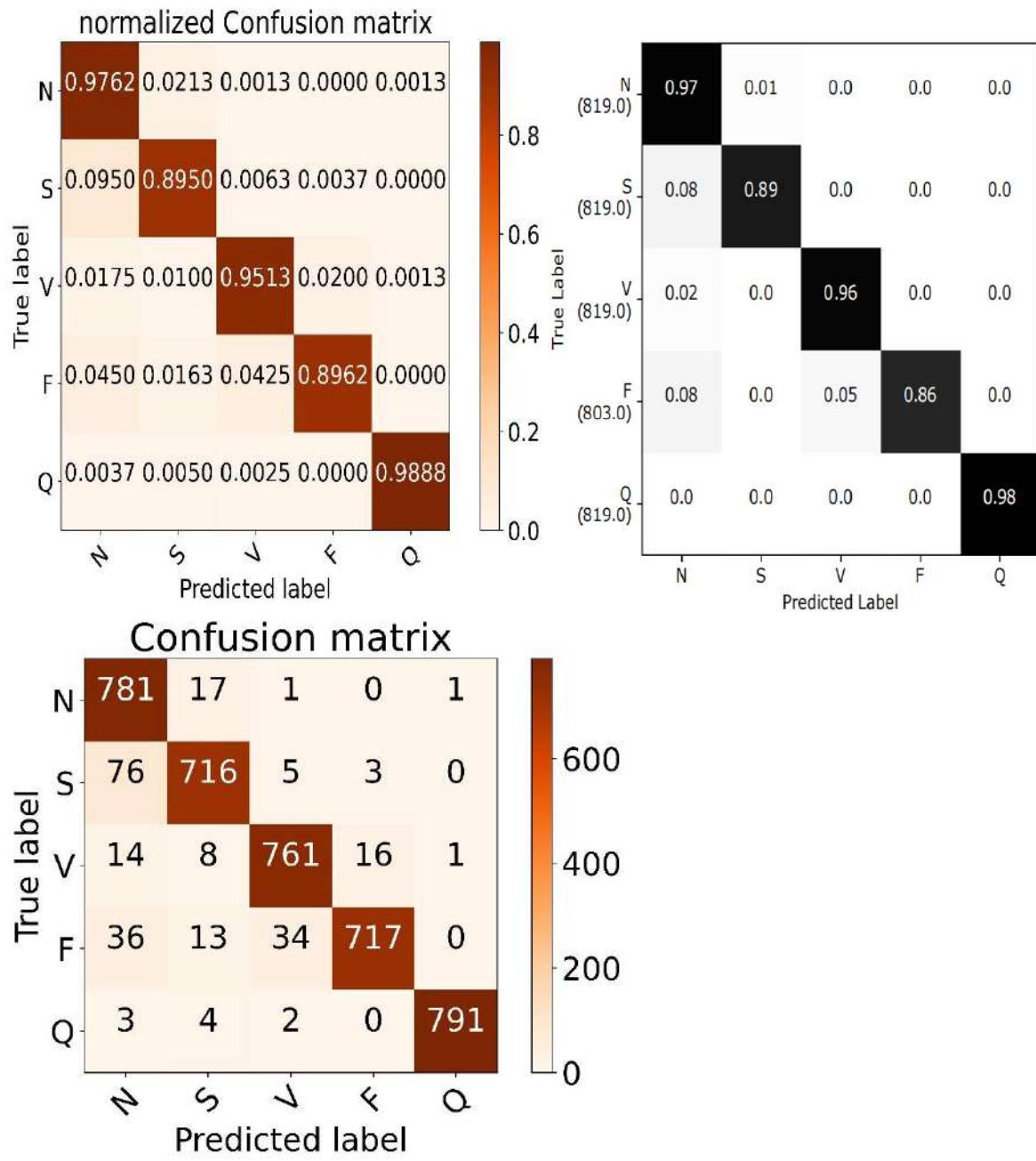


Figure 4-37 version 3 EXP. (4) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (5)

Number of heads = 6

Table 4-28 testing results for version 3 EXP. (5)

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	93.75%	93.77%	93.70%	0.9758	93.68%	92.19%
[15]	93.4%					

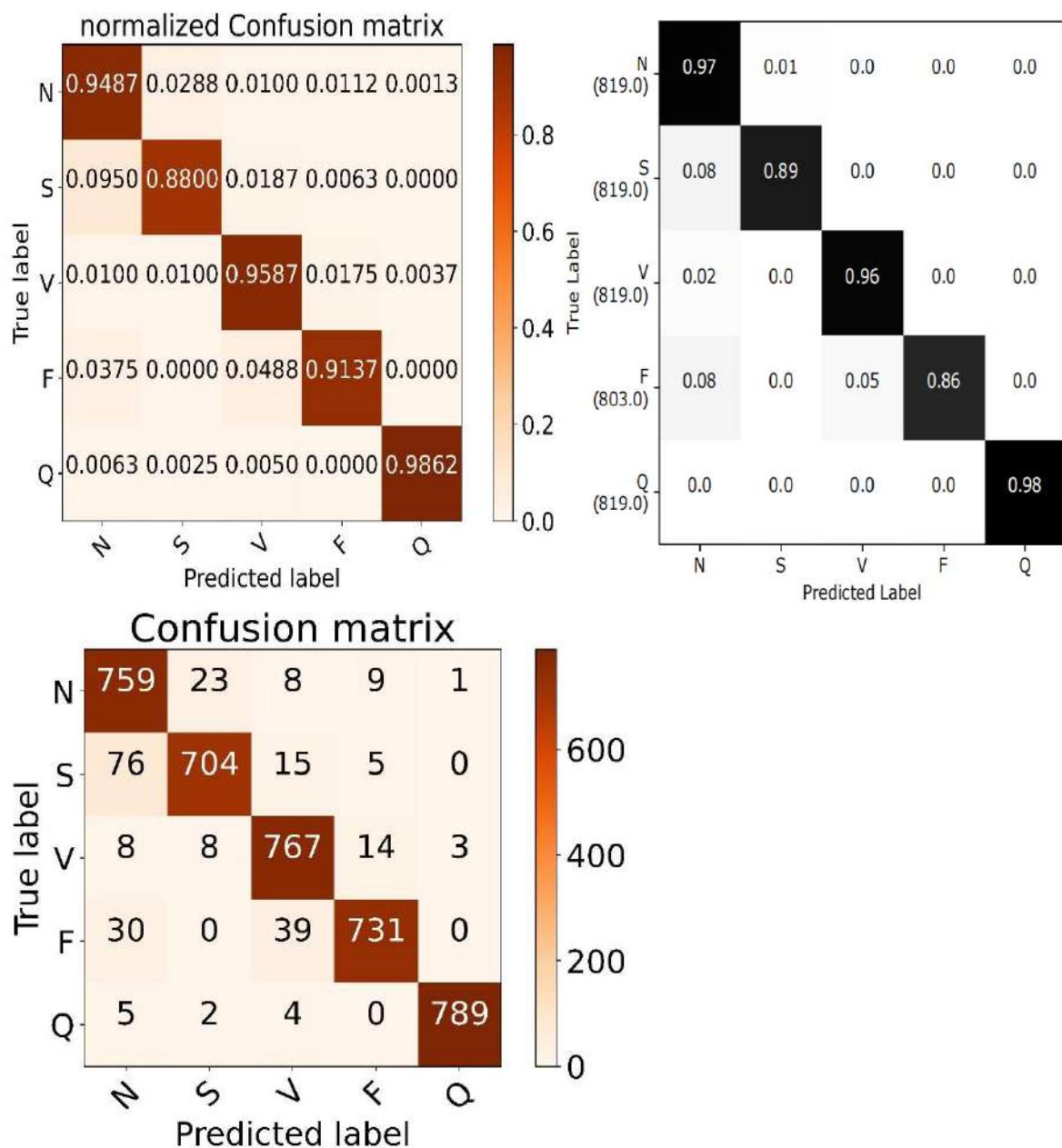


Figure 4-38 version 3 EXP. (5) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-29 version 3 EXP. (5) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8645	0.9487	0.9046	800
1	S	0.9552	0.88	0.9161	800
2	V	0.9208	0.9587	0.9394	800
3	F	0.9631	0.9137	0.9378	800
4	Q	0.995	0.9862	0.9906	800
accuracy				0.9375	4000
macro avg		0.9397	0.9375	0.9377	4000
weighted avg		0.9397	0.9375	0.9377	4000

## **EXP. (6)**

Number of heads = 9

*Table 4-30 testing results for version 3 EXP. (6)*

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	94.20%	94.22%	94.20%	0.9745	94.13%	92.75%
[15]	93.4%					

*Table 4-31 version 3 EXP. (6) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8674	0.965	0.9136	800
1	S	0.9659	0.8862	0.9244	800
2	V	0.9213	0.9663	0.9433	800
3	F	0.9733	0.91	0.9406	800
4	Q	0.9962	0.9825	0.9893	800
accuracy				0.9420	4000
macro avg		0.9448	0.9420	0.9422	4000
weighted avg		0.9448	0.9420	0.9422	4000

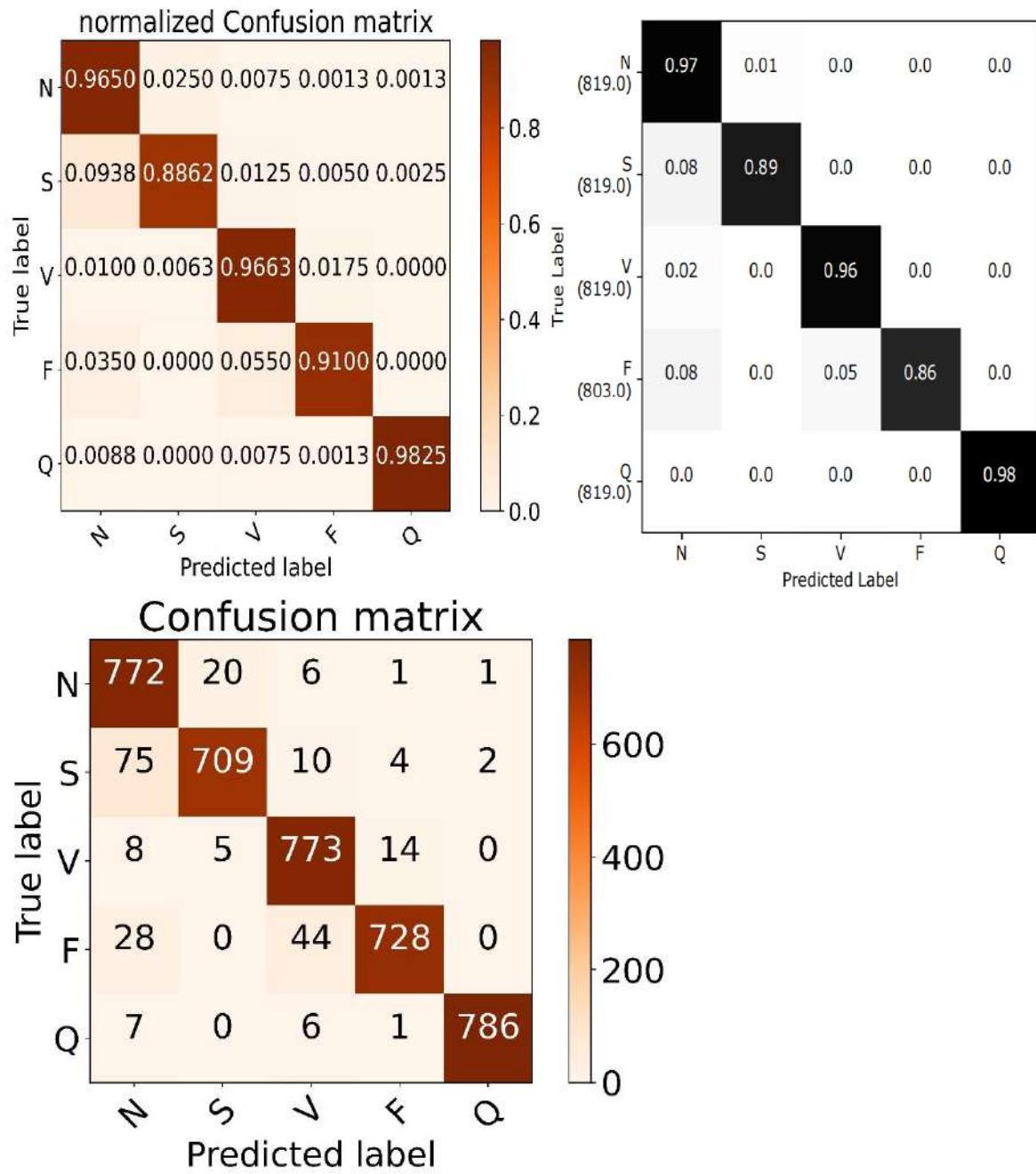


Figure 4-39 version 3 EXP. (6) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (7)

Number of heads = 10

Table 4-32 testing results for version 3 EXP. (7)

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	94.25%	94.53%	94.23%	0.9891	94.18%	92.81%
[15]	93.4%					

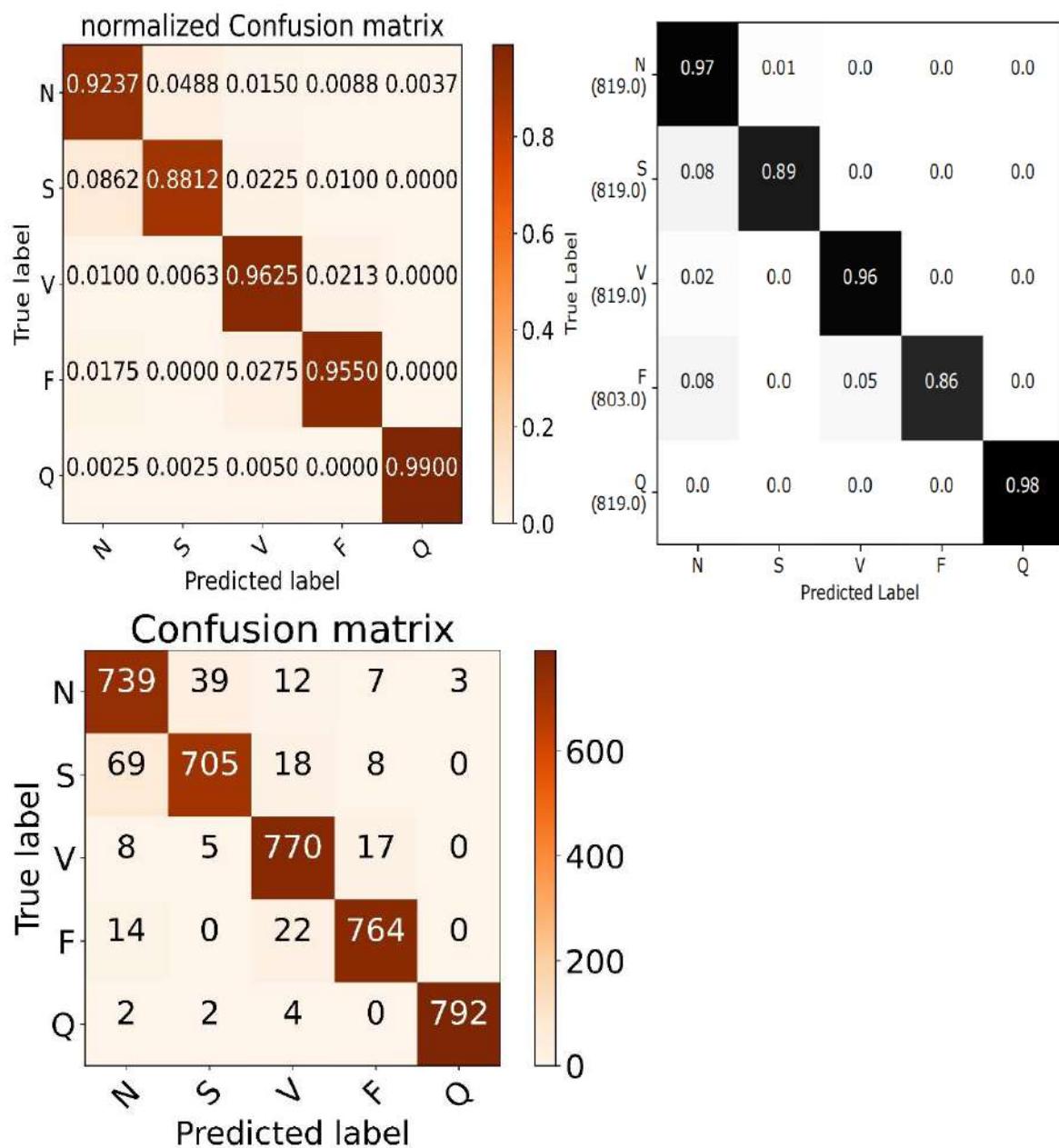


Figure 4-40 version 3 EXP. (7) confusion matrices in orange VS [15] confusion matrix in black

Table 4-33 version 3 EXP. (7) classification report

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8882	0.9237	0.9056	800
1	S	0.9387	0.8812	0.9091	800
2	V	0.9322	0.9625	0.9471	800
3	F	0.9598	0.955	0.9574	800
4	Q	0.9962	0.99	0.9931	800
accuracy				0.9425	4000
macro avg		0.9430	0.9425	0.9425	4000
weighted avg		0.9430	0.9425	0.9425	4000

The following experiments will be based on two parallel branches each with a MHA layer, one with a low number of heads and the other with a high number of heads which will help capture the long term dependencies as well as the short term ones.

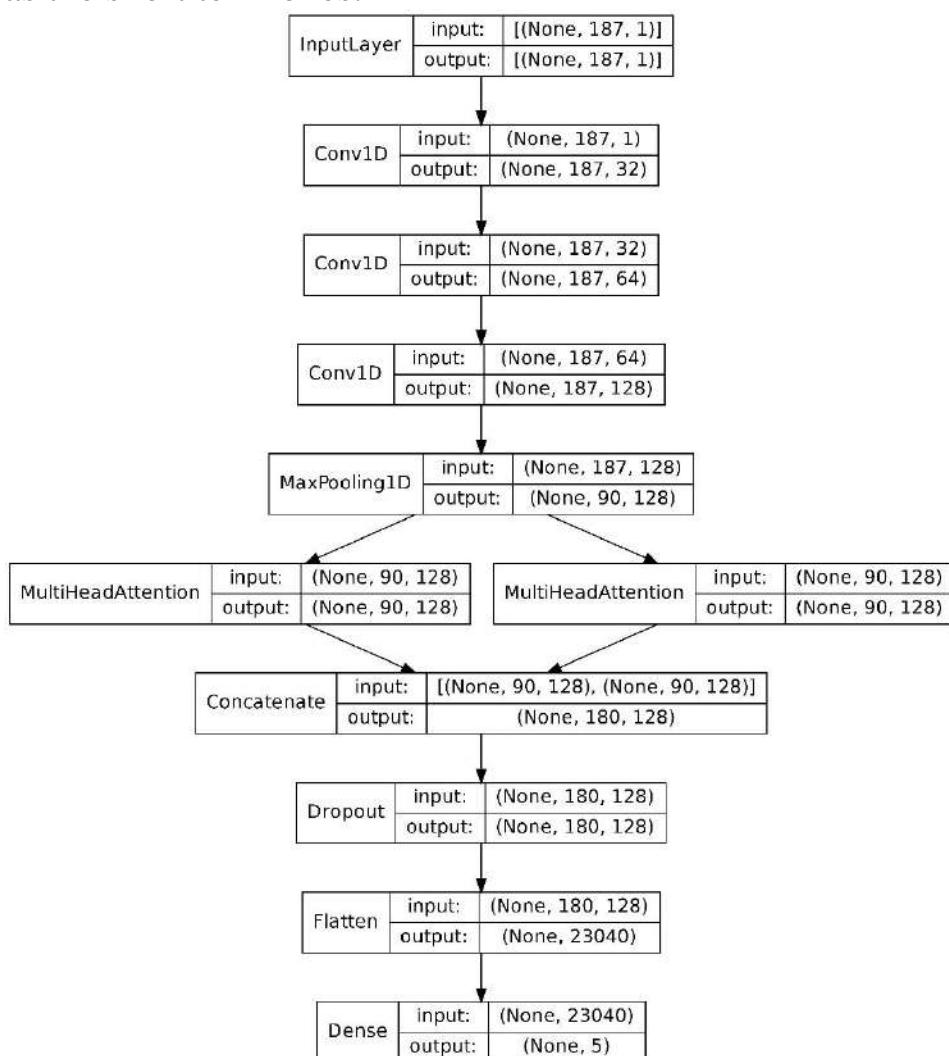


Figure 4-41 version 3 EXP. (8) proposed network architecture

## EXP. (8)

Number of heads = 2-6

Table 4-34 testing results for version 3 EXP. (8)

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	94.82%	95.03%	94.73%	0.9855	94.77%	93.53%
[15]	93.4%					

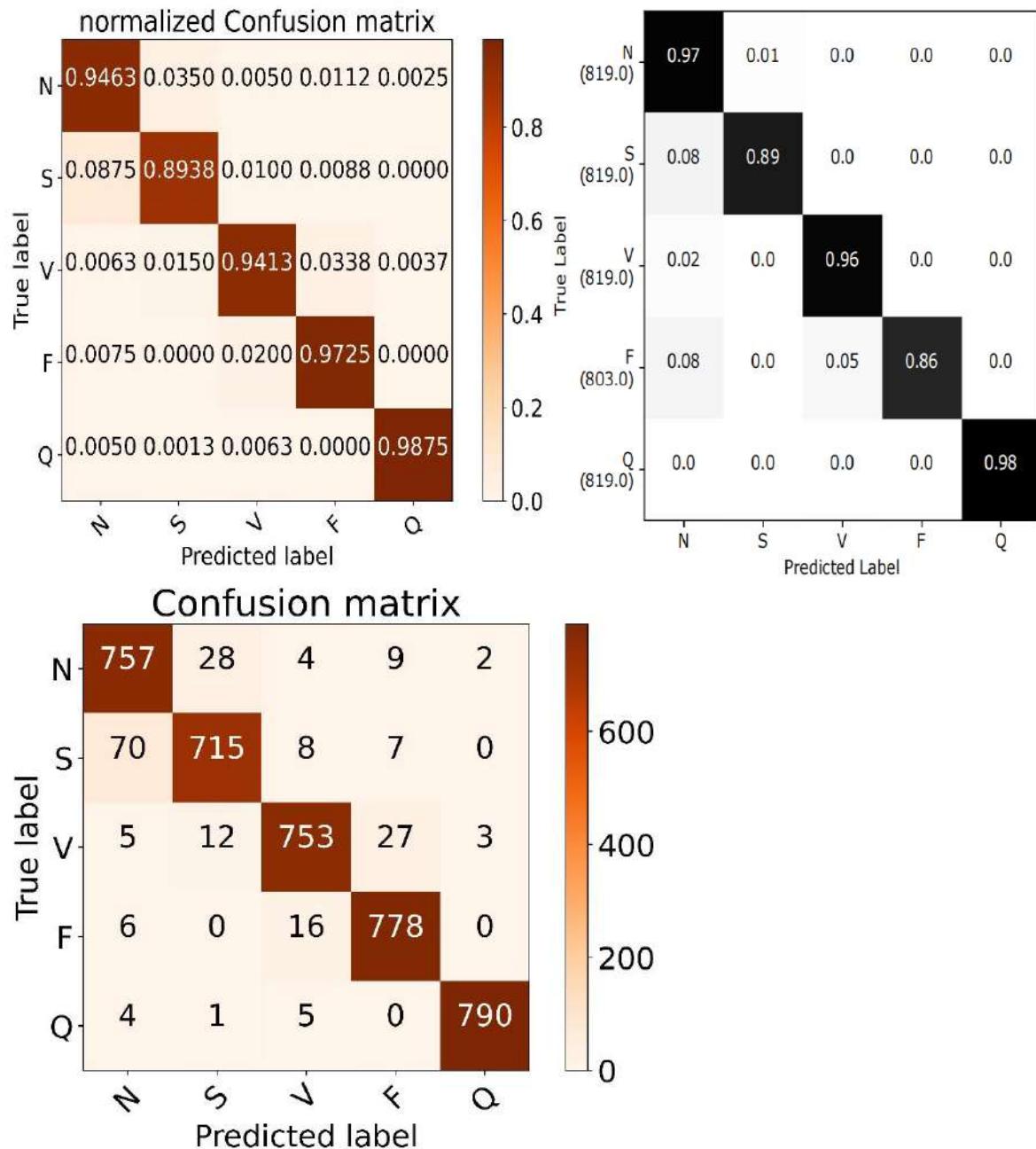


Figure 4-42 version 3 EXP. (8) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-35 version 3 EXP. (8) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.9043	0.945	0.9242	800
1	S	0.9458	0.8725	0.9077	800
2	V	0.9144	0.9613	0.9372	800
3	F	0.9491	0.9325	0.9407	800
4	Q	0.99	0.9888	0.9894	800
accuracy				0.9483	4000
macro avg		0.9488	0.9483	0.9482	4000
weighted avg		0.9488	0.9483	0.9482	4000

## **EXP. (9)**

Number of heads = 3-9

*Table 4-36 testing results for version 3 EXP. (9)*

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	94.00%	94.07%	93.98%	0.9793	93.92%	92.50%
[15]	93.4%					

*Table 4-37 version 3 EXP. (9) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.9043	0.945	0.9242	800
1	S	0.9458	0.8725	0.9077	800
2	V	0.9144	0.9613	0.9372	800
3	F	0.9491	0.9325	0.9407	800
4	Q	0.99	0.9888	0.9894	800
accuracy				0.9400	4000
macro avg		0.9407	0.9400	0.9398	4000
weighted avg		0.9407	0.9400	0.9398	4000

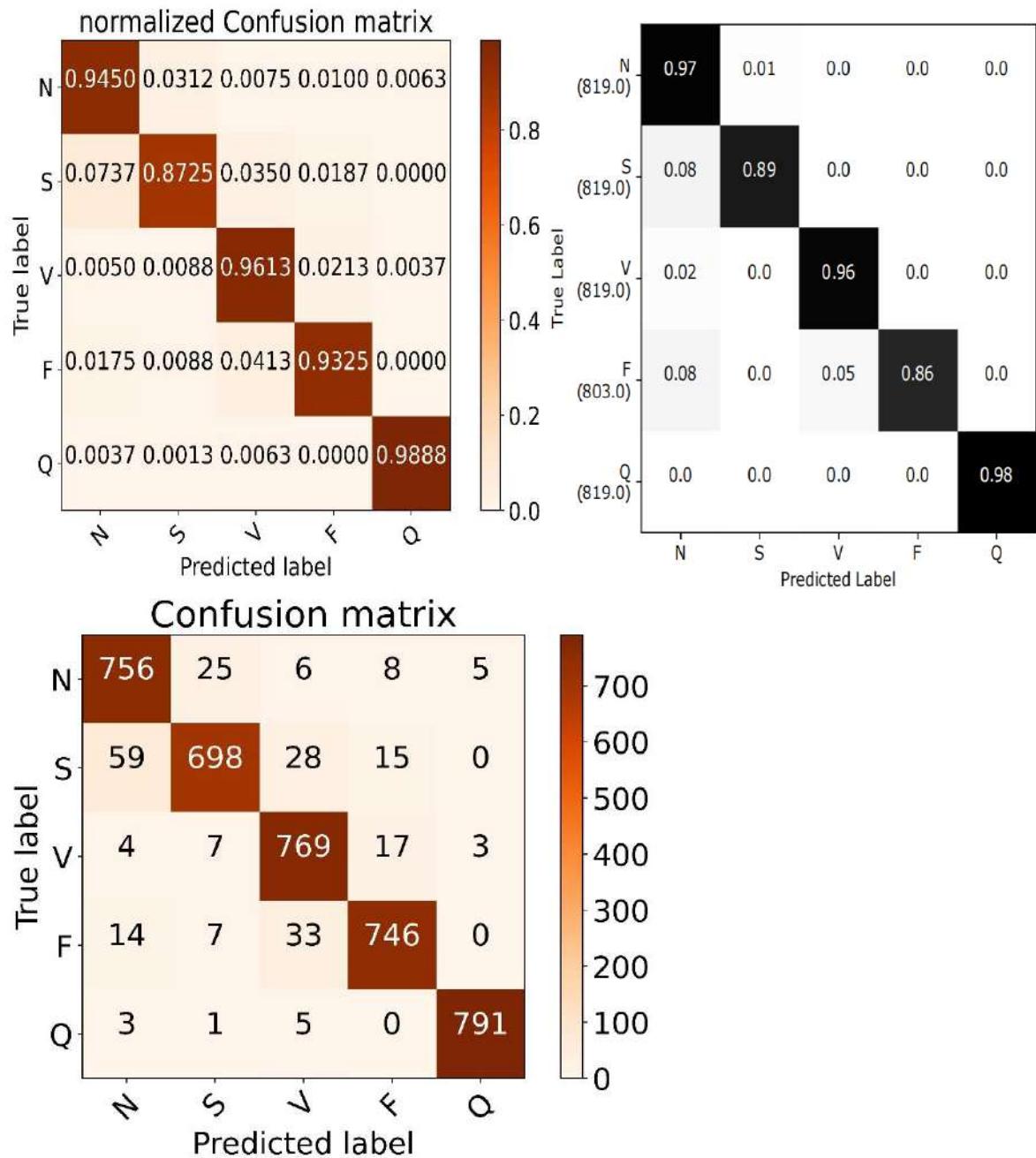


Figure 4-43 version 3 EXP. (9) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (10)

Number of heads = 5-10

Table 4-38 testing results for version 3 EXP. (10)

	Accuracy	precision	Recall	AUC	G-mean	kappa
Proposed	94.25%	94.27%	94.25%	0.9789	94.19%	92.81%
[15]	93.4%					

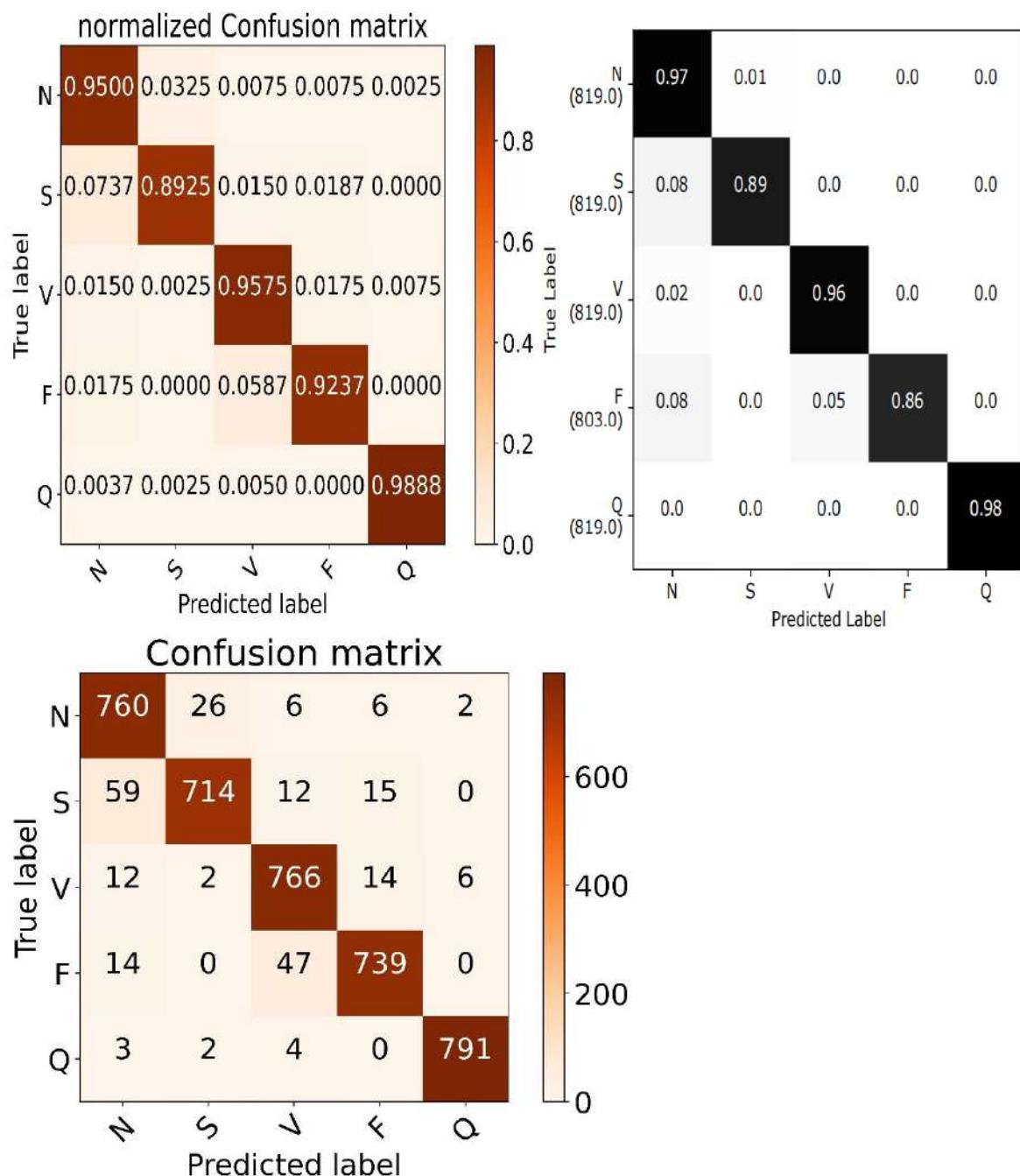


Figure 4-44 version 3 EXP. (10) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-39 version 3 EXP. (10) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8962	0.95	0.9223	800
1	S	0.9597	0.8925	0.9249	800
2	V	0.9174	0.9575	0.937	800
3	F	0.9548	0.9237	0.939	800
4	Q	0.99	0.9888	0.9894	800
accuracy				0.9425	4000
macro avg		0.9436	0.9425	0.9425	4000
weighted avg		0.9436	0.9425	0.9425	4000

## EXP. (11)

In this experiment the output of the max-pooling layer was passed as it is and inverted to two branches of MHA layer to imitate the idea of a bidirectional layer, the outputs of the two branches were concatenated, the previous process was repeated again with different number of heads as in EXP. (9), and the outputs of each concatenated branch was concatenated again to have a single output.

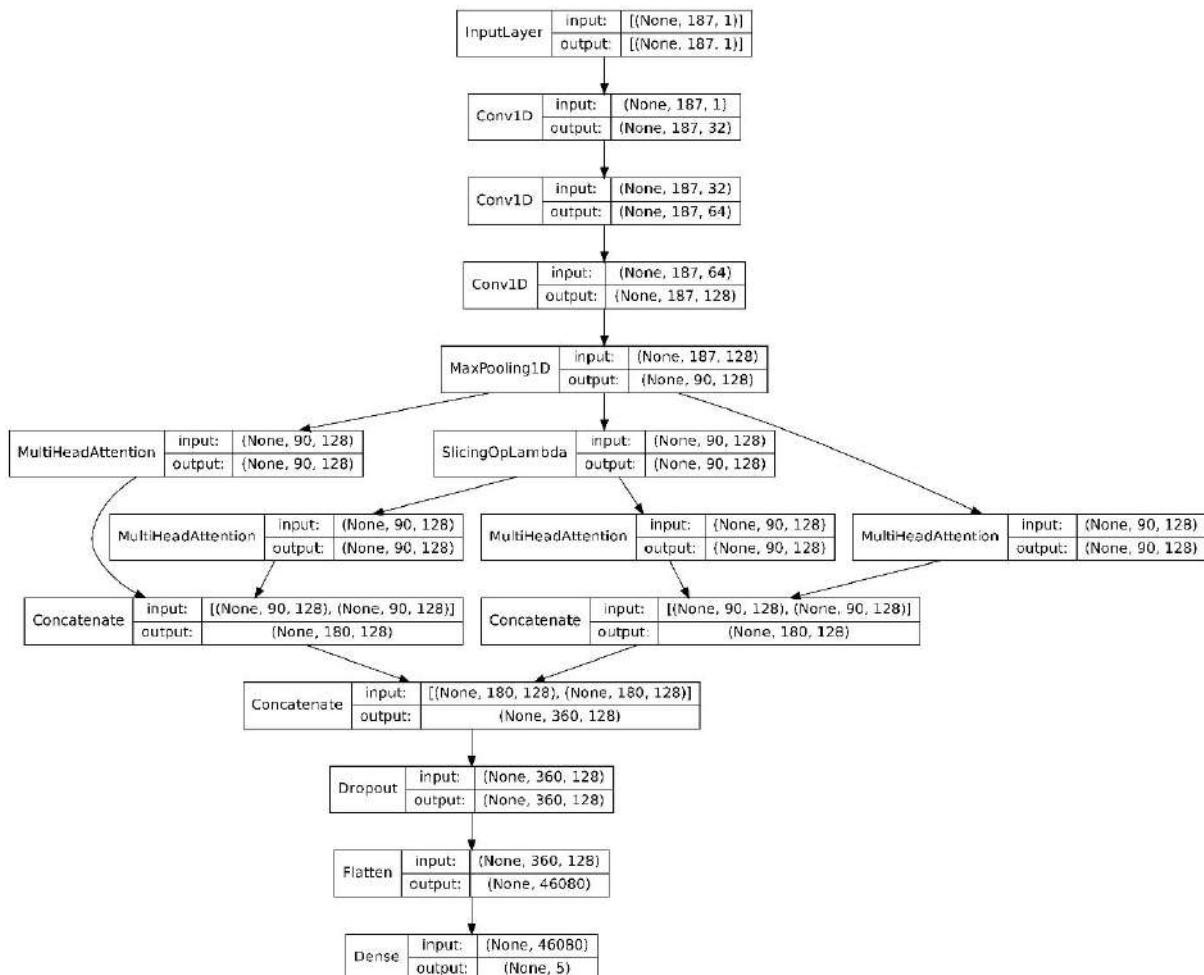


Figure 4-45 version 3 EXP. (11) proposed network architecture

Number of heads = 2-6

Table 4-40 testing results for version 3 EXP. (11)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.15%	94.15%	94.15%	94.15%	0.9704	94.09%	92.69%
[15]	93.4%						

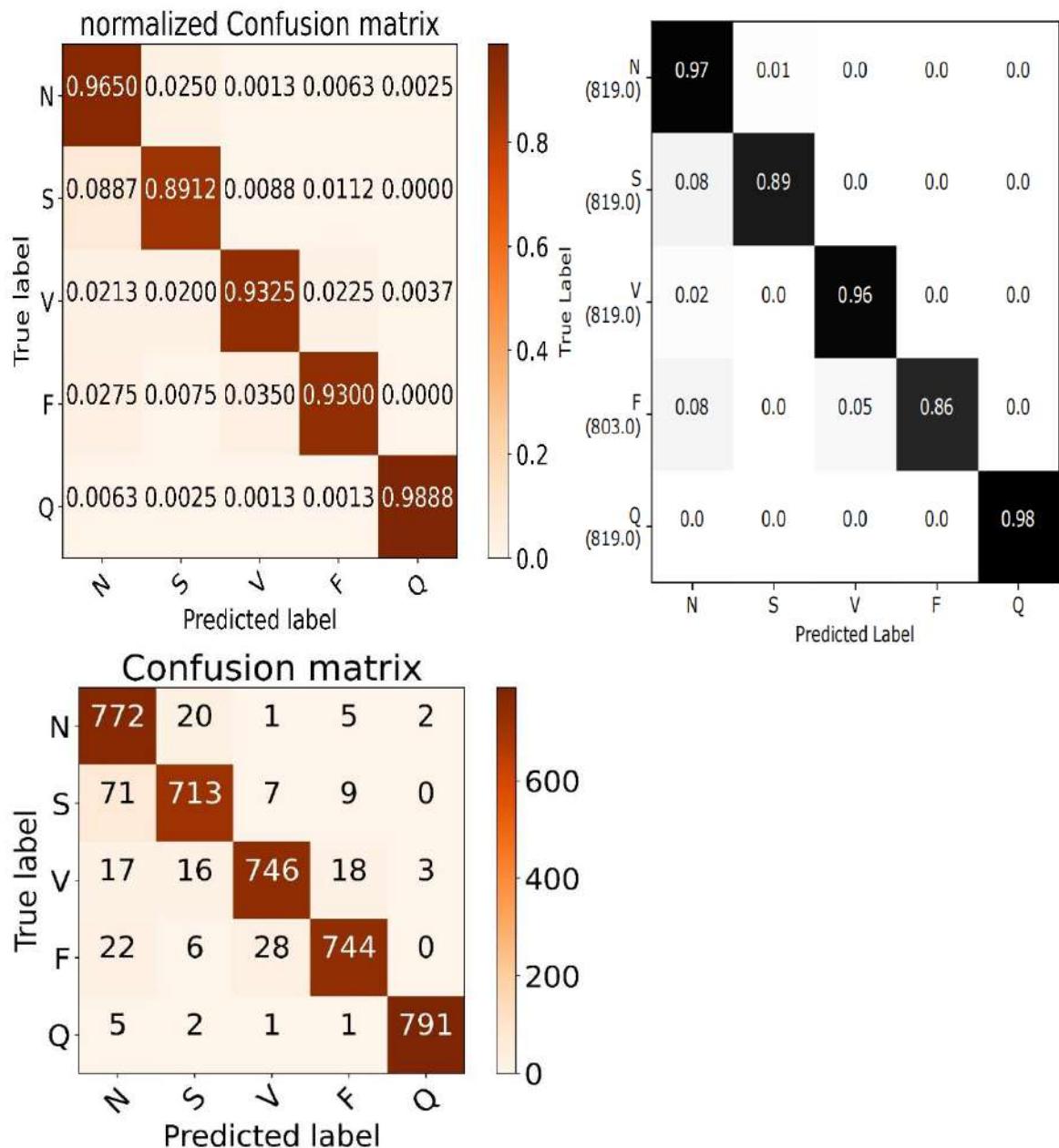


Figure 4-46 version 3 EXP. (11) confusion matrices in orange VS [15] confusion matrix in black

Table 4-41 version 3 EXP. (11) classification report

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8703	0.965	0.9152	800
1	S	0.9419	0.8912	0.9159	800
2	V	0.9527	0.9325	0.9425	800
3	F	0.9575	0.93	0.9436	800
4	Q	0.9937	0.9888	0.9912	800
accuracy				0.9415	4000
macro avg		0.9432	0.9415	0.9417	4000
weighted avg		0.9432	0.9415	0.9417	4000

## Tuning the number of heads of the MHA layer

Table 4-42 tuning the number of heads of the MHA layer

No. of heads	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
<b>Single multi-head attention branch</b>							
<b>1</b>	93.48%	93.64%	93.45%	93.54%	0.9806	93.38%	91.84%
<b>2</b>	94.05%	94.14%	94.02%	94.08%	0.9838	93.93%	92.56%
<b>3</b>	93.70%	93.77%	93.65%	93.71%	0.9811	93.62%	92.12%
<b>5</b>	94.15%	94.20%	94.15%	94.17%	0.9782	94.07%	92.69%
<b>6</b>	93.75%	93.77%	93.70%	93.73%	0.9758	93.68%	92.19%
<b>9</b>	94.20%	94.22%	94.20%	94.22%	0.9745	94.13%	92.75%
<b>10</b>	94.25%	94.53%	94.23%	94.39%	0.9891	94.18%	92.81%
<b>Parallel multi-head attention branches</b>							
<b>2-6</b>	94.82%	95.03%	94.73%	94.88%	0.9855	94.77%	93.53%
<b>3-9</b>	94.00%	94.07%	93.98%	94.02%	0.9793	93.92%	92.50%
<b>5-10</b>	94.25%	94.27%	94.25%	94.26%	0.9789	94.19%	92.81%
<b>Parallel bidirectional multi-head attention branches</b>							
<b>2-6</b>	94.15%	94.15%	94.15%	94.15%	0.9704	94.09%	92.69%

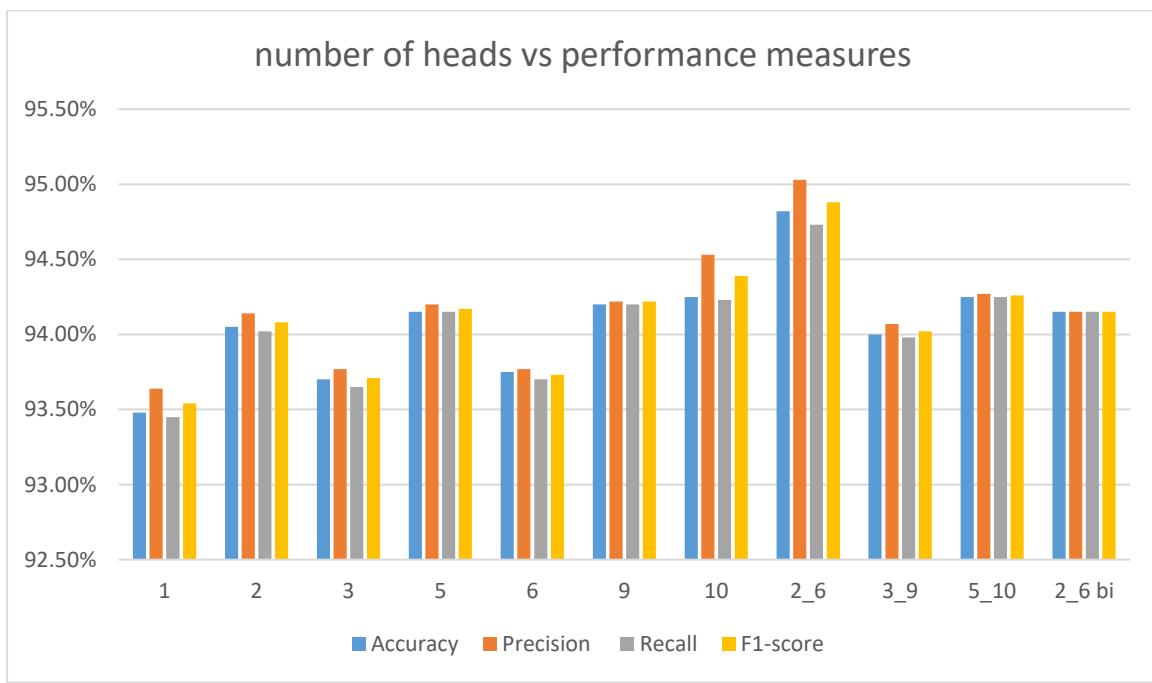


Figure 4-47 no. of heads vs performance measures

The numbers under each 4 bars represent the number of heads used in that experiment, 2\_6 indicates that a parallel branch was used one with 2 heads and the other with 6 heads, 2\_6 bi indicates the bidirectional branches.

## Version 4

In this version an adaptive feature recalibration (AFR) layer was used after the max-pooling layer, according to [95] the AFR layer is a neural network layer that aims to improve the performance of deep learning models using the **squeeze and excitation** technique that selectively emphasizing informative features and suppressing less important ones, **the squeezing** operation involves reducing the number of channels in a feature map by applying global average pooling, which averages the feature maps along the spatial dimensions. This results in a feature vector that contains information about the presence or absence of certain features across the entire feature map, **the excitation** operation involves learning a set of weights that represent the importance of each channel, this is done by applying a set of fully connected layers to the squeezed feature vector, followed by a sigmoid or relu activation function to ensure positive weights, the resulting weights are then used to rescale the feature maps, emphasizing informative features and suppressing less important ones, **the reduction ratio** is parameter that refers to the ratio of the number of input channels to the number of output channels in the squeezing operation. A higher reduction ratio means that more channels are being compressed into a smaller number of channels, which can help to reduce the computational cost and improve the efficiency of the network

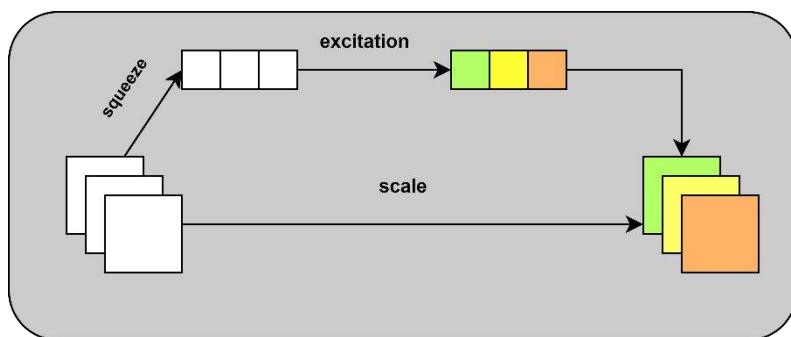


Figure 4-48 Adaptive feature recalibration layer block diagram

## Data preprocessing

The training set was re-sampled to 20000 and the test set was re-sampled to 800.

## Proposed framework

The same as version 2 but with an AFR layer instead of the LSTM layer

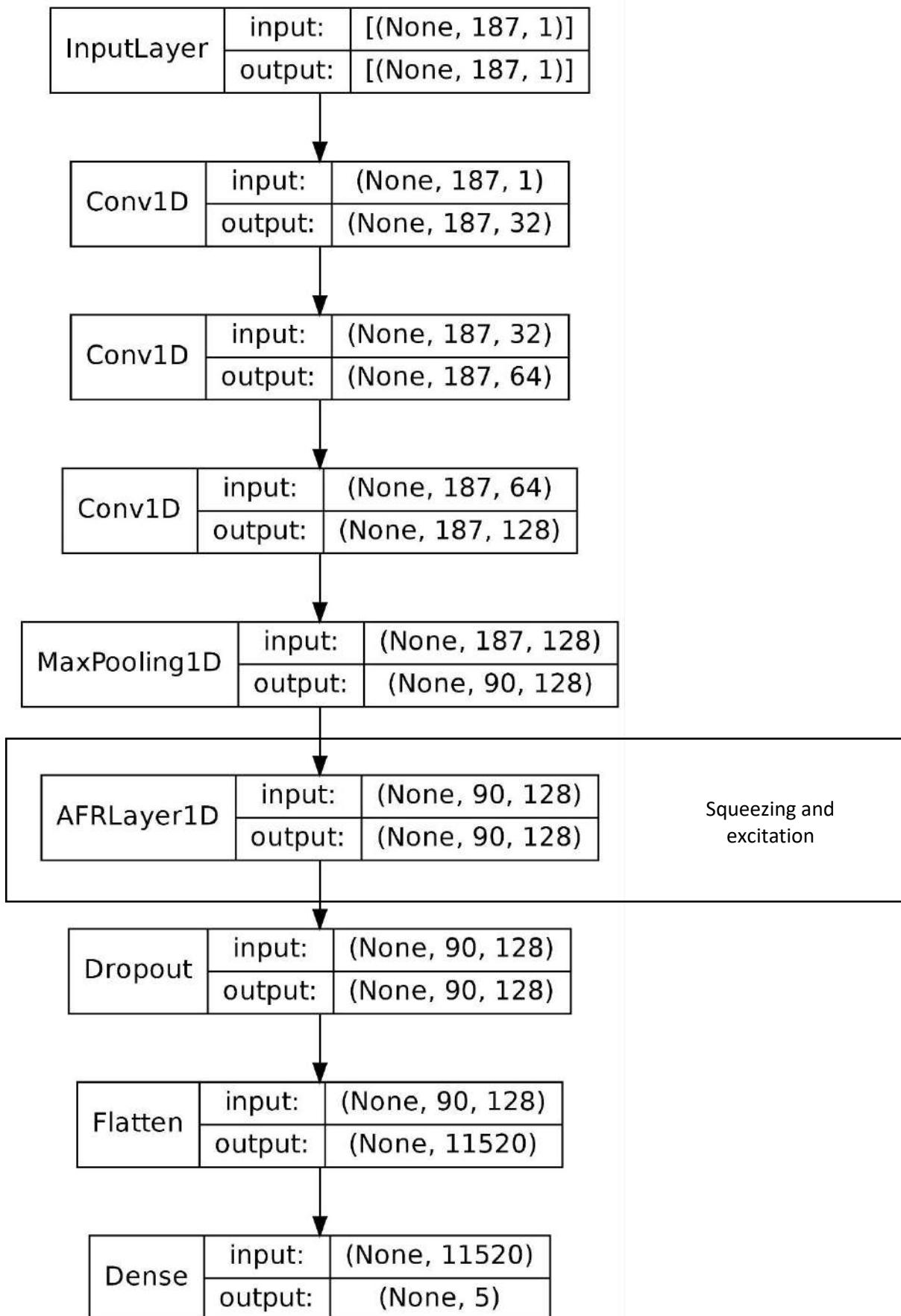


Figure 4-49 version 4 proposed network architecture

## EXP. (1)

Reduction ratio = 4

Table 4-43 testing results for version 4 EXP. (1)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.55%	94.67%	94.52%	94.55%	0.9849	94.48%	93.19%
[15]	93.4%						

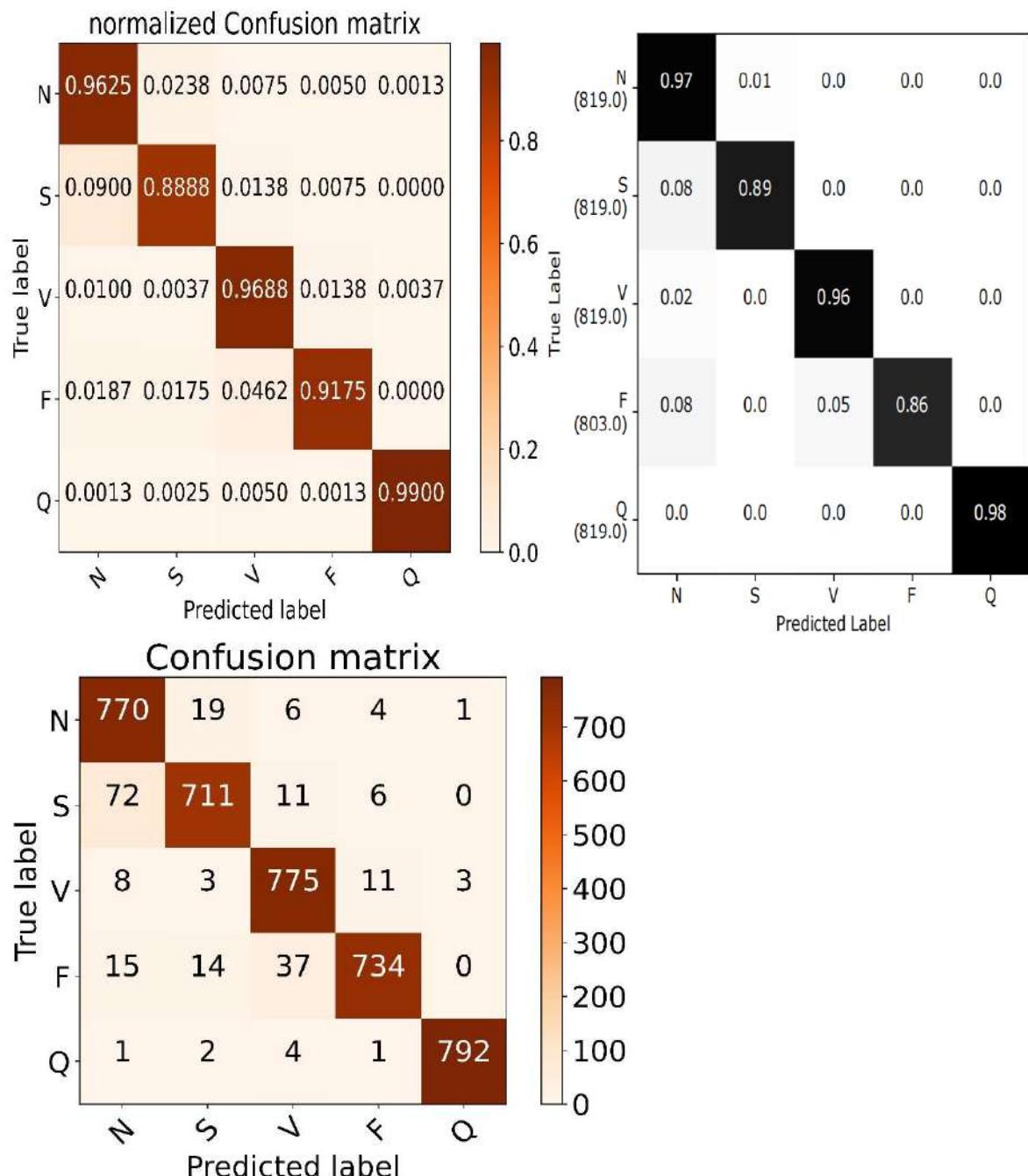


Figure 4-50 version 4 EXP. (1) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-44 version 4 EXP. (1) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8891	0.9625	0.9244	800
1	S	0.9493	0.8888	0.918	800
2	V	0.9304	0.9688	0.9492	800
3	F	0.9709	0.9175	0.9434	800
4	Q	0.995	0.99	0.9925	800
accuracy				0.9455	4000
macro avg		0.9469	0.9455	0.9455	4000
weighted avg		0.9469	0.9455	0.9455	4000

## EXP. (2)

Reduction ratio = 8

*Table 4-45 testing results for version 4 EXP. (2)*

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.55%	94.67%	94.52%	94.55%	0.9849	94.48%	93.19%
[15]	93.4%						

*Table 4-46 version 4 EXP. (2) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8699	0.9613	0.9133	800
1	S	0.9427	0.885	0.913	800
2	V	0.9533	0.9688	0.9609	800
3	F	0.9763	0.925	0.9499	800
4	Q	0.9975	0.99	0.9937	800
accuracy				0.9460	4000
macro avg		0.9479	0.9460	0.9462	4000
weighted avg		0.9479	0.9460	0.9462	4000

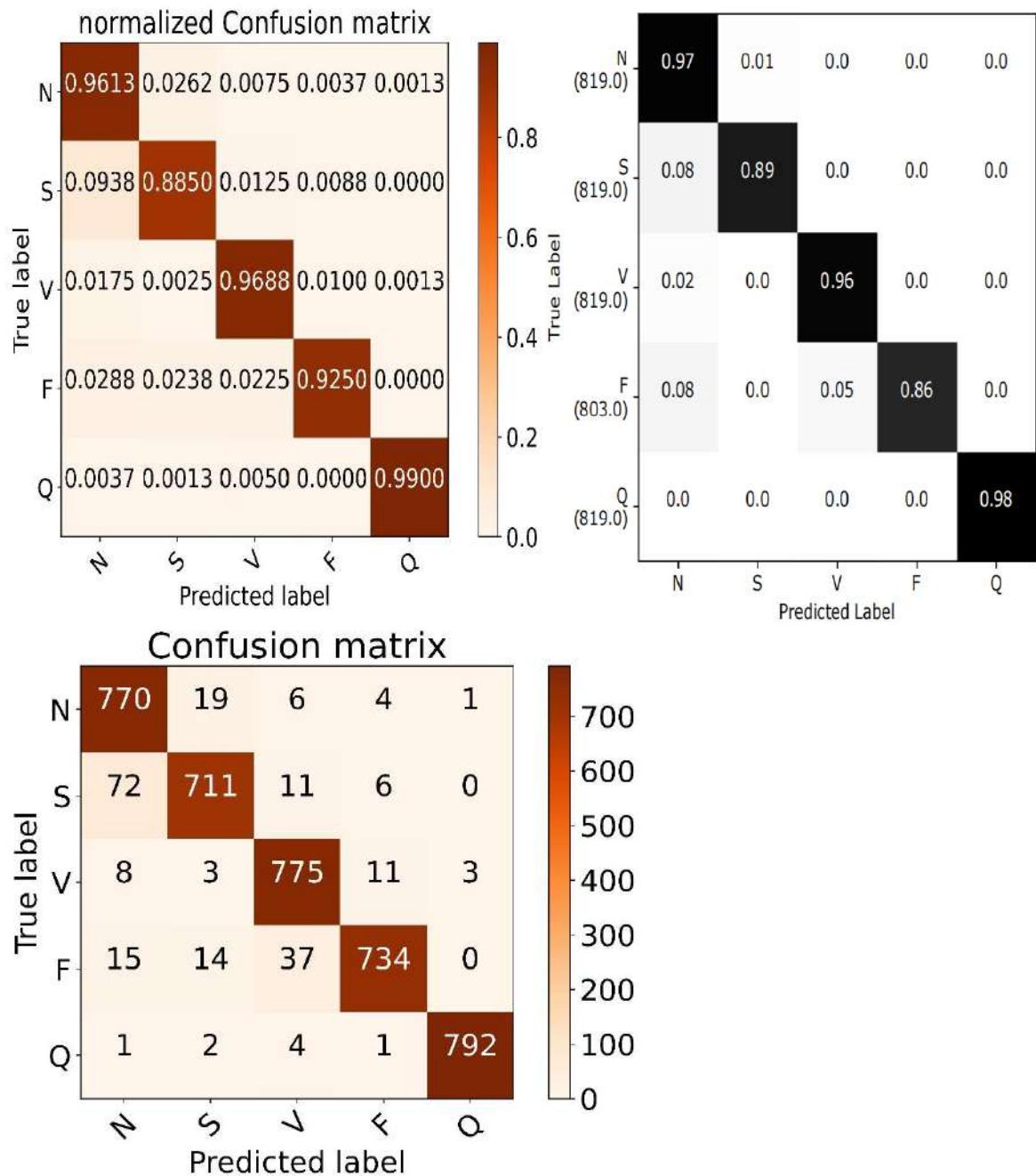


Figure 4-51 version 4 EXP. (2) confusion matrices in orange VS [15] confusion matrix in black

### EXP. (3)

Reduction ratio = 12

Table 4-47 testing results for version 4 EXP. (3)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.40%	94.54%	94.35%	94.40%	0.9806	94.32%	93.00%
[15]	93.4%						

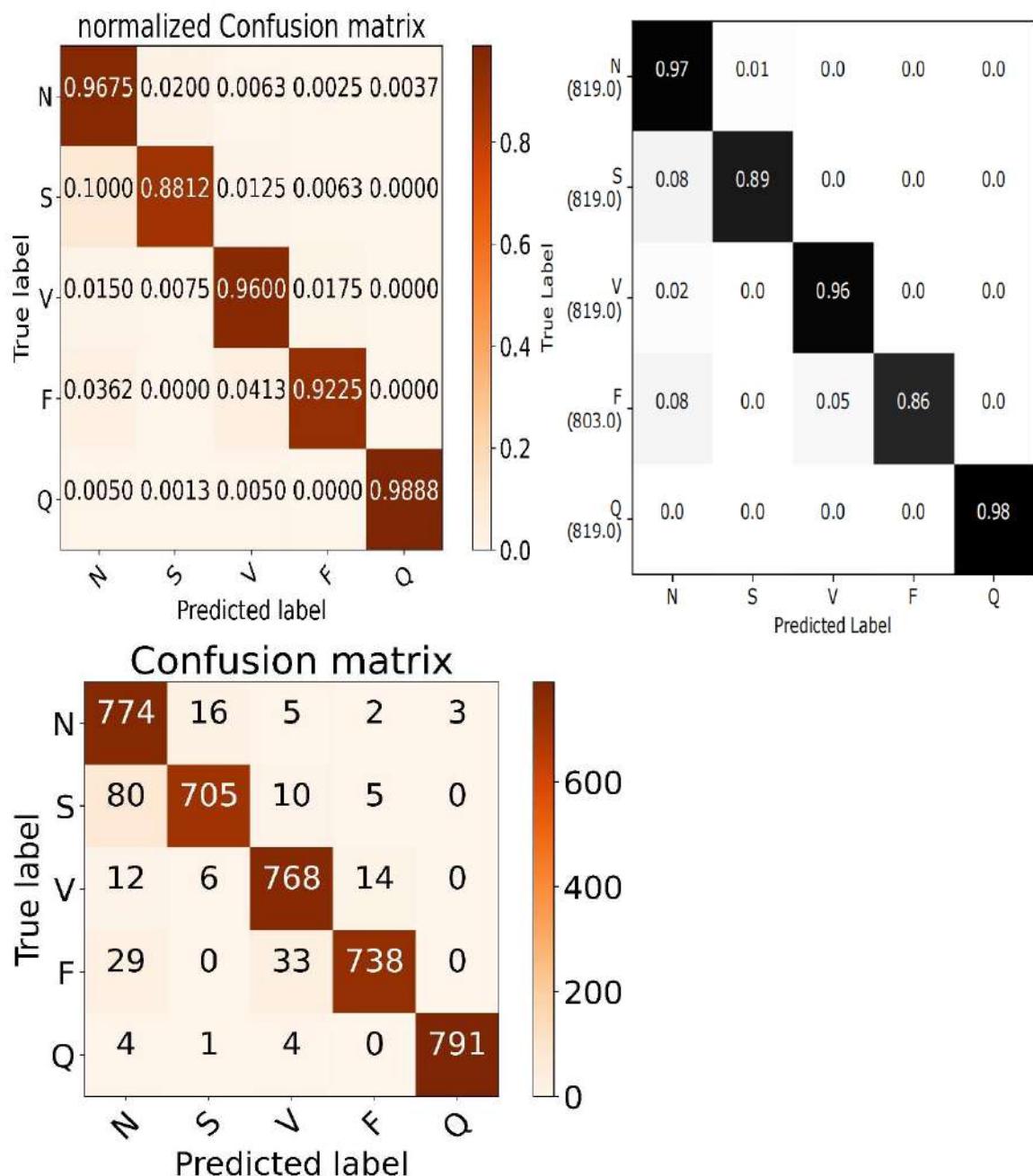


Figure 4-52 version 4 EXP. (3) confusion matrices in orange VS [15] confusion matrix in black

*Table 4-48 version 4 EXP. (3) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.861	0.9675	0.9111	800
1	S	0.9684	0.8812	0.9228	800
2	V	0.9366	0.96	0.9481	800
3	F	0.9723	0.9225	0.9468	800
4	Q	0.9962	0.9888	0.9925	800
accuracy				0.9440	4000
macro avg		0.9469	0.9440	0.9443	4000
weighted avg		0.9469	0.9440	0.9443	4000

#### **EXP. (4)**

Reduction ratio = 16

*Table 4-49 testing results for version 4 EXP. (4)*

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.02%	94.10%	94.02%	94.03%	0.9802	93.92%	92.53%
[15]	93.4%						

*Table 54 version 4 EXP. (4) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8551	0.8551	0.8551	800
1	S	0.8551	0.8551	0.8551	800
2	V	0.8551	0.8551	0.8551	800
3	F	0.8551	0.8551	0.8551	800
4	Q	0.8551	0.8551	0.8551	800
accuracy				0.9403	4000
macro avg		0.9437	0.9403	0.9404	4000
weighted avg		0.9437	0.9403	0.9404	4000

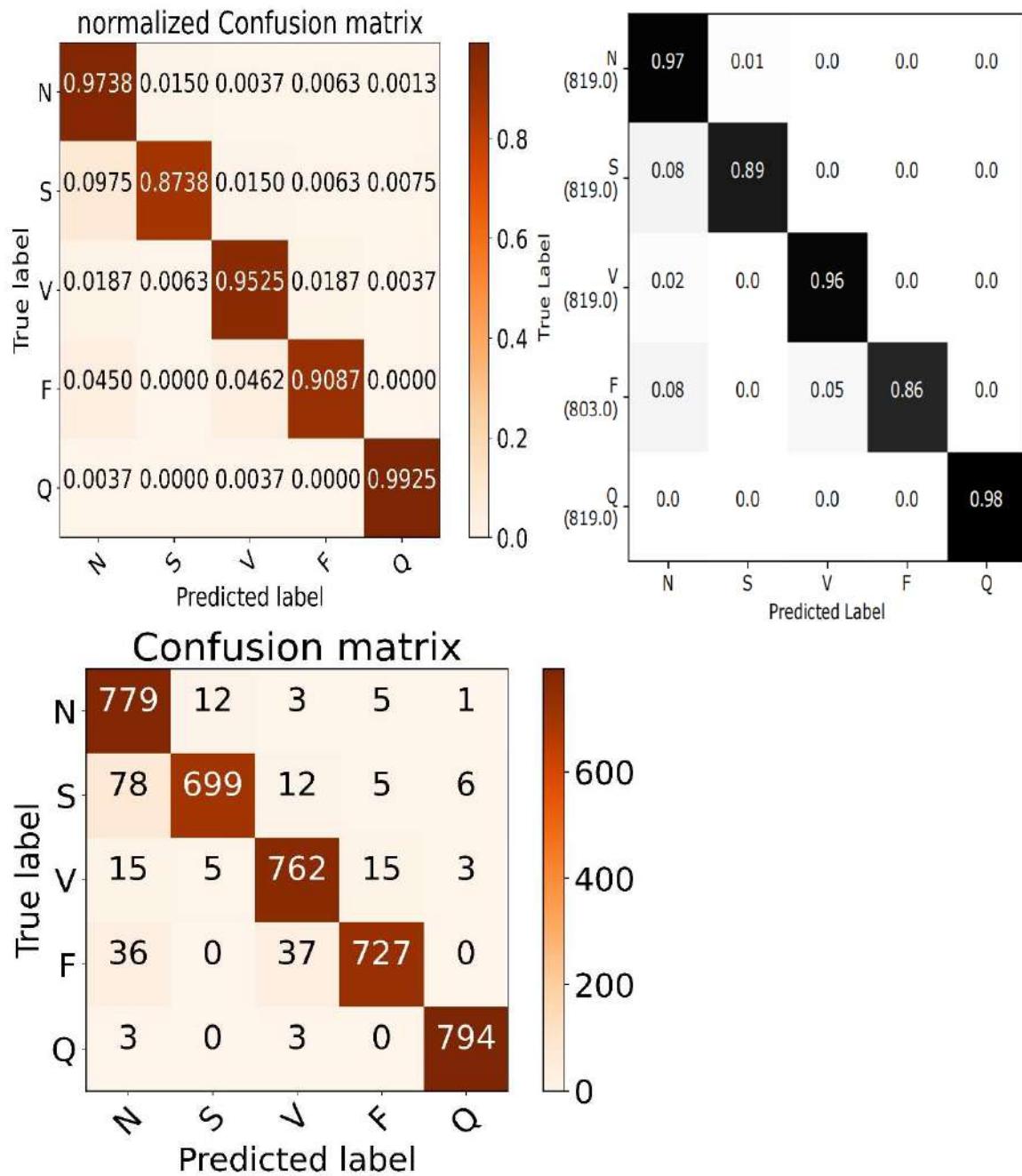
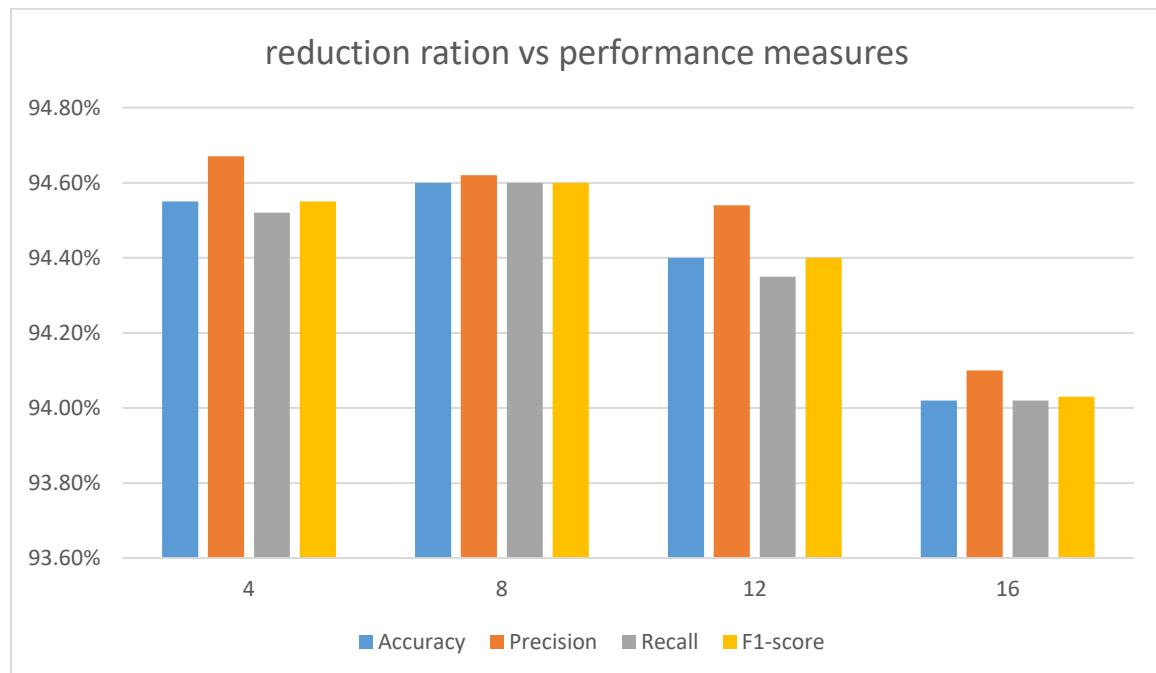


Figure 4-53 version 4 EXP. (4) confusion matrices in orange VS [15] confusion matrix in black

## Tuning the reduction ratio of the AFR layer

Table 4-50 tuning the reduction ratio

Reduction ratio	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
<b>4</b>	94.55%	<b>94.67%</b>	94.52%	94.55%	<b>0.9849</b>	94.48%	93.19%
<b>8</b>	<b>94.60%</b>	94.62%	<b>94.60%</b>	<b>94.60%</b>	0.9839	<b>94.53%</b>	<b>93.25%</b>
<b>12</b>	94.40%	94.54%	94.35%	94.40%	0.9806	94.32%	93.00%
<b>16</b>	94.02%	94.10%	94.02%	94.03%	0.9802	93.92%	92.53%



The number under each 4 bars indicates the reduction ratio.

## Version 5

In this version an adaptive kernel size convolution layer (AKSC) was utilized instead of the ordinary Conv1D, the AKSC layer makes the kernel size a trainable parameter changing during the training process which allows the model gains the flexibility to capture both local and global features effectively.

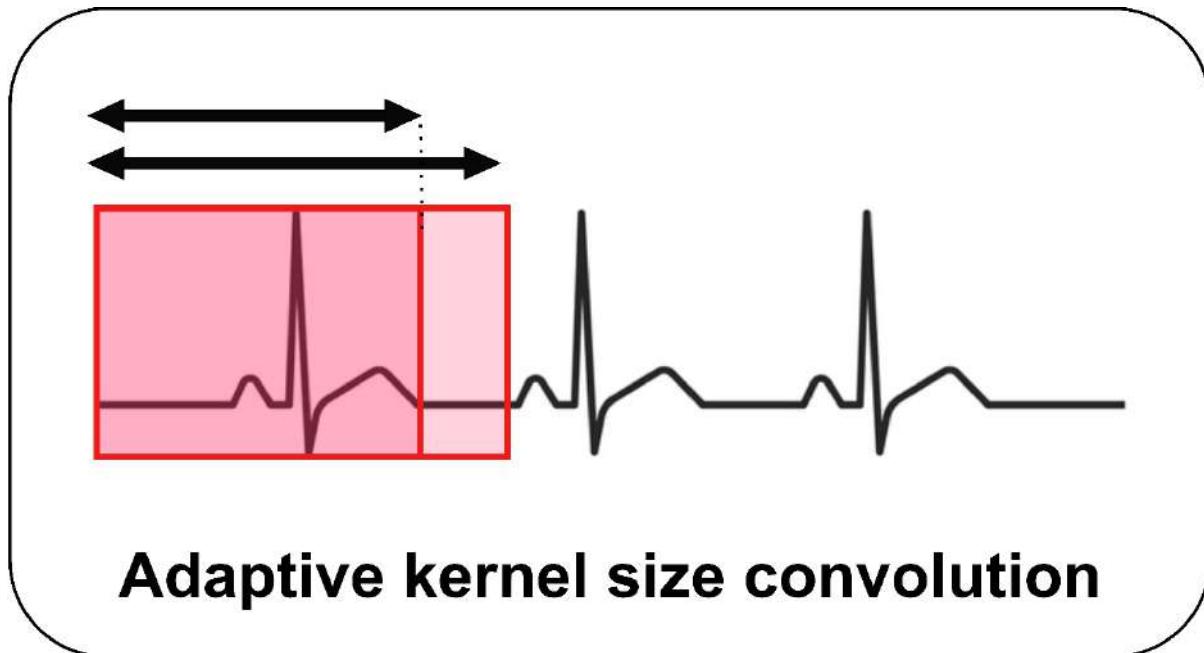


Figure 4-54 Adaptive kernel size convolution layer block diagram

## Data preprocessing

The training set was re-sampled to 20000 and the test set was re-sampled to 800.

## Proposed framework

The same as version 2 but with an AdaptiveConv1D layers

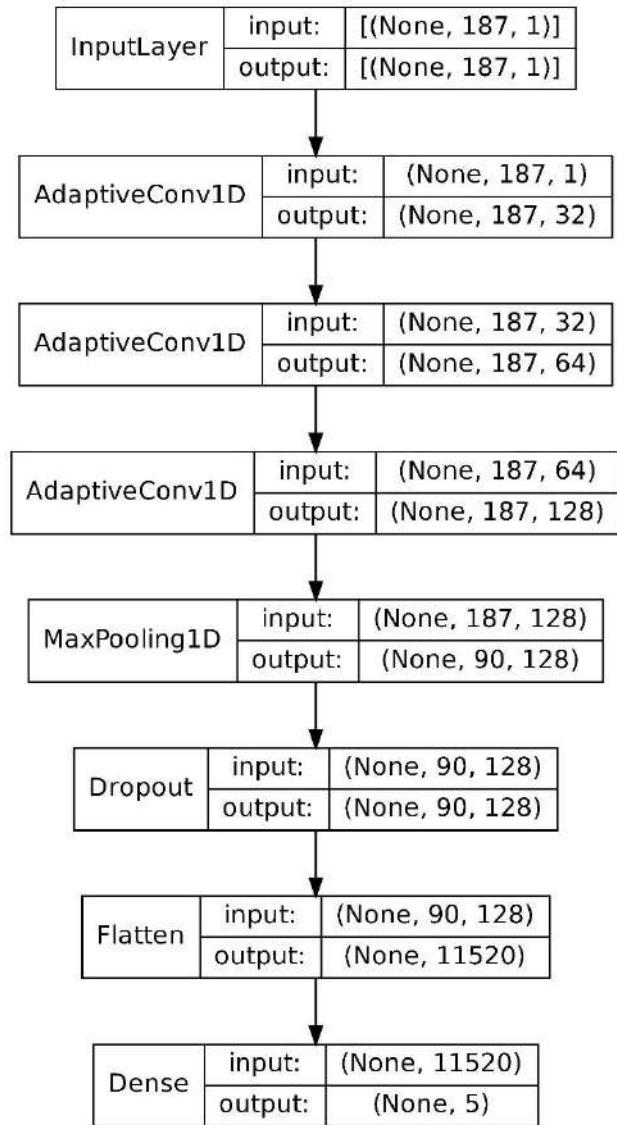


Figure 4-55 version 5 proposed network architecture

Table 4-51 testing results for version 5

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	93.80%	93.80%	93.80%	93.80%	0.9793	93.69%	92.25%
[15]	93.4%						

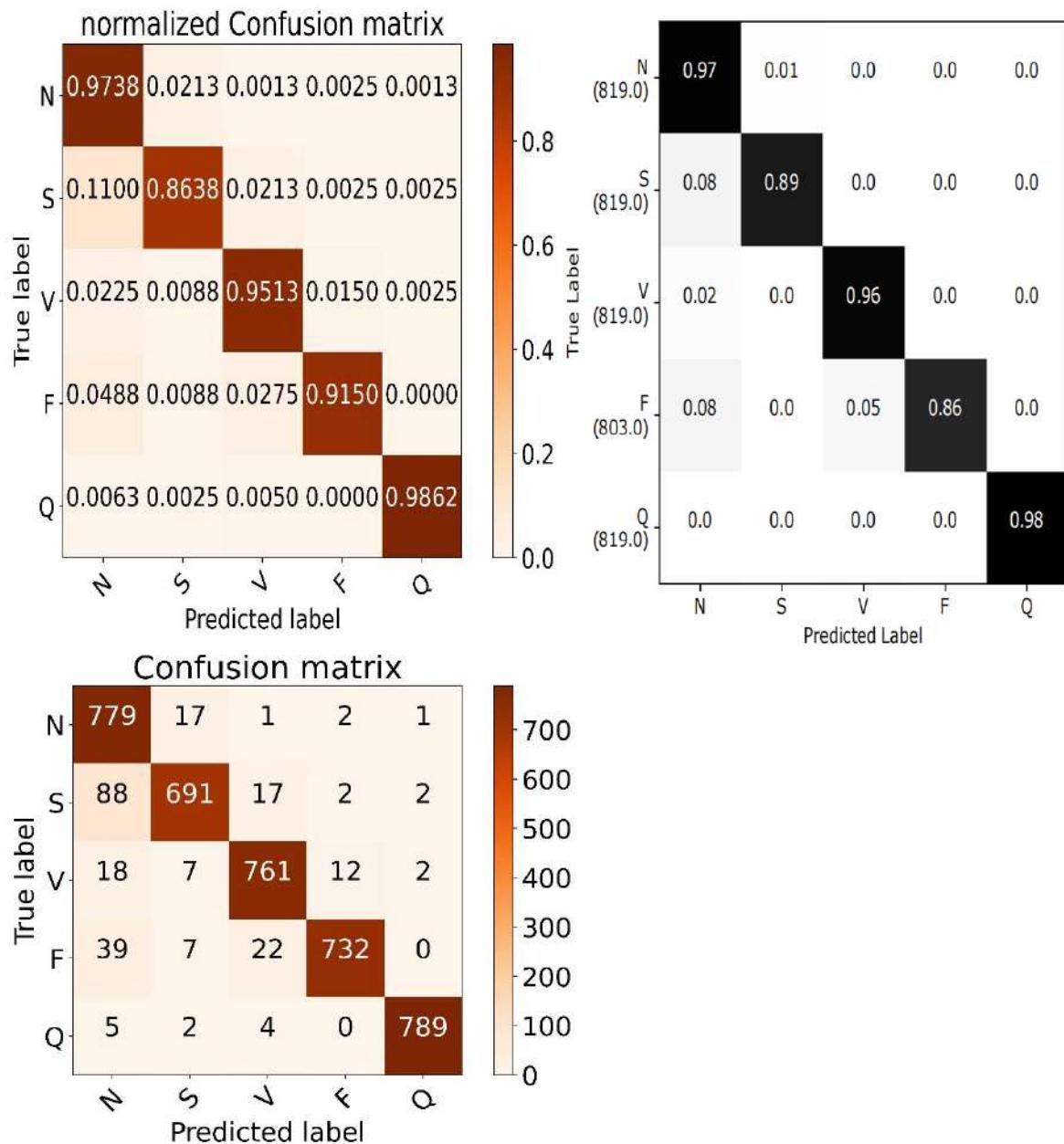


Figure 4-56 version 5 confusion matrices in orange VS [15] confusion matrix in black

Table 4-52 version 5 classification report

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8385	0.9738	0.9011	800
1	S	0.9544	0.8638	0.9068	800
2	V	0.9453	0.9513	0.9483	800
3	F	0.9786	0.915	0.9457	800
4	Q	0.9937	0.9862	0.99	800
accuracy					0.9380
macro avg					0.9384
weighted avg					0.9384

## Version 6

In this version a combination of CNN, AFR (squeezing and excitation), and MHA was utilized.

### Data preprocessing

The training set was re-sampled to 20000 and the test set was re-sampled to 800.

### Proposed framework

The same as version 2 but with an AFR layer followed by MHA layer both tuned from the previous experiments.

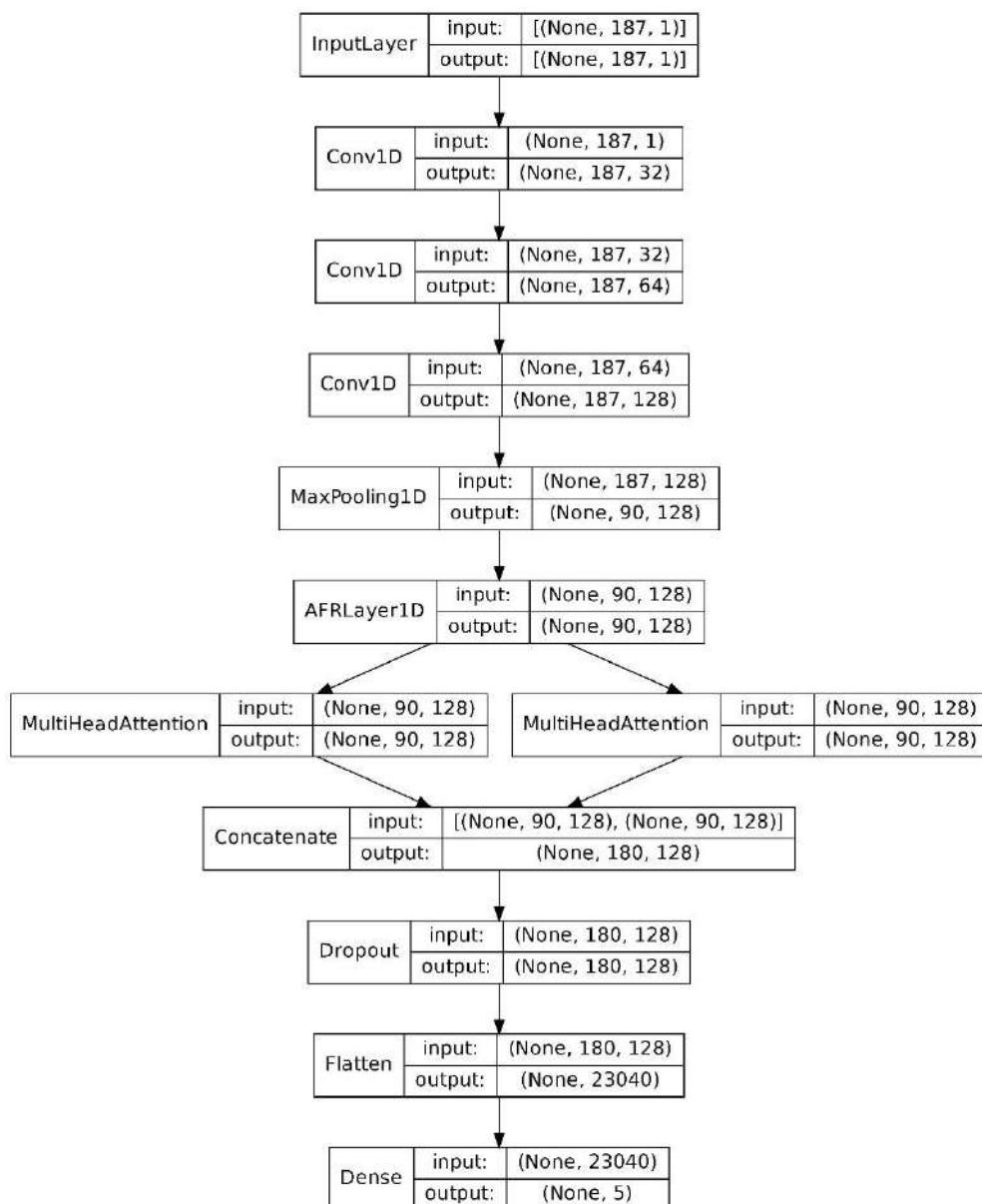


Figure 4-57 version 6 proposed network architecture

Table 4-53 testing results for version 6

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.02%	94.17%	94.02%	94.03%	0.9806	93.95%	92.53%
[15]	93.4%						

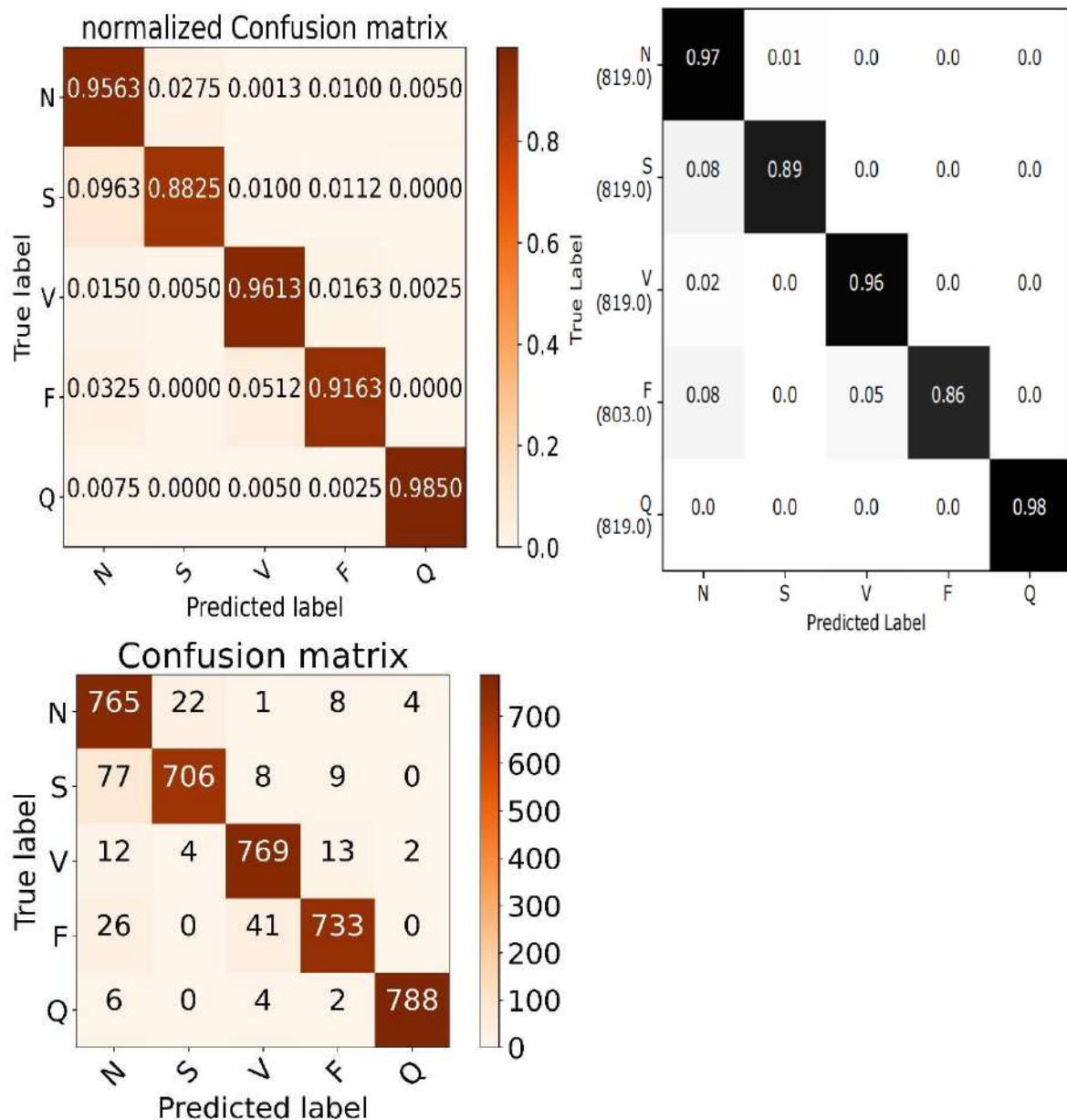


Figure 4-58 version 6 confusion matrices in orange VS [15] confusion matrix in black

*Table 4-54 version 6 classification report*

<b>index</b>	<b>Class name</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
<b>0</b>	N	0.8634	0.9563	0.9075	800
<b>1</b>	S	0.9645	0.8825	0.9217	800
<b>2</b>	V	0.9344	0.9613	0.9476	800
<b>3</b>	F	0.9582	0.9163	0.9367	800
<b>4</b>	Q	0.9924	0.985	0.9887	800
<b>accuracy</b>				0.9403	4000
<b>macro avg</b>		0.9426	0.9403	0.9404	4000
<b>weighted avg</b>		0.9426	0.9403	0.9404	4000

## **Version 7**

In this version a combination of adaptive kernel size convolution ( AKSC), and multi-head attention (MHA) was utilized.

### **Data preprocessing**

The training set was re-sampled to 20000 and the test set was re-sampled to 800.

### **Proposed framework**

The same as version 2 but with an AKSC layers followed by MHA layer Stuned from the previous experiments.

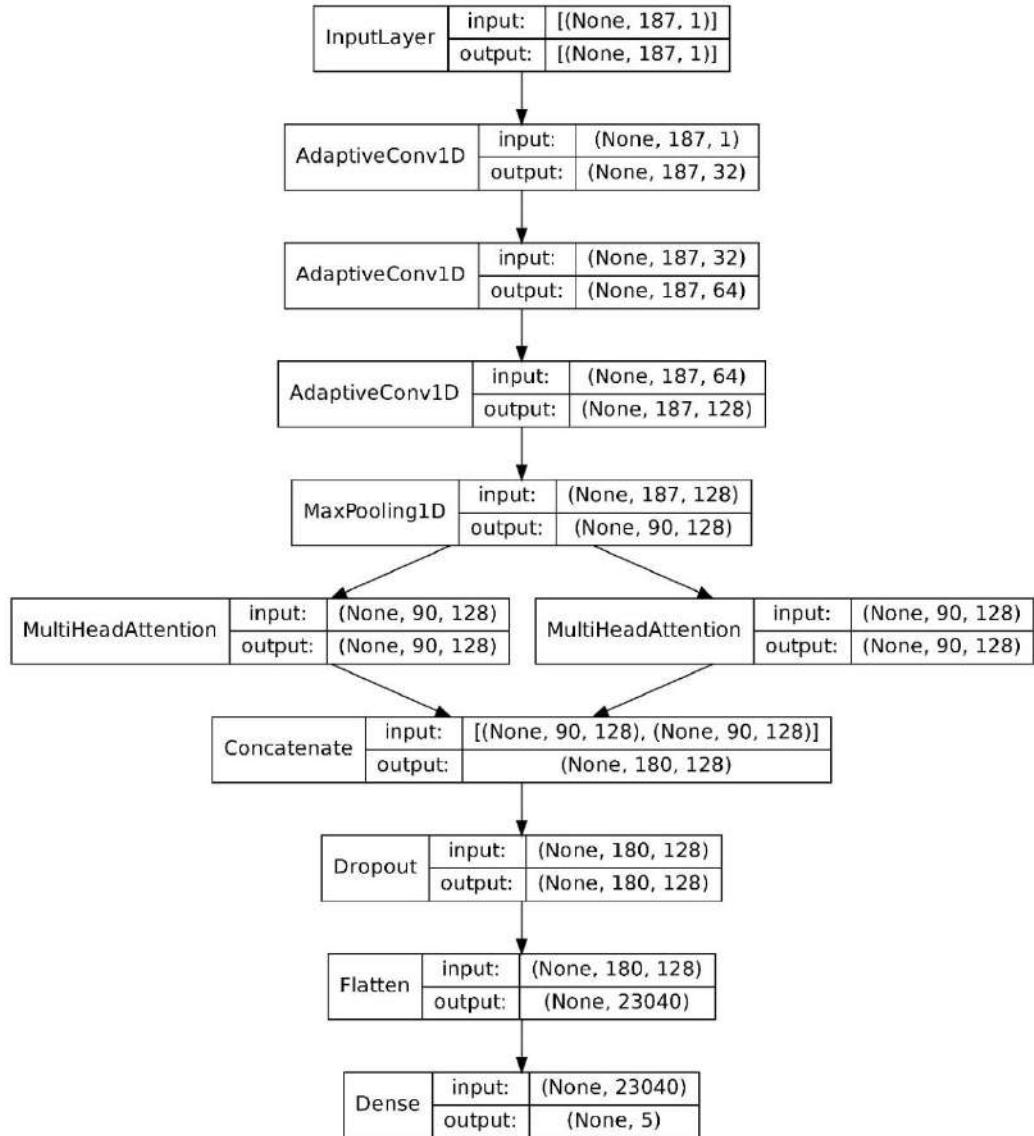


Figure 4-59 version 7 proposed network architecture

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	93.88%	93.88%	93.88%	93.88%	0.9717	93.76%	92.34%
[15]	93.4%						

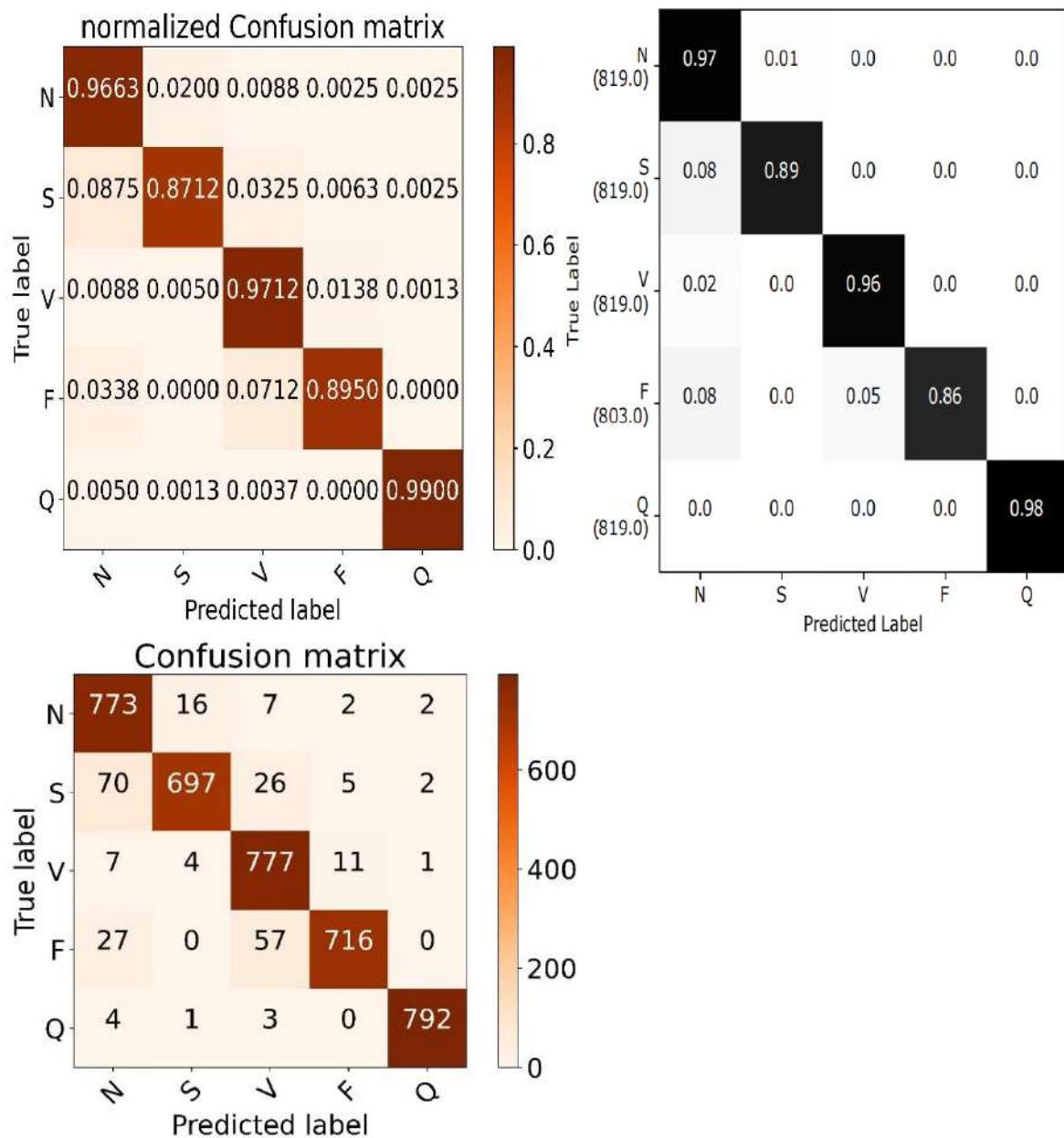


Figure 4-60 version 7 confusion matrices in orange VS [15] confusion matrix in black

Table 4-55 version 7 classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8774	0.9663	0.9197	800
<b>1</b>	S	0.9708	0.8712	0.9183	800
<b>2</b>	V	0.8931	0.9712	0.9305	800
<b>3</b>	F	0.9755	0.895	0.9335	800
<b>4</b>	Q	0.9937	0.99	0.9919	800
<hr/>					
<b>accuracy</b>				0.9387	4000
<b>macro avg</b>		0.9421	0.9387	0.9388	4000
<b>weighted avg</b>		0.9421	0.9387	0.9388	4000

## Experiments summary with over-sampling

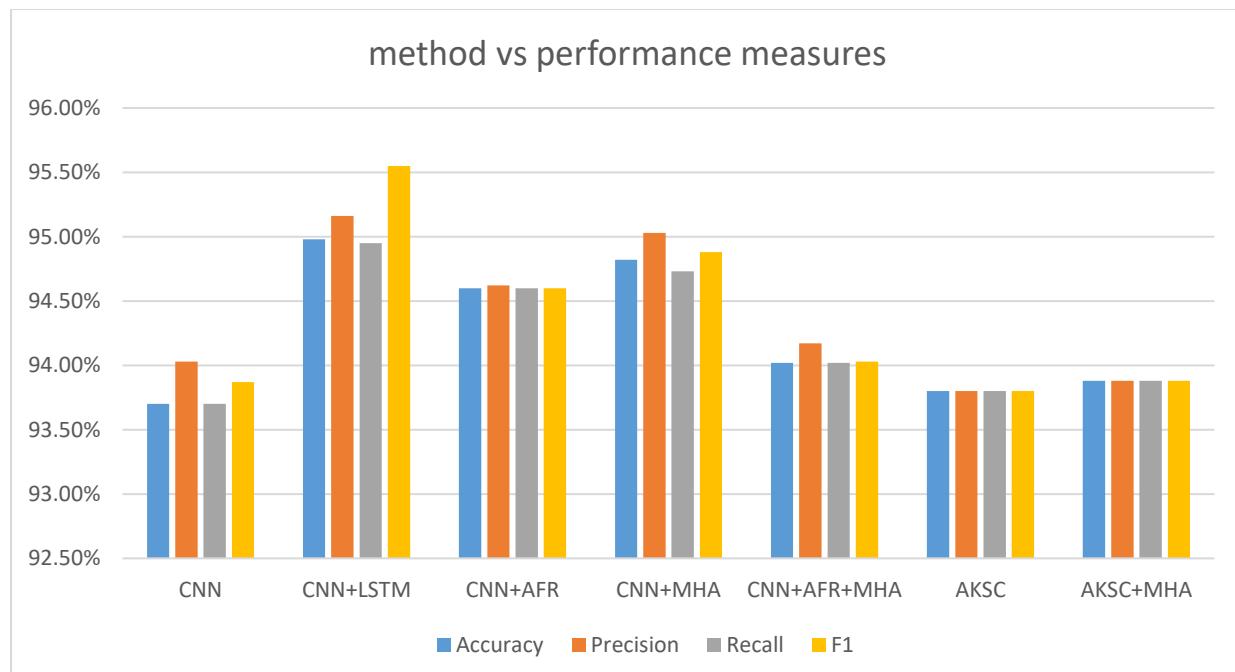
Table 4-56 results of each method with over sampling

Method	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
CNN	93.70%	94.03%	93.70%	93.87%	0.9762	93.25%	92.05%
CNN+LSTM	94.98%	95.16%	94.95%	95.55%	0.9858	94.76%	93.65%
CNN+AFR	94.60%	94.62%	94.60%	94.60%	0.9839	94.53%	93.25%
CNN+MHA	94.82%	95.03%	94.73%	94.88%	0.9855	94.77%	93.53%
CNN+AFR+MHA	94.02%	94.17%	94.02%	94.03%	0.9806	93.95%	92.53%
AKSC	93.80%	93.80%	93.80%	93.80%	0.9793	93.69%	92.25%
AKSC+MHA	93.88%	93.88%	93.88%	93.88%	0.9717	93.76%	92.34%

Table 4-57 Abbreviations

<b>LSTM</b>	Long-short time memory
<b>AFR</b>	Adaptive Feature Recalibration (squeeze and excitation)
<b>MHA</b>	Multi-Head Attention
<b>AKSC</b>	Adaptive kernel Size Convolution

As show in the table, best results was achieved using the CNN+LSTM combination with 94.98% accuracy, the CNN+MHA combination comes in the second place with 94.82% and the CNN alone had the least results with 93.70% accuracy.



## Version 8

In this version different types of loss functions will be utilized to deal with the imbalance problem without any additional computation.

### Data preprocessing

Training set was used as it is.

### Proposed framework

The same as version 2.

### EXP. (1)

In this experiment the weighted cross entropy (WCE) loss function was utilized which assigns a weight to each class based on the inverse class frequency so that smaller classes are given a higher weight.

*Table 4-58 testing results for version 8 EXP. (1)*

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	93.35%	94.33%	93.23%	93.35%	0.9885	93.30%	91.69%
[15]	93.4%						

*Table 4-59 version 8 EXP. (1) classification report*

index	Class name	Precision	Recall	F1-score	Support
0	N	0.8439	0.9325	0.886	800
1	S	0.9265	0.8975	0.9117	800
2	V	0.9521	0.945	0.9486	800
3	F	0.9592	0.91	0.9339	800
4	Q	0.9975	0.9825	0.9899	800
accuracy				0.9335	4000
macro avg		0.9358	0.9335	0.9340	4000
weighted avg		0.9358	0.9335	0.9340	4000

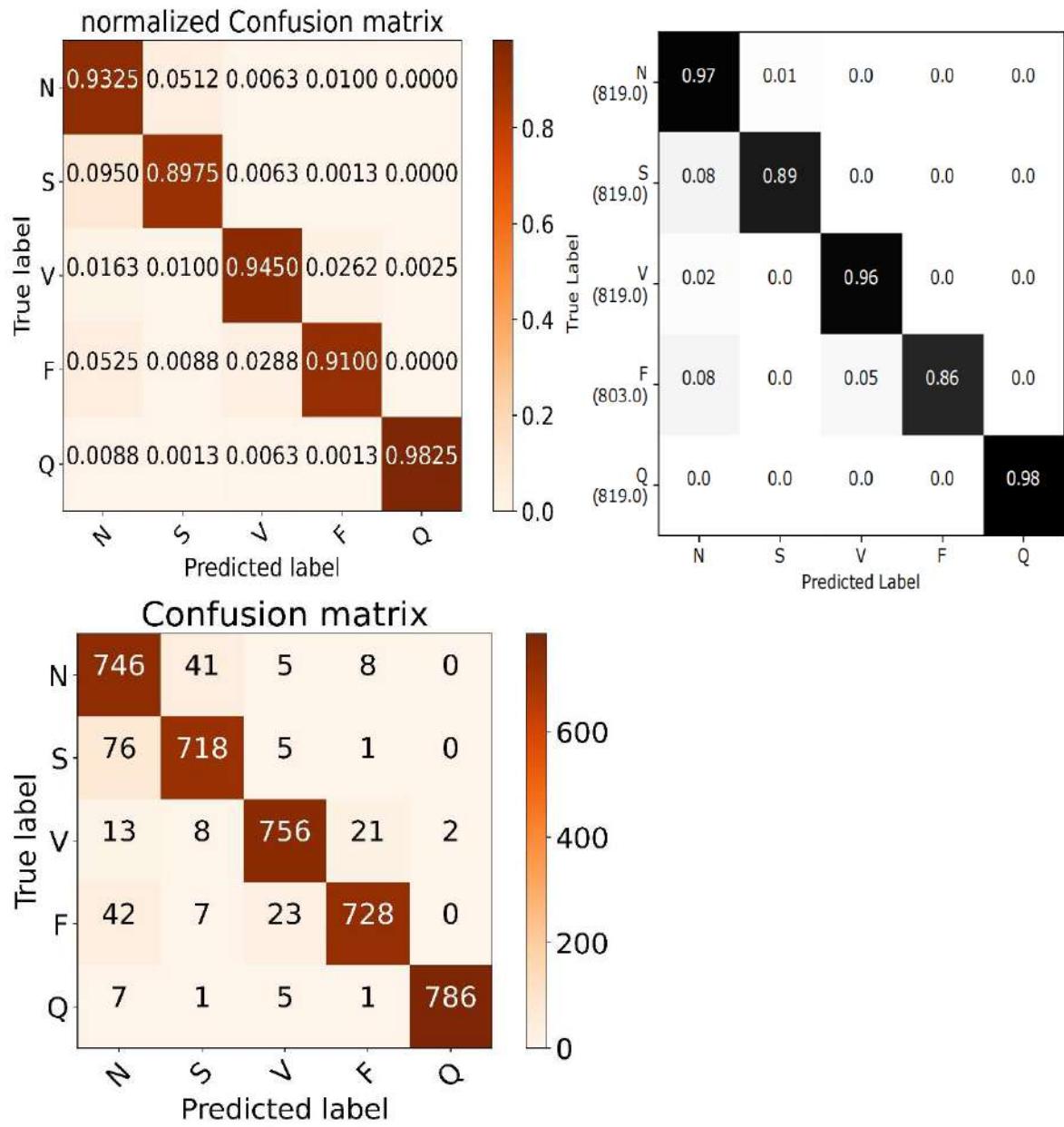


Figure 4-61 version 8 EXP. (1) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (2)

In this experiment focal loss (FL) function was utilized, the focal loss function is designed to address the problem of class imbalance by giving more weight to hard-to-classify examples, while down weighting easy-to-classify examples. It does this by introducing a modulating factor (the focal factor) that decreases the contribution of well-classified examples to the loss function, it has two parameters, **the gamma** parameter used for is a scalar parameter that controls the degree of focus on hard-to-classify examples, **the alpha** parameter which is a vector of class weights that controls the balance between classes, it is often set to the inverse class frequency, so that less frequent classes are given more weight.

Table 4-60 testing results for version 8 EXP. (2)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	86.48%	91.01%	79.75%	86.48%	0.9782	85.36%	83.09%
[15]	93.4%						

Table 4-61 version 8 EXP. (2) classification report

index	Class name	Precision	Recall	F1-score	Support
0	N	0.6519	0.995	0.7877	800
1	S	0.9929	0.7013	0.822	800
2	V	0.8773	0.9475	0.9111	800
3	F	0.9947	0.6987	0.8209	800
4	Q	0.9962	0.9812	0.9887	800
<b>accuracy</b>				0.8648	4000
<b>macro avg</b>		0.9026	0.8647	0.8661	4000
<b>weighted avg</b>		0.9026	0.8648	0.8661	4000

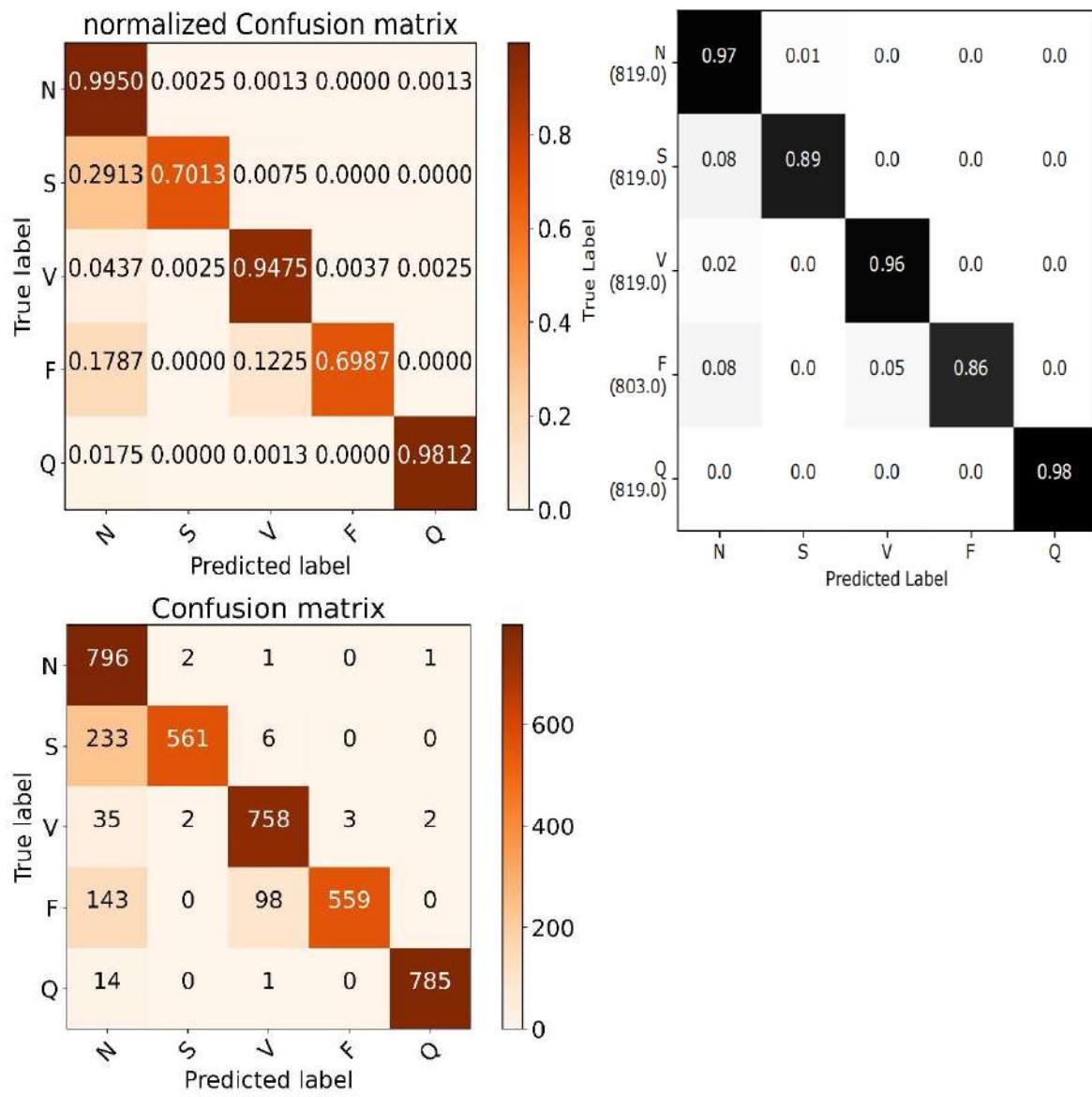
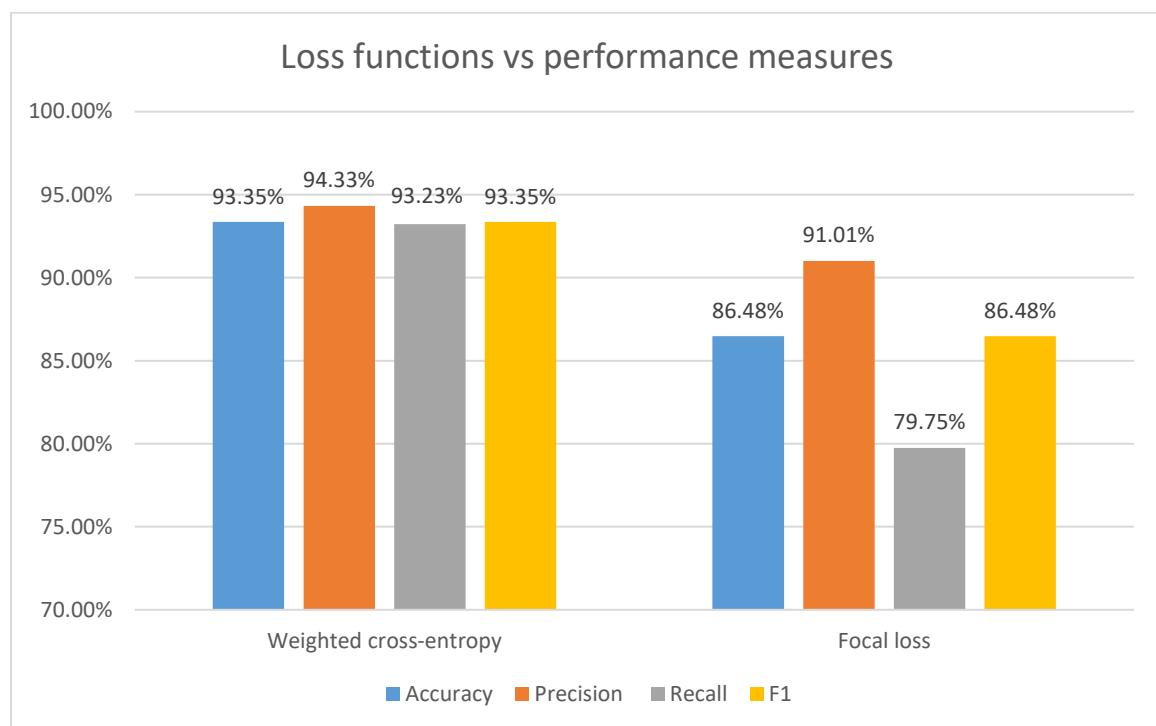


Figure 4-62 version 8 EXP. (2) confusion matrices in orange VS [15] confusion matrix in black

## Weighted cross entropy vs focal loss

Table 4-62 testing results for different loss functions

Loss function	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
<b>Weighted cross-entropy</b>	93.35%	94.33%	93.23%	93.35%	0.9885	93.30%	91.69%
<b>Focal loss</b>	86.48%	91.01%	79.75%	86.48%	0.9782	85.36%	83.09%



## Version 9

In this version the top performing methods will be re-experimented with weighted cross entropy loss function to see their performance on the dataset without any additional computation.

### EXP (1)

The CNN+LSTM combination with the weighted cross entropy.

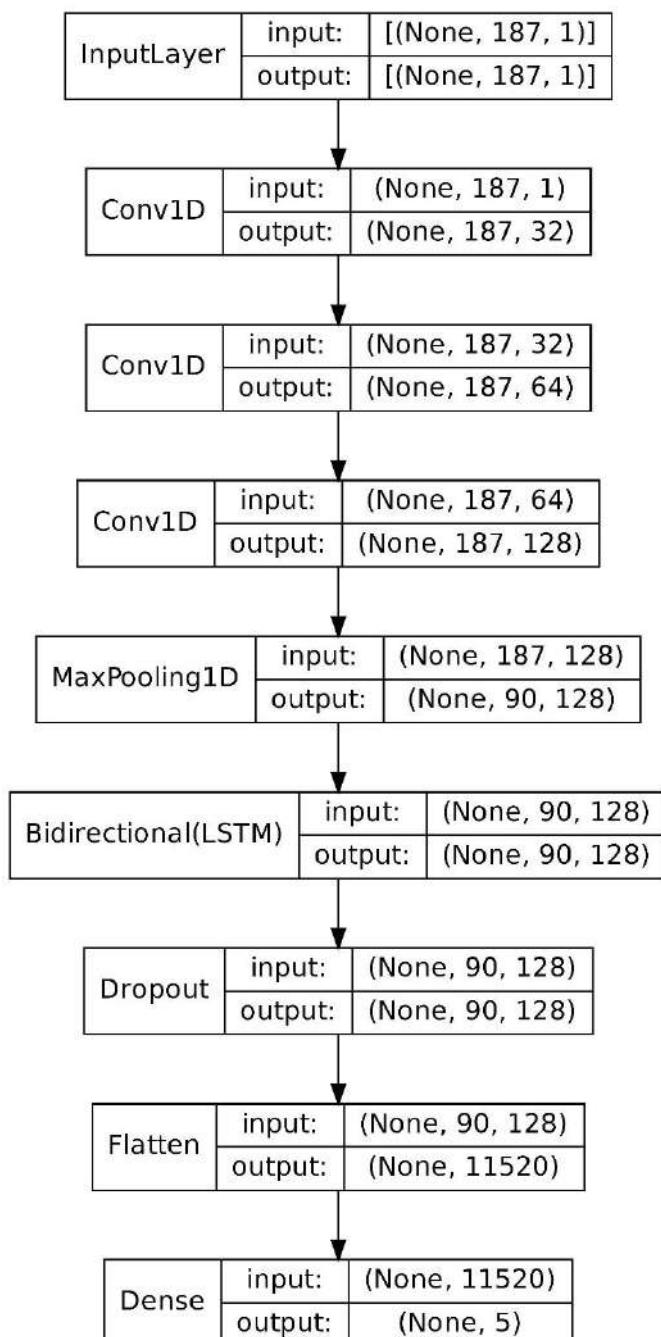


Figure 4-63 version 9 EXP (1) proposed network architecture

Table 4-63 testing results for version 9 EXP. (1)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.77%	95.08%	94.75%	94.77%	0.9876	94.72%	93.47%
[15]	93.4%						

Table 4-64 version 9 EXP (1) classification report

index	Class name	Precision	Recall	F1-score	Support
0	N	0.88	0.9625	0.9194	800
1	S	0.9595	0.8875	0.9221	800
2	V	0.9368	0.9637	0.9501	800
3	F	0.9766	0.94	0.958	800
4	Q	0.9949	0.985	0.9899	800
<b>accuracy</b>				0.9477	4000
<b>macro avg</b>		0.9496	0.9477	0.9479	4000
<b>weighted avg</b>		0.9496	0.9477	0.9479	4000

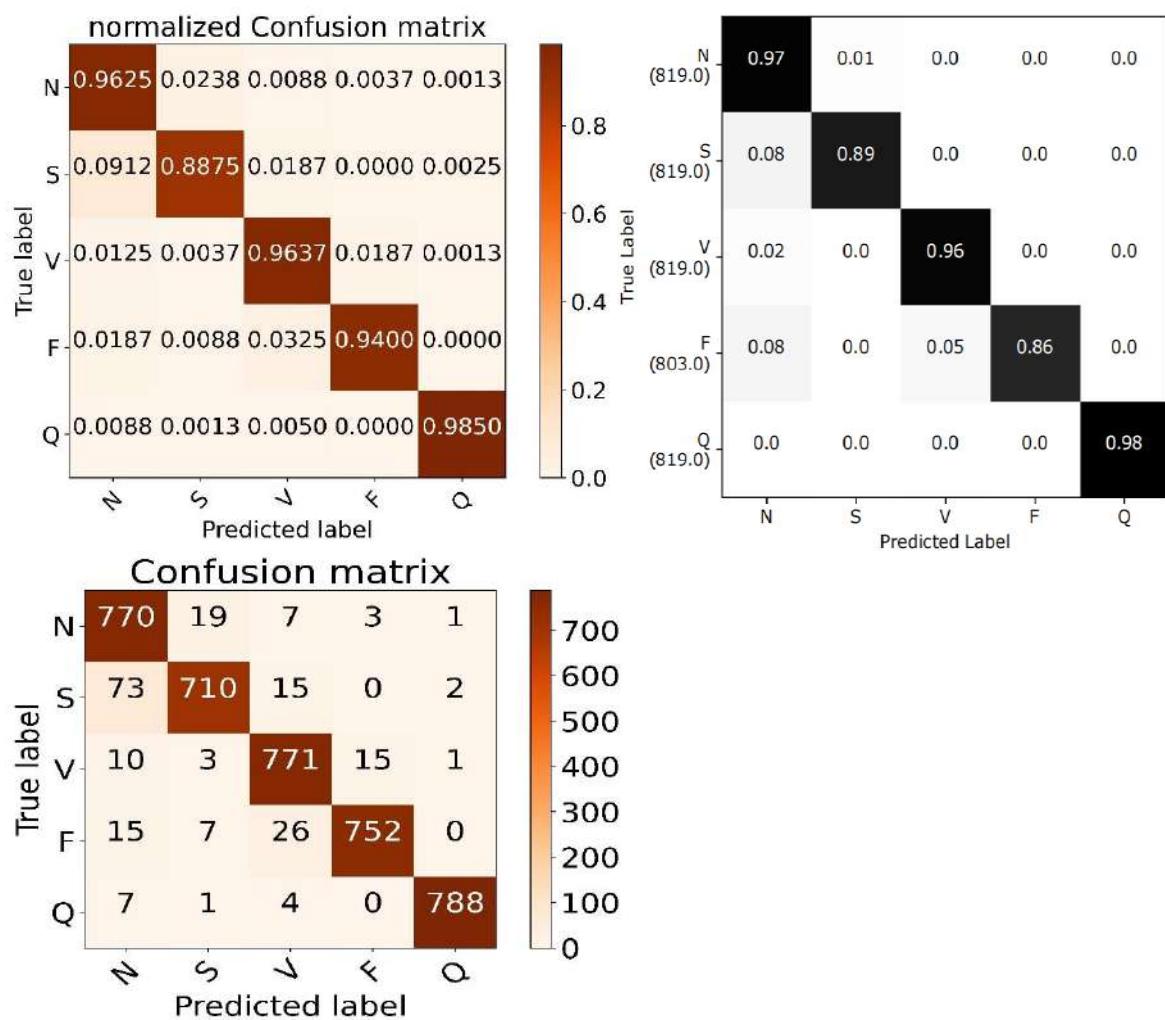


Figure 4-64 version 9 EXP. (1) confusion matrices in orange VS [15] confusion matrix in black

## EXP (2)

The CNN+MHA combination with weighted cross entropy.

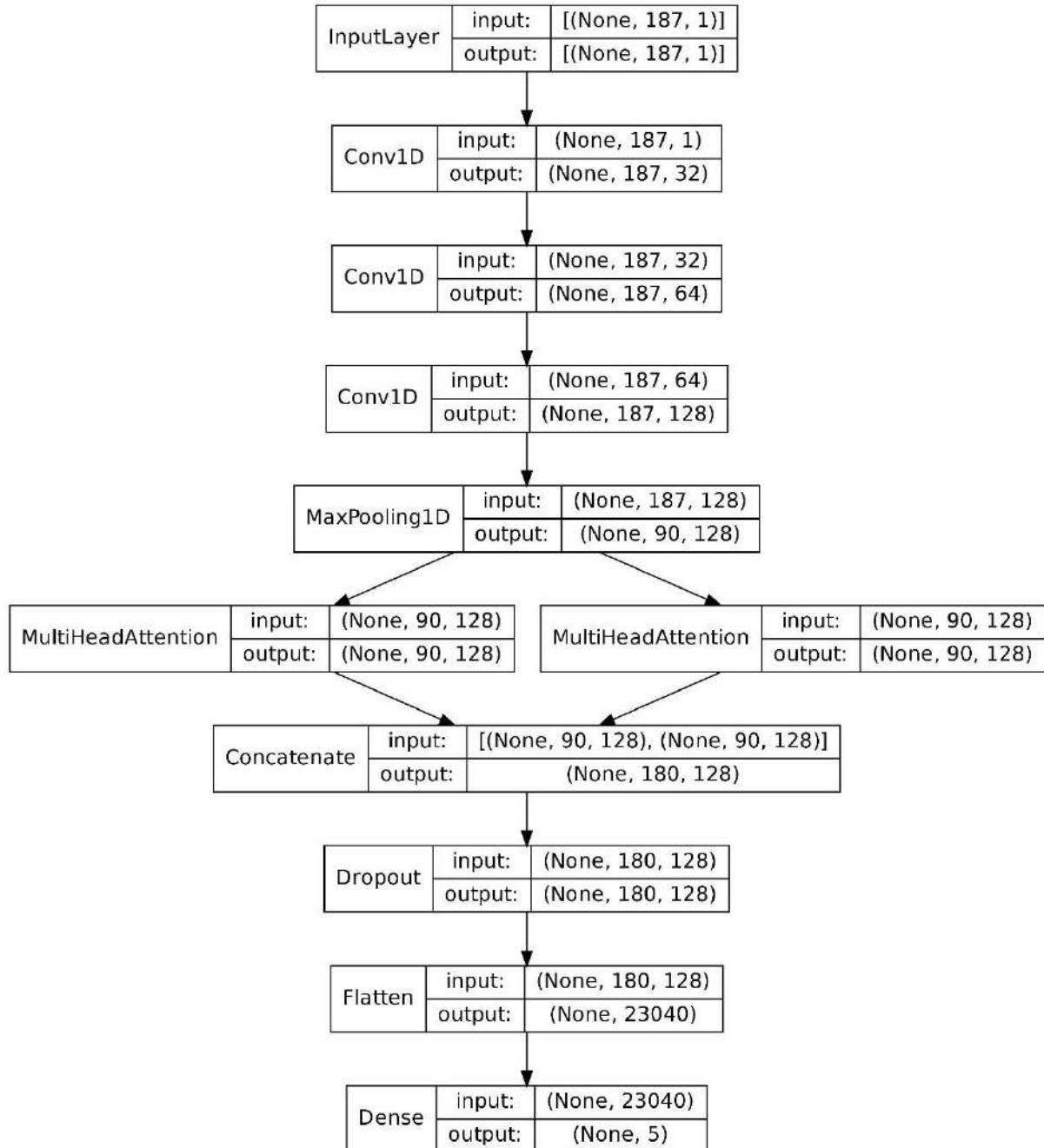


Figure 4-65 version 9 EXP (2) proposed network architecture

Table 4-65 testing results for version 9 EXP. (2)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	92.65%	92.65%	92.65%	92.65%	0.9816	92.59%	90.81%
[15]	93.4%						

Table 4-66 version 9 EXP (2) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8853	0.8975	0.8914	800
<b>1</b>	S	0.9009	0.8975	0.8992	800
<b>2</b>	V	0.8911	0.9613	0.9248	800
<b>3</b>	F	0.9664	0.9	0.932	800
<b>4</b>	Q	0.9962	0.9762	0.9861	800
<hr/>					
<b>accuracy</b>				0.9265	4000
<b>macro avg</b>		0.9280	0.9265	0.9267	4000
<b>weighted avg</b>		0.9280	0.9265	0.9267	4000

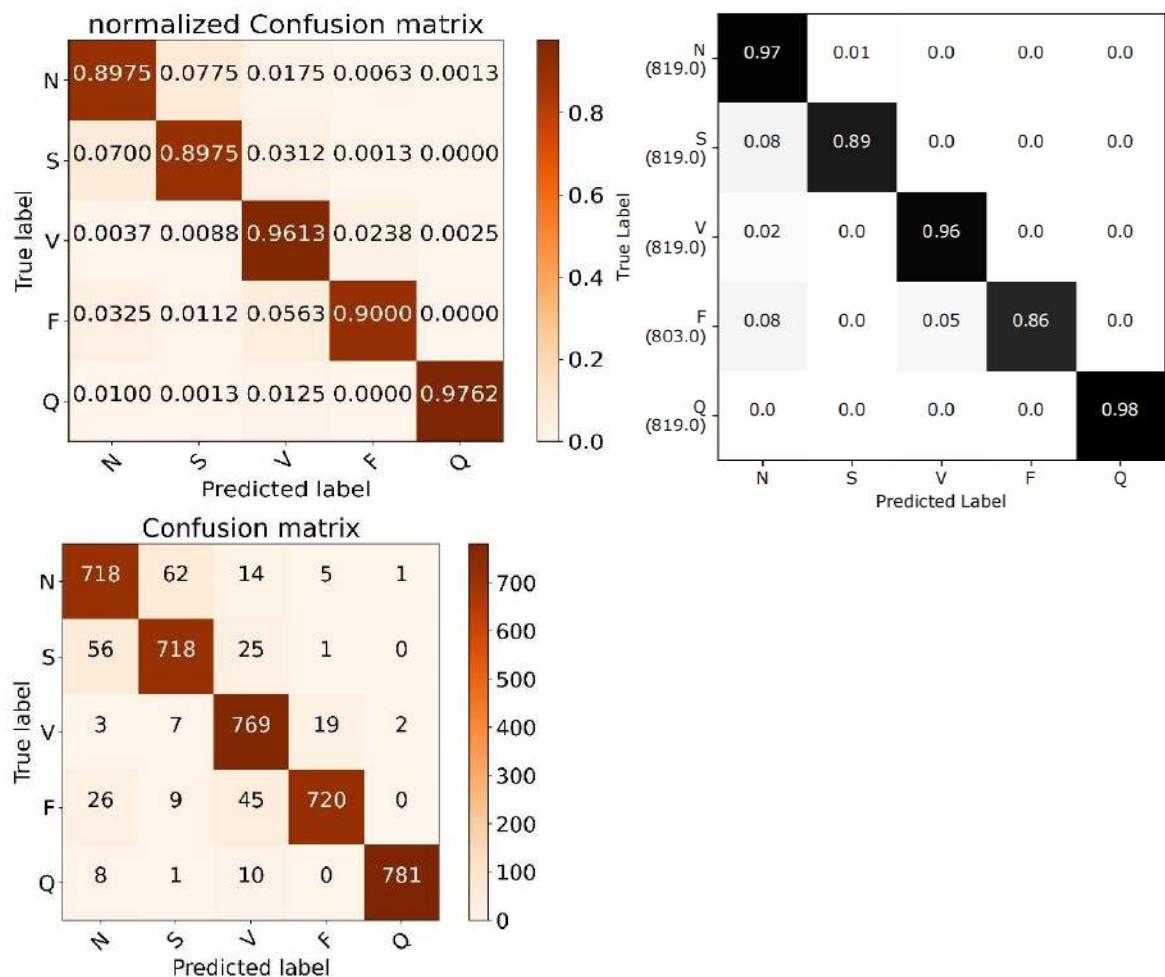


Figure 4-66 version 9 EXP. (2) confusion matrices in orange VS [15] confusion matrix in black

### EXP. (3)

The CNN+AFR combination with weighted cross entropy

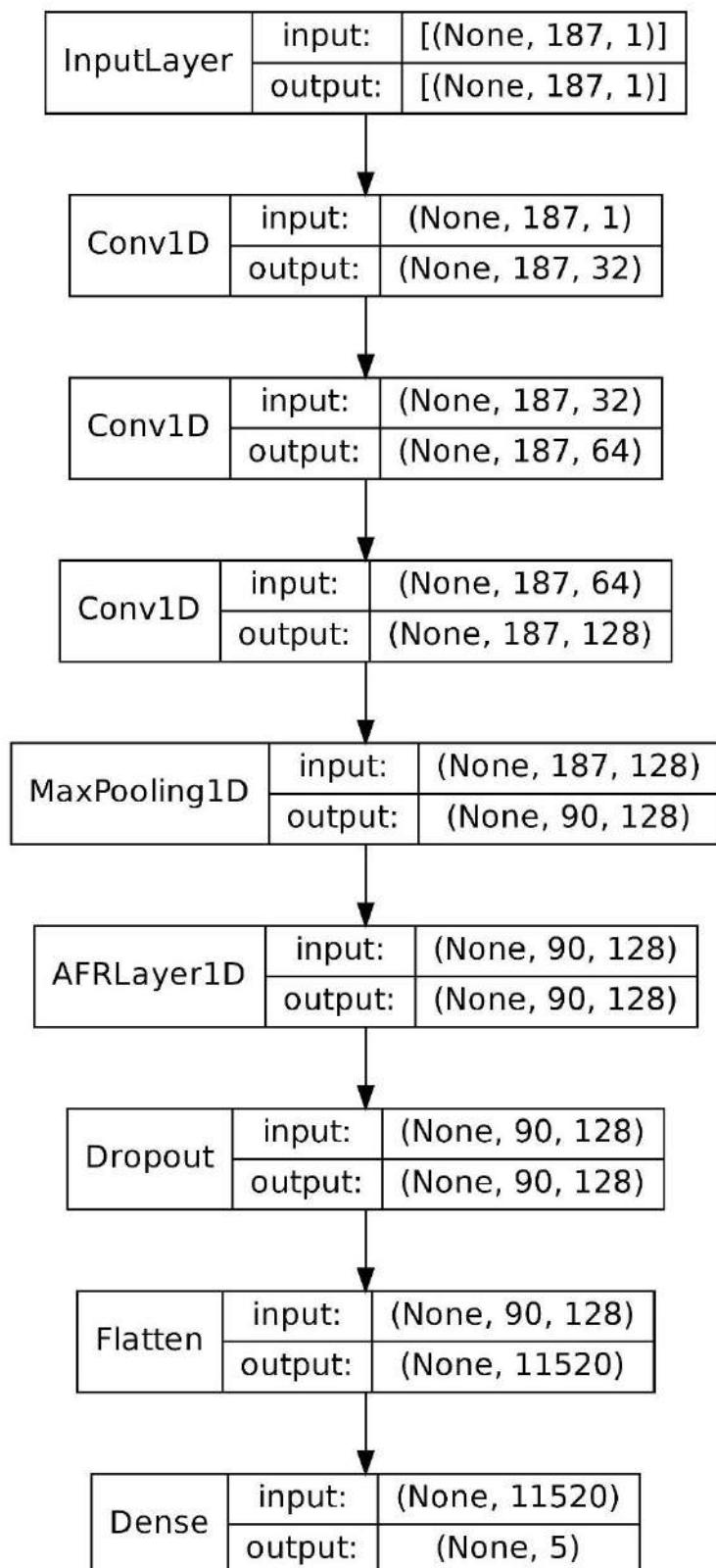


Figure 4-67 version 9 EXP (3) proposed network architecture

Table 4-67 testing results for version 9 EXP. (3)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	93.98%	94.22%	93.67%	93.97%	0.9885	93.93%	92.47%
[15]	93.4%						

Table 4-68 version 9 EXP (3) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8995	0.9287	0.9139	800
<b>1</b>	S	0.922	0.9012	0.9115	800
<b>2</b>	V	0.9445	0.9575	0.951	800
<b>3</b>	F	0.9462	0.9237	0.9349	800
<b>4</b>	Q	0.9875	0.9875	0.9875	800
<hr/>					
<b>accuracy</b>				0.9397	4000
<b>macro avg</b>		0.9399	0.9398	0.9397	4000
<b>weighted avg</b>		0.9399	0.9397	0.9397	4000

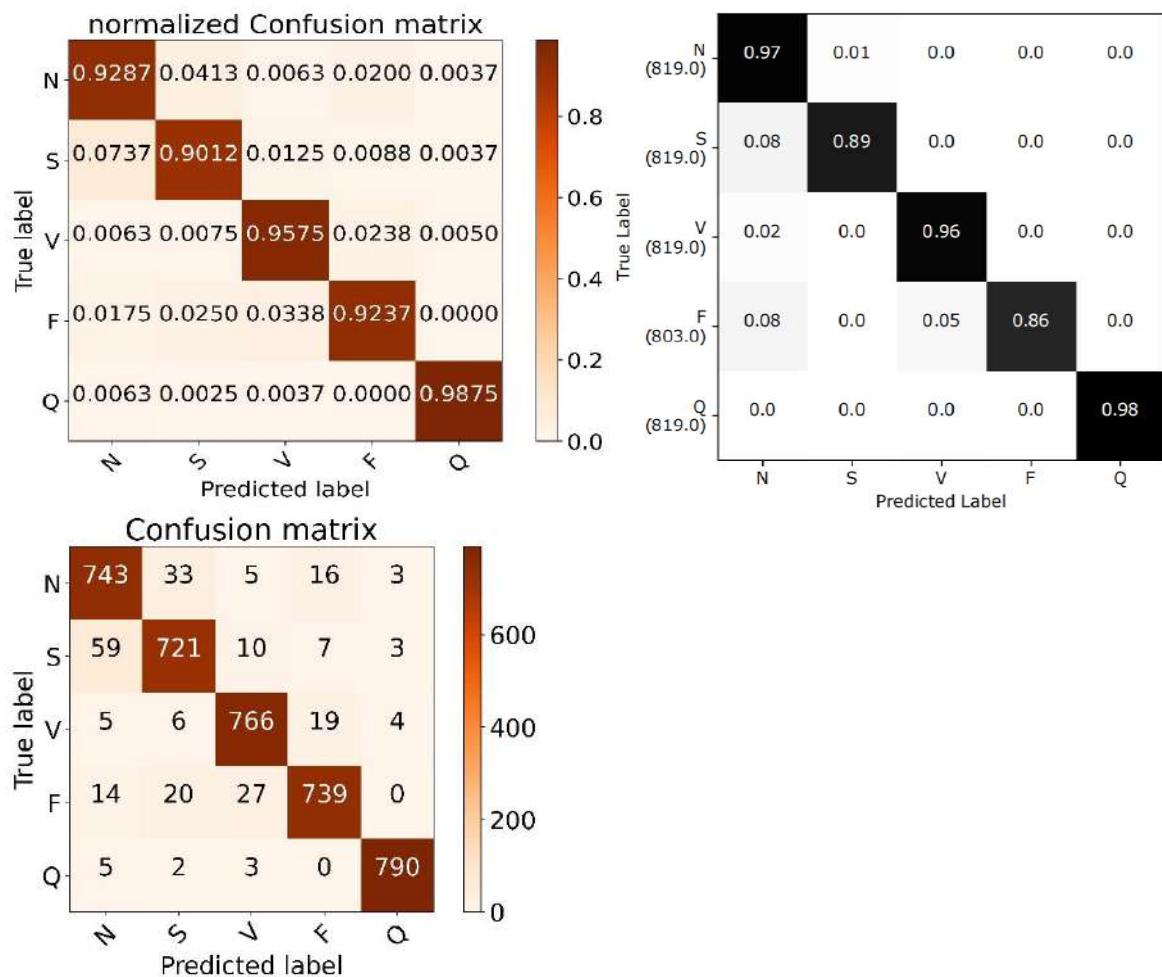


Figure 4-68 version 9 EXP. (3) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (4)

The CNN+AFR+MHA with the weighted cross entropy

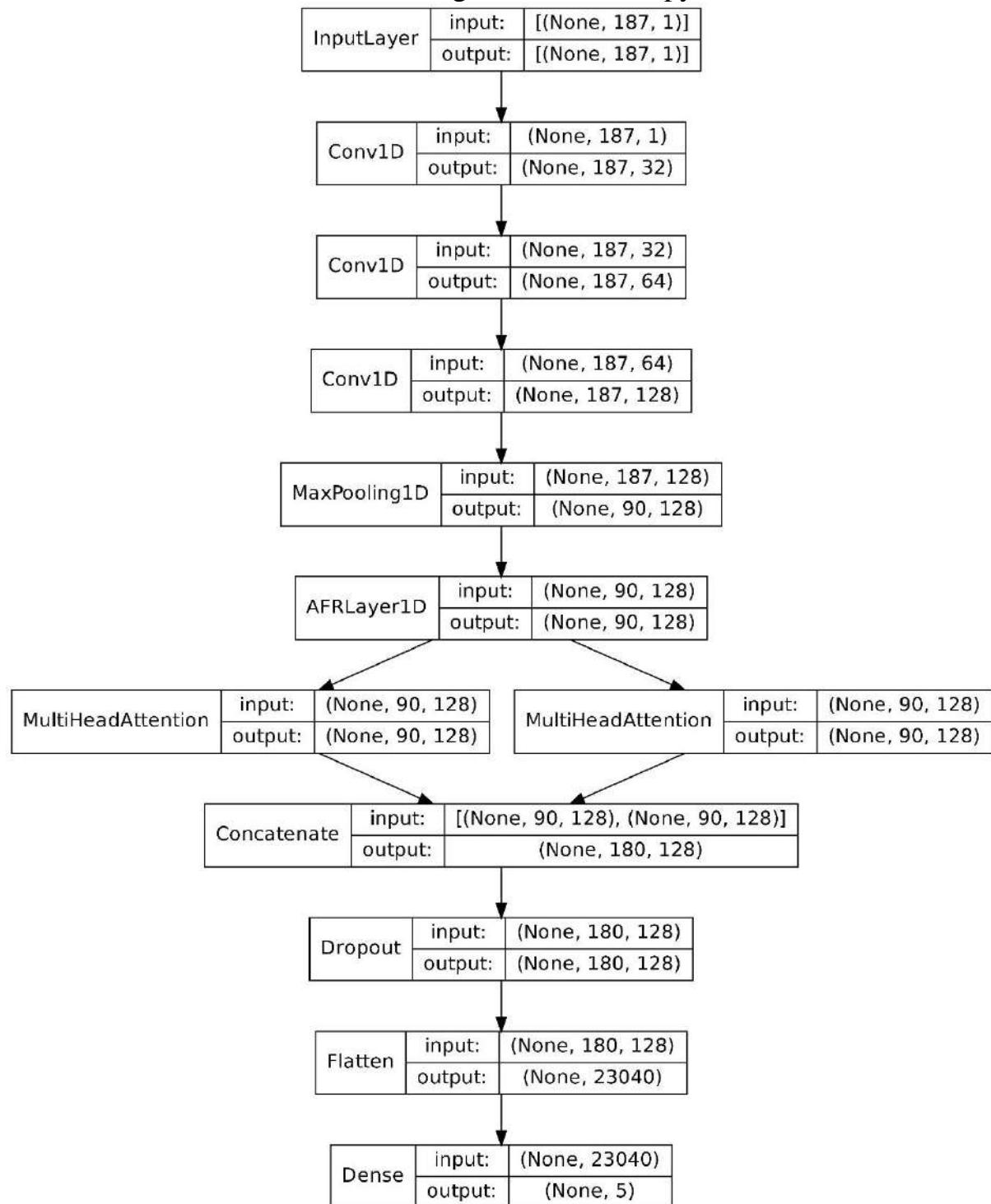


Figure 4-69 version 9 EXP (4) proposed network architecture

Table 4-69 testing results for version 9 EXP. (4)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	93.45%	93.67%	93.27%	93.45%	0.9852	93.34%	91.81%
[15]	93.4%						

Table 4-70 version 9 EXP (4) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8473	0.95	0.8957	800
<b>1</b>	S	0.9645	0.85	0.9037	800
<b>2</b>	V	0.9457	0.9363	0.941	800
<b>3</b>	F	0.9362	0.9537	0.9449	800
<b>4</b>	Q	0.9937	0.9825	0.9881	800
<hr/>					
<b>accuracy</b>				0.9345	4000
<b>macro avg</b>		0.9375	0.9345	0.9347	4000
<b>weighted avg</b>		0.9375	0.9345	0.9347	4000

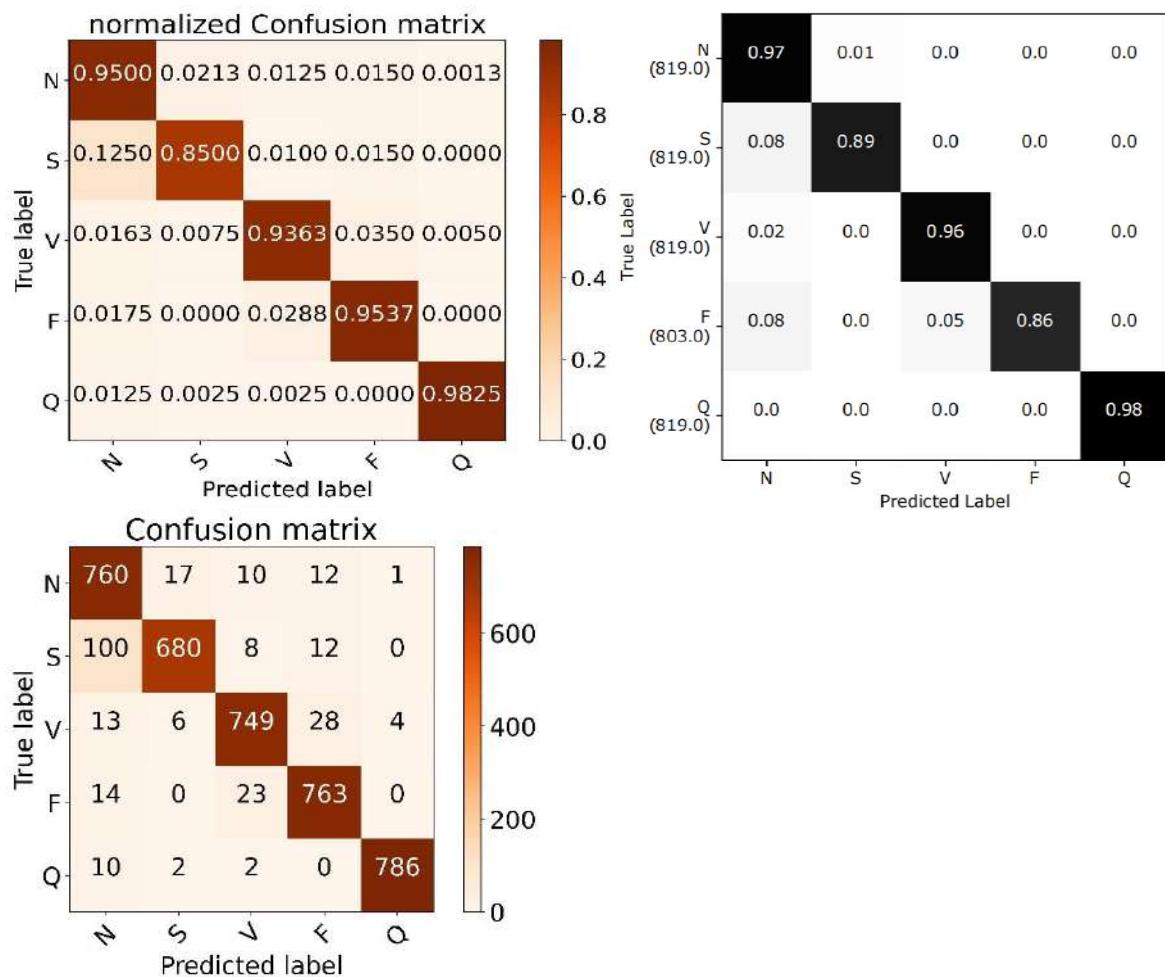


Figure 4-70 version 9 EXP. (4) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (5)

AKSC with weighted cross entropy.

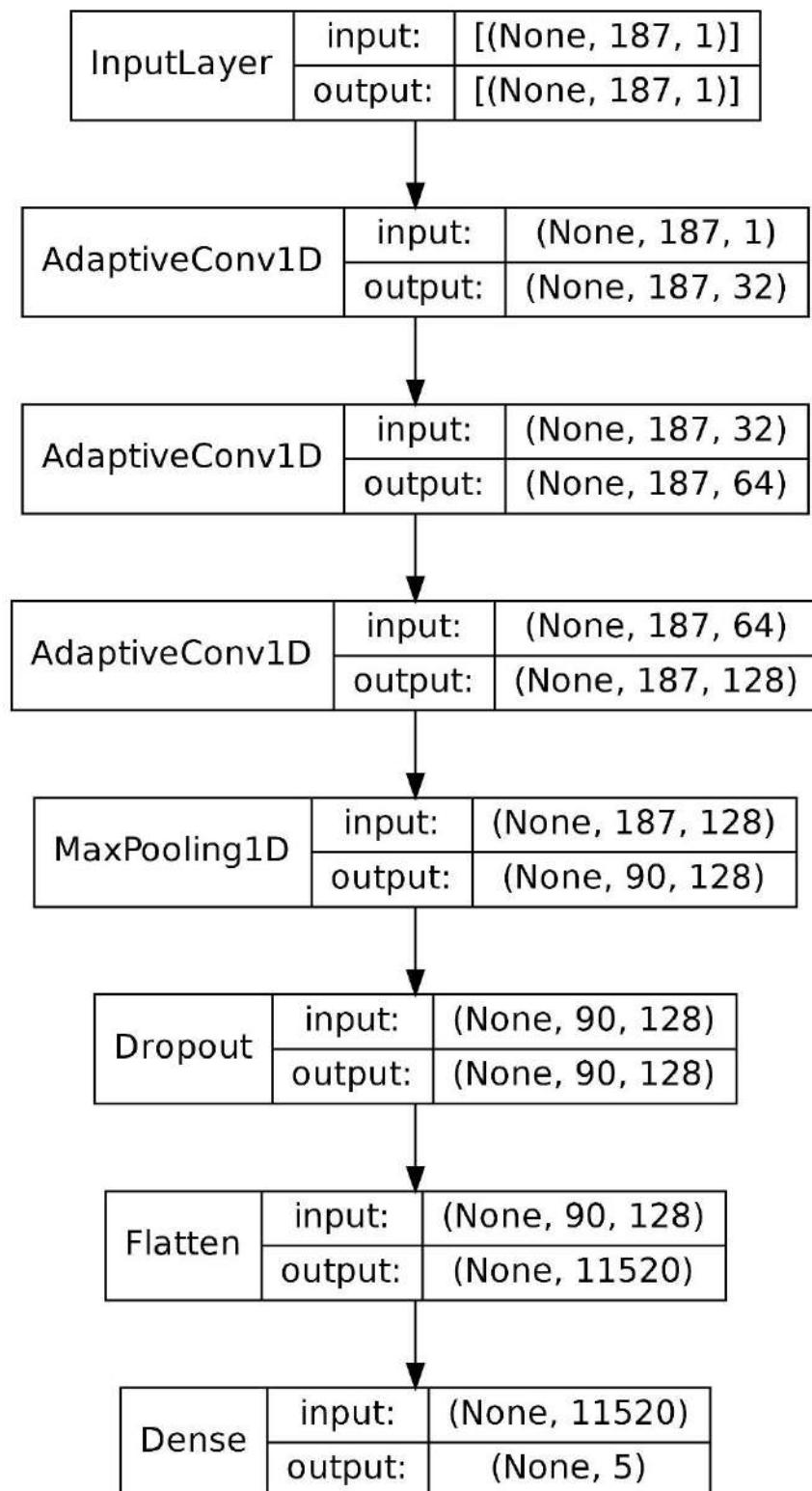


Figure 4-71 version 9 EXP (5) proposed network architecture

Table 4-71 testing results for version 9 EXP. (5)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	92.95%	93.65%	92.87%	92.95%	0.9871	92.83%	91.19%
[15]	93.4%						

Table 4-72 version 9 EXP (5) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8504	0.945	0.8952	800
<b>1</b>	S	0.9264	0.85	0.8866	800
<b>2</b>	V	0.9217	0.9563	0.9387	800
<b>3</b>	F	0.9656	0.9113	0.9376	800
<b>4</b>	Q	0.9949	0.985	0.9899	800
<hr/>					
<b>accuracy</b>				0.9295	4000
<b>macro avg</b>		0.9318	0.9295	0.9296	4000
<b>weighted avg</b>		0.9318	0.9295	0.9296	4000

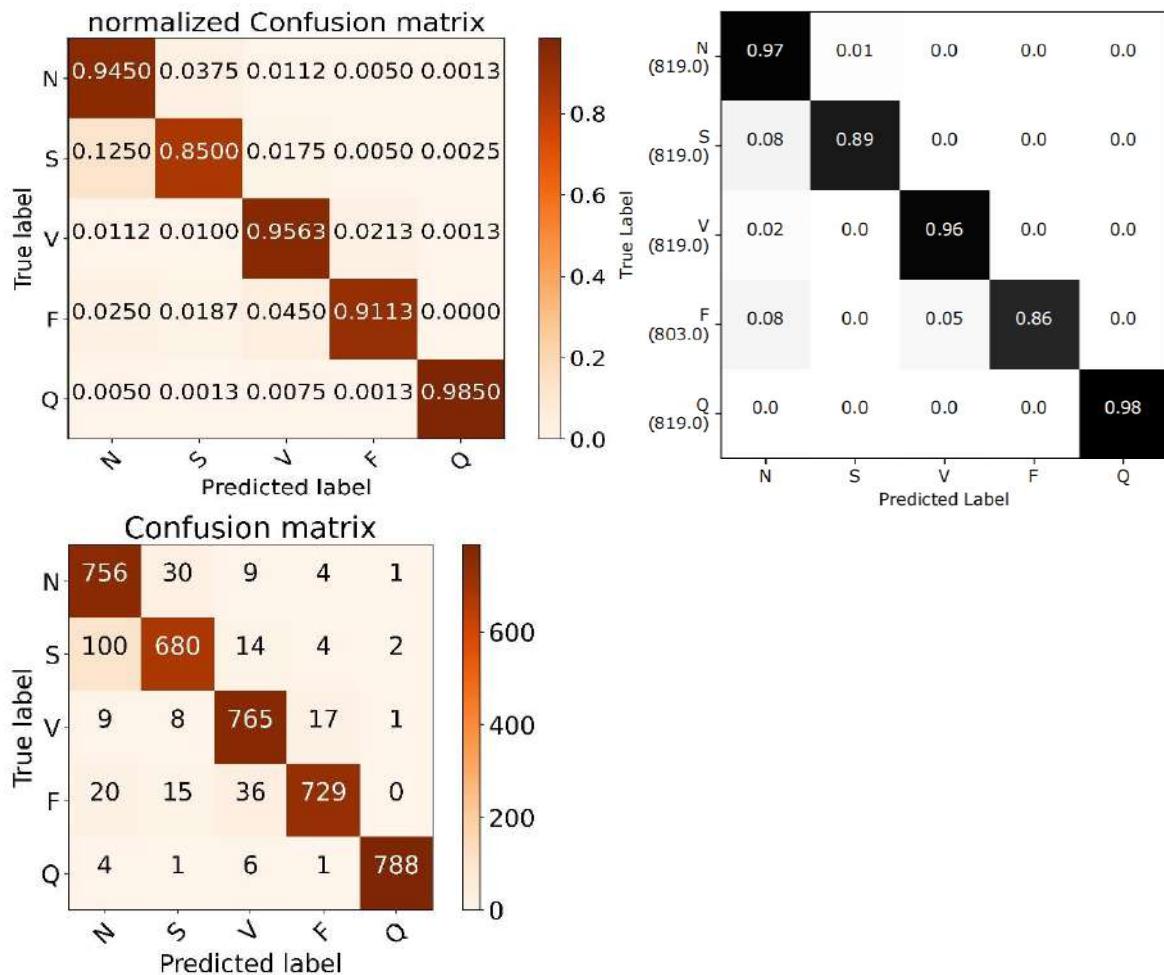


Figure 4-72 version 9 EXP. (5) confusion matrices in orange VS [15] confusion matrix in black

## EXP (6)

The AKSC+MHA with the weighted cross entropy.

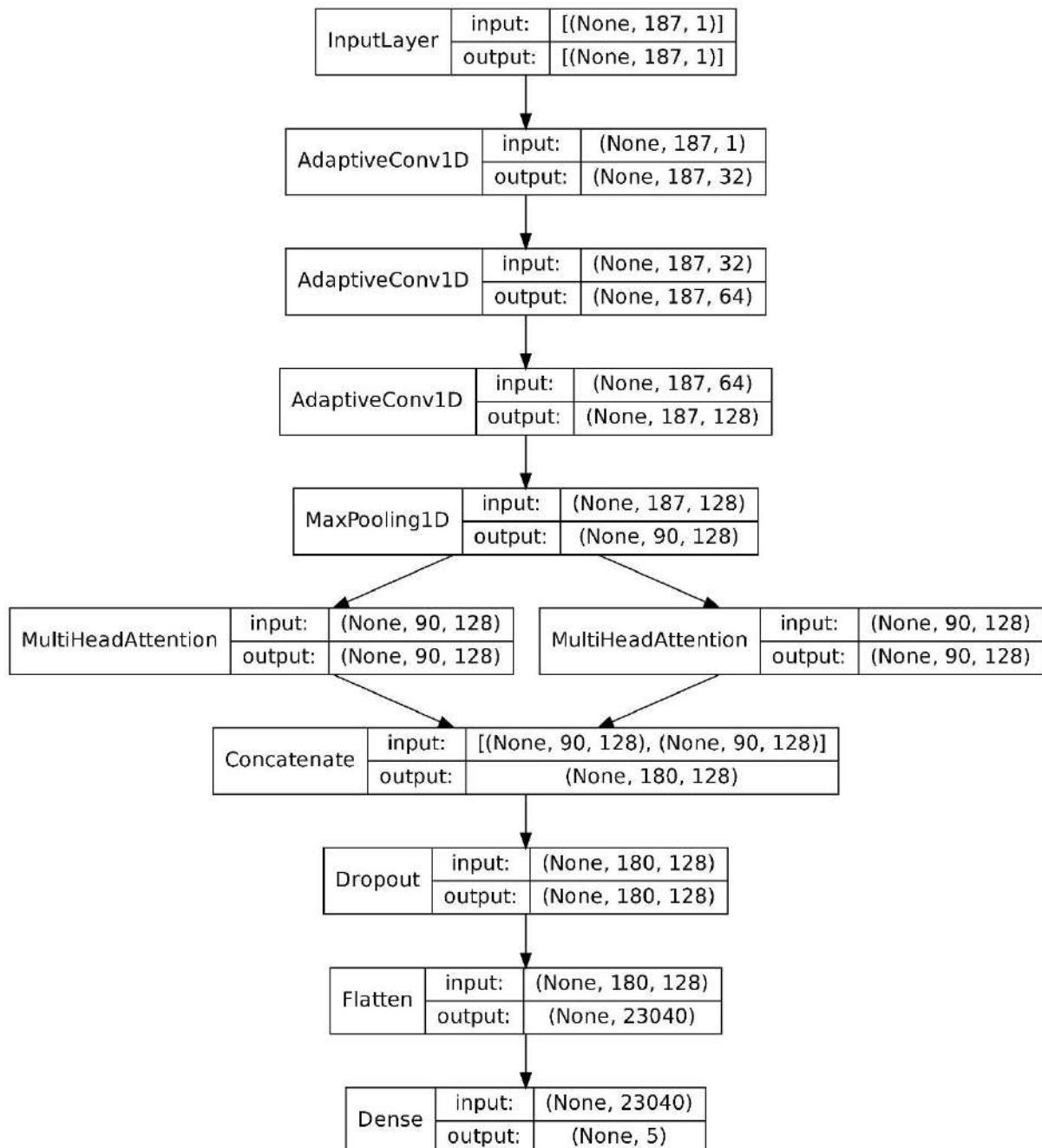


Figure 4-73 version 9 EXP (6) proposed network architecture

Table 4-73 testing results for version 9 EXP. (6)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	92.68%	93.04%	92.22%	92.67%	0.9871	92.56%	90.84%
[15]	93.4%						

Table 4-74 version 9 EXP (6) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8875	0.8775	0.8825	800
<b>1</b>	S	0.9138	0.875	0.894	800
<b>2</b>	V	0.9392	0.9275	0.9333	800
<b>3</b>	F	0.9312	0.9637	0.9472	800
<b>4</b>	Q	0.96	0.99	0.9748	800
<b>accuracy</b>				0.9267	4000
<b>macro avg</b>		0.9263	0.9267	0.9263	4000
<b>weighted avg</b>		0.9263	0.9267	0.9263	4000

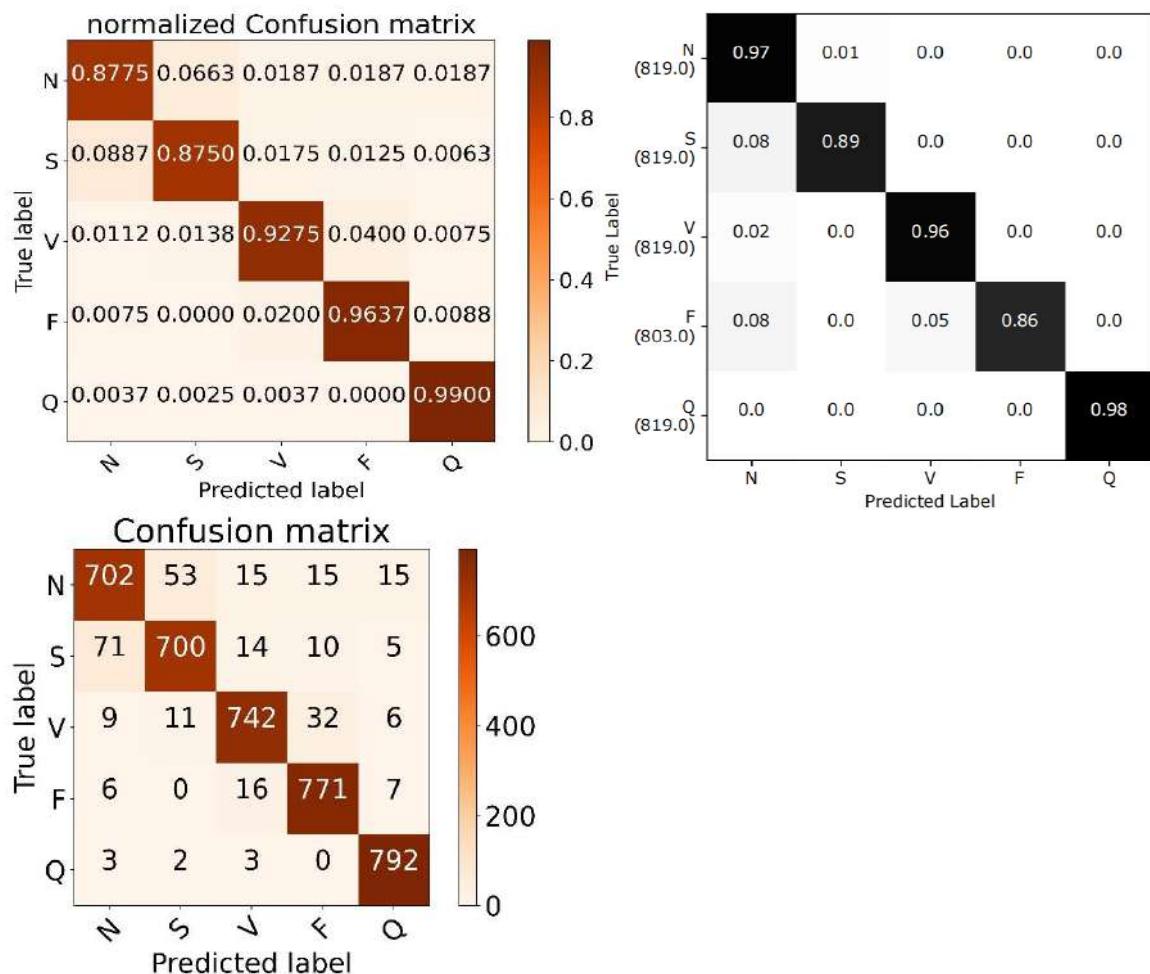


Figure 4-74 version 9 EXP. (6) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (7)

The CNN+MHA combination was utilized with positional encoding before the MHA layer, the basic idea behind positional encoding is to add a fixed vector of sine and cosine component to the embedding of each token in the input sequence, this vector is computed based on the position of the token in the sequence and is designed to be different for each position

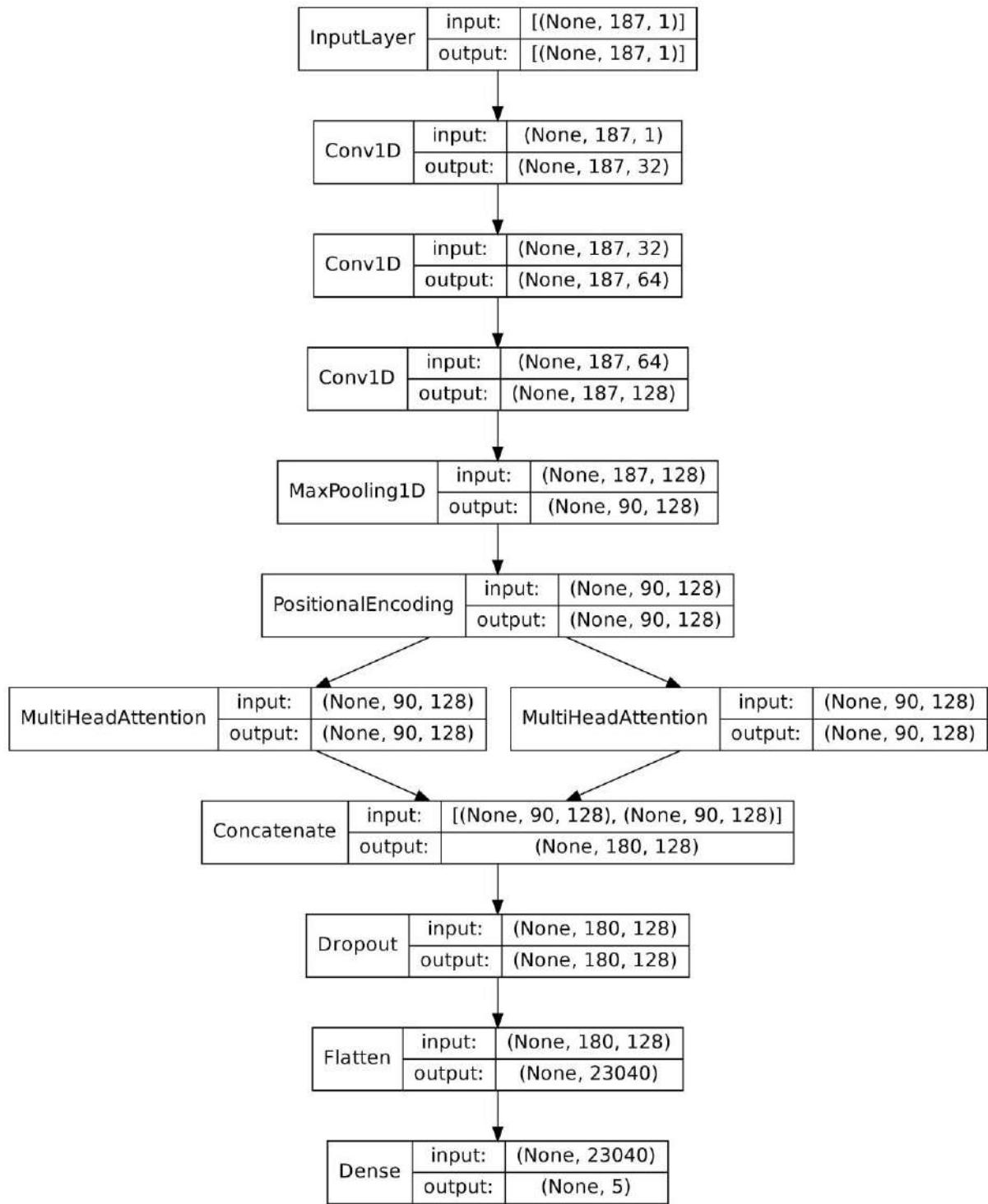


Figure 4-75 version 9 EXP (7) proposed network architecture

Table 4-75 testing results for version 9 EXP. (7)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	91.53%	91.53%	91.53%	91.53%	0.9481	91.45%	89.41%
[15]	93.4%						

Table 4-76 version 9 EXP (7) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8741	0.885	0.8795	800
<b>1</b>	S	0.8656	0.91	0.8873	800
<b>2</b>	V	0.9407	0.8725	0.9053	800
<b>3</b>	F	0.9071	0.9275	0.9172	800
<b>4</b>	Q	0.9949	0.9812	0.988	800
<hr/>					
<b>accuracy</b>				0.9153	4000
<b>macro avg</b>		0.9165	0.9153	0.9155	4000
<b>weighted avg</b>		0.9165	0.9153	0.9155	4000

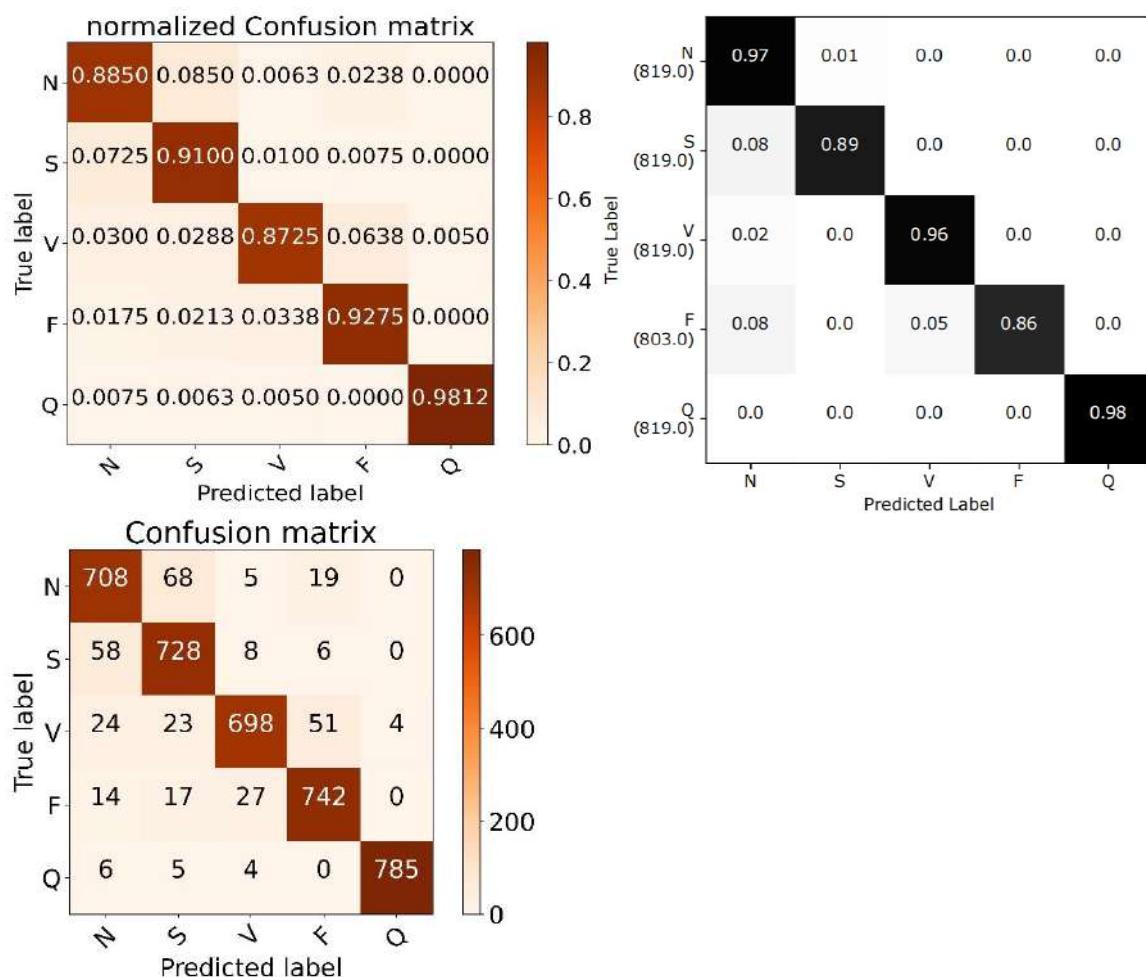


Figure 4-76 version 9 EXP. (7) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (8)

The CNN+LSTM combination was utilized with a residual connection over the LSTM.

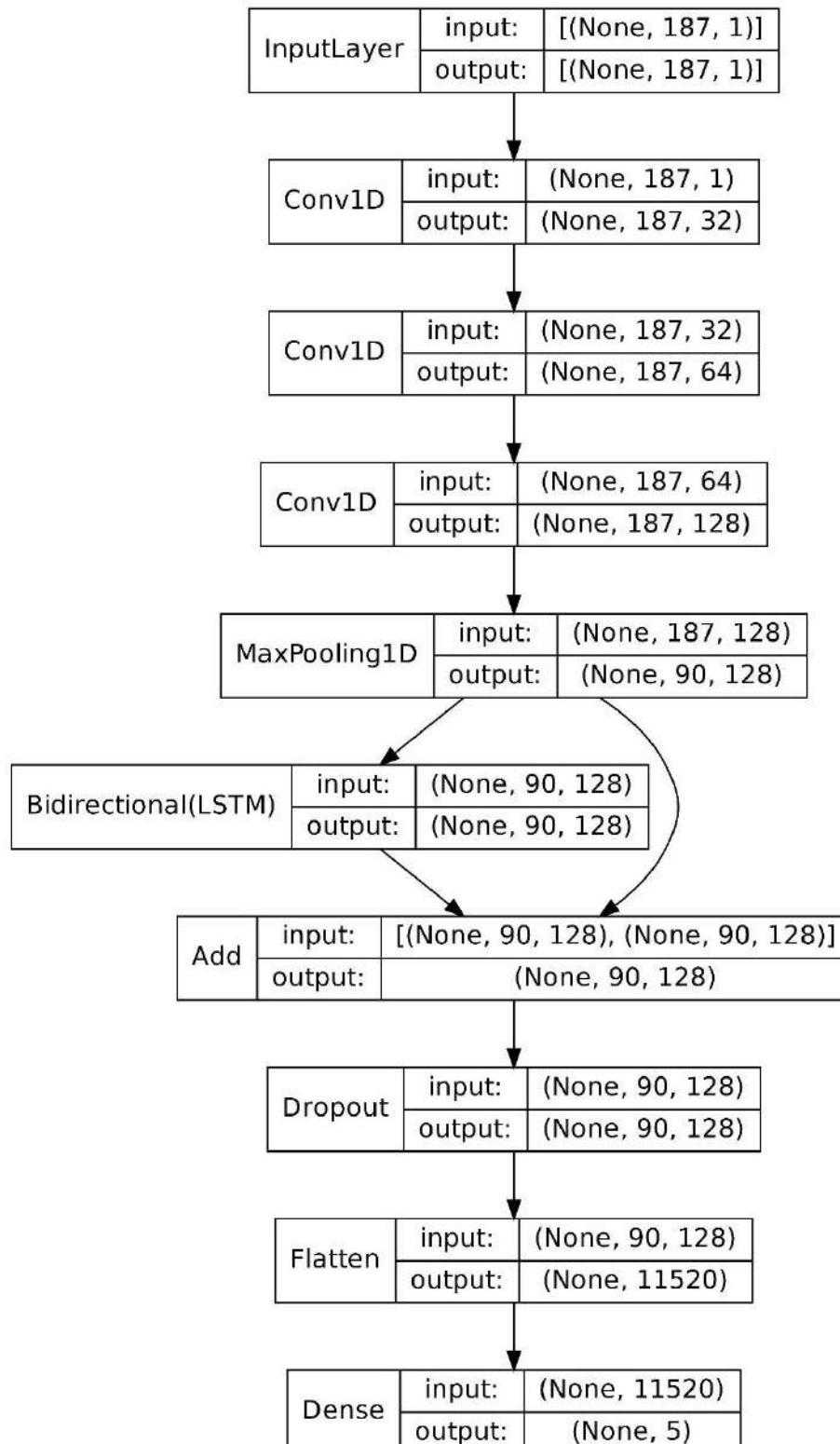


Figure 4-77 version 9 EXP (8) proposed network architecture

Table 4-77 testing results for version 9 EXP. (8)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.77%	94.79%	94.52%	94.77%	0.9882	94.73%	93.47%
[15]	93.4%						

Table 4-78 version 9 EXP (8) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.869	0.945	0.9054	800
<b>1</b>	S	0.946	0.8975	0.9211	800
<b>2</b>	V	0.9694	0.9513	0.9603	800
<b>3</b>	F	0.972	0.9563	0.9641	800
<b>4</b>	Q	0.99	0.9888	0.9894	800
<hr/>					
<b>accuracy</b>				0.9477	4000
<b>macro avg</b>		0.9493	0.9477	0.9480	4000
<b>weighted avg</b>		0.9493	0.9477	0.9480	4000

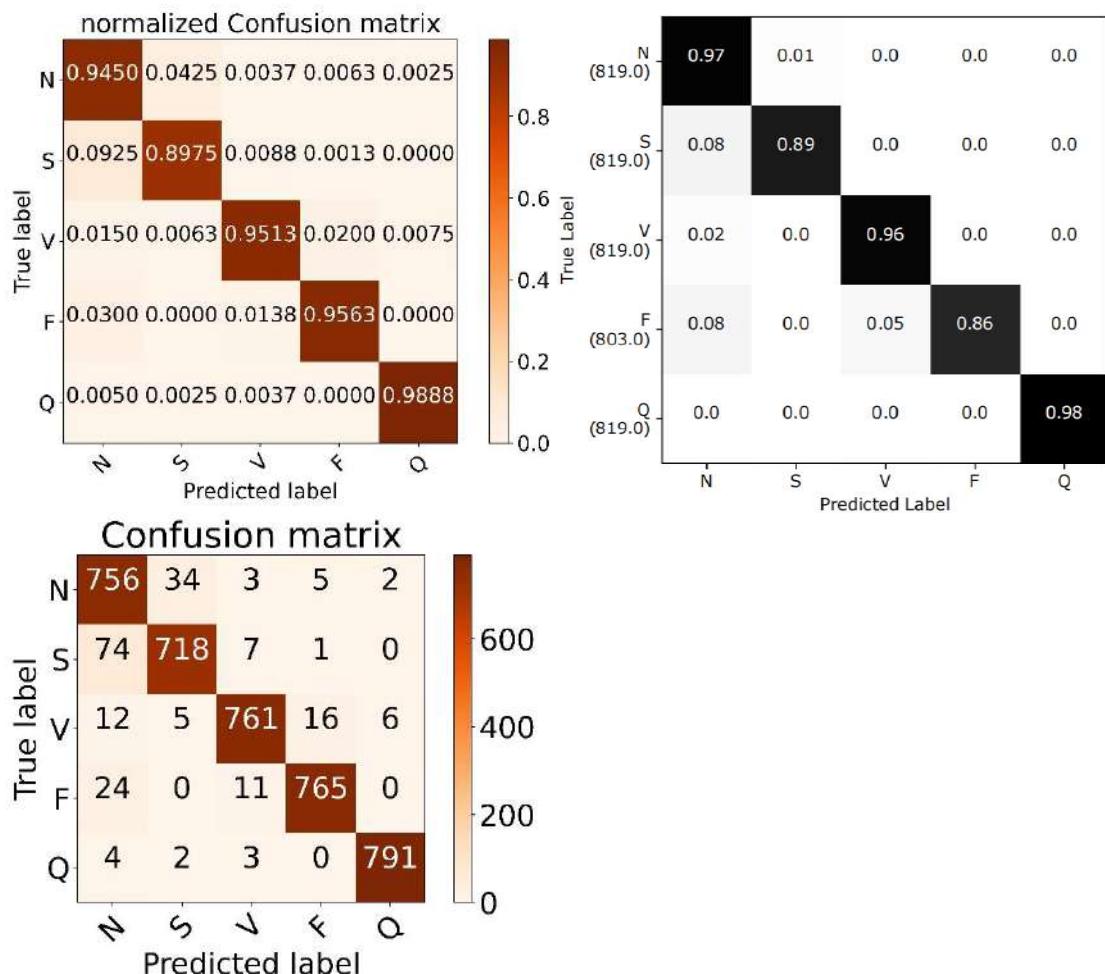


Figure 4-78 version 9 EXP. (8) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (9)

The CNN+AFR+LSTM combination was utilized.

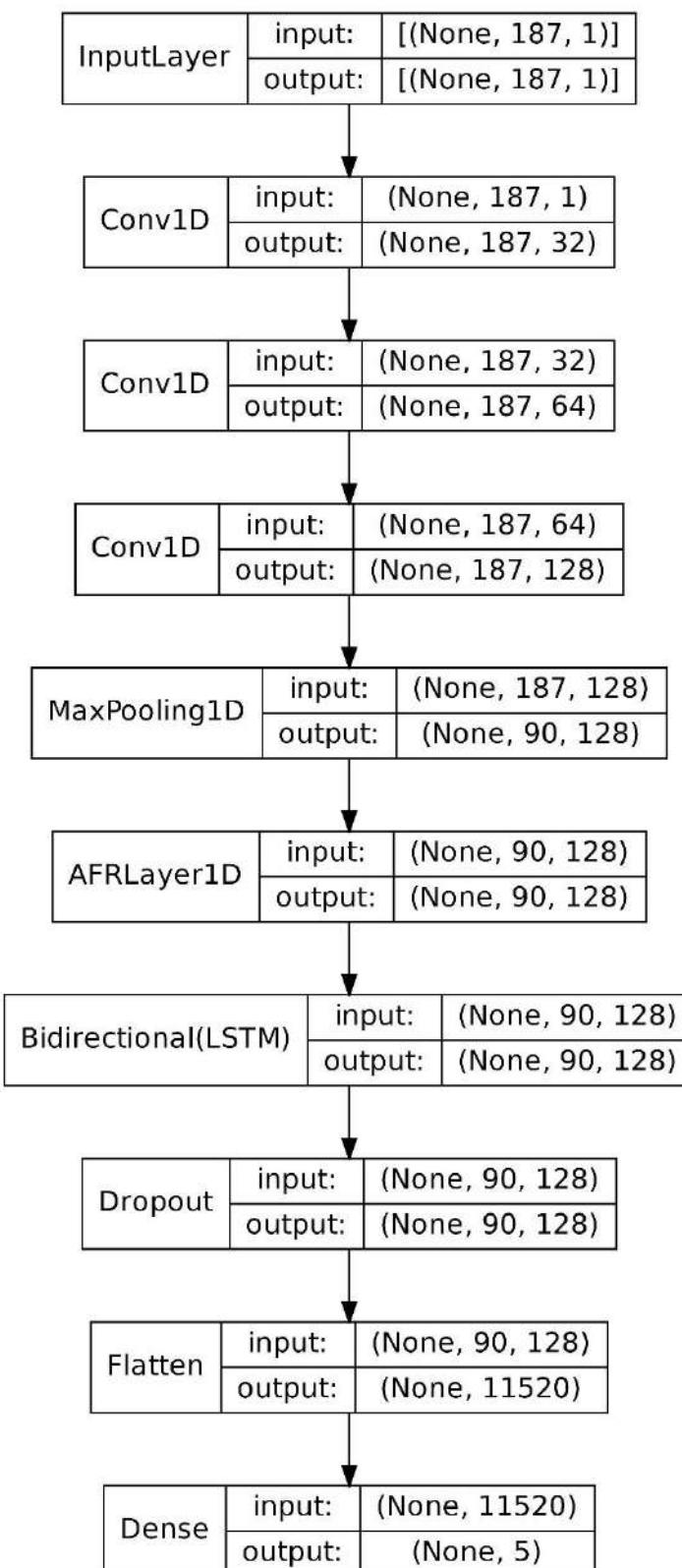


Figure 4-79 version 9 EXP (9) proposed network architecture

Table 4-79 testing results for version 9 EXP. (9)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.88%	95.09%	94.82%	94.88%	0.9885	94.82%	93.59%
[15]	93.4%						

Table 4-80 version 9 EXP (9) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8802	0.9738	0.9246	800
<b>1</b>	S	0.9551	0.905	0.9294	800
<b>2</b>	V	0.9526	0.955	0.9538	800
<b>3</b>	F	0.9685	0.9213	0.9443	800
<b>4</b>	Q	0.9962	0.9888	0.9925	800
<b>accuracy</b>				0.9488	4000
<b>macro avg</b>		0.9505	0.9488	0.9489	4000
<b>weighted avg</b>		0.9505	0.9487	0.9489	4000

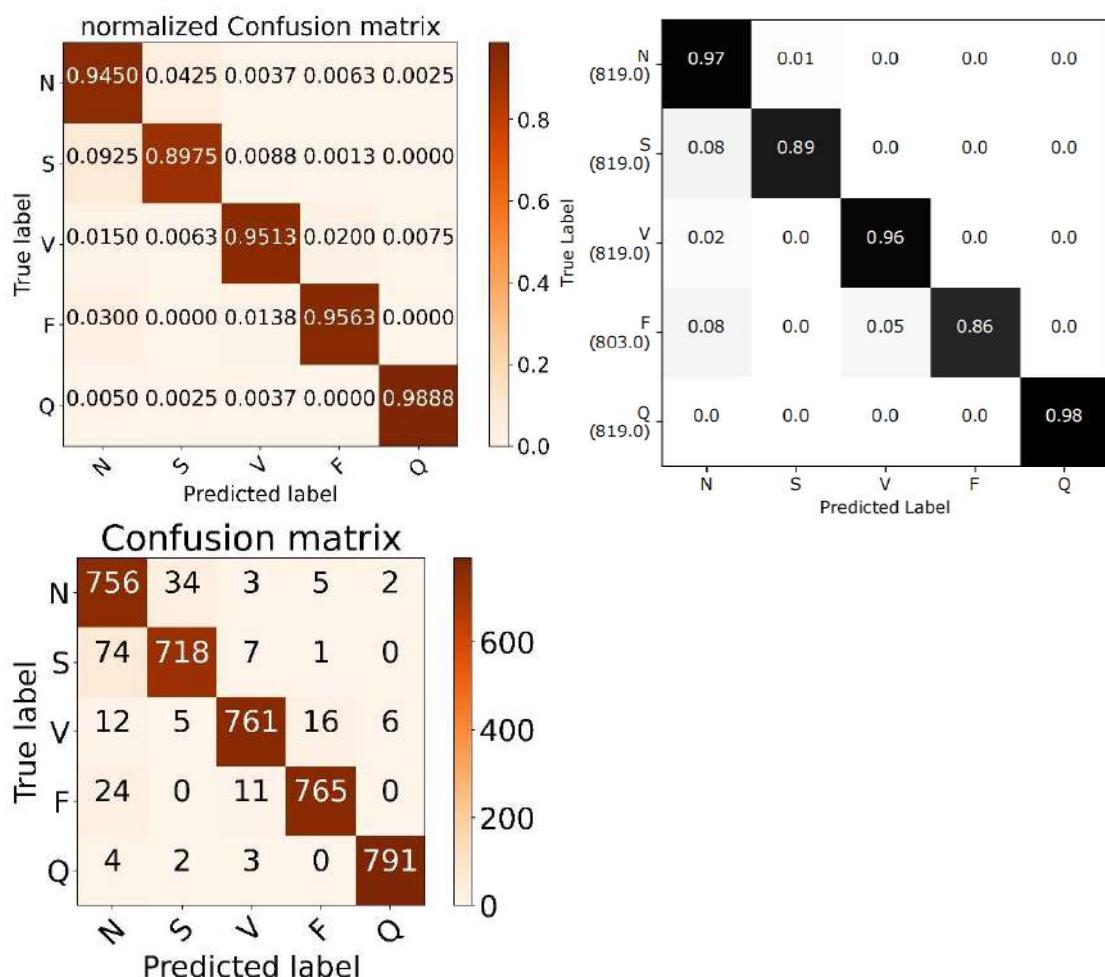


Figure 4-80 version 9 EXP. (9) confusion matrices in orange VS [15] confusion matrix in black

## EXP. (10)

The CNN+AFR+LSTM combination was utilized with adam optimizer and a **reducible learning rate** that starts at 0.001 and get reduced by a factor of 0.4 if the validation loss didn't improve for 5 epoch.

Table 4-81 testing results for version 9 EXP. (10)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	95.10%	95.10%	95.00%	95.10%	0.9853	95.03%	93.88%
[15]	93.4%						

Table 4-82 version 9 EXP (10) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.8818	0.97	0.9238	800
<b>1</b>	S	0.9726	0.8862	0.9274	800
<b>2</b>	V	0.9436	0.9625	0.953	800
<b>3</b>	F	0.9718	0.9487	0.9602	800
<b>4</b>	Q	0.995	0.9875	0.9912	800
<b>accuracy</b>				0.9510	4000
<b>macro avg</b>		0.9511	0.9510	0.9511	4000
<b>weighted avg</b>		0.9511	0.9510	0.9511	4000

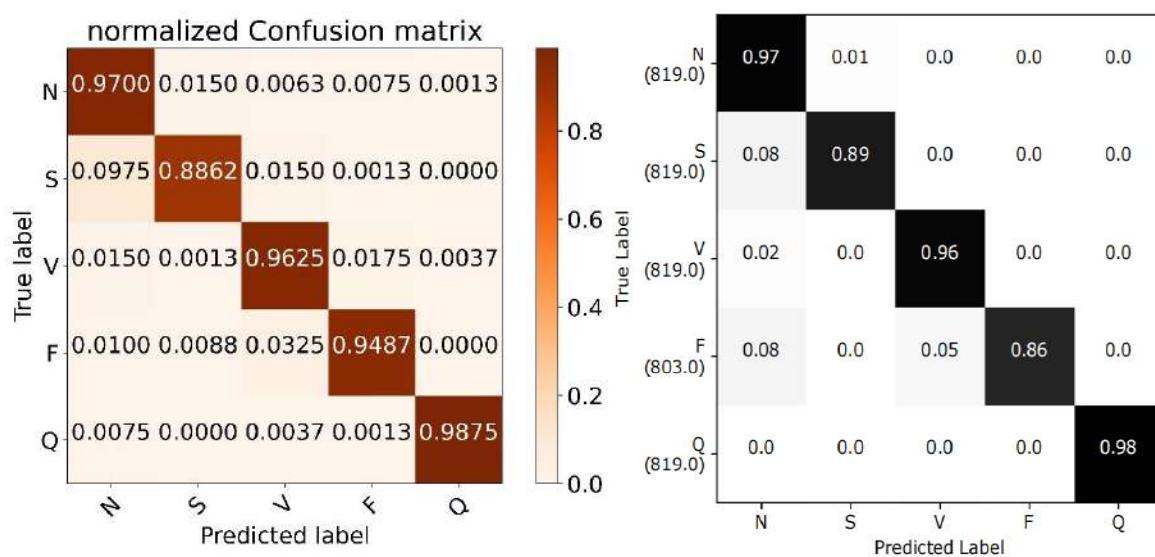


Figure 4-81 version 9 EXP. (10) confusion matrices in orange VS [15] confusion matrix in black

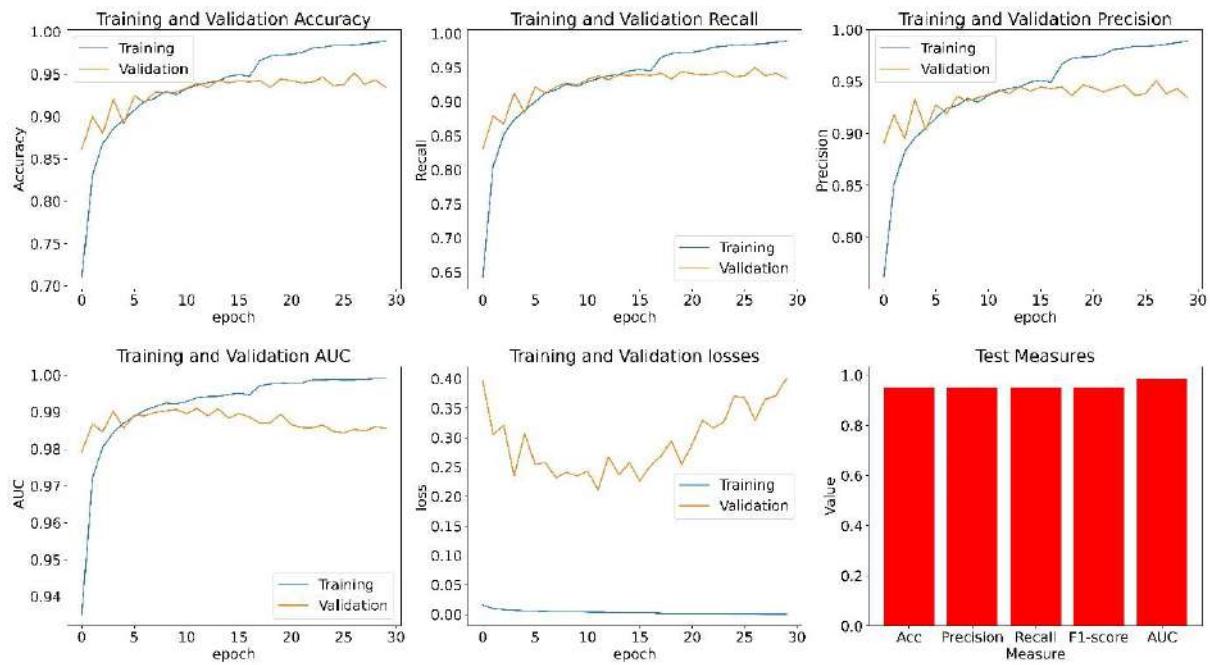


Figure 4-82 training and validation curves of version 9 EXP (10)

## EXP. (11)

The trained model from the last experiment was tested with the test data as it is without over-sampling, note that in the previous experiments the test data was over-sampled for the comparison with [15] as the author over-sampled the test data.

Table 4-83 testing results for version 9 EXP. (10)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	97.11%	97.17%	97.09%	97.11%	0.9960	94.62%	90.94%

Table 4-84 version 9 EXP (10) classification report

index	Class name	Precision	Recall	F1-score	Support
<b>0</b>	N	0.9946	0.9738	0.9841	18118
<b>1</b>	S	0.6993	0.8615	0.772	556
<b>2</b>	V	0.9347	0.9593	0.9468	1448
<b>3</b>	F	0.4185	0.9506	0.5811	162
<b>4</b>	Q	0.9882	0.9913	0.9898	1608
<b>accuracy</b>				0.9711	21892
<b>macro avg</b>		0.8071	0.9473	0.8548	21892
<b>weighted avg</b>		0.9784	0.9711	0.9737	21892

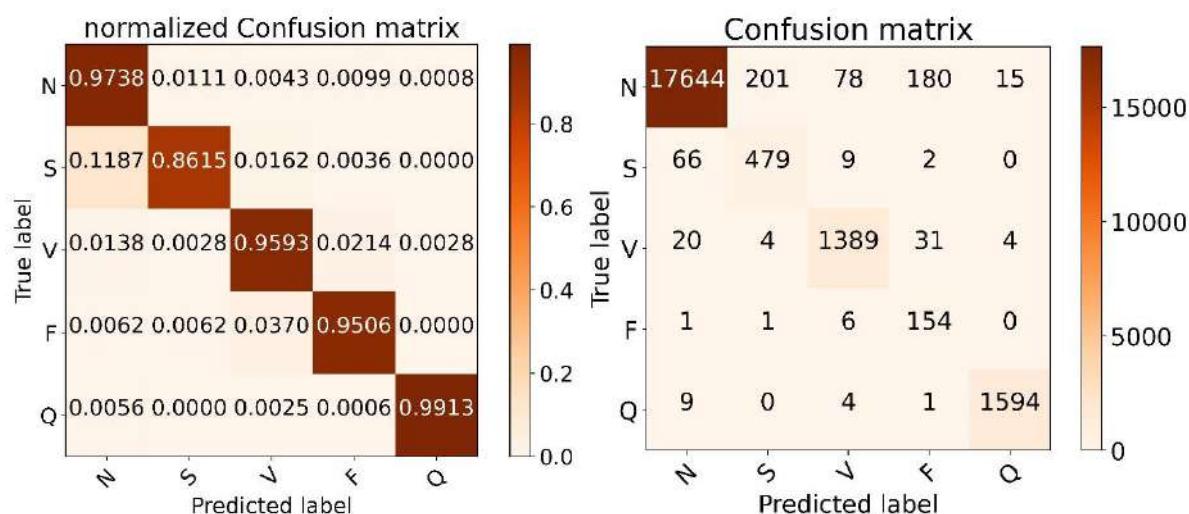


Figure 4-83 version 9 EXP. (11) confusion matrices

## Experiments summary with weighted cross entropy

Table 4-85 different methods with weighted cross entropy

Method	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
<b>Over-sampling the test data to 800</b>							
<b>CNN</b>	93.35%	94.33%	93.23%	93.35%	0.9885	93.30%	91.69%
<b>CNN+ Bi-LSTM</b>	94.77%	95.08%	94.75%	94.77%	0.9876	94.72%	93.47%
<b>CNN+ Res- Bi-LSTM</b>	94.77%	94.79%	94.52%	94.77%	0.9882	94.73%	93.47%
<b>CNN+AFR</b>	93.98%	94.22%	93.67%	93.97%	0.9885	93.93%	92.47%
<b>CNN+MHA</b>	92.65%	92.65%	92.65%	92.65%	0.9816	92.59%	90.81%
<b>CNN+PE+ MHA</b>	91.53%	91.53%	91.53%	91.53%	0.9481	91.45%	89.41%
<b>CNN+AFR+ MHA</b>	93.45%	93.67%	93.27%	93.45%	0.9852	93.34%	91.81%
<b>AKSC</b>	92.95%	93.65%	92.87%	92.95%	0.9871	92.83%	91.19%
<b>AKSC+MHA</b>	92.68%	93.04%	92.22%	92.67%	0.9871	92.56%	90.84%
<b>CNN+AFR+ Bi-LSTM</b>	94.88%	95.09%	94.82%	94.88%	0.9885	94.82%	93.59%
<b>CNN+AFR+ Bi-LSTM+RLR</b>	95.10%	95.10%	95.00%	95.10%	0.9853	95.03%	93.88%
<b>[15]</b>	93.4%						
<b>Using the test data as it is</b>							
<b>CNN+AFR+ Bi-LSTM+RLR</b>	97.11%	97.17%	97.09%	97.11%	0.9960	94.62%	90.94%

Table 4-86 Abbreviations

<b>Bi-LSTM</b>	Bidirectional Long-short time memory
<b>AFR</b>	Adaptive Feature Recalibration (squeeze and excitation)
<b>MHA</b>	Multi-Head Attention
<b>AKSC</b>	Adaptive kernel Size Convolution
<b>PE</b>	Positional Encoding
<b>Res- Bi-LSTM</b>	Bidirectional LSTM with residual connection
<b>RLR</b>	Reducible learning rate

As shown in the table the best results with the over-sampled test data was achieved by the CNN+AFR+LSTM+RLR method with 95.10% accuracy, the least results was achieved by the CNN+MHA method with accuracy of 92.65%, using the test data as it is results in an accuracy of 97.11%.

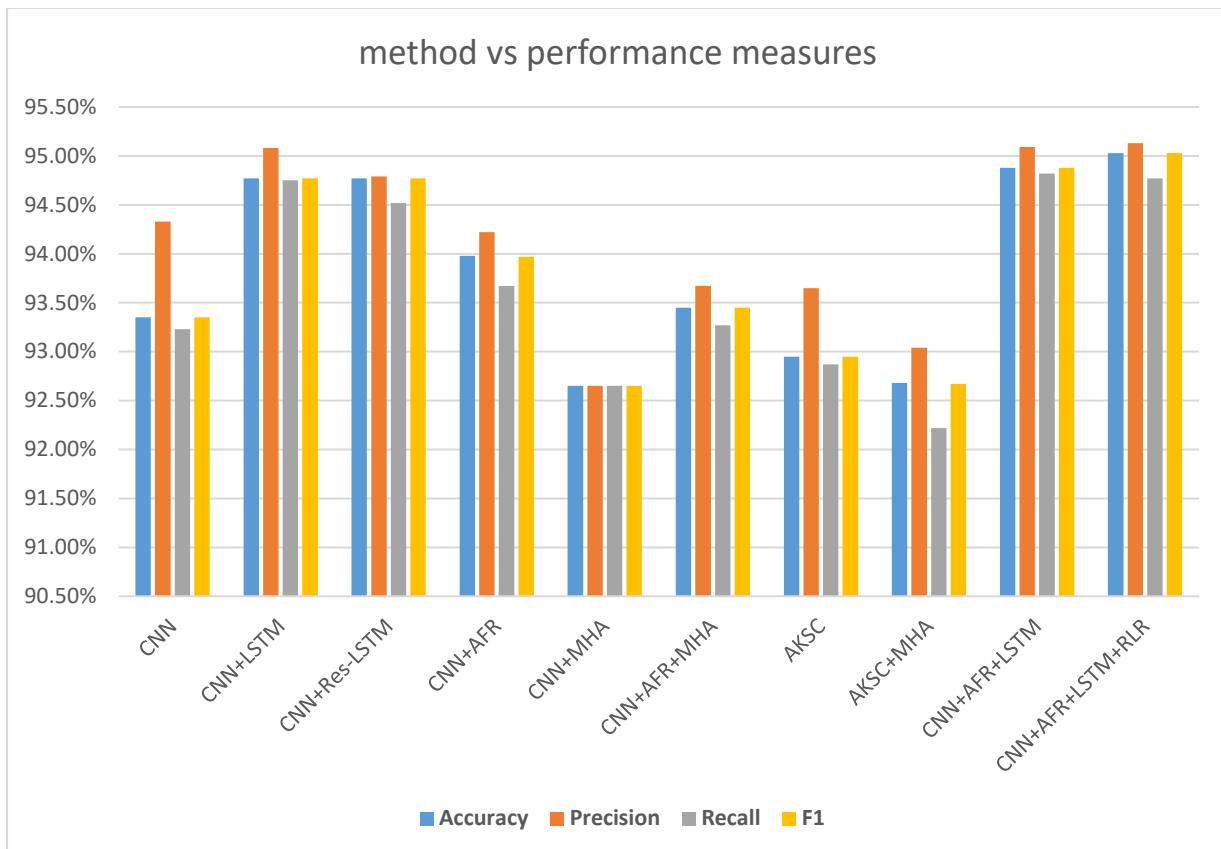


Figure 4-84 method vs performance measure chart with the test data over-sampled to 800

## GPU specs

NVIDIA T4 GPU

Architecture: Turing

CUDA Cores: 2,560

Tensor Cores: 320

Memory Size: 16 GB GDDR6

Memory Bandwidth: 320 GB/s

Memory Interface: 256-bit

Max Power Consumption: 70 Watts

PCIe Interface: PCIe 3.0 x16

Form Factor: Single-slot, Full-height

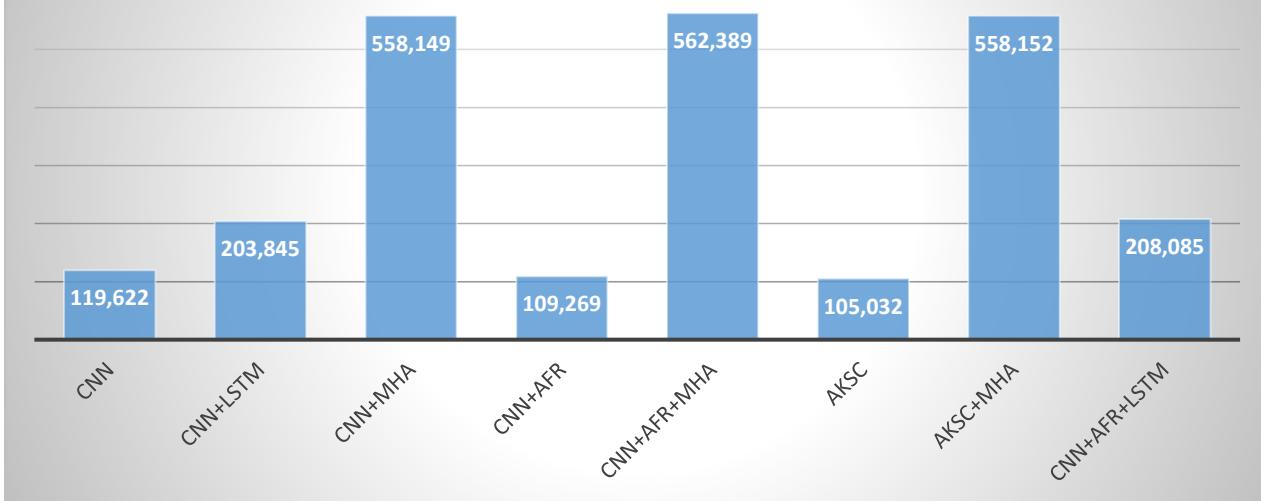
Compute Capability: 7.5

GPU Boost Clock: 1,590 MHz

## Training time and number of parameters

Method	Training time	Total parameters	Trainable parameters	Non trainable parameters
CNN	8min	105,029	105,029	0
CNN+LSTM	29min	203,845	203,845	0
CNN+MHA	16min 40s	558,149	558,149	0
CNN+AFR	8min 39s	109,269	109,269	0
CNN+AFR+MHA	18min 56s	562,389	562,389	0
AKSC	8min	105,032	105,032	0
AKSC+MHA	16min 31s	558,152	558,152	0
CNN+AFR+LSTM	29min 30s	208,085	208,085	0

Total parameters



Training time in minutes



#### 4.2.1.7 Final proposed framework

Out of all the methods that was experimented with the combination of 1D-CNN, adaptive feature recalibration and long-short time memory had the best results with an accuracy of 97.11% using the test data as it is

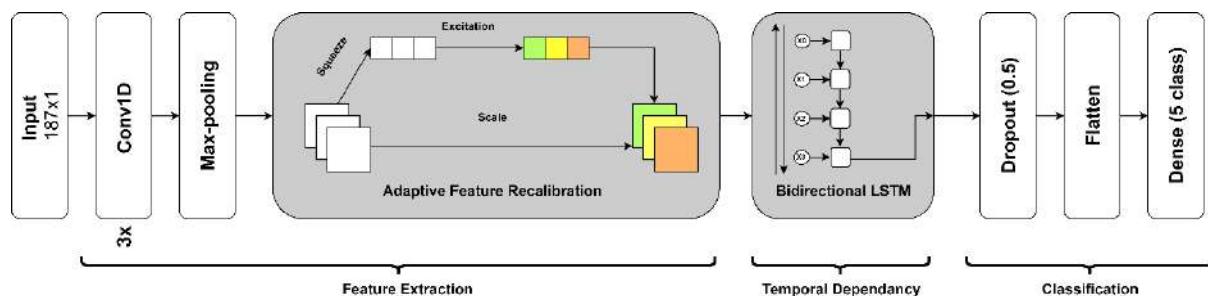


Figure 4-85 final proposed network architecture

	Architecture	Temporal dependency	Handling data imbalance	Remarks	Accuracy	Macro F1
<b>Resampling the test data to 800</b>						
[15]	Residual CNN	-	Oversampling	Low recall on S,F classes	93.4%	
<b>Ours</b>	<b>CNN+AFR</b>	<b>Bi-LSTM</b>	<b>Weighted loss</b>	<b>Good recall</b>	<b>95.10%</b>	
<b>Test data as it is</b>						
[71]	CNN	-	Oversampling	Low recall on S,F classes	93.47%	
[72]	SVM	-	-	Low recall		82%
<b>Ours</b>	<b>CNN+AFR</b>	<b>Bi-LSTM</b>	<b>Weighted loss</b>	<b>Good recall</b>	<b>97.11%</b>	<b>85.48%</b>

#### 4.2.1.8 Method validation

For validating the final model the PTB Diagnostic ECG Database was used to train and test the model's performance with the same preprocessing as the dataset which was used for all the previous experiments

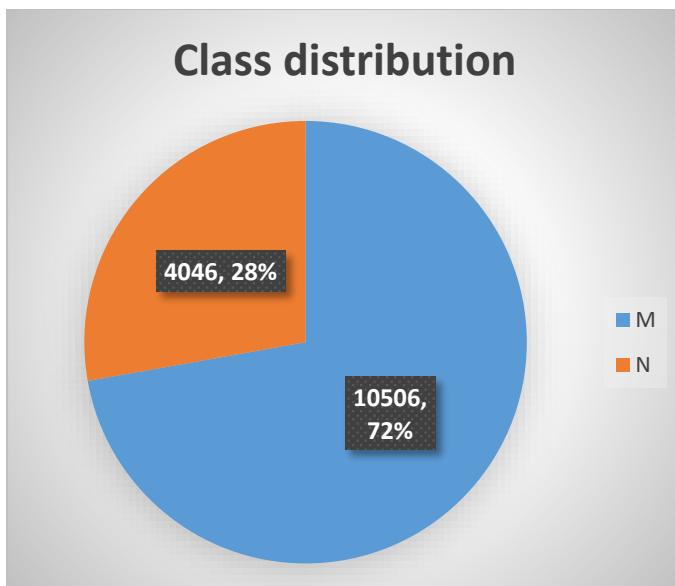


Figure 4-86 class distribution of PTB Diagnostic ECG Database

Table 4-87 testing results comparison with the state of the art

	Method	Accuracy	precision	Recall
Proposed	CNN+AFR+ Bi-LSTM+RLR	99.79%	99.82%	99.66%
[15]	Transfer learning Res-CNN	95.9%	95.1%	95.2%

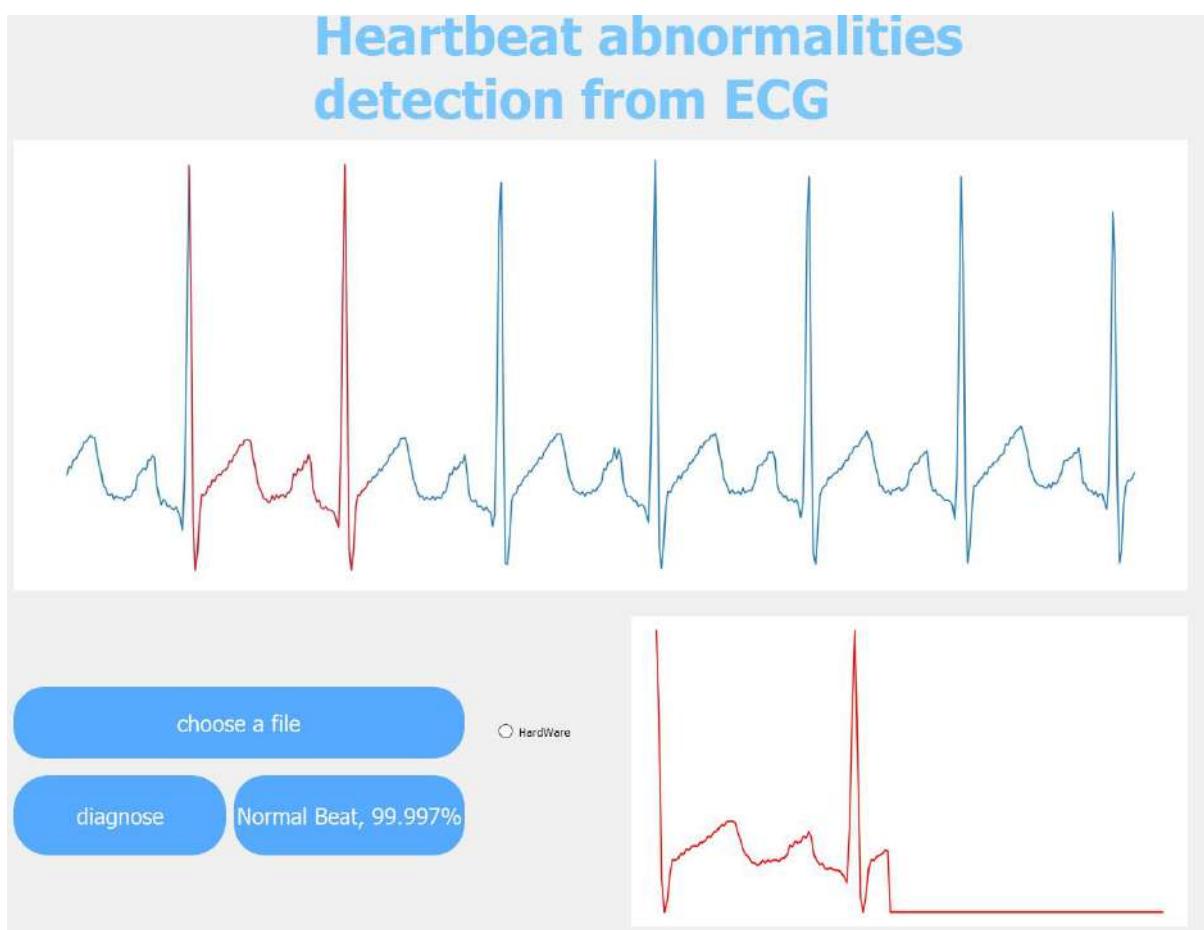
Table 4-88 abbreviations

<b>Bi-LSTM</b>	Bidirectional Long-short time memory
<b>AFR</b>	Adaptive feature recalibration (squeeze and excitation)
<b>RLR</b>	Reducible learning rate
<b>Res-CNN</b>	Residual CNN

#### 4.2.1.9 Testing with real world cases (online testing)

After the training process is done and the final model was chosen to be the combination of the 1D-CNN, the adaptive feature recalibration (AFR), the bidirectional loge-short time memory (bi-LSTM), reducible learning rate (RLR) and the weighted loss function (WL) (CNN+AFR+Bi-LSTM+RLR+WL) which achieved 97.11% accuracy, it is time to test it on some samples completely out side of the dataset used in the training and testing, another dataset with different source form the one used for training was utilized, this dataset have different classes than the one used for training except for the normal class so the online testing was only performed on 10 normal samples, the model diagnosed all 10 samples to be normal and most of them with confidence above 99%.

Please note that the GUI will be discussed in details in section 4.3



## Heartbeat abnormalities detection from ECG

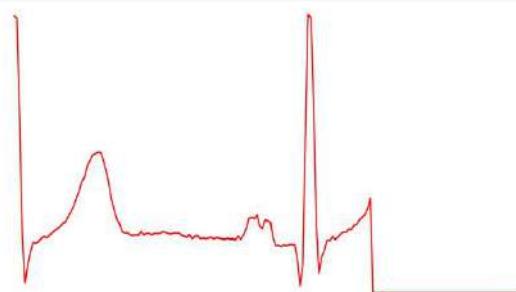


choose a file

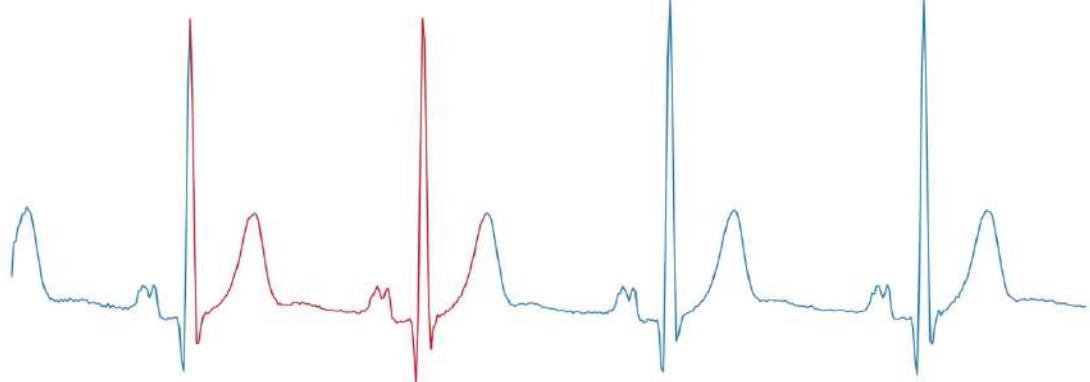
HardWare

diagnose

Normal Beat, 99.938%



## Heartbeat abnormalities detection from ECG

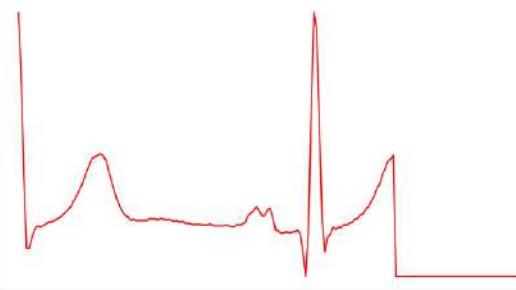


choose a file

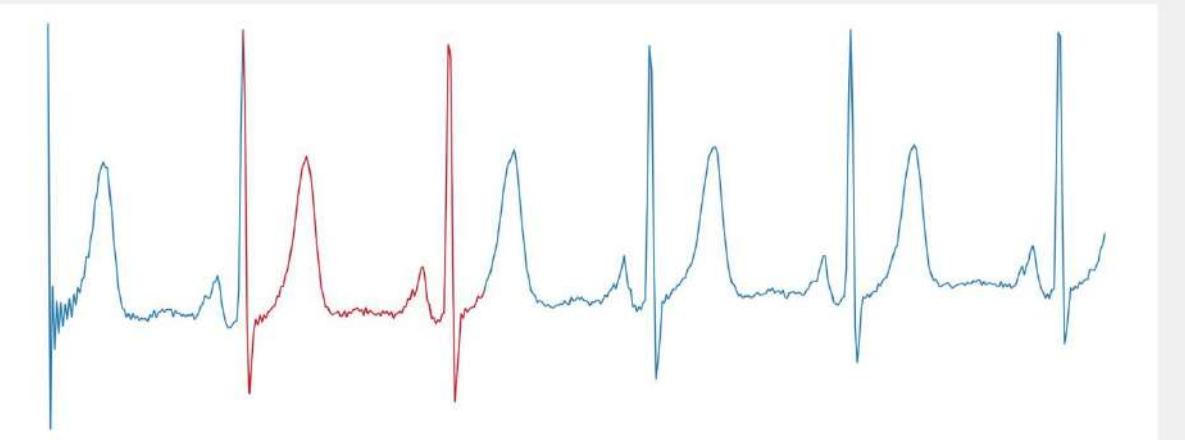
HardWare

diagnose

Normal Beat, 99.959%



## Heartbeat abnormalities detection from ECG

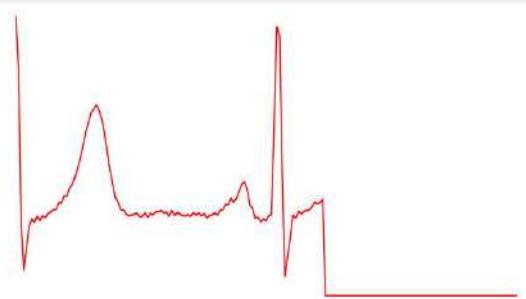


choose a file

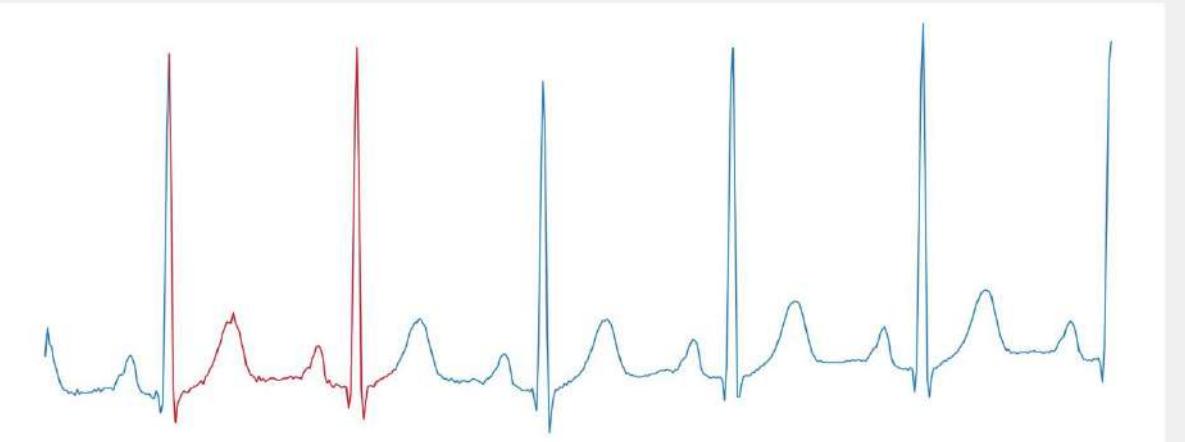
HardWare

diagnose

Normal Beat, 90.183%



## Heartbeat abnormalities detection from ECG

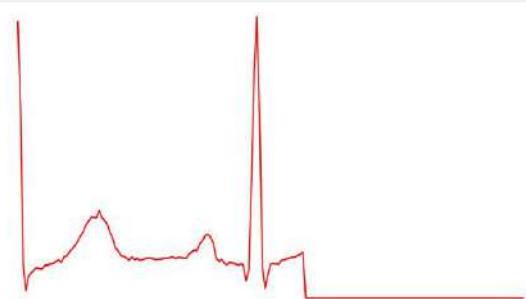


choose a file

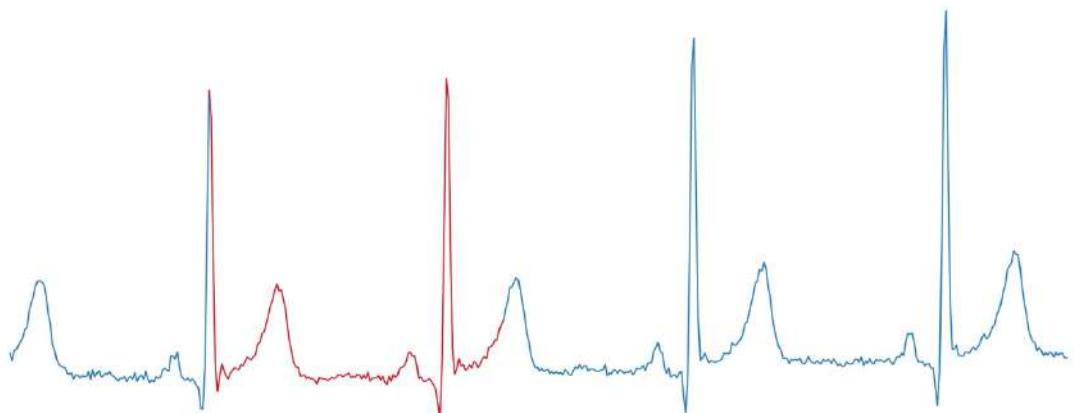
HardWare

diagnose

Normal Beat, 99.892%



## Heartbeat abnormalities detection from ECG

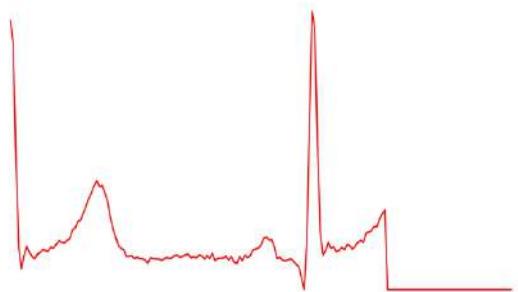


choose a file

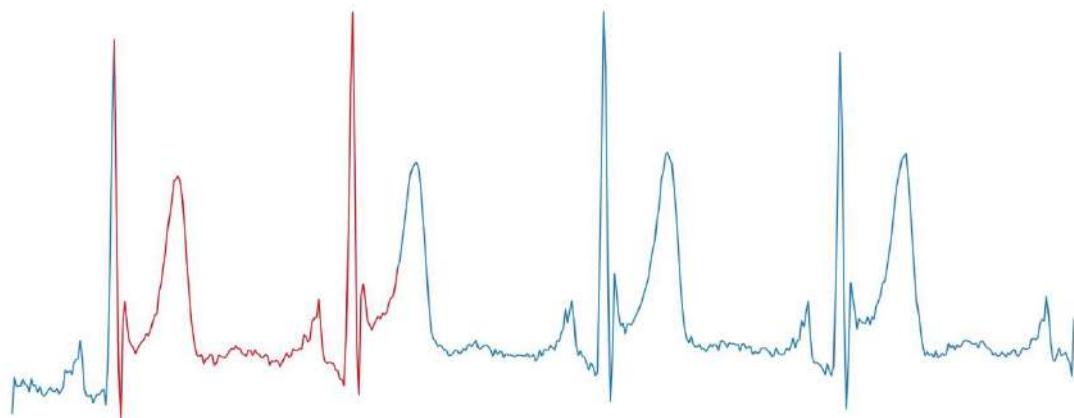
diagnose

Normal Beat, 99.939%

○ HardWare



## Heartbeat abnormalities detection from ECG

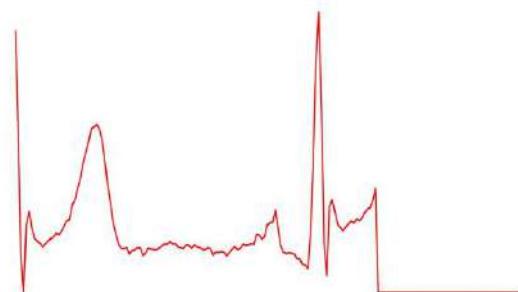


choose a file

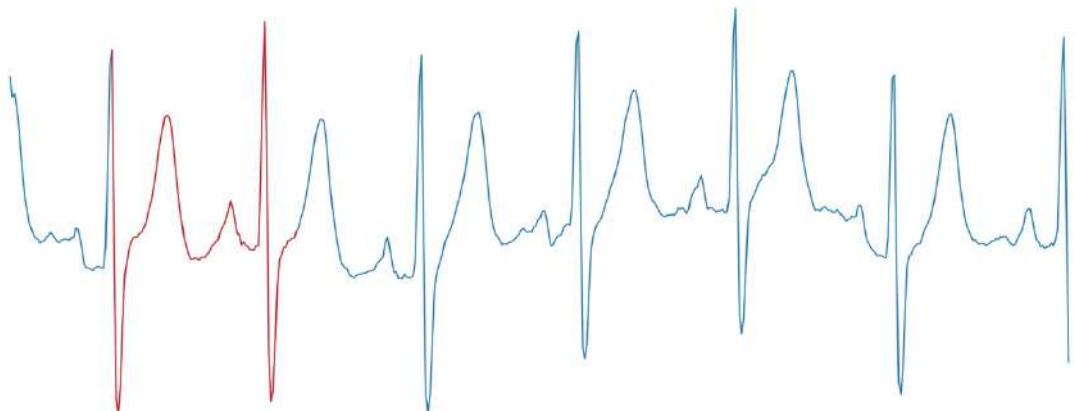
diagnose

Normal Beat, 99.998%

○ HardWare



## Heartbeat abnormalities detection from ECG

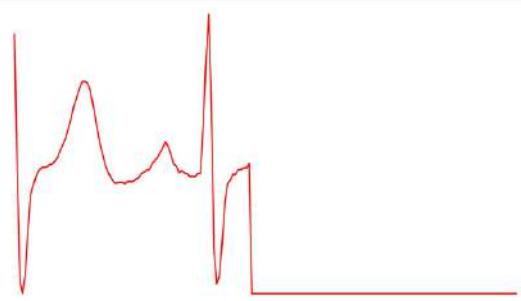


choose a file

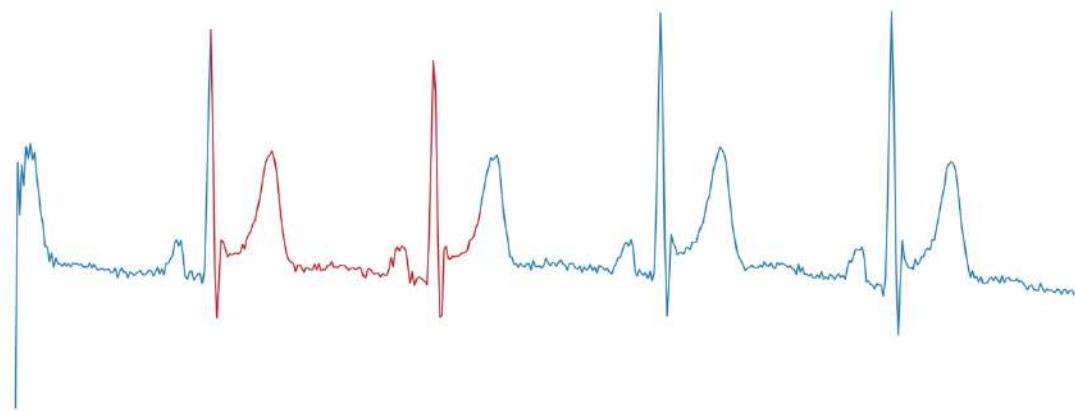
diagnose

Normal Beat, 99.998%

○ HardWare



## Heartbeat abnormalities detection from ECG



choose a file

diagnose

Normal Beat, 63.924%

○ HardWare



## Heartbeat abnormalities detection from ECG



### 4.2.1.10 Hardware system for real time diagnosis of ECG signal

The purpose of the hardware system is to take the ECG signal from a test subject and send it to the application over the Wi-Fi to be diagnosed in real time, the reason for using Wi-Fi is that the doctors has a separate room in their clinics for the ECG and the patient would perform the ECG test and waits for the results before meeting the doctor, for the proposed system to be compatible with that a wireless connection is required to send the ECG reading to the doctor's PC on which the application would be running without disturbing the process of managing patients at the clinic.

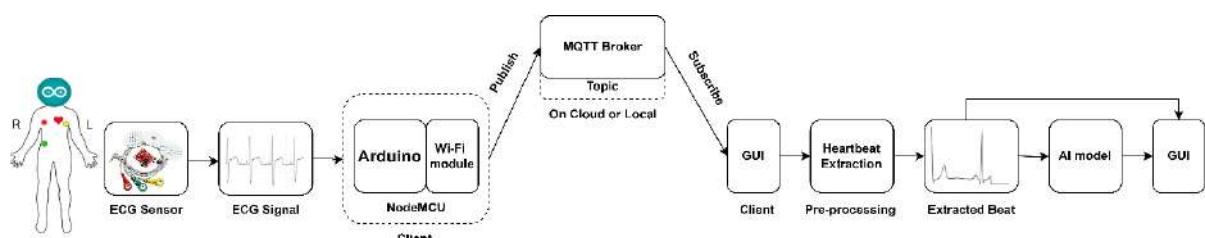


Figure 4-87 block diagram of the hardware system

As seen in the previous figure, the real time diagnosis system of the ECG signal consists of three main parts, first the hardware part which consists of an AD8232

ECG sensor module connected with a NodeMCU which is an Arduino microcontroller with a built in Wi-Fi module which will receive the analog output of the sensor (time series data) and sends it over the Wi-Fi to the second part which is the MQTT server (broker) which could run on cloud such as Hivemq broker or run on a local machine such as mosquito broker, the broker will receive the time series data from the NodeMCU and sends it to the third part which is the application to be pre-processed, visualized and classified by the developed AI model.

## Part 1: The Hardware

The hardware mainly consists of firstly an ECG sensor, there were 3 different modules available Max300, PPG and AD8232, the first two are IR based sensors which reads the ECG signal by measuring the changes of the blood volume in the fingertip vessels which in turn affects the amount of light that is transmitted or reflected back to the sensor.

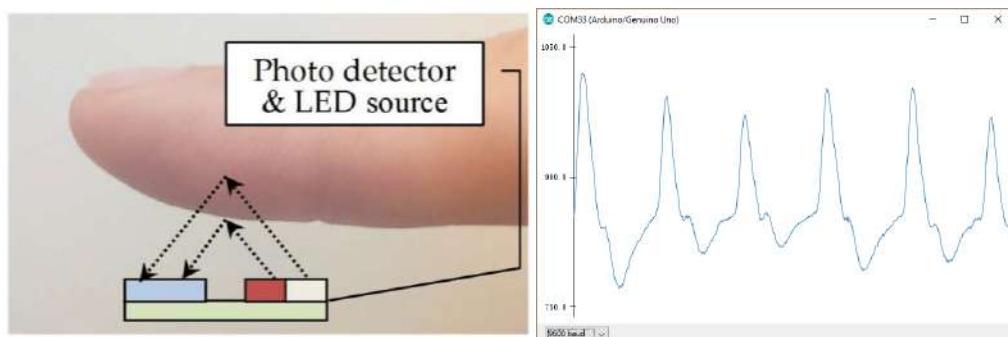


Figure 4-88 concept of reading ECG signals using IR beams and output signal of ppg sensor

This method could be fine for measuring the number of beats per minutes but the resulting ECG signal was very noisy compared to an actual ECG signal

The AD8232 module was the better choice for this as it uses the same concept used in medical ECG monitoring devices, the AD8232 sensor captures the electrical signals generated by the heart using electrodes placed on the body it then enhances and strengthens these signals, reduces unwanted noise, and removes interference from the surroundings, the final processed ECG signal is available at the sensor's output for further analysis or display.

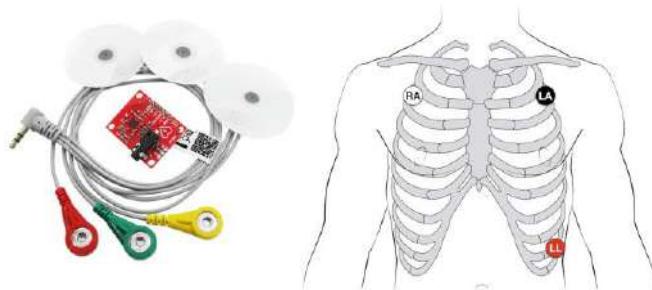


Figure 4-89 AD8232 module and electrode placement

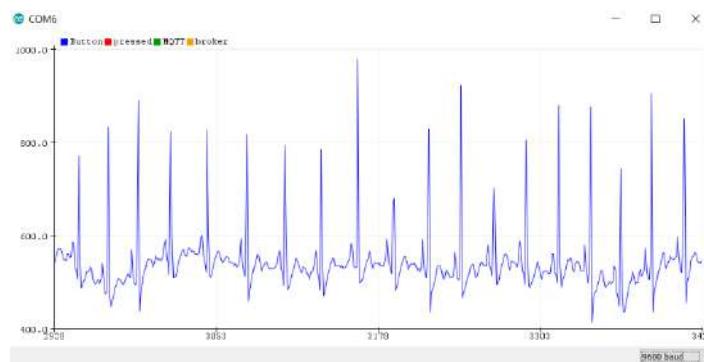


Figure 4-90 output of AD8232 module

Secondly, we have a NodeMCU which is an open-source development board that combines the functionality of an ESP8266 Wi-Fi module with an integrated Arduino microcontroller.

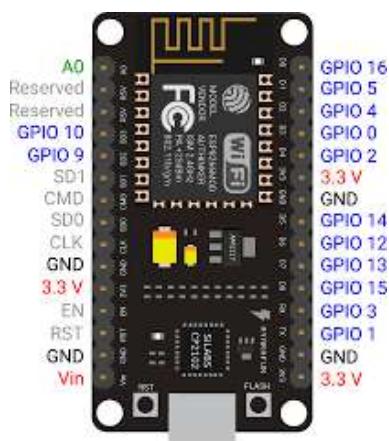


Figure 4-91 NodeMCU pinout

Thirdly, we have an Adafruit oled displays (SSD1306) which provide support for graphics and text display, which is connected with the NodeMCU to visualize the output signal from the sensor, this display has two input pins which are the SCL (serial clock) used to synchronize data transfers between the microcontroller and the oled display and SDA (serial data) which carries the actual data being transmitted between the microcontroller and the OLED display



Figure 4-92 SSD1306 oled display

Fourthly, we have a push button, when the button is pressed the NodeMCU start sending the output of the sensor in real time for four seconds which is enough for the heartbeat extraction process.

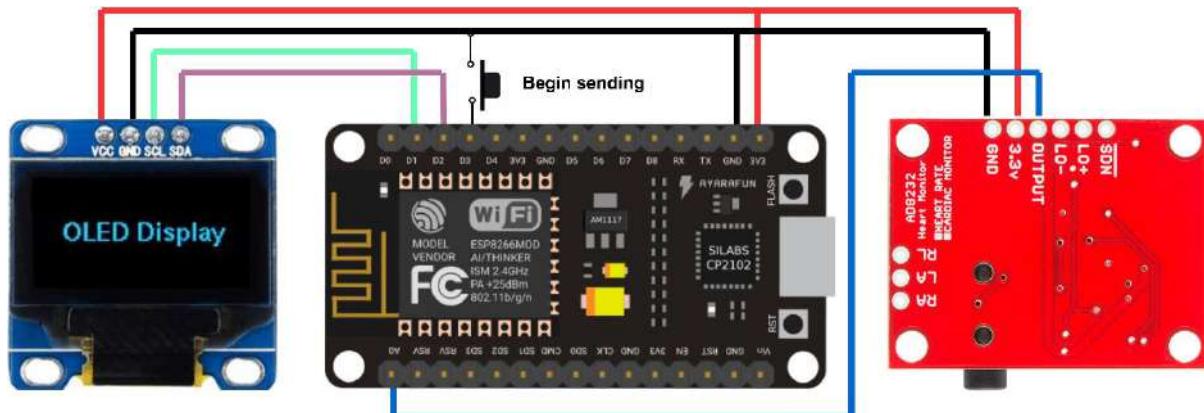


Figure 4-93 connection diagram

As shown in the previous figure, the sensor's output pin is connected to the A0 analog pin of the NodeMCU, the Adafruit display SCL and SDA are connected to D1 and D2 pins of the NodeMCU, the push button is connected to D3.

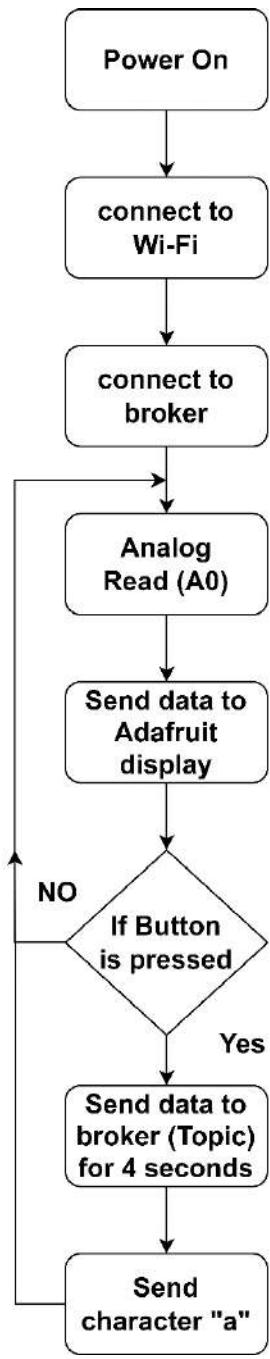


Figure 4-94 control flow of the hardware system

As shown in the previous figure, once the hardware system is powered on the NodeMCU connects to a Wi-Fi network of the following credentials

**Ssid:** MDS **Password:** 12345678

Then it connects to the broker using its IP address and then reads the analog readings on the A0 pin which is the ECG signal sent from the sensor and visualizes it on the Adafruit display, once the push button is pressed, the NodeMCU starts publishing (sending) the analog readings to the broker on a specific topic for four seconds then the broker will send these readings to the GUI which is subscribed to the same topic, when the four seconds are over the NodeMCU will send the character “a” to indicate that the sending is done.

## **Part 2: The Broker (server)**

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for efficient communication between devices in low-bandwidth, it is widely used in applications where devices need to exchange data in a publish-subscribe manner.

Overview of MQTT and how to send and receive data using it:

### **MQTT Basics:**

- Publish-Subscribe Model: MQTT follows a publish-subscribe model, devices can publish messages to topics, and other devices (subscribers) can subscribe to specific topics to receive those messages.
- Broker: MQTT communication relies on a central message broker, devices connect to the broker, publish messages to it, and subscribe to topics to receive messages from it.

### **Establishing Connection:**

- Connect to Broker: Devices establish a TCP/IP connection with an MQTT broker, typically on port 1883 (or 8883 for encrypted connections).
- Client ID: Each MQTT client must have a unique client ID to identify itself to the broker.

### **Publish Messages:**

- Topic Structure: MQTT messages are published to topics, topics are hierarchical, using forward slashes ("/") to create a topic structure for example "sensors/temperature".
- Payload: Messages can contain any binary or text data as the payload,

### **Subscribe and Receive Messages:**

- Subscribe to Topics: Devices can subscribe to specific topics or use wildcards to receive messages, for example, subscribing to "sensors/+" would match all subtopics under "sensors".
- Message Callback: Devices typically define a callback function to handle incoming messages, the callback is triggered whenever a subscribed message arrives.

### **Disconnect:**

- Properly disconnect: When a device is finished using MQTT, it should send a disconnect message to the broker and close the network connection.

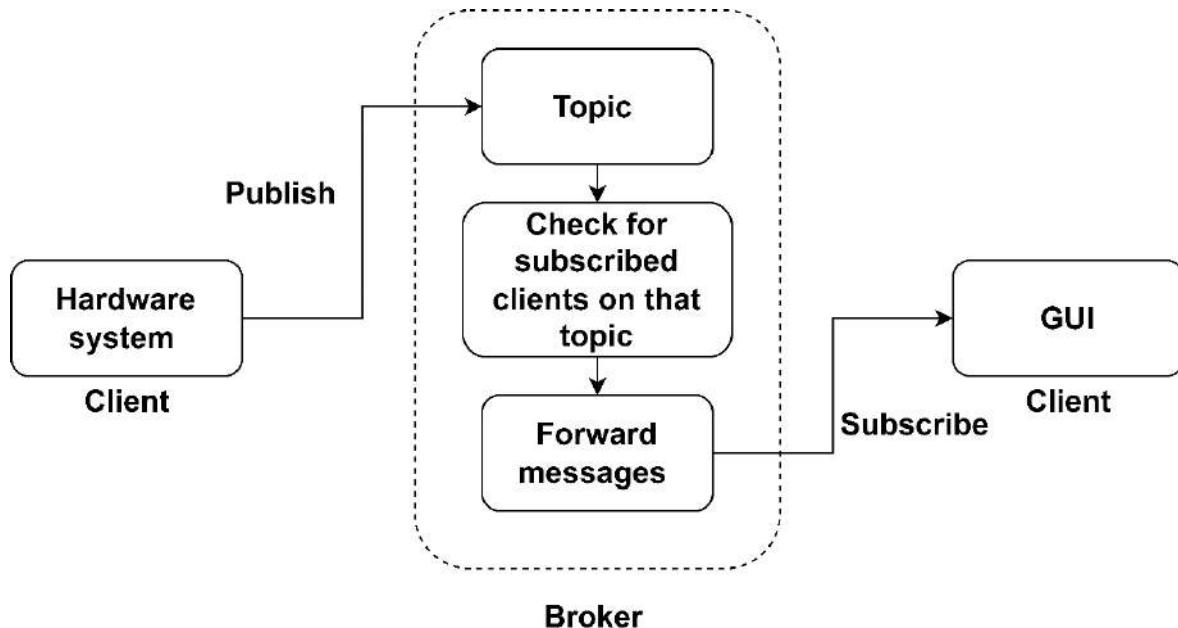


Figure 4-95 message flow through the broker

As shown in the previous figure, when the hardware system starts sending the analog readings of the sensor to the broker for 4 seconds on a specific topic which is “MDS”, the broker checks for subscribed clients on that topic and forwards these messages to them, the GUI receives these readings, visualizes them before and after the preprocessing, then passes the pre-processed signal to the AI model to make predictions on it.

### Part 3: The GUI

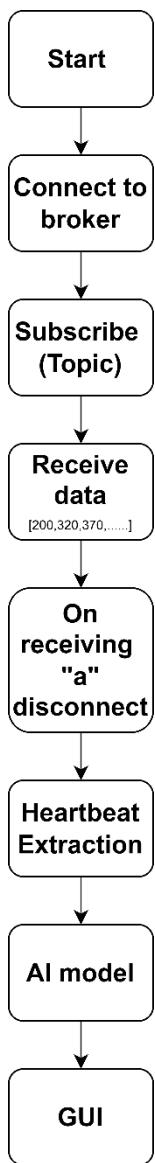


Figure 4-96 GUI flow

As shown in the previous figure, the user starts the application and navigates to the ECG service page and click a button to connect to the broker and subscribe to the previously determined topic “MDS”, the hardware system starts sending the analog readings one by one for four seconds and the GUI receives them and stores them in an array till the hardware sends a character “a” to indicate that the sending process is done so that the GUI can disconnect from the broker and starts the pre-processing on the received signal to be passed to the AI model.



Figure 4-97 snap shot of the GUI receiving the ECG signal from the hardware

The previous figure shows the sensor output using the serial plotter on the Arduino IDE and the received signal on the GUI with the pre-processed part highlighted, note that the GUI and deployment part will be discussed in details in section 4.3.

## Summary

To sum up all the previously demonstrated part, firstly the broker (server) is initialized with the same IP as the machine it's running on, then the GUI which should be running on the same machine connects to the broker and subscribes to the “MDS” topic, then the hardware is powered on and the sensor is connected to the test subject and once the pushbutton is pressed the hardware will start sending the analog reading of the sensor to the GUI for 4 seconds and after it's done it sends the character ‘a’ to indicate that the sending process is done, upon receiving ‘a’ the GUI disconnects from the broker and starts the heartbeat extraction process to prepare the received signal to be classified by the AI model.

## **4.2.2 COVID-19 detection from chest x-ray images**

The standard diagnostic method for detecting coronavirus infection is reverse transcription polymerase chain reaction (RT-PCR), the results of which can be obtained within a few hours or two days, but this detection method is not only time-consuming and expensive, but the detection process also requires close contact between the doctor and the person to be tested, so the best solution is to use the modern computer science techniques to detect covid-19 from x-rays

### **4.2.2.1 Motivation**

It's important to note that chest X-ray images alone are not sufficient to differentiate between Covid-19 and pneumonia and additional testing and clinical assessments are typically needed for a more accurate diagnosis so it can be challenging for a doctor to differentiate between Covid-19 and pneumonia from a chest X-ray image alone without examining the patient or knowing any information about their medical history or symptoms because Covid-19 and pneumonia can cause similar changes in the lungs that can be seen on a chest X-ray, there are some subtle differences that a trained radiologist or physician may be able to identify. However, the accuracy of the diagnosis would be greatly improved if additional information such as the patient's medical history, clinical symptoms, and other diagnostic tests (such as a PCR test for Covid-19) were available.

AI models can contribute to differentiating between Covid-19 and pneumonia from chest X-ray images by processing the pixel values in the image and using a convolutional neural network to identify patterns and features that are indicative of COVID-19 or pneumonia

### **4.2.2.2 Dataset description**

COVID19-Pneumonia-Normal-chest-xray-pa-dataset is a public dataset published on Kaggle which contains 3 classes (Normal, Pneumonia, COVID19), this datasets has 6939 total samples with an balanced number of samples for each class, it has 2313 sample for each class, all samples are chest x-ray images collected from different sources including the Italian Society of Radiology (SIRM), the Radiopaedia database and Figshare data repository.

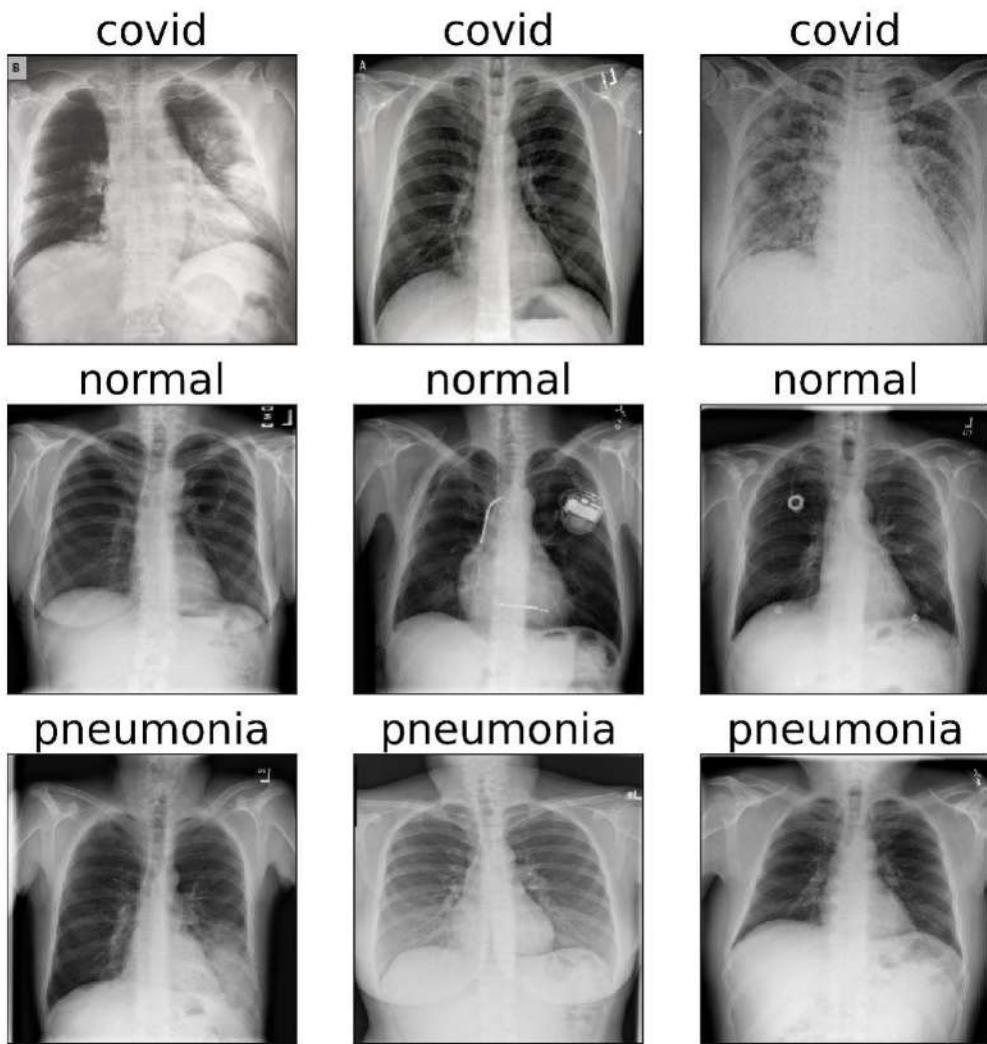


Figure 4-98 samples from COVID19-Pneumonia-Normal-chest-xray-pa-dataset

## Data source

This dataset was collected from multiple sources such as

**Radiopedia.org:** Radiopaedia.org is a free, open-access, peer-reviewed, and collaborative radiology resource, it was founded in 2005 by a group of radiologists and radiology trainees from around the world, multiple medical organizations recognize the value of Radiopaedia.org as a resource for radiology education and practice, they support the site by providing financial and in-kind contributions, as well as by promoting Radiopaedia.org to their members and the wider medical community.

**Società Italiana di Radiologia Medica e Interventistica (SIRM):** which is the Italian Society of Medical and Interventional Radiology, it is a non-profit organization that was founded in 1913, SIRM has over 10,000 members, including radiologists, radiology trainees, and other healthcare professionals, SIRM's mission is to promote the highest standards of radiology education, research, and practice in Italy.

#### 4.2.2.3 Reference papers for result comparison

Results of experiments which will be performed in the project will be compared with [44] in which the author proposed three pre-trained models and combined their output using confidence fusion method achieving an accuracy of 95.56%

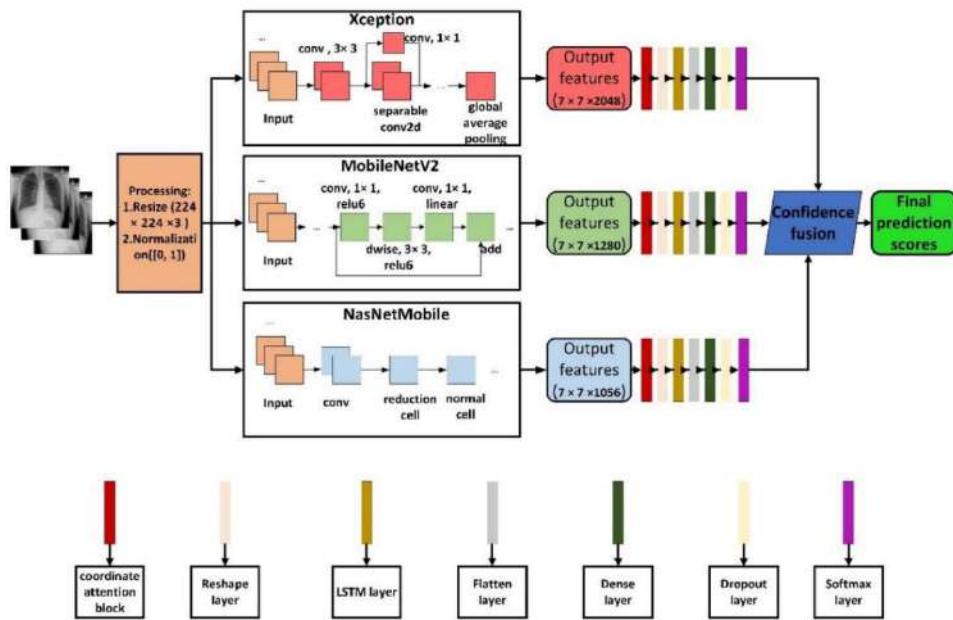


Figure 4-99 [44] proposed model architecture

In [76] the author utilized the EfficientNet-b0 pre-trained model with the Linear Discriminant Analysis (LDA) classifier achieving an accuracy of 94.7%.

#### 4.2.2.4 Development methodology

The suitable type of neural networks for this dataset is 2-D convolution neural networks as it is more useful with images due to the feature extraction process of the convolution layers, different pre-trained models could be experimented as samples of this dataset are in RGB color mode.

#### 4.2.2.5 2-D convolution methodology

- **Version 1**

The VGG-16 pre-trained model was utilized in this version

#### Data preprocessing

300 sample was taking as testing set and the rest as a training set with 0.2 validation split all images were resized to 224\*224 and normalized.

## Proposed framework

As shown in Fig. 24, Vgg-16 pre-trained model with pooling layers set to average pooling followed by a dropout layer of 0.2 followed by a customized classification top consisting of three Dense layers to match the problem, relu activation function is used in all the hidden Dense layers and softmax is used in the output layer.

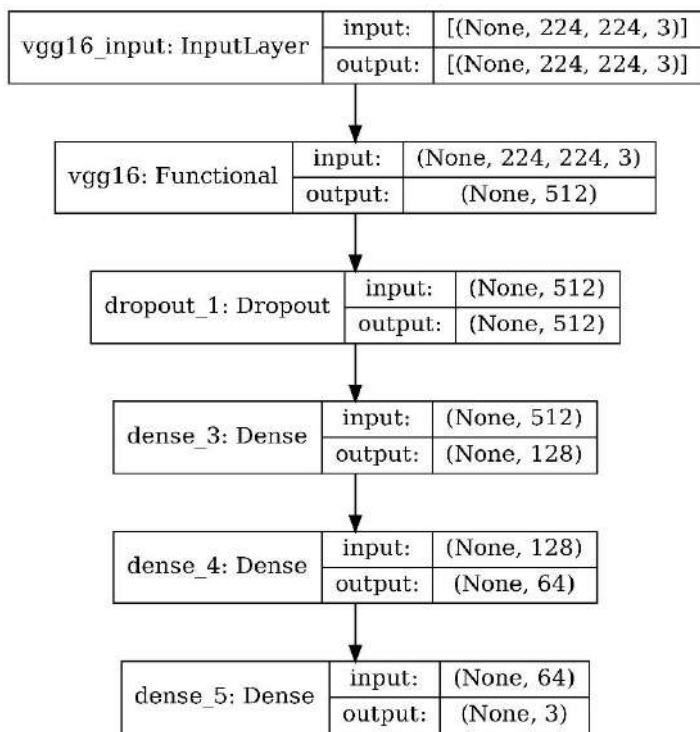


Figure 4-100 version 1 proposed network architecture

## Result

Table 4-89 Results of version 1

	Train acc	Val acc	Test acc
proposed	98.69%	94.83%	93.86%
[44]			95.56%

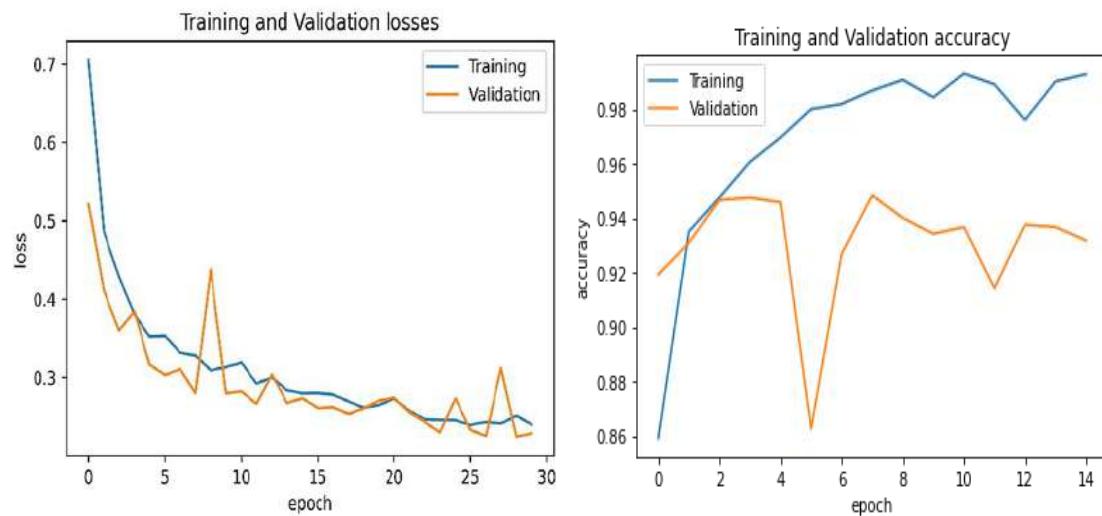


Figure 4-101 training accuracy and loss curves for version 1

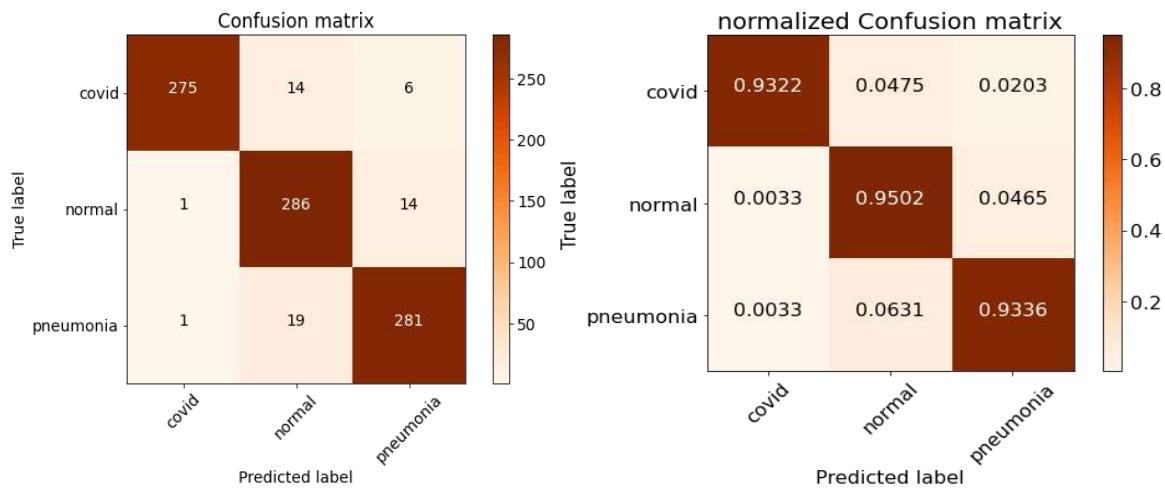


Figure 4-102 confusion matrices of version 1

Table 4-90 classification report of version 1

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.96	0.94	0.95	295
1	Normal	0.89	0.91	0.90	301
2	Pneumonia	0.93	0.93	0.93	301
<hr/>					
accuracy				0.94	897
macro avg	0.94	0.94	0.94	0.94	897
weighted avg	0.94	0.94	0.94	0.94	897

## Version 1 conclusion

This model is not the best model for this classification problem, higher accuracy could be achieved using other pre-trained models.

## • Version 2

The VGG-19 pre-trained model was utilized in this version

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized

## Proposed framework

As shown the Figure, Vgg-16 pre-trained model with pooling layers set to average pooling followed by a max-pooling layer then a flatten layer then a Dense layer to match the classification problem, softmax activation function is used in the output layer.

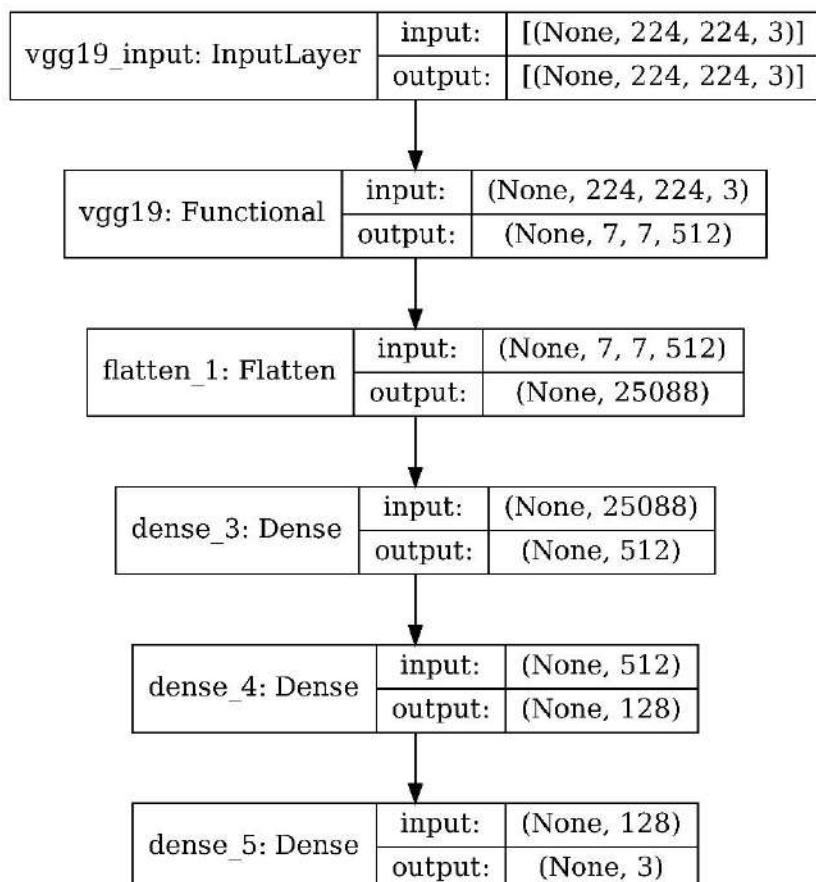


Figure 4-103 version 2 proposed network architecture

## Result

Table 4-91 Results of version 2

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	95.8751%	95.9821%	95.8751%	0.9797
[44]	95.56%			

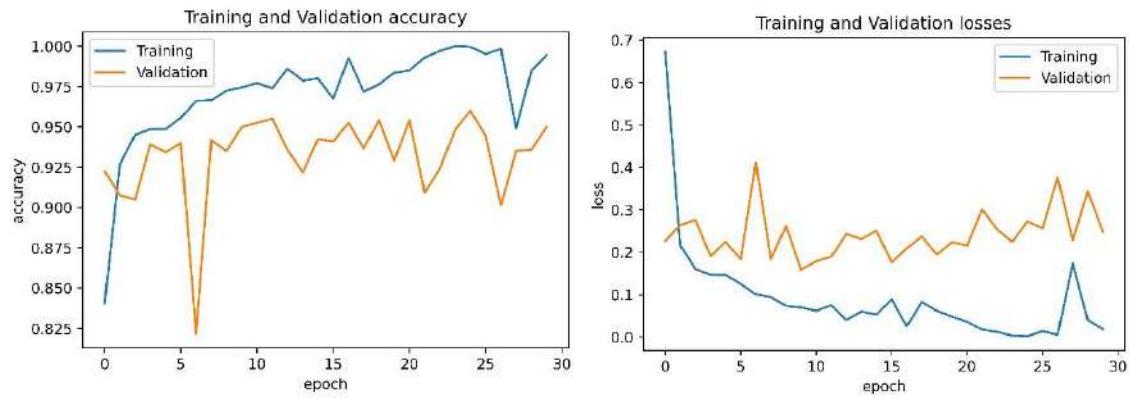


Figure 4-104 training accuracy and loss curves for version 2

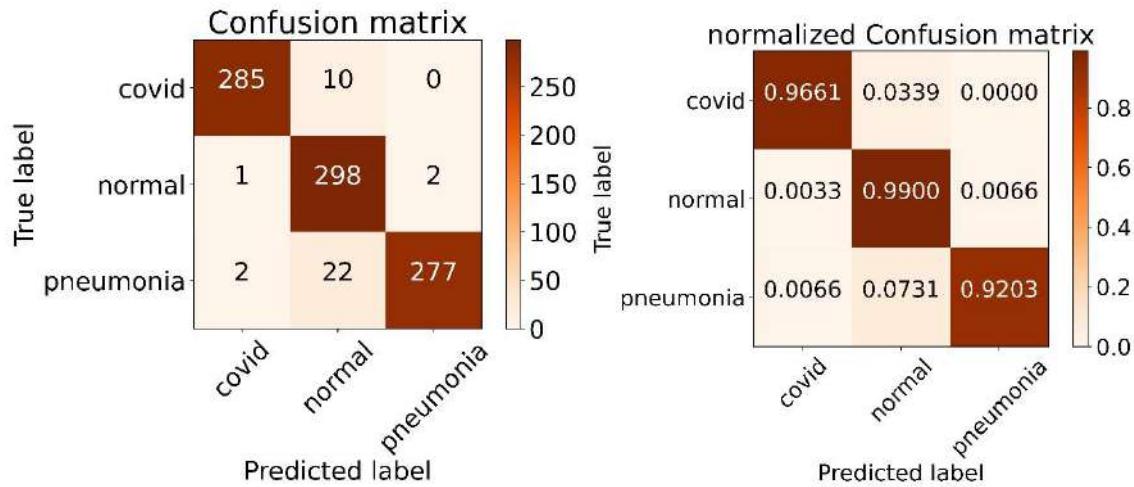


Figure 4-105 confusion matrices of version 2

Table 4-92 classification report of version 1

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9896	0.9661	0.9777	295
1	Normal	0.903	0.99	0.9445	301
2	pneumonia	0.9928	0.9203	0.9552	301
accuracy				0.9588	897
macro avg		0.9618	0.9588	0.9591	897
weighted avg		0.9616	0.9588	0.9590	897

## Version 2 conclusion

Increasing the depth of the model help improve the results.

## Version 3

In this version, ResNet50 pre-trained model was utilized.

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

### Proposed framework

Resnet50 pre-trained model was used with a customized classification top to match our classification problem, relu activation function was used in all hidden Dense layers and softmax activation function was used in output layer.

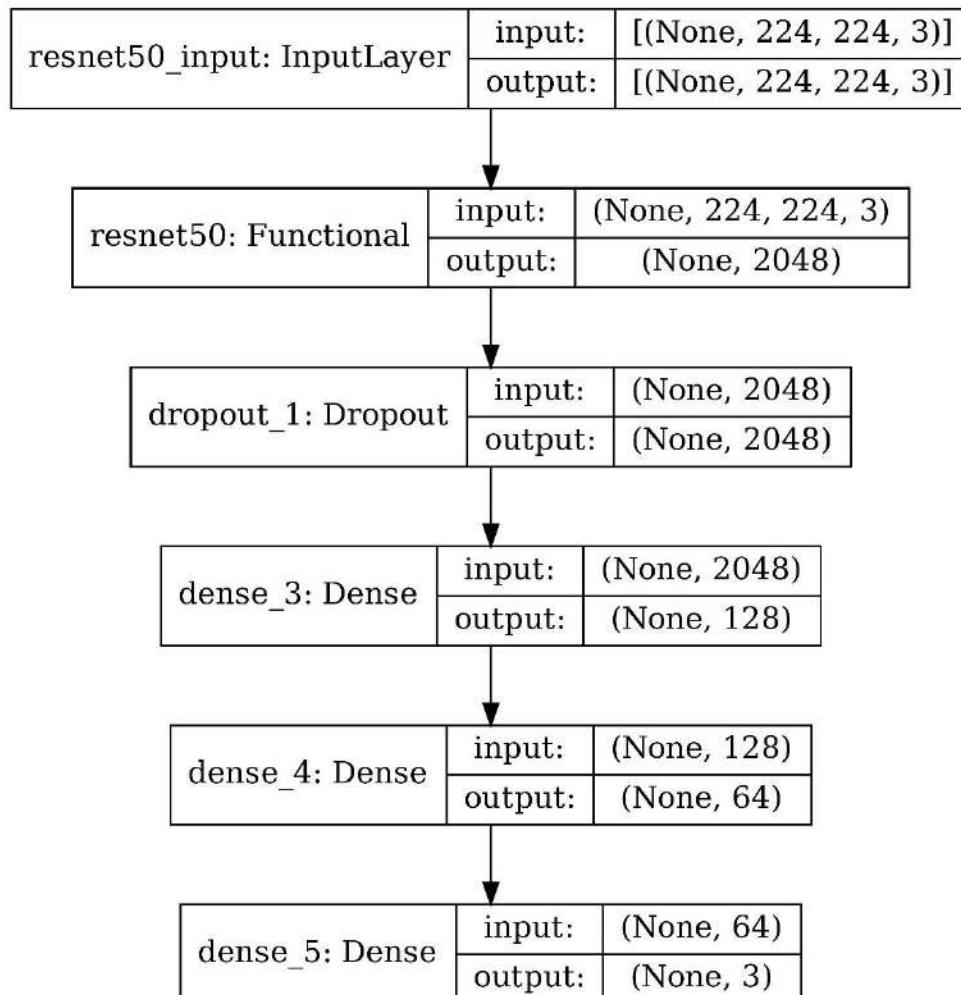


Figure 4-106 version 3 proposed network architecture

## Result

Table 4-93 results of version 3

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	82.6087%	84.5972%	79.5987%	0.9408
[44]	95.56%			

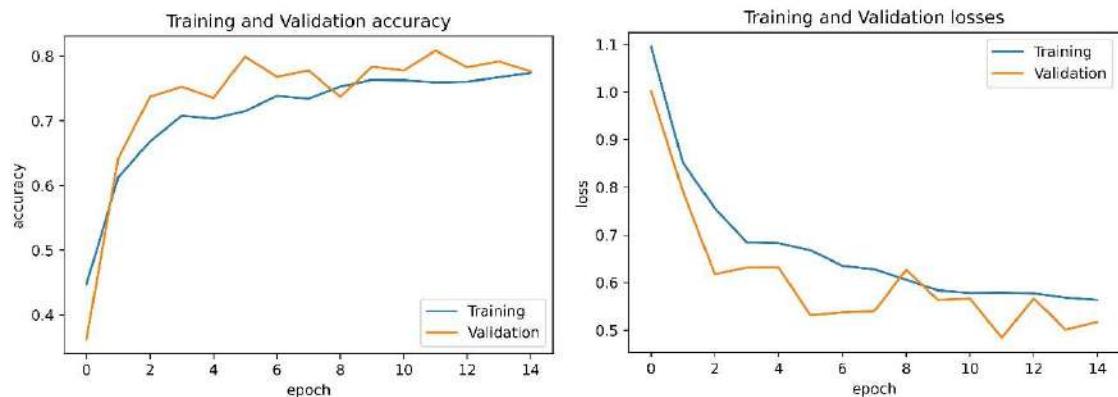


Figure 4-107 version 3 training accuracy and loss curves

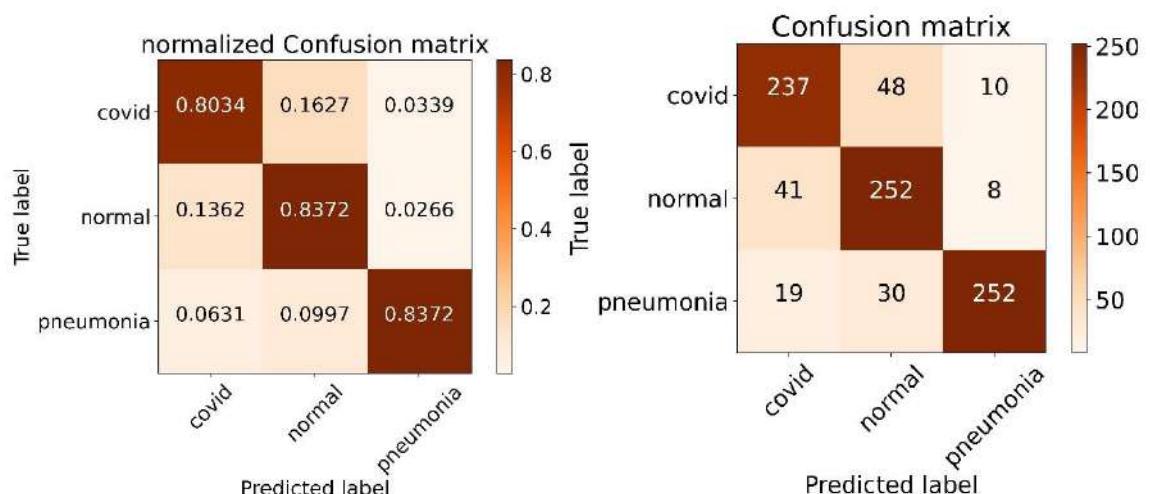


Figure 4-108 version 3 confusion matrices

Table 4-94 version 3 classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.8	0.8	0.8	295
1	Normal	0.76	0.84	0.8	301
2	pneumonia	0.93	0.84	0.88	301
accuracy				0.83	897
macro avg	0.83	0.83	0.83	0.83	897
weighted avg	0.83	0.83	0.83	0.83	897

## Version 4

In this version, DeneseNet121 pre-trained model was utilized.

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

### Proposed framework

DeneseNet121 pre-trained model was used with a customized classification top to match our classification problem, relu activation function was used in all hidden Dense layers and softmax activation function was used in output layer.

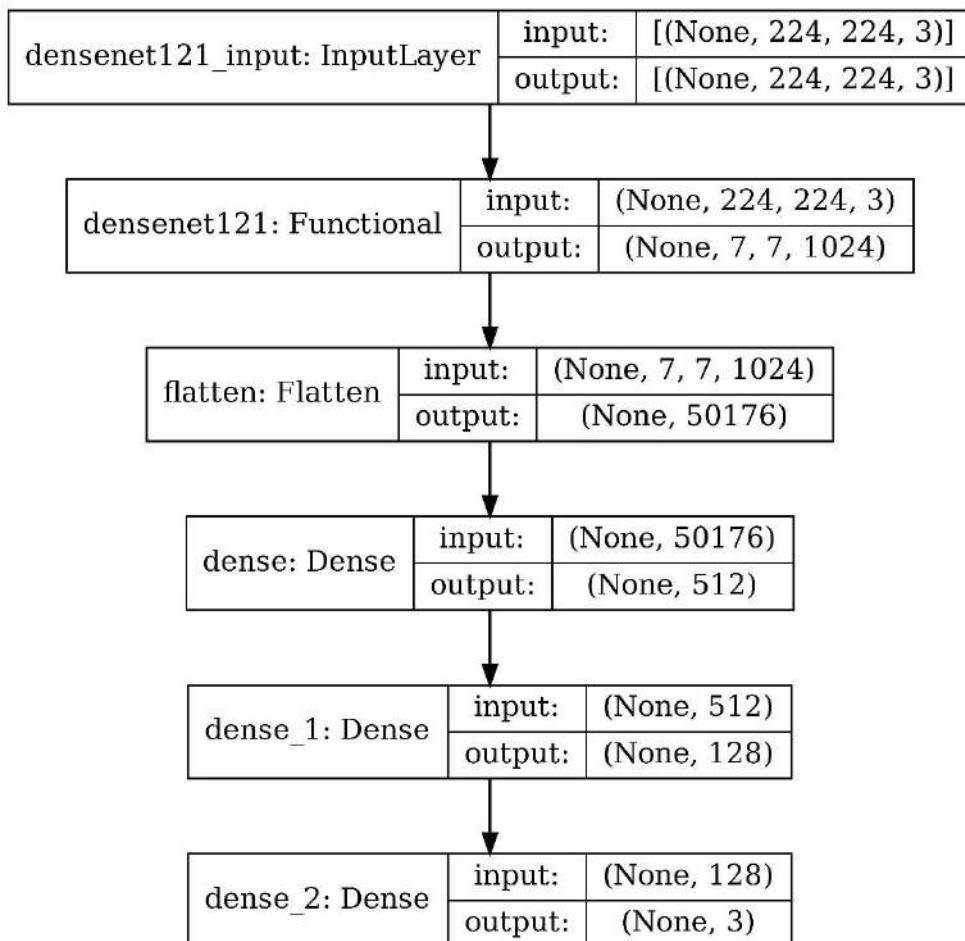


Figure 4-109 version 4 proposed network architecture

## Result

Table 4-95 Result of version 4

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	93.5340%	93.8272%	93.1996%	0.9853
[44]	95.56%			

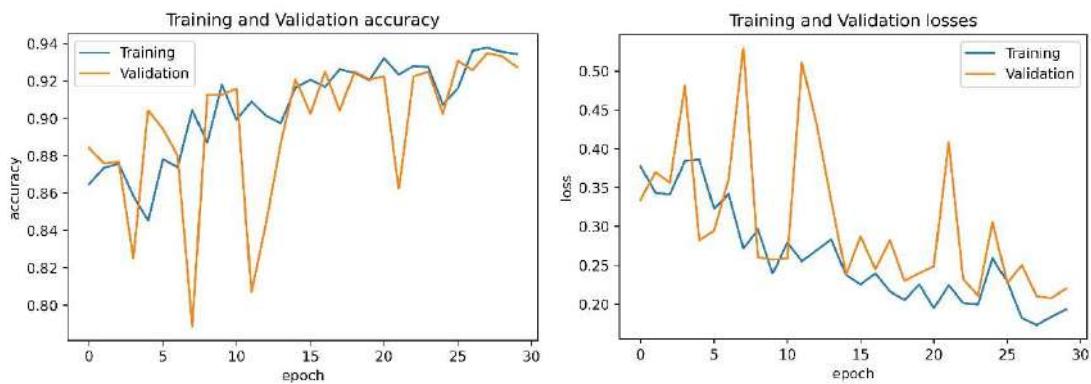


Figure 4-110 version 4 training accuracy and loss curves

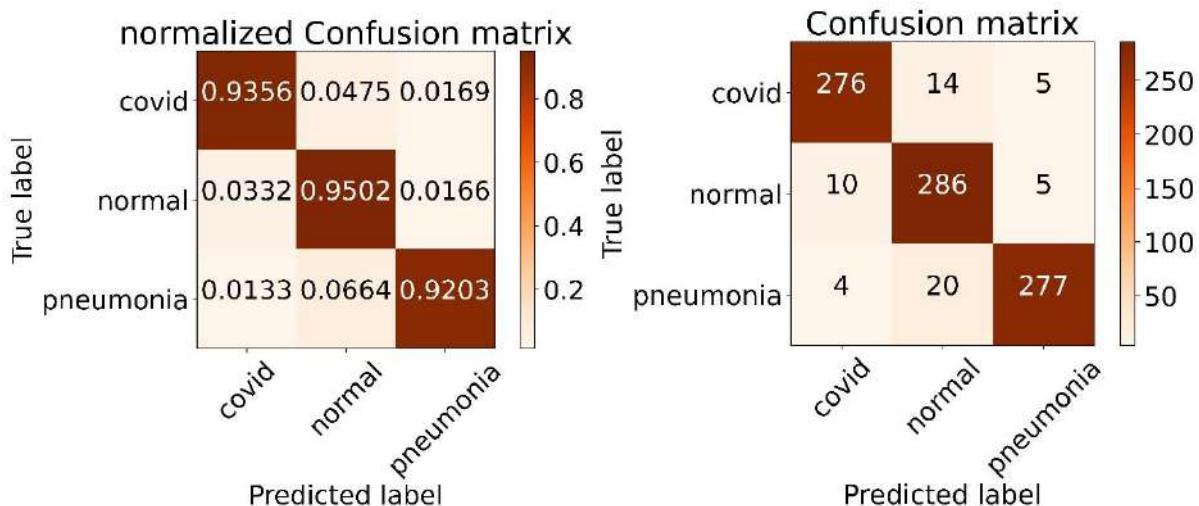


Figure 4-111 version 4 confusion matrices

Table 4-96 version 4 classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9517	0.9356	0.9436	295
1	Normal	0.8938	0.9502	0.9211	301
2	pneumonia	0.9652	0.9203	0.9422	301
accuracy				0.9353	897
macro avg		0.9369	0.9353	0.9356	897
weighted avg		0.9369	0.9353	0.9356	897

## Version 5

In this version, MobileNet pre-trained model was utilized.

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

### Proposed framework

MobileNet pre-trained model was used with a customized classification top to match our classification problem, relu activation function was used in all hidden Dense layers and softmax activation function was used in output layer.

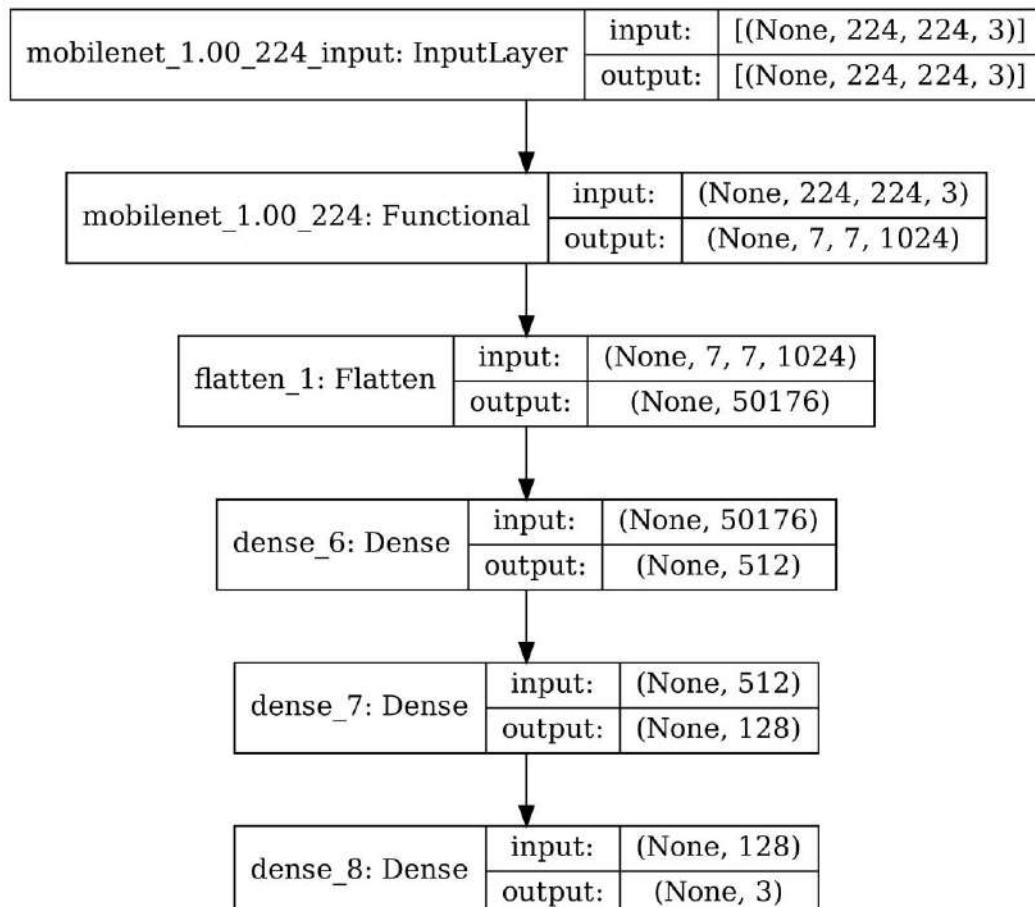


Figure 4-112 version 5 proposed network architecture

## Result

Table 4-97 Results of version 5

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	90.8584%	91.3386%	90.5240%	0.9777
[44]	95.56%			

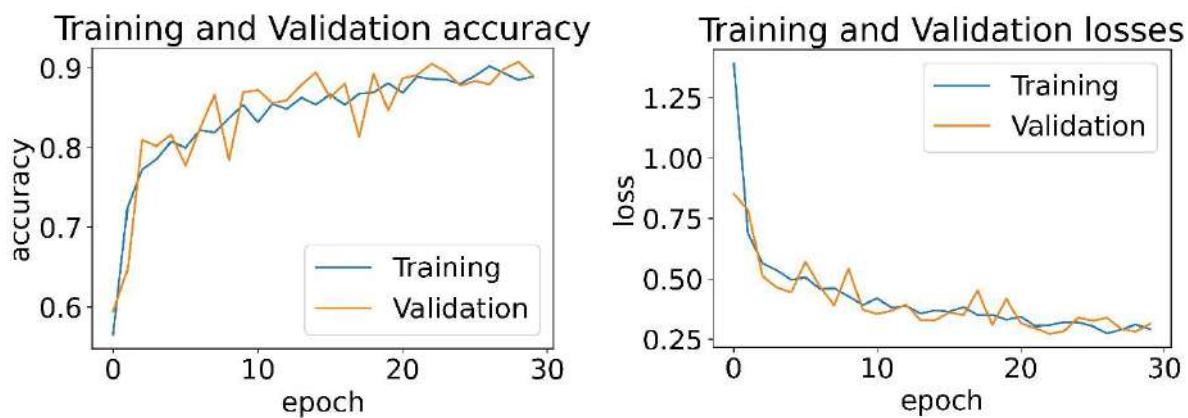


Figure 4-113 version 5 training accuracy and loss curves

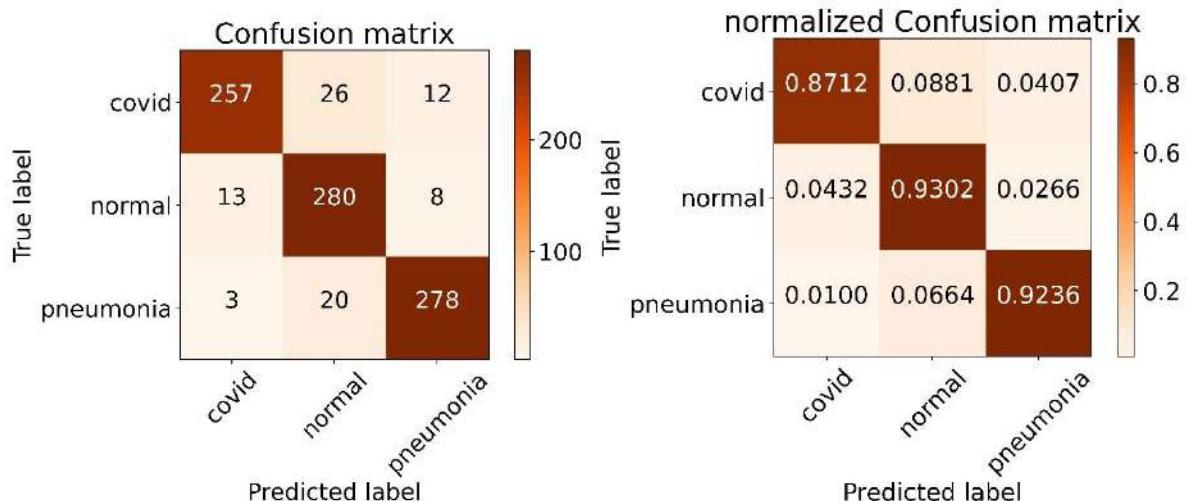


Figure 4-114 version 5 confusion matrices

Table 4-98 version 5 classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9414	0.8712	0.9049	295
1	Normal	0.8589	0.9302	0.8931	301
2	pneumonia	0.9329	0.9236	0.9282	301
accuracy				0.9086	897
macro avg		0.9111	0.9083	0.9088	897
weighted avg		0.9109	0.9086	0.9088	897

## Version 6

In this version, Xception pre-trained model was utilized.

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

### Proposed framework

Xception pre-trained model was used with a customized classification top to match our classification problem, relu activation function was used in all hidden Dense layers and softmax activation function was used in output layer.

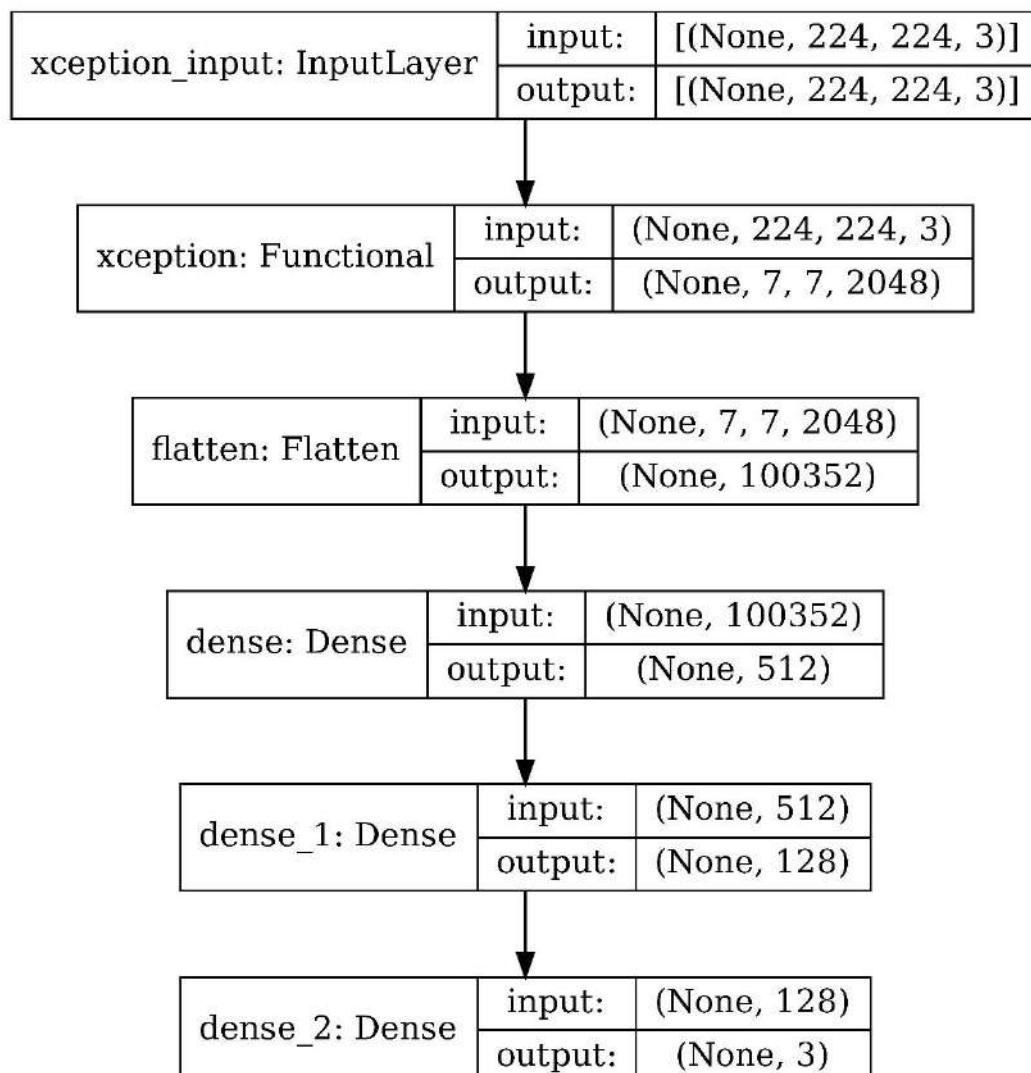


Figure 4-115 version 6 proposed network architecture

## Result

Table 4-99 results of version 6

	Test accuracy	Test precision	Test Recall	Test AUC
Proposed	93.9799%	93.9799%	93.9799%	0.9725
[44]	95.56%			

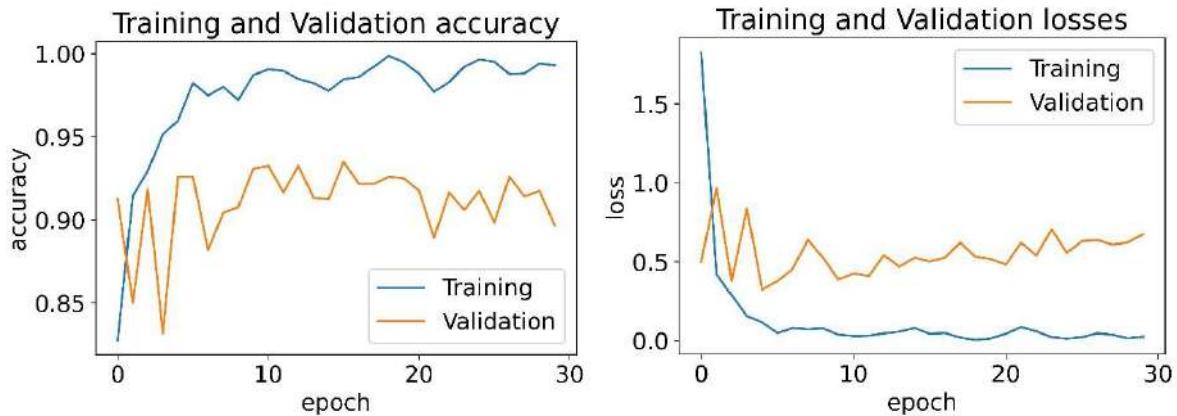


Figure 4-116 version 6 training accuracy and loss curves

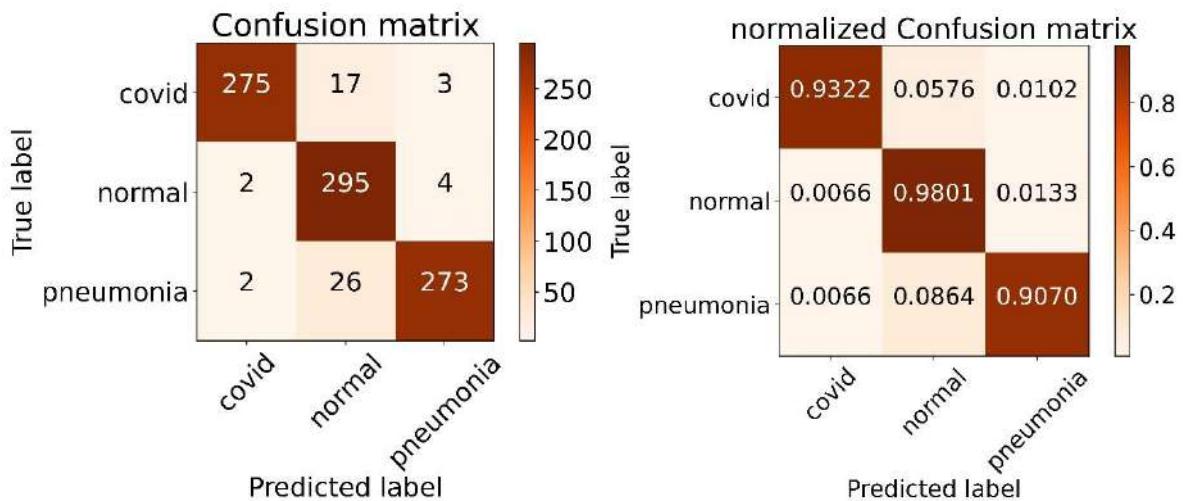
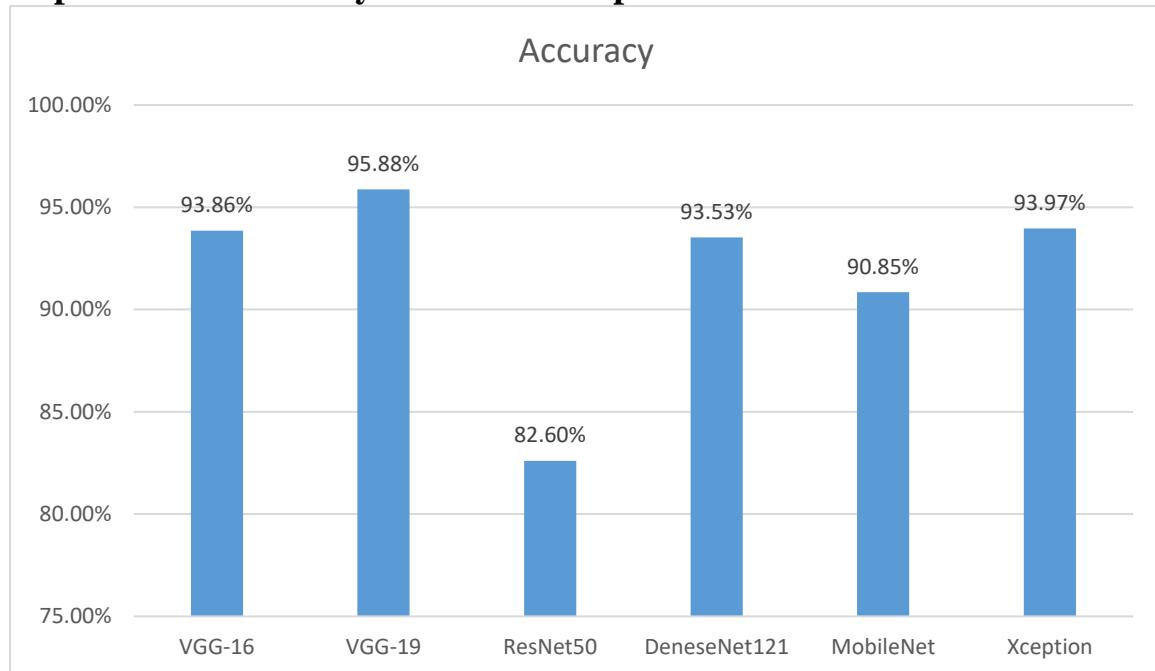


Figure 4-117 version 6 confusion matrices

Table 4-100 version 6 classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9857	0.9322	0.9582	295
1	Normal	0.8728	0.9801	0.9233	301
2	pneumonia	0.975	0.907	0.9398	301
accuracy				0.9398	897
macro avg		0.9445	0.9397	0.9404	897
weighted avg		0.9442	0.9398	0.9403	897

## Experiments summary with different pre-trained models



## Experiment on version 2

All images was converted to grayscale and passed throw a convolution layer which outputs 3 channel of the same input size to be compatible with the VGG-19, the reason for this is to test the model performance on the gray scale images as it is easier to apply image enhancement techniques on gray images.

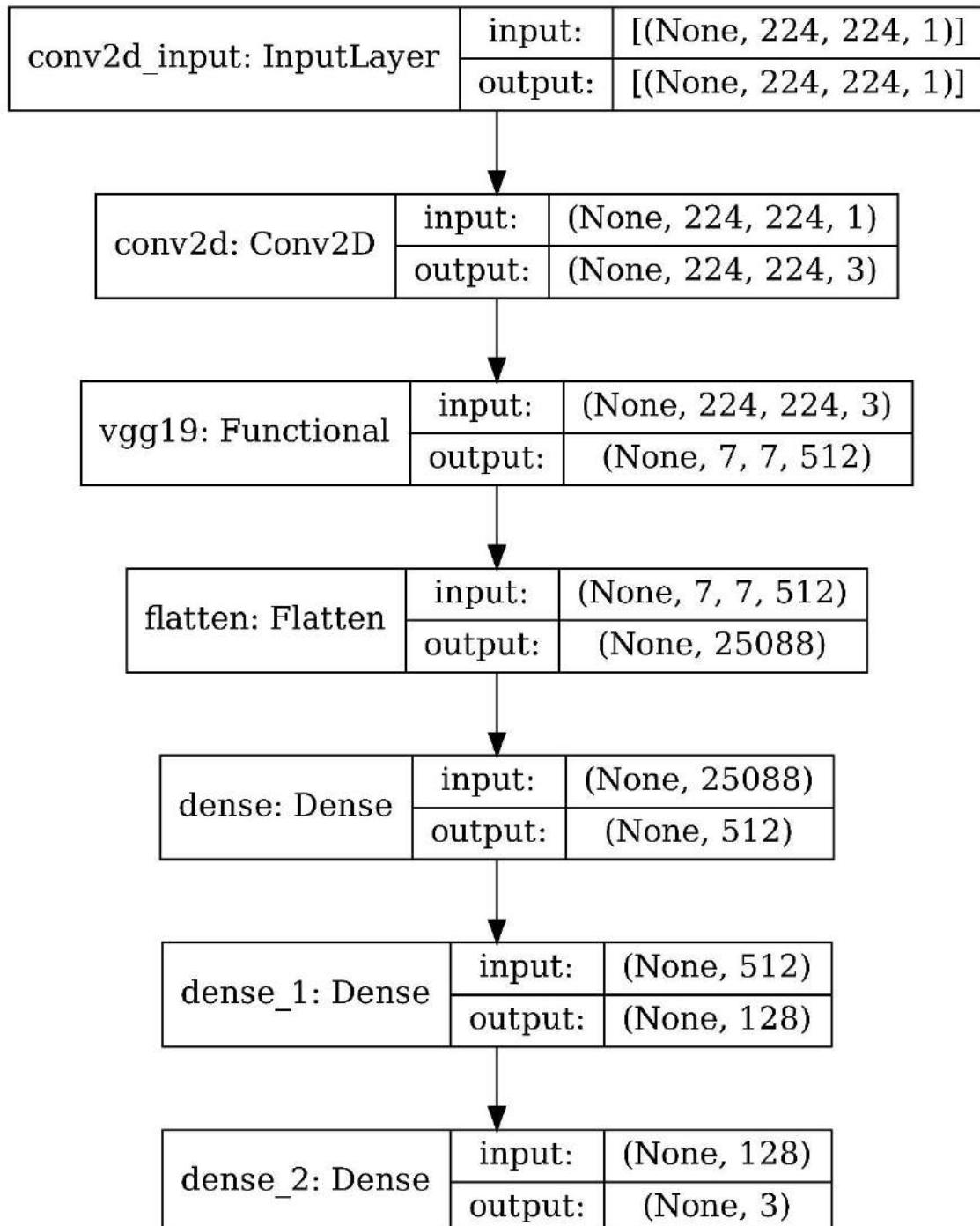


Figure 4-118 version 2 EXP. (3) proposed network architecture

Table 4-101 version 2 EXP. (3) results

	Test accuracy	Test precision	Test Recall	Test AUC
VGG-19 (gray images)	95.54%	95.86%	95.54%	0.9889
VGG-19	95.87%	95.98%	95.87%	0.9797
[44]	95.56%			

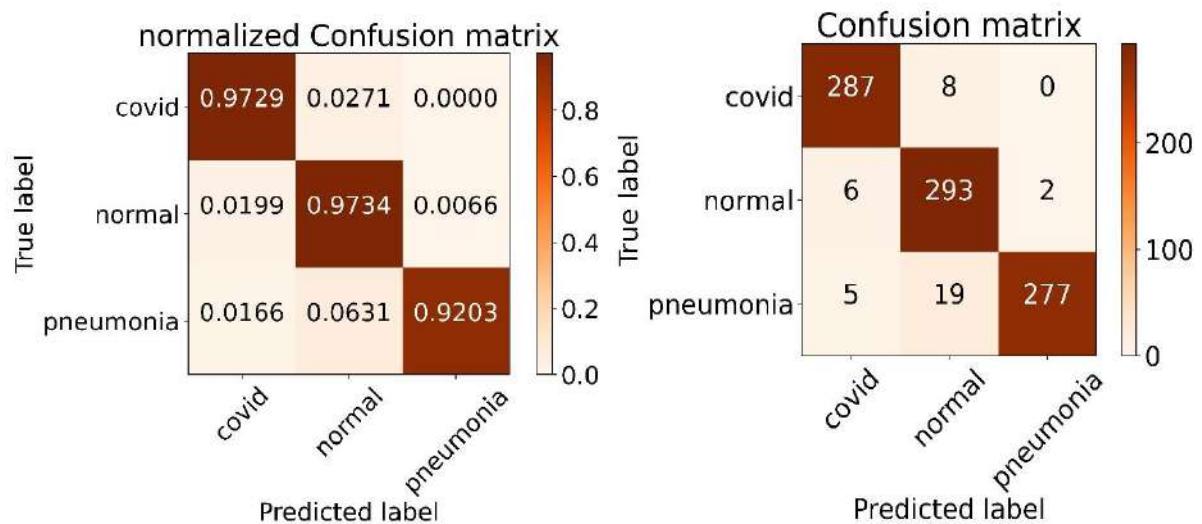


Figure 4-119 EXP (3) confusion matrices

## Version 7

In this version a multi-head attention (MHA) layer was utilized with the VGG-19 model, as the input of the MHA layer is in the shape of (7, 7, 512), the suitable number of heads will be 1, 7

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

### Proposed framework

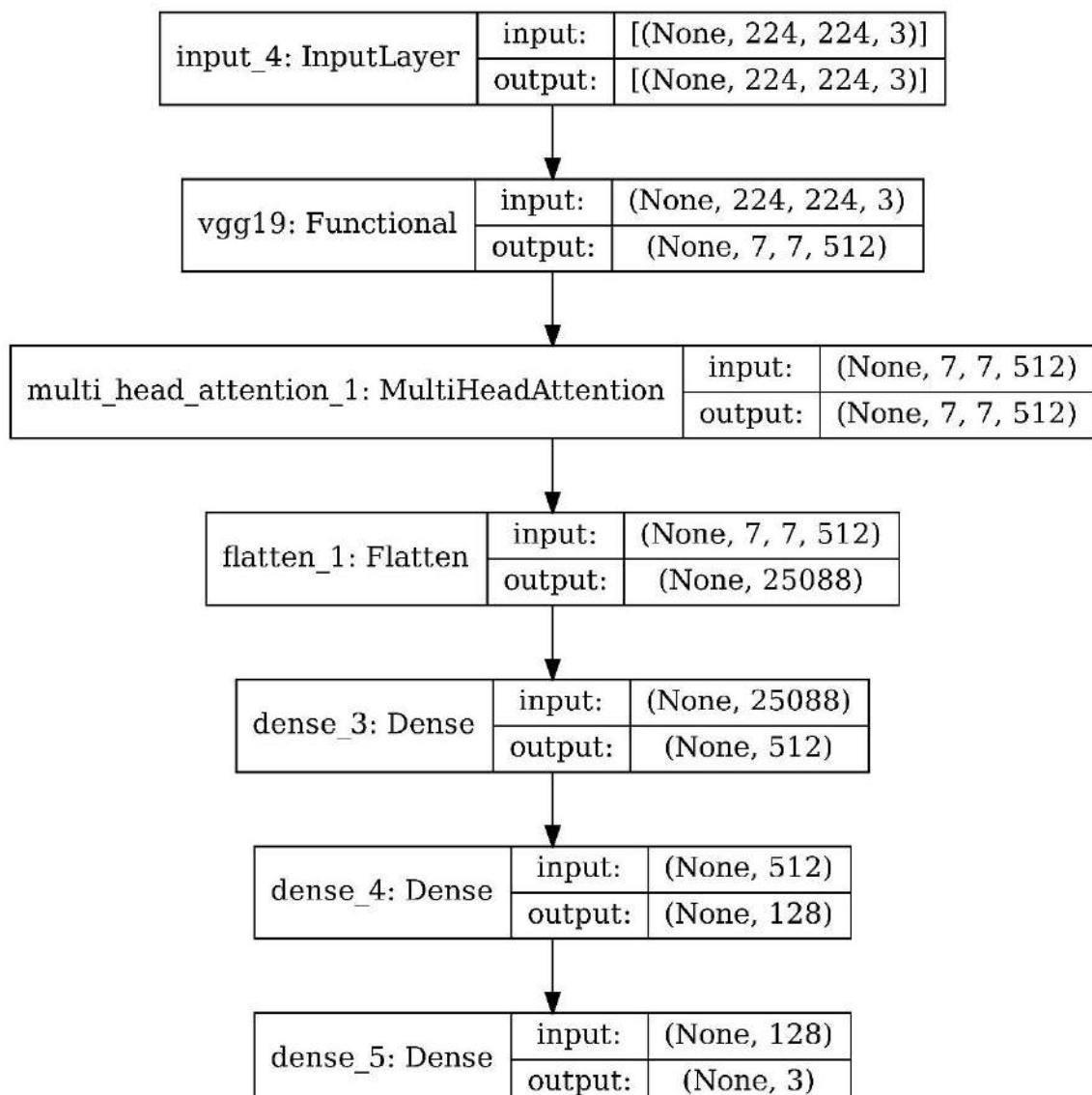


Figure 4-120 version 7 proposed network architecture

## EXP. (1)

Number of heads = 1

Table 4-102 testing results for version 7 EXP (1)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	92.53%	93.03%	92.31%	92.53%	0.9815	92.53%	88.80%
[44]	95.56%						

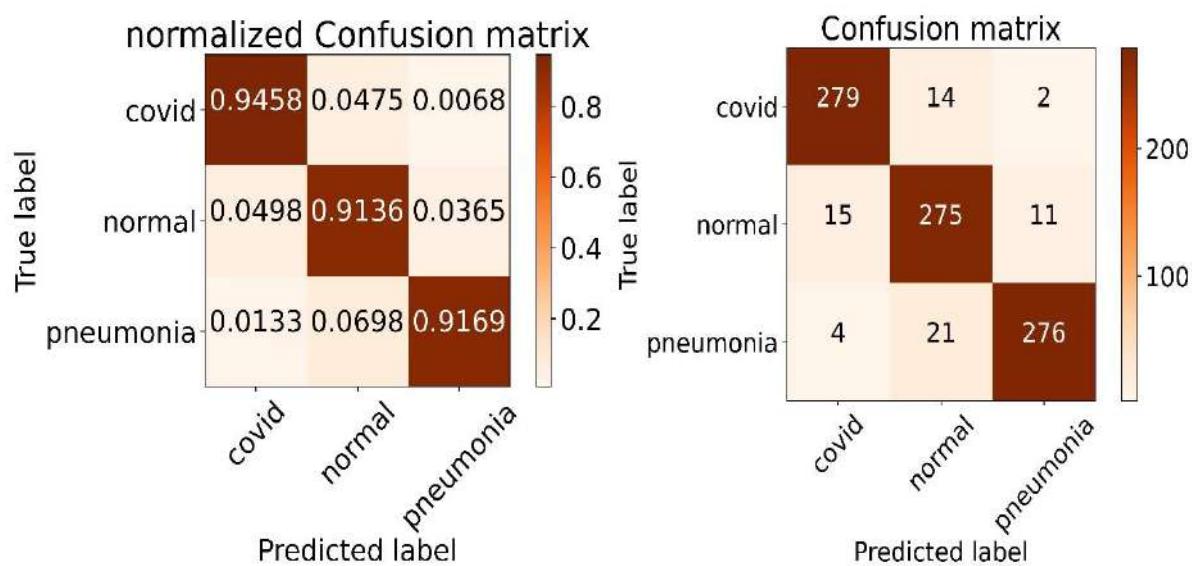


Figure 4-121 version 7 EXP (1) confusion matrices

Table 4-103 version 7 EXP (1) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9362	0.9458	0.941	295
1	Normal	0.8871	0.9136	0.9002	301
2	pneumonia	0.955	0.9169	0.9356	301
accuracy				0.9253	897
macro avg		0.9261	0.9254	0.9256	897
weighted avg		0.9261	0.9253	0.9255	897

## EXP. (2)

Number of heads = 7

Table 4-104 testing results for version 7 EXP (2)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	94.09%	94.17%	93.65%	94.09%	0.9857	94.08%	91.14%
[44]	95.56%						

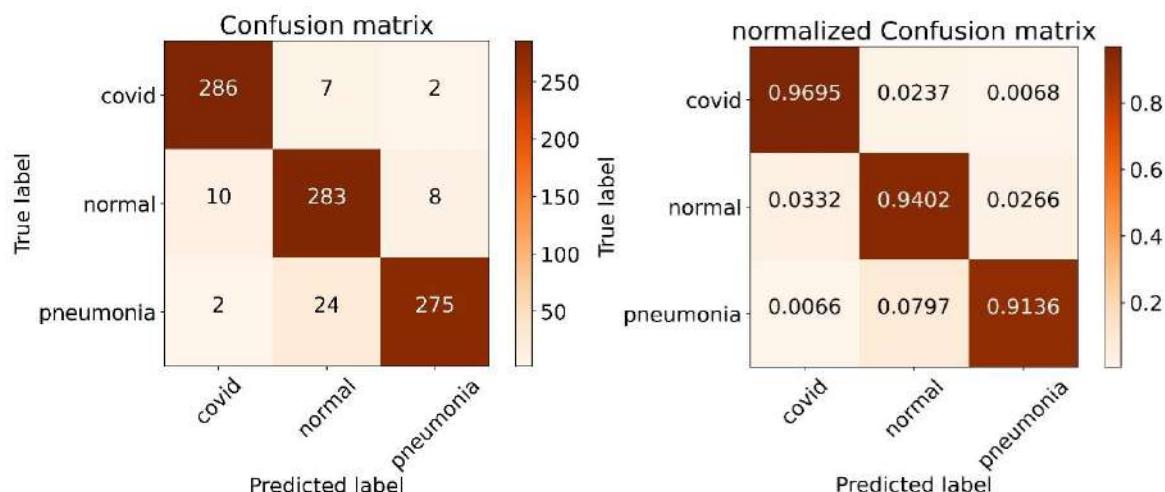


Figure 4-122 version 7 EXP (2) confusion matrices

Table 4-105 version 7 EXP (2) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9597	0.9695	0.9646	295
1	Normal	0.9013	0.9402	0.9203	301
2	pneumonia	0.9649	0.9136	0.9386	301
accuracy				0.9409	897
macro avg		0.9420	0.9411	0.9412	897
weighted avg		0.9419	0.9409	0.9410	897

### EXP. (3)

Number of heads = 1-7

Table 4-106 testing results for version 7 EXP (3)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	92.08%	92.18%	91.97%	92.08%	0.9724	92.09%	88.13%
[44]	95.56%						

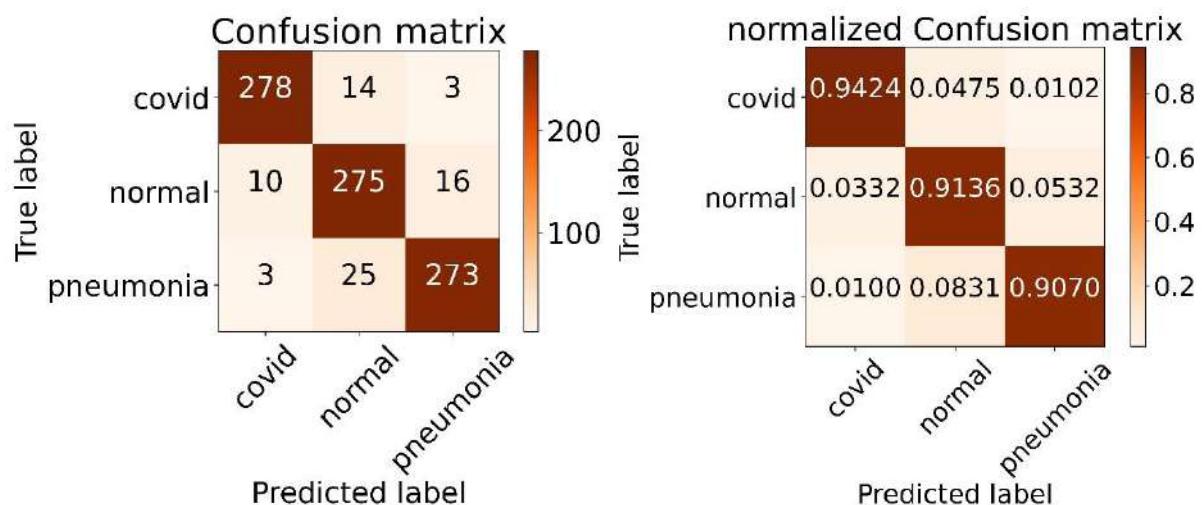


Figure 4-123 version 7 EXP (3) confusion matrices

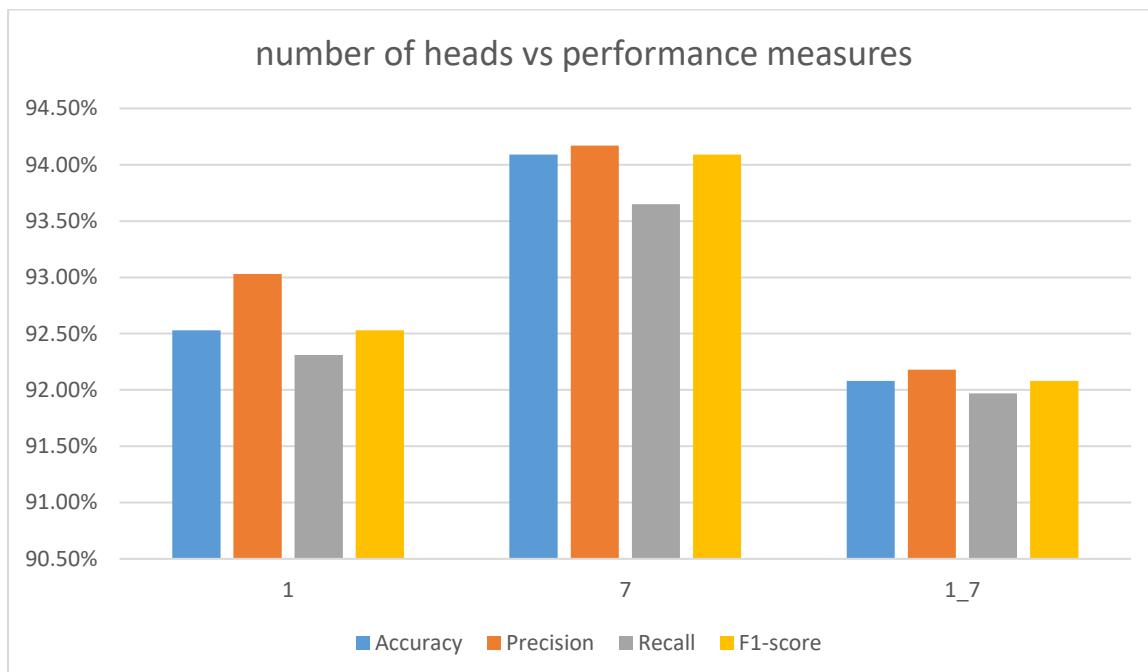
Table 4-107 version 7 EXP (2) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9553	0.9424	0.9488	295
1	Normal	0.8758	0.9136	0.8943	301
2	pneumonia	0.9349	0.907	0.9207	301
accuracy				0.9208	897
macro avg		0.9220	0.9210	0.9213	897
weighted avg		0.9218	0.9208	0.9211	897

## Tuning the number of heads of the MHA layer

Table 4-108 tuning the number of heads

No. of heads	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
<b>Single multi-head attention branch</b>							
1	92.53%	93.03%	92.31%	92.53%	0.9815	92.53%	88.80%
7	94.09%	94.17%	93.65%	94.09%	0.9857	94.08%	91.14%
<b>Parallel multi-head attention branches</b>							
1_7	92.08%	92.18%	91.97%	92.08%	0.9724	92.09%	88.13%



The numbers under each 4 bars represent the number of heads used in that experiment, 1\_7 indicates that a parallel branch was used one with 1 head and the other with 7 heads,

## Version 8

In this version an adaptive feature recalibration (AFR) layer was utilized after the VGG-19 model, the AFR layer has a parameter that needs tuning which is the reduction ratio, a good range for it is between 4 and 16.

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

### Proposed framework

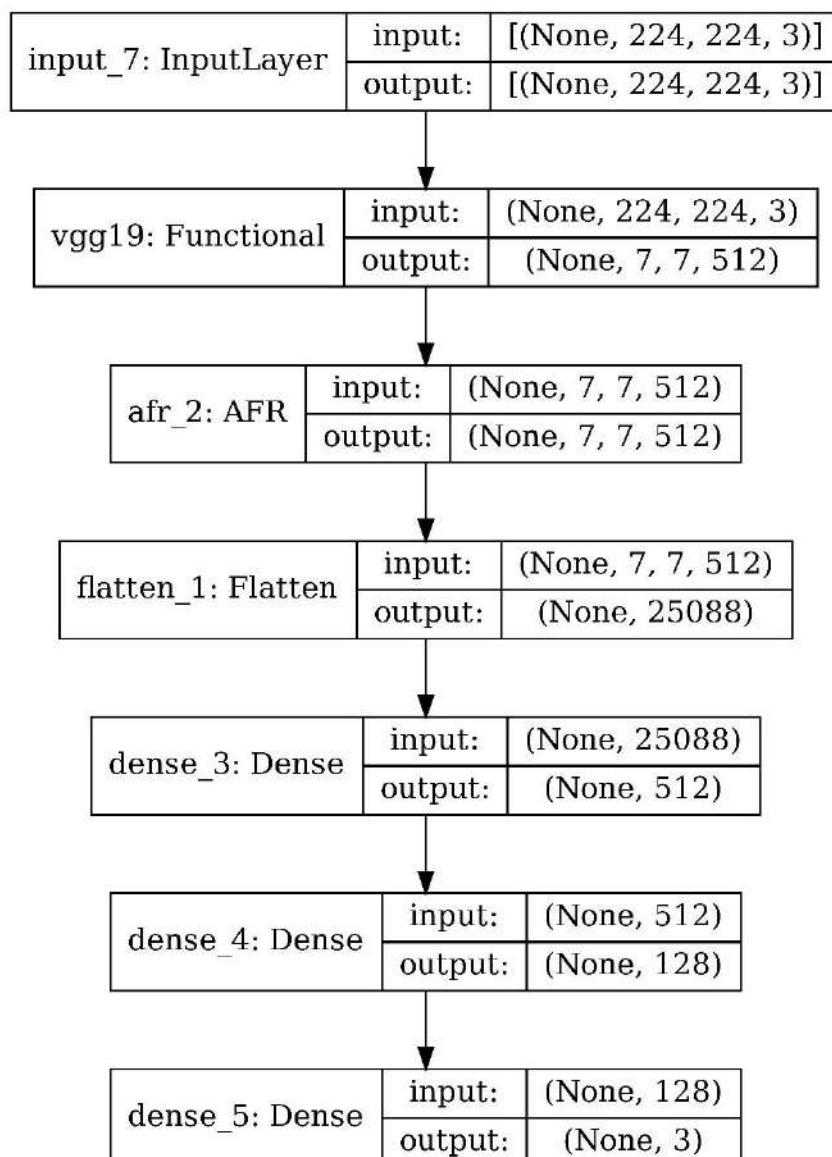


Figure 4-124 version 8 proposed network architecture

## EXP. (1)

Reduction ratio = 4

Table 4-109 testing results for version 8 EXP (1)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	95.99%	95.99%	95.99%	95.99%	0.9799	95.95%	93.98%
[44]	95.56%						

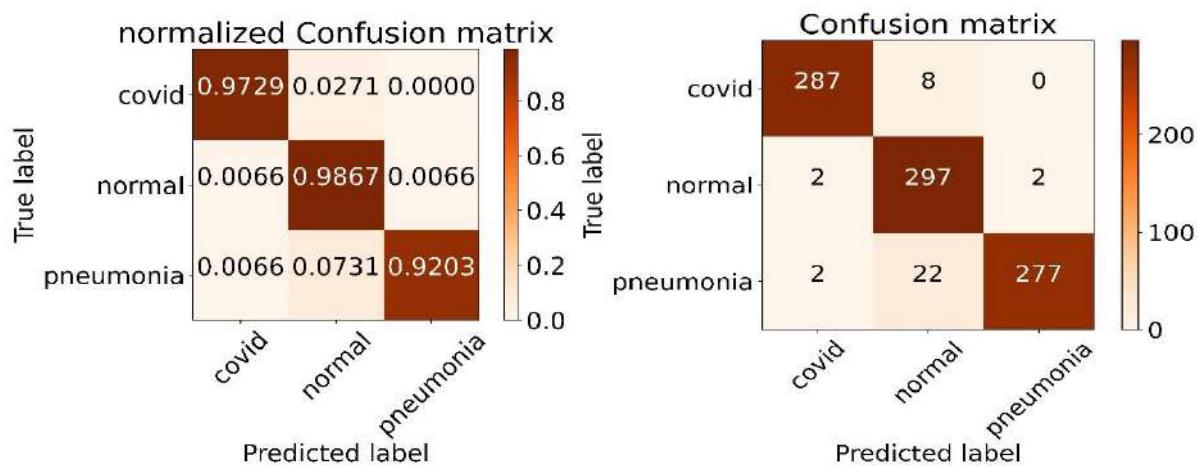


Figure 4-125 version 8 EXP (1) confusion matrices

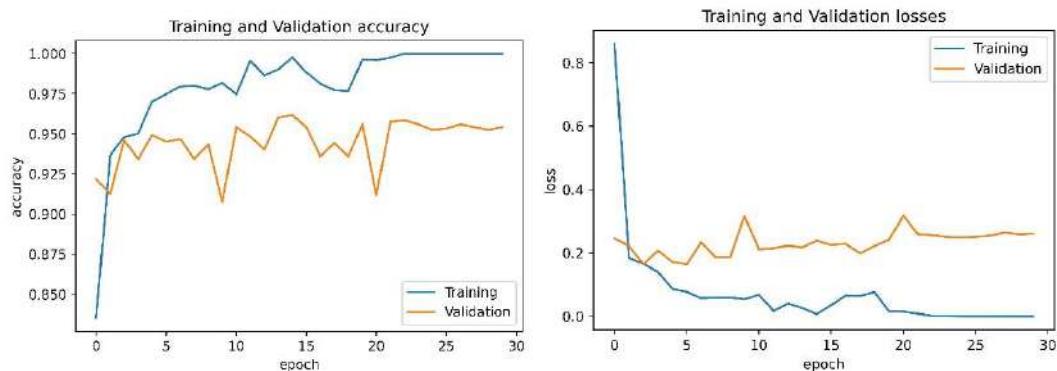


Figure 4-126 training accuracy and loss curves

Table 4-110 version 8 EXP (1) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9863	0.9729	0.9795	295
1	Normal	0.9083	0.9867	0.9459	301
2	pneumonia	0.9928	0.9203	0.9552	301
accuracy				0.9599	897
macro avg		0.9624	0.9600	0.9602	897
weighted avg		0.9623	0.9599	0.9601	897

## EXP. (2)

Reduction ratio = 8

Table 4-111 testing results for version 8 EXP (2)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	96.10%	96.10%	96.10%	96.10%	0.9836	96.07%	94.15%
[44]	95.56%						

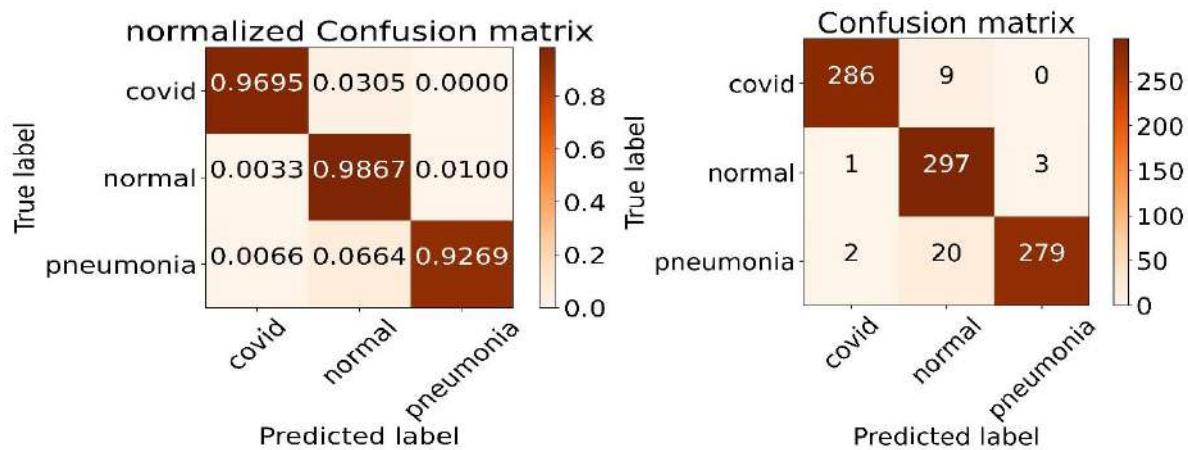


Figure 4-127 version 8 EXP (2) confusion matrices

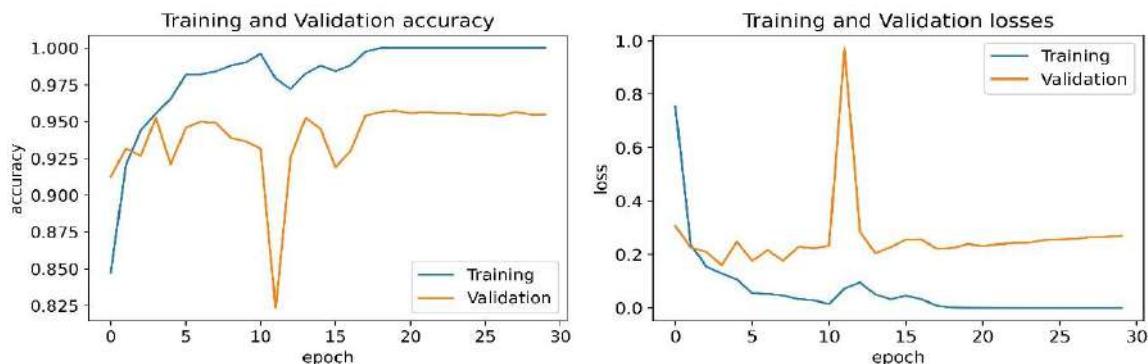


Figure 4-128 training accuracy and loss curves

Table 4-112 version 8 EXP (2) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9896	0.9695	0.9795	295
1	Normal	0.911	0.9867	0.9474	301
2	pneumonia	0.9894	0.9269	0.9571	301
accuracy				0.9610	897
macro avg		0.9633	0.9610	0.9613	897
weighted avg		0.9632	0.9610	0.9612	897

### EXP. (3)

Reduction ratio = 12

Table 4-113 testing results for version 8 EXP (3)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	95.21%	95.20%	95.09%	95.21%	0.9792	95.16%	92.81%
[44]	95.56%						

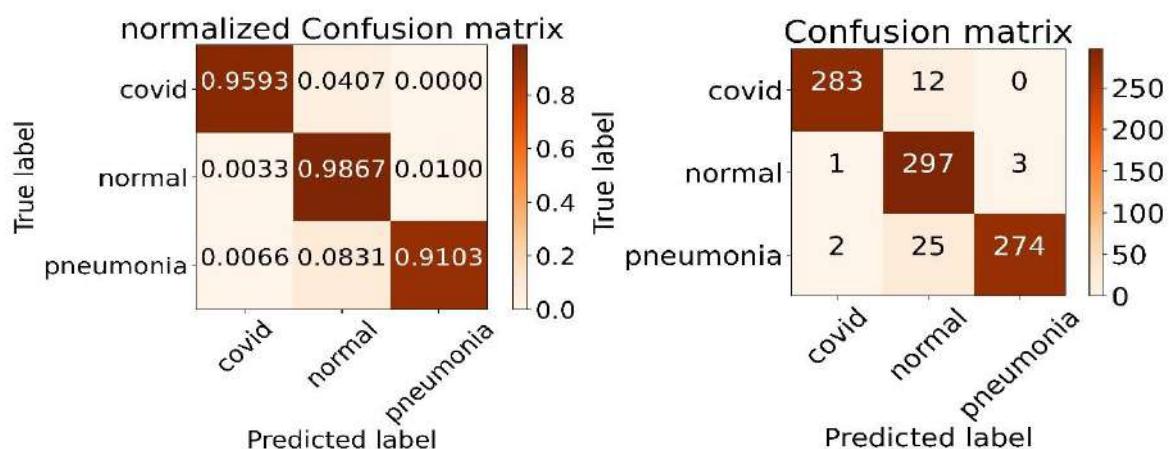


Figure 4-129 version 8 EXP (3) confusion matrices

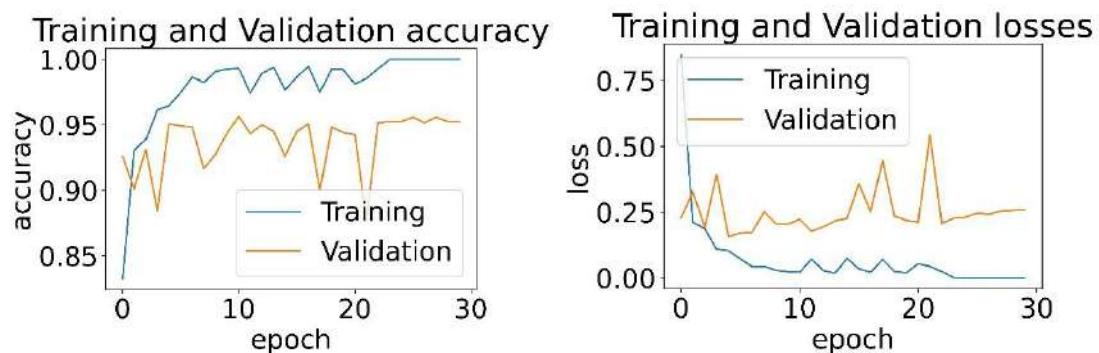


Figure 4-130 training accuracy and loss curves

Table 4-114 version 8 EXP (3) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9895	0.9593	0.9742	295
1	Normal	0.8892	0.9867	0.9354	301
2	pneumonia	0.9892	0.9103	0.9481	301
accuracy				0.9521	897
macro avg		0.9560	0.9521	0.9526	897
weighted avg		0.9557	0.9521	0.9524	897

## EXP. (4)

Reduction ratio = 16

Table 4-115 testing results for version 8 EXP (4)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	95.65%	95.75%	95.54%	95.65%	0.9844	95.64%	93.48%
[44]	95.56%						

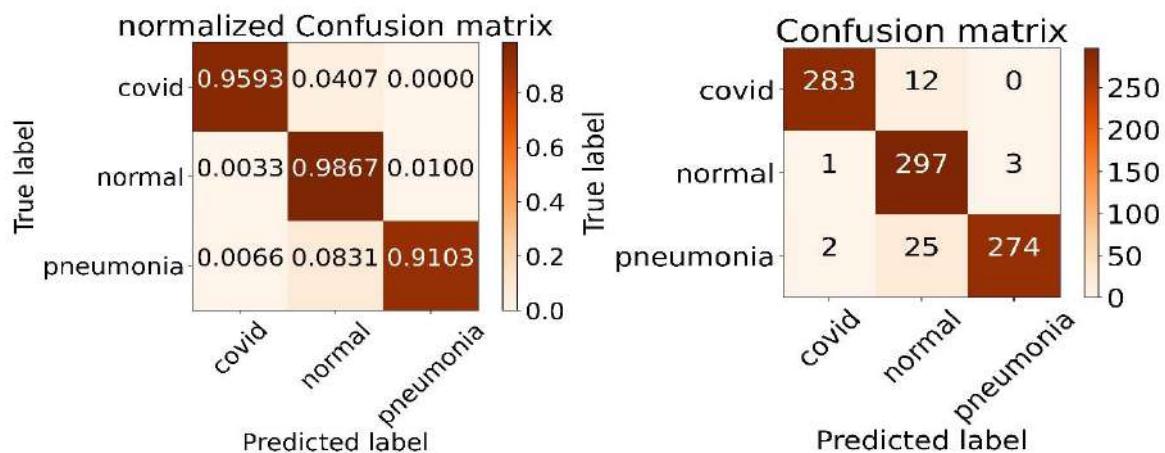


Figure 4-131 version 8 EXP (4) confusion matrices

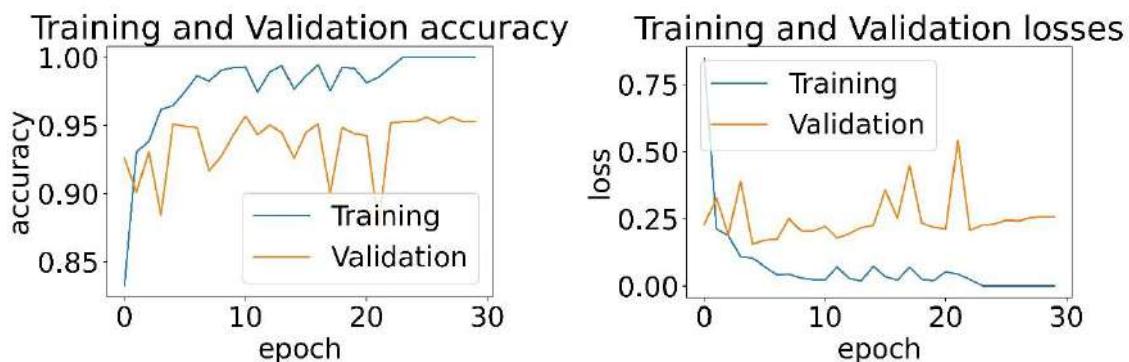


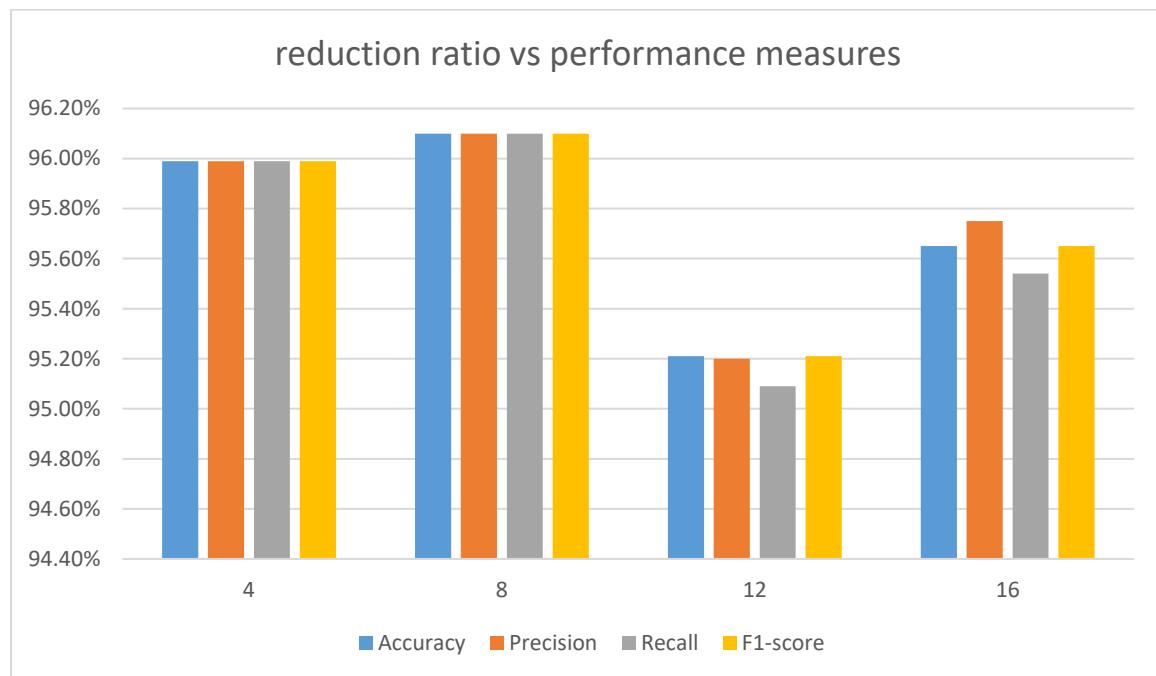
Figure 4-132 training accuracy and loss curves

Table 4-116 version 8 EXP (4) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9862	0.9695	0.9778	295
1	Normal	0.9156	0.9734	0.9436	301
2	pneumonia	0.9721	0.9269	0.949	301
accuracy				0.9565	897
macro avg		0.9580	0.9566	0.9568	897
weighted avg		0.9578	0.9565	0.9567	897

## Tuning the reduction ratio of the AFR layer

Reduction ratio	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
4	95.99%	95.99%	95.99%	95.99%	0.9799	95.95%	93.98%
8	<b>96.10%</b>	<b>96.10%</b>	<b>96.10%</b>	<b>96.10%</b>	0.9836	<b>96.07%</b>	<b>94.15%</b>
12	95.21%	95.20%	95.09%	95.21%	0.9792	95.16%	92.81%
16	95.65%	95.75%	95.54%	95.65%	0.9844	95.64%	93.48%



## Version 9

In this version a combination of AFR and MHA layers was utilized with the VGG-19

### Data preprocessing

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

### Proposed framework

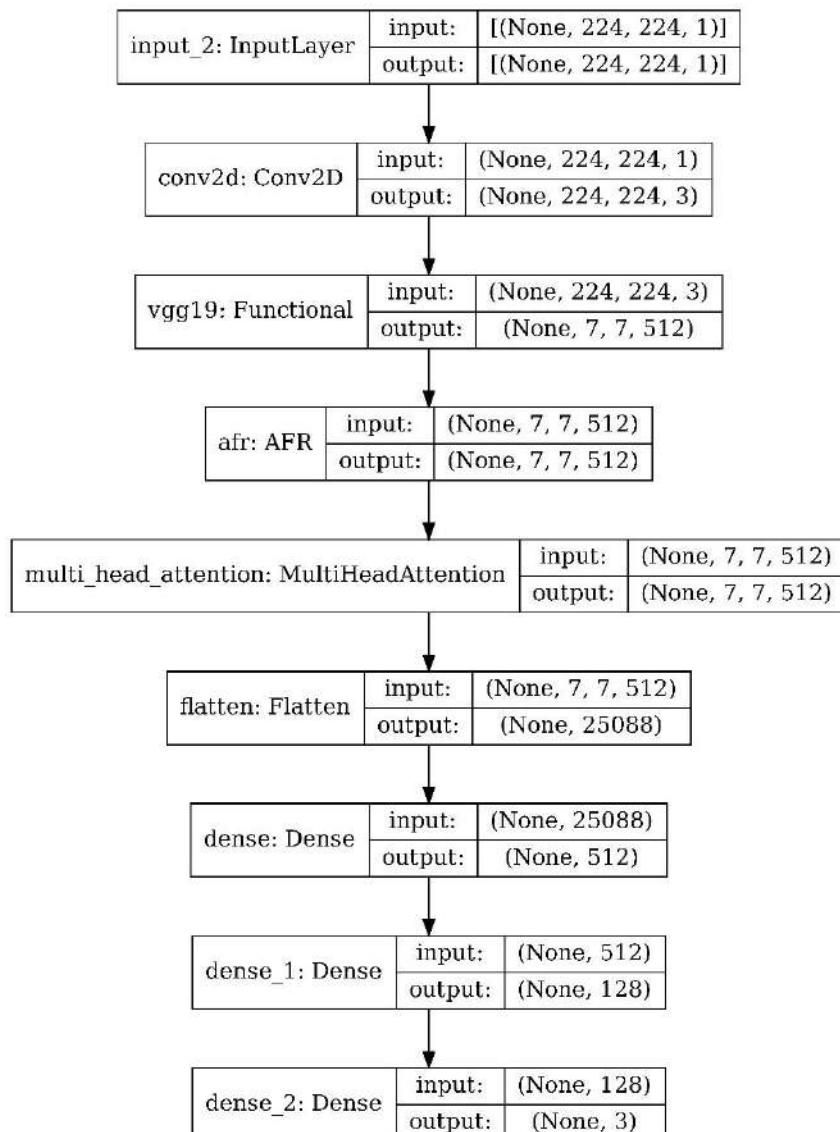


Figure 4-133 version 9 proposed network architecture

Table 4-117 testing results for version 9

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	92.20%	92.48%	91.86%	92.20%	0.9753	92.09%	88.29%
[44]	95.56%						

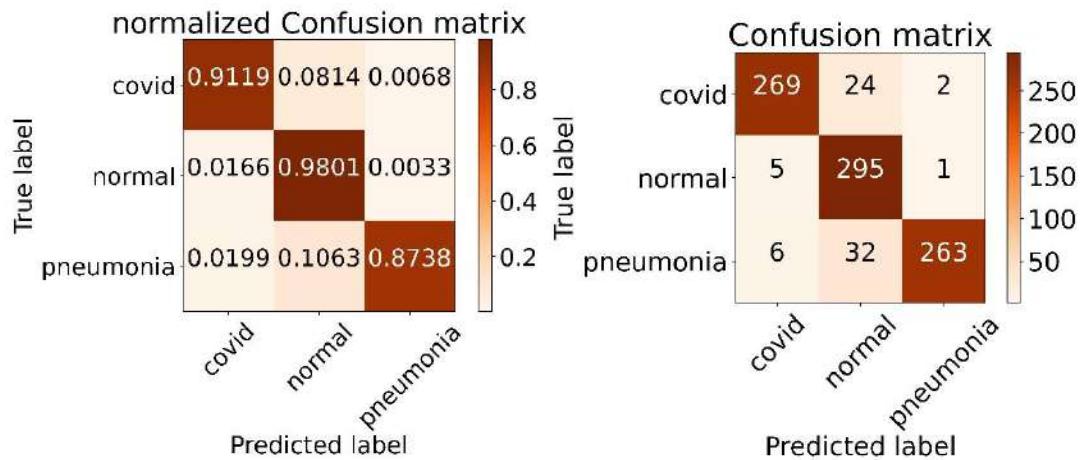


Figure 4-134 version 9 confusion matrices

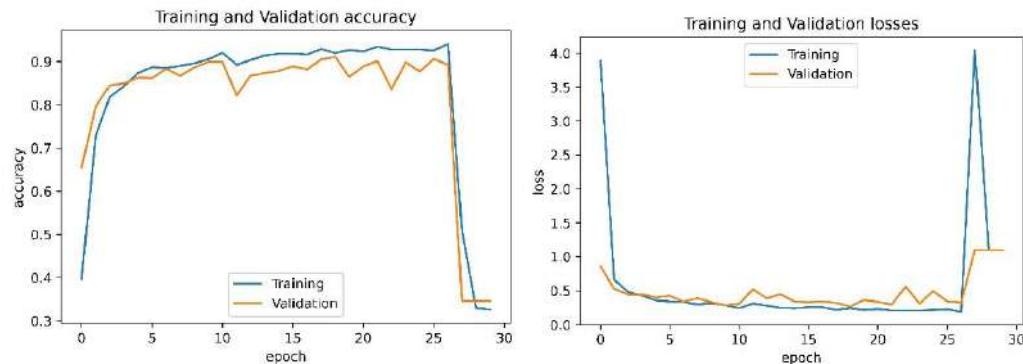


Figure 4-135 training accuracy and loss curves

Table 4-118 version (9) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.9607	0.9119	0.9357	295
1	Normal	0.8405	0.9801	0.9049	301
2	pneumonia	0.9887	0.8738	0.9277	301
accuracy				0.9220	897
macro avg		0.9300	0.9219	0.9227	897
weighted avg		0.9298	0.9220	0.9227	897

## **Version 10**

In this version a vision transformer (VT) classifier was utilized, the key idea behind VT is to represent an image as a sequence of patches, and apply the transformer architecture to these patches to learn their features and classify the image, VT takes as input an image and first divides it into a set of smaller patches. Each patch is then linearly projected to a lower-dimensional space, called an embedding, these patch embeddings are then fed into a transformer encoder, which processes the sequence of patches to learn their features, finally, a classification head is added to the network to predict the class label for the image, the transformer has a hyper-parameter called **patch size** which controls the size of the parts that the input image will be divided into.

## **Data preprocessing**

300 images from each class was taken as a testing set and the rest was used for training with a 0.2 validation split, all images were resized to 224\*224 and normalized.

## **Proposed framework**

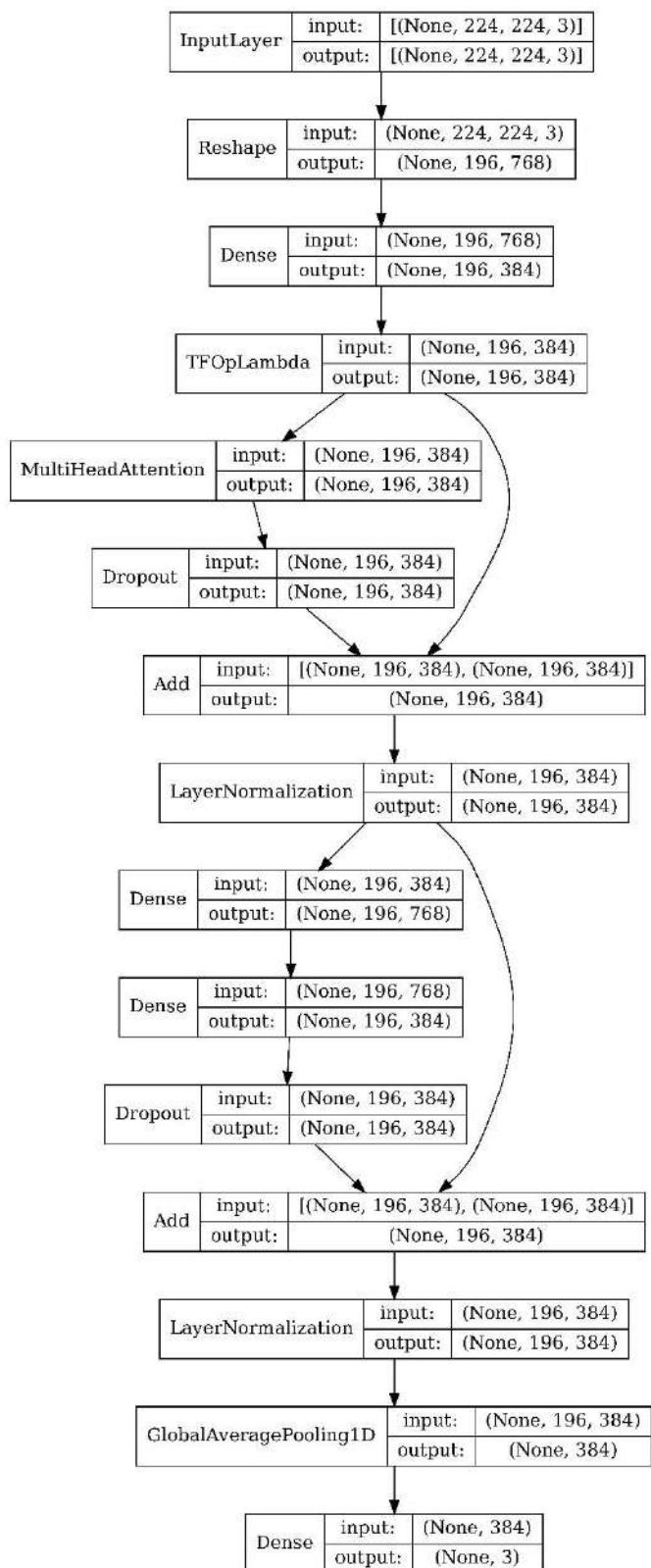


Figure 4-136 version 9 proposed network architecture

## EXP. (1)

Vision transformer model with 1 transformer, MHA layer with 7 heads, patch size 16.

Table 4-119 testing results for version 10 EXP 1

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	72.46%	72.67%	72.02%	72.46%	0.8815	72.13%	58.68%
[44]	95.56%						

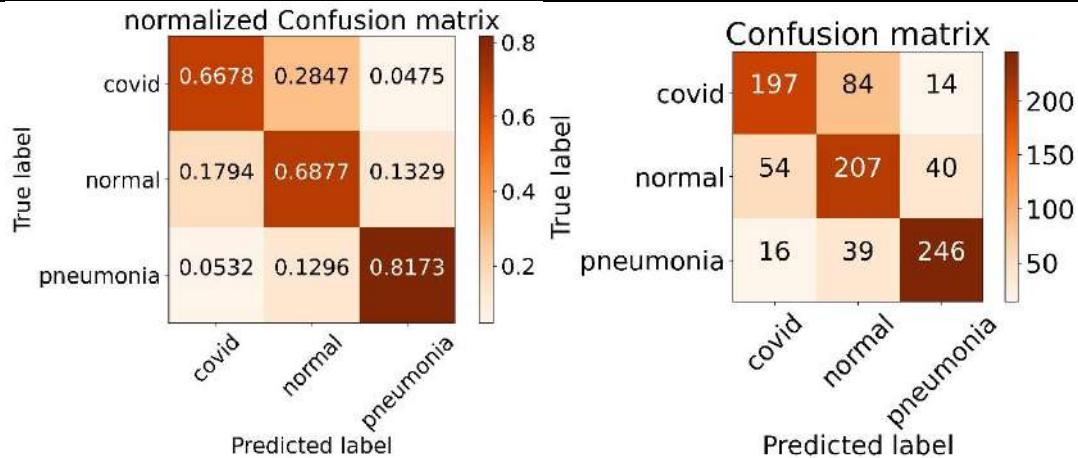


Figure 4-137 version 10 EXP (1) confusion matrices

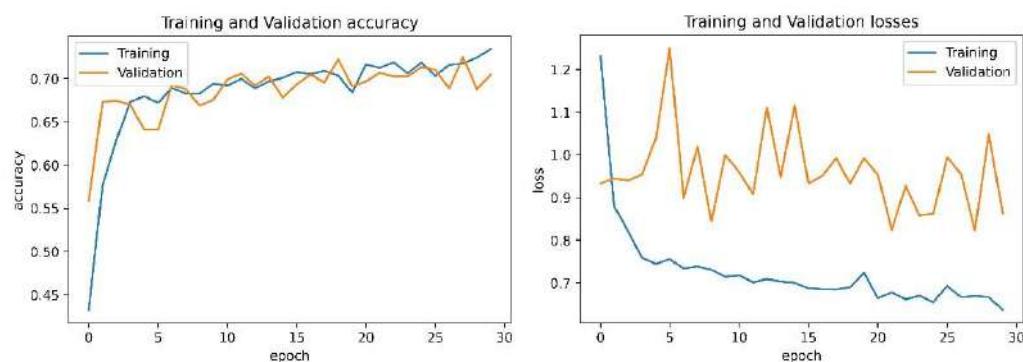


Figure 4-138 training accuracy and loss curves

Table 4-120 version 10 EXP (1) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.7378	0.6678	0.7011	295
1	Normal	0.6273	0.6877	0.6561	301
2	pneumonia	0.82	0.8173	0.8186	301
accuracy				0.7246	897
macro avg		0.7284	0.7243	0.7253	897
weighted avg		0.7283	0.7246	0.7254	897

## EXP. (2)

Vision transformer model with 1 transformer, MHA layer with 7 heads, patch size 32.

Table 4-121 testing results for version 10 EXP (2)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	70.57%	71.36%	70.01%	70.57%	0.8592	69.66%	55.82%
[44]	95.56%						

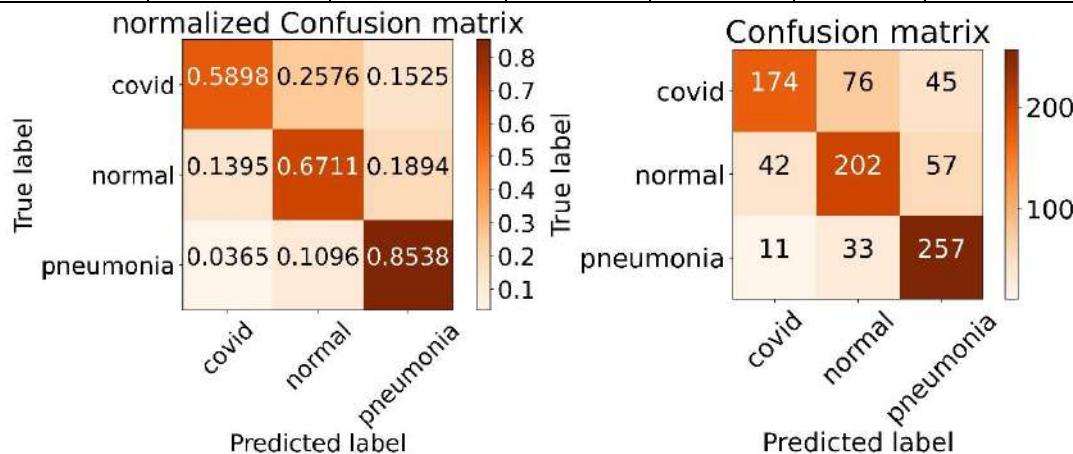


Figure 4-139 version 10 EXP (2) confusion matrices

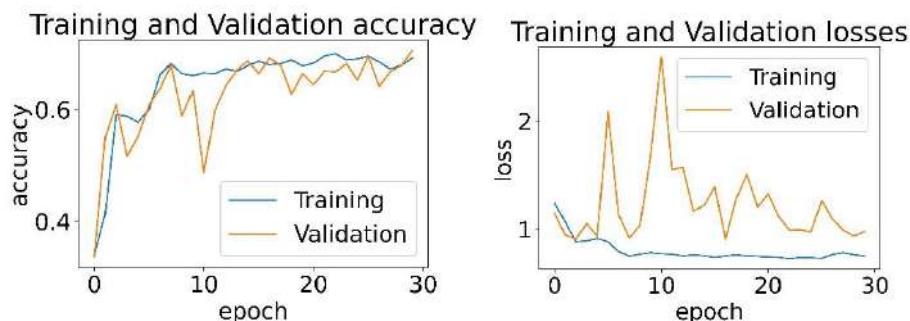


Figure 4-140 training accuracy and loss curves

Table 4-122 version 10 EXP (2) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.7665	0.5898	0.6667	295
1	Normal	0.6495	0.6711	0.6601	301
2	pneumonia	0.7159	0.8538	0.7788	301
accuracy				0.7057	897
macro avg		0.7106	0.7049	0.7019	897
weighted avg		0.7103	0.7057	0.7021	897

### EXP. (3)

Vision transformer model with 1 transformer, MHA layer with 7 heads, patch size 8.

Table 4-123 testing results for version 10 EXP (3)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	68.34%	69.19%	67.34%	68.34%	0.8332	62.83%	52.53%
[44]	95.56%						

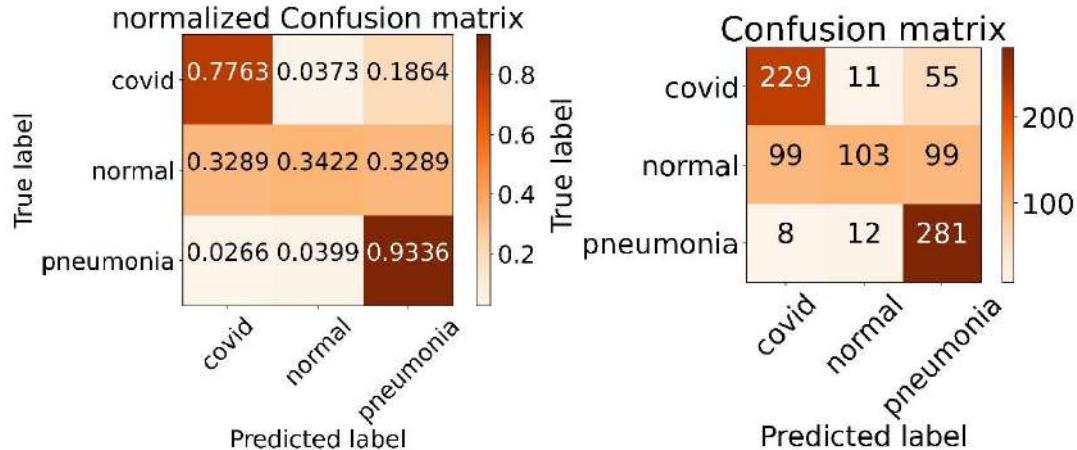


Figure 4-141 version 10 EXP (3) confusion matrices

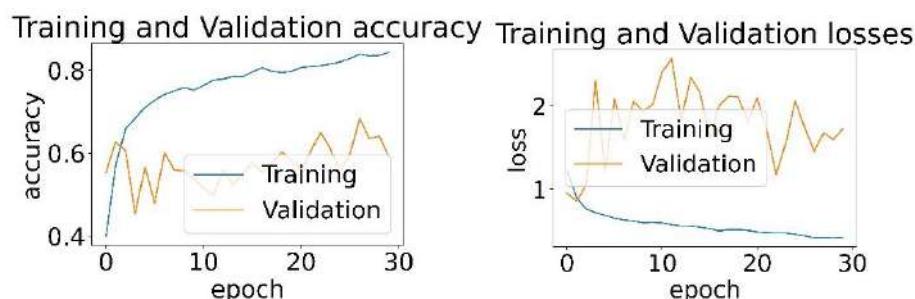


Figure 4-142 training accuracy and loss curves

Table 4-124 version 10 EXP (3) classification report

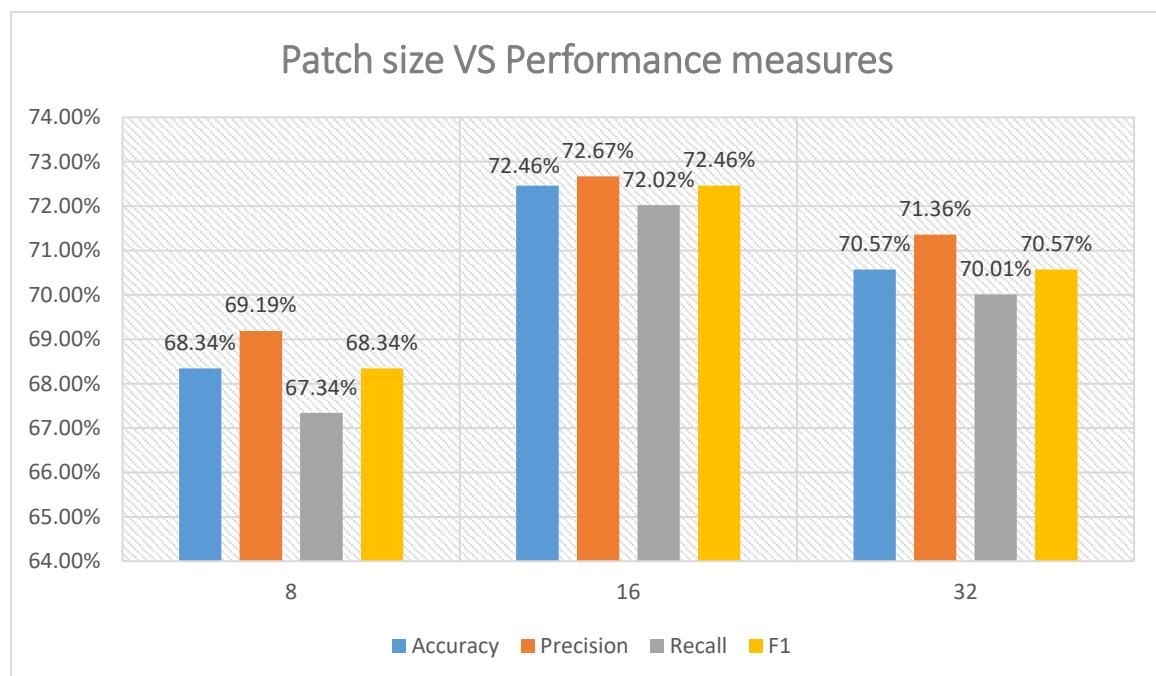
index	Class name	Precision	Recall	F1-score	Support
0	Covid-19	0.6815	0.7763	0.7258	295
1	Normal	0.8175	0.3422	0.4824	301
2	pneumonia	0.646	0.9336	0.7636	301
accuracy				0.6834	897
macro avg		0.7150	0.6840	0.6573	897
weighted avg		0.7152	0.6834	0.6568	897

## Tuning the patch size of the vision transformer

Table 4-125 the effect of patch size

Patch size	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
8	68.34%	69.19%	67.34%	68.34%	0.8332	62.83%	52.53%
16	72.46%	72.67%	72.02%	72.46%	0.8815	72.13%	58.68%
32	70.57%	71.36%	70.01%	70.57%	0.8592	69.66%	55.82%

The patch size factor determines the size of small parts that the input image will be divided into.



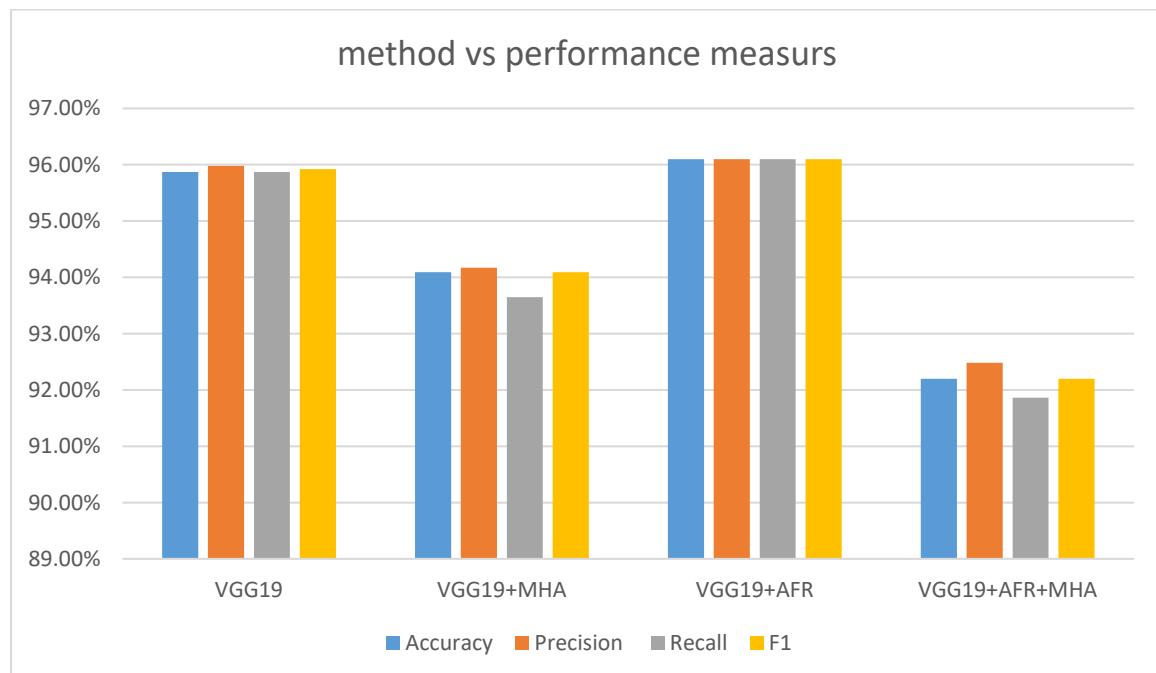
## Experiments summary

Method	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
VGG19	95.87%	95.98%	95.87%	95.92%	0.9823	95.85%	93.96%
VGG19+MHA	94.09%	94.17%	93.65%	94.09%	0.9857	94.08%	91.14%
VGG19+AFR	96.10%	96.10%	96.10%	96.10%	0.9836	96.07%	94.15%
VGG19+AFR+MHA	92.20%	92.48%	91.86%	92.20%	0.9753	92.09%	88.29%
VT	72.46%	72.67%	72.02%	72.46%	0.8815	72.13%	58.68%

Table 4-126 Abbreviations

<b>LSTM</b>	Long-short time memory
<b>AFR</b>	Adaptive Feature Recalibration (squeeze and excitation)
<b>MHA</b>	Multi-Head Attention
<b>VT</b>	Vision transformer

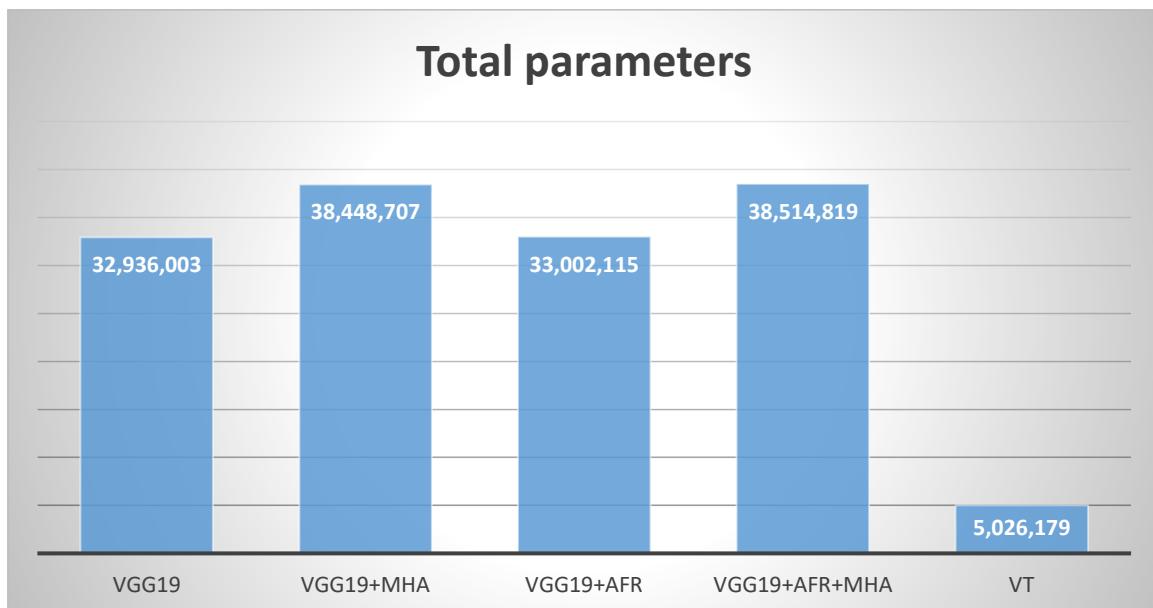
As shown in the table the best results was achieved by the VGG-19+AFR combination with 96.10% accuracy which is higher than the VGG-19 alone which achieved 95.87%, meanwhile, the VT had the least results with 72.46% accuracy.



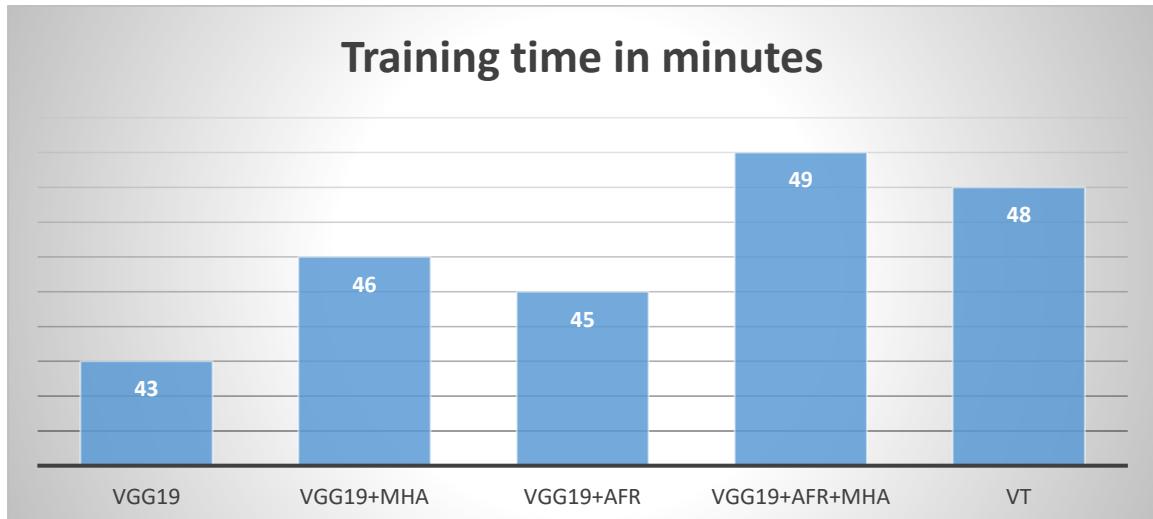
## Training time and number of parameters

Method	Training time	Total parameters	Trainable parameters	Non trainable parameters
VGG19	43min 40s	32,936,003	12,911,619	20,024,384
VGG19+MHA	46min 13s	38,448,707	18,424,323	20,024,384
VGG19+AFR	45min 30s	33,002,115	12,977,731	20,024,384
VGG19+AFR+MHA	48min 22s	38,514,819	18,490,435	20,024,384
VT	48min 3s	5,026,179	5,026,179	0

**Total parameters**



**Training time in minutes**



Note that these experiments was carried out using the same GPU as the one desctried in the heartbeat abnormalities service

#### 4.2.2.6 Final proposed framework

As shown in the previous section, some pre-trained models was experimented with and the best results was achieved by the VGG-19 model at 95.87% accuracy, then some additional methods was experimented with the VGG-19 to improve the results even more including the multi-head attention (MHA) layer, the adaptive feature recalibration (AFR) and a combination of both and the best results was achieve be the VGG-19+AFR combination with 96.10% accuracy.

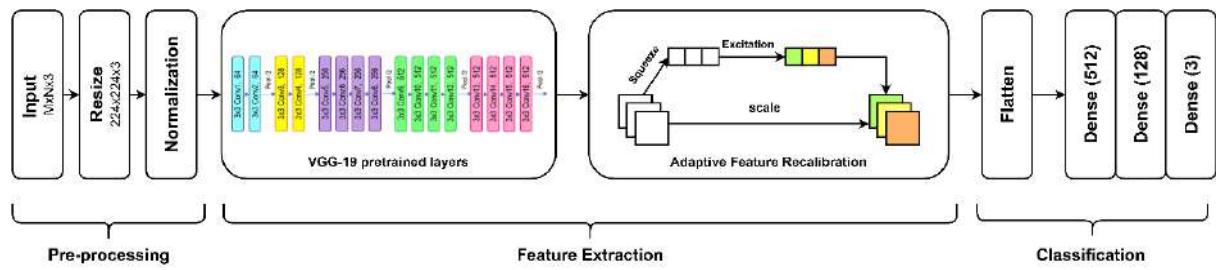


Figure 4-143 final proposed network architecture

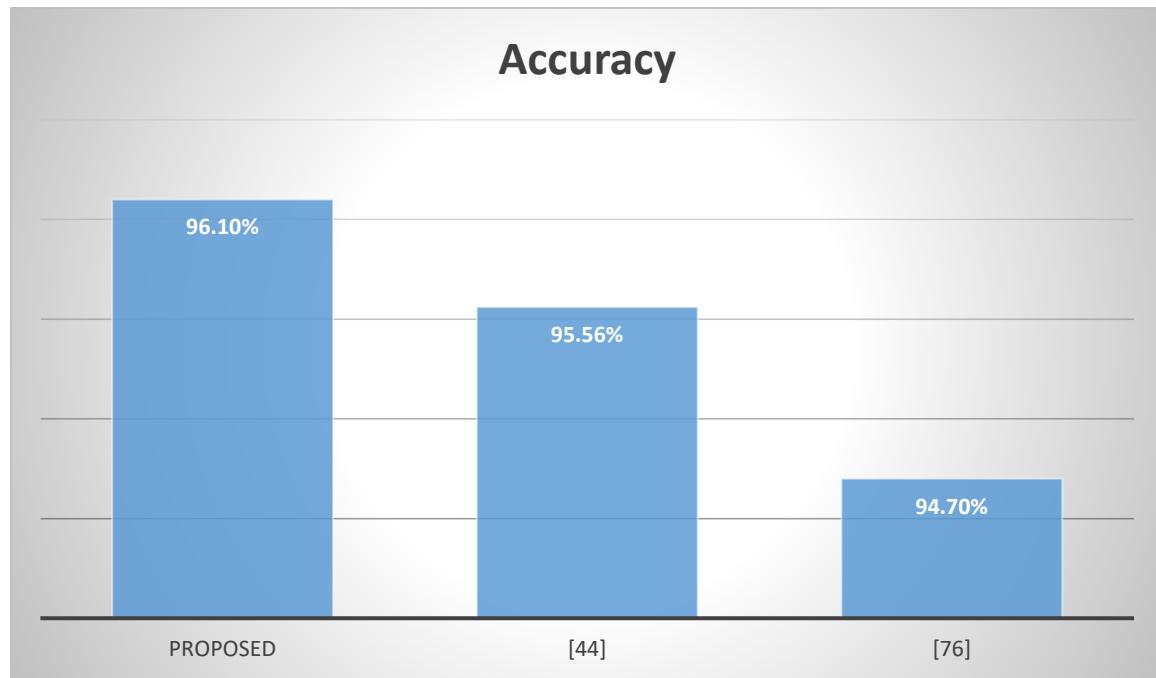


Figure 4-144 accuracy comparison with [44] and [76]

#### 4.2.2.7 Method validation

For validating the final proposed model the COVID-19 Radiography Database for covid-19 detection from chest x-ray images was used to train the model and test its performance, this dataset consist of three classes covid-19, normal and pneumonia

Table 4-127 class distribution of COVID-19 Radiography Database

Class	Covid-19	Normal	Viral Pneumonia
Number of samples	3616	10.2 k	1354

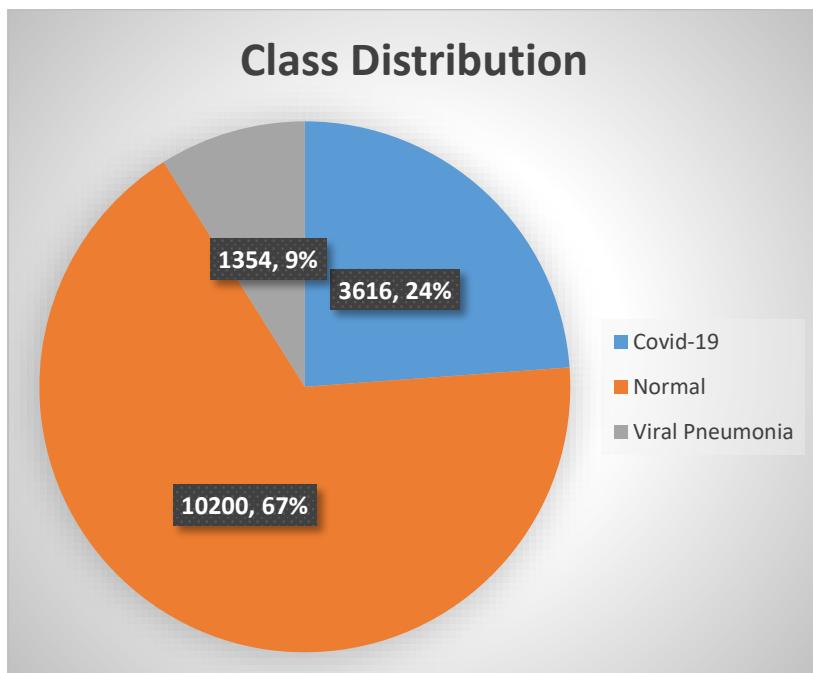


Figure 4-145 class distribution of COVID-19 Radiography Database

Table 4-128 result comparison with the state-of-the-art

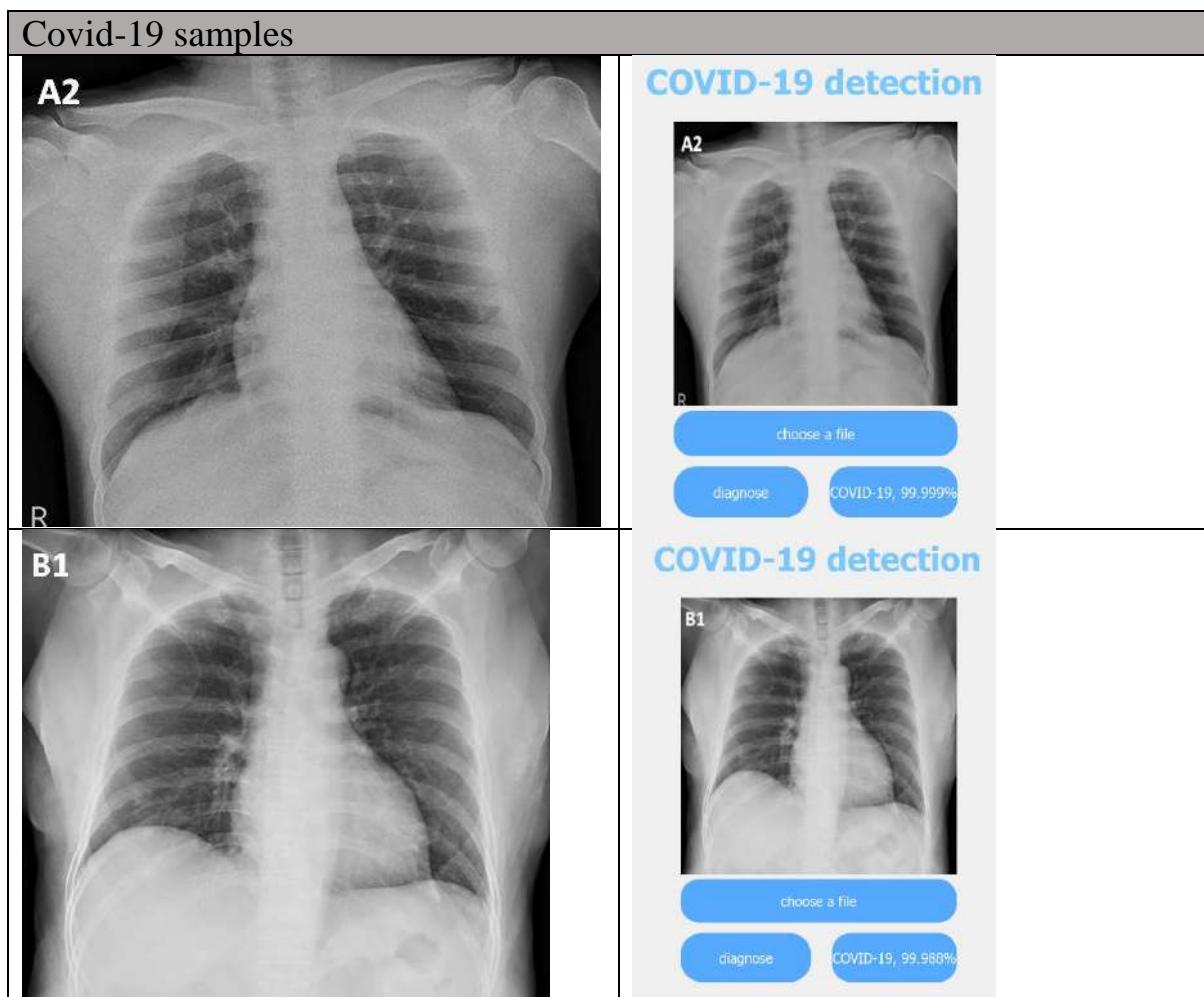
	Handling data imbalance	Train : Test	Model	Accuracy	Recall	Precision	F1-score	Kappa
[77]	Augmentation	70% : 30%	CovidDetNet	98.40%	96.66%	97.00%	96.82%	95.0%
Ours	Weighted loss		VGG-19+AFR	98.22%	97.37%	98.08%	97.72%	96.29%
<hr/>								
[78]	Augmentation	90% : 10%	AlexNet	97.59%	95.45%	98.55%	96.9%	
Ours	Weighted loss		VGG19+AFR	98.02%	97.54%	98.08%	97.81%	95.89%

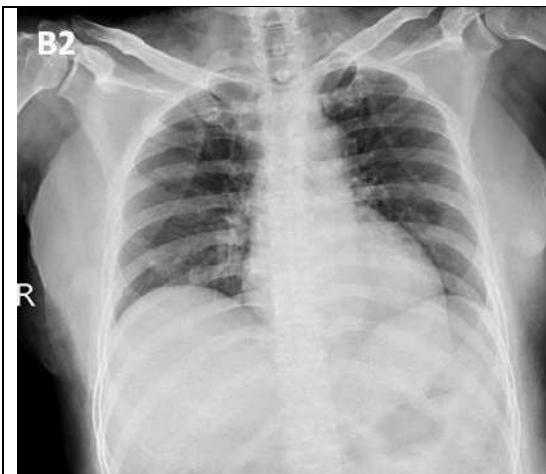
#### 4.2.2.8 Testing with real world cases (online testing)

After the training process is done and the final model was chosen to be the combination of the VGG-19 and the adaptive feature recalibration (VGG-19+AFR) which achieved 96.10% accuracy, it is time to test it on some samples completely out side of the dataset used in the training and testing, the model was tested with 8 samples from each class (covid-19, normal, pneumonia) and it miss diagnosed only one normal sample to be pneumonia, you can find links for datasets used in this test in the appendix.

The following table shows each sample on the left size and the model's diagnosis in the right side along with the confidence for the diagnosis.

Please note that the GUI will be discussed in details in section 4.3





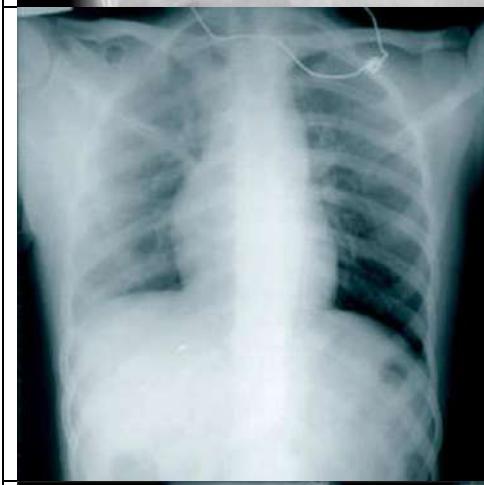
### COVID-19 detection



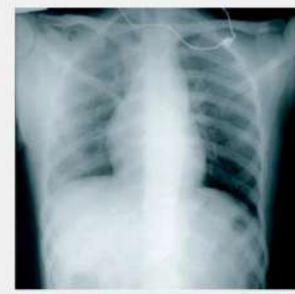
choose a file

diagnose

COVID-19, 99.840%



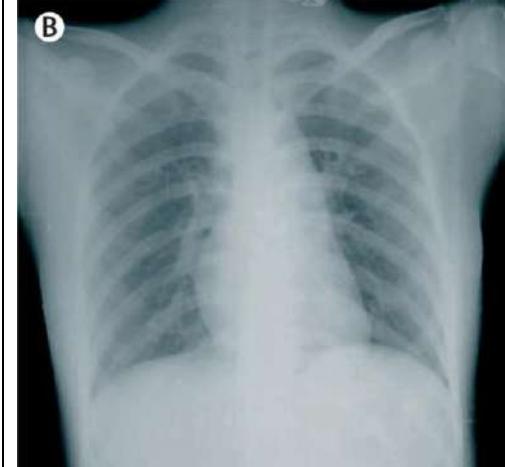
### COVID-19 detection



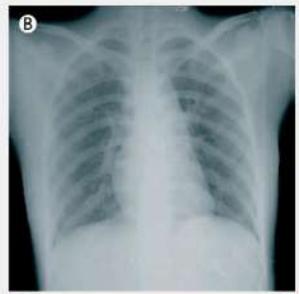
choose a file

diagnose

COVID-19, 98.082%



### COVID-19 detection



choose a file

diagnose

COVID-19, 100.0%



### COVID-19 detection



choose a file

diagnose

COVID-19, 99.999%



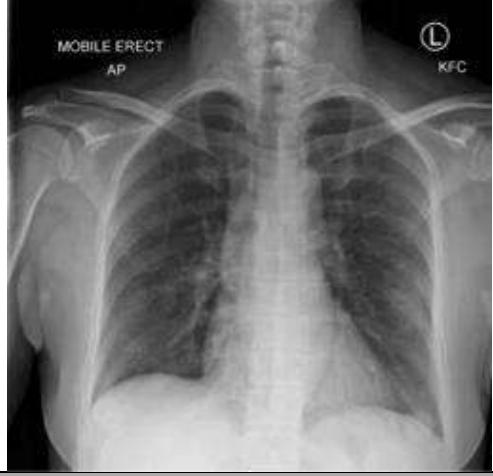
### COVID-19 detection



choose a file

diagnose

COVID-19, 99.993%



### COVID-19 detection



choose a file

diagnose

COVID-19, 99.999%

Normal test samples



### COVID-19 detection



choose a file

diagnose

Pneumonia,  
85.823%



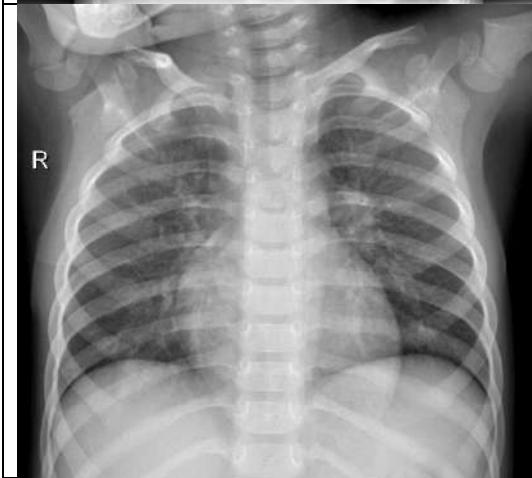
### COVID-19 detection



choose a file

diagnose

Normal, 100.0%



### COVID-19 detection



choose a file

diagnose

Normal, 99.814%



### COVID-19 detection



choose a file

diagnose

Normal, 99.999%



### COVID-19 detection



choose a file

diagnose

Normal, 99.984%



### COVID-19 detection



choose a file

diagnose

Normal, 96.157%



### COVID-19 detection



choose a file

diagnose

Normal, 99.945%



### COVID-19 detection



choose a file

diagnose

Normal, 100.0%

### Pneumonia test samples



### COVID-19 detection



choose a file

diagnose

Pneumonia,

99.999%



### COVID-19 detection



choose a file

diagnose

Pneumonia,  
99.998%



### COVID-19 detection



choose a file

diagnose

Pneumonia,  
99.994%



### COVID-19 detection



choose a file

diagnose

Pneumonia,  
99.999%



### COVID-19 detection



choose a file

diagnose

Pneumonia,  
99.999%



### COVID-19 detection



choose a file

diagnose

Pneumonia,  
99.999%



### COVID-19 detection



choose a file

diagnose

Pneumonia,  
99.999%

### **4.2.3 Brain tumor categorization from MRI images**

Brain tumor is one of the most dangerous aggressive diseases, the low number of skillful doctors and min of knowledge about tumors make it challenging and time-consuming to generate reports from MRI images, so the solution to this problem is to use AI automated classification models.

#### **4.2.3.1 Motivation**

To diagnose brain tumors, doctors typically rely on a combination of imaging studies, physical exams, and patient history, in many cases, MRI imaging is an essential component of the diagnostic process, providing detailed images of the brain and surrounding structures, however, making a definitive diagnosis based only on MRI imaging studies can be challenging, as different types of brain tumors can present with similar features on MRI images, both meningioma and schwannoma tumors can appear as well-defined, round, and enhancing masses on MRI images, making it difficult to differentiate between the two, to overcome these challenges, doctors may use additional imaging modalities such as MRS, DWI, or positron emission tomography (PET) imaging, which can provide additional information about the biochemical and metabolic features of brain tumors. Doctors may also consider the patient's age, medical history, and presenting symptoms when making a diagnosis.

AI models can assist doctors in the diagnostic process by analyzing large datasets of MRI images and learning to recognize features that are indicative of different types of brain tumors, this can provide doctors with an additional tool to help make a faster and more accurate diagnosis, leading to better patient outcomes.

#### **4.2.3.2 Dataset description**

Brain Tumor MRI Dataset is a public dataset available on Kaggle, this dataset is a combination of three datasets, Brain Tumor Classification (MRI), Br35H and brain tumor dataset, it contain a total of 7023 sample most of them are in grayscale color-mode.

*Table 4-129 class distribution of the Brain Tumor MRI Dataset*

	glioma	meningioma	no tumor	pituitary	total	
train	1312	1339	1595	1457	5712	7023
test	300	306	405	300	1311	

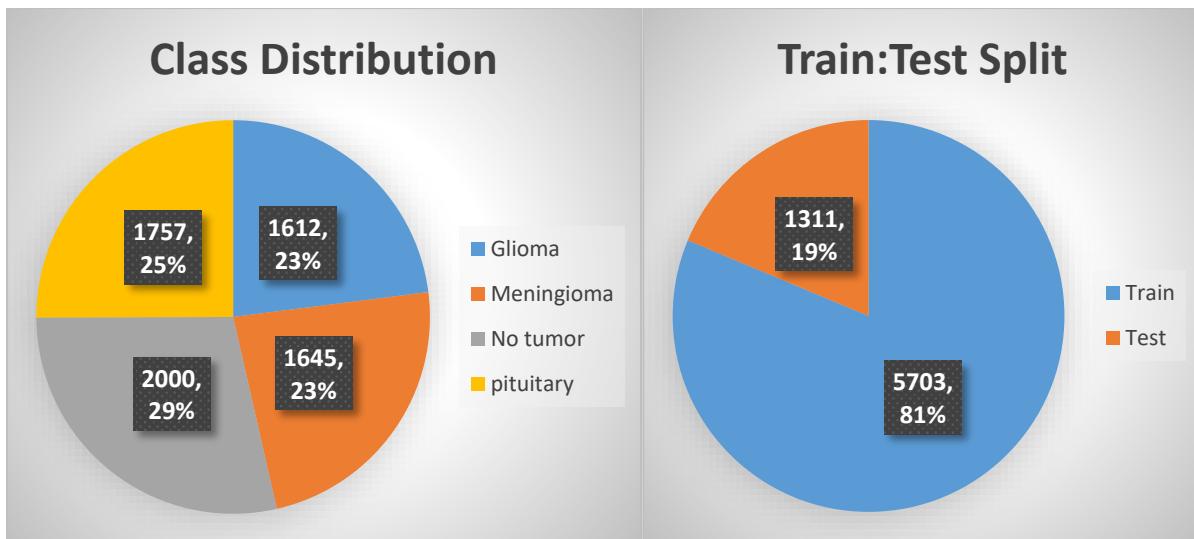


Figure 4-146 class distribution percentage and train test split percentage

### Data source

This dataset is a combination of two datasets, the first one is **SARTAJ dataset** which was collected by the **Indian Institute of Technology Bombay (IIT Bombay)** which is a public research university and technical institute in Powai, Mumbai, Maharashtra, India, IIT Bombay is ranked among the most prestigious academic institutions in India, IIT Bombay has a long history of collaboration with medical organizations in India and abroad, the second dataset is **Br35h dataset** which is a public dataset of brain MRI images that was collected by researchers at the **University of California, San Francisco (UCSF)**, it is part of the University of California system as its Medical Department and is dedicated entirely to health science and life science, it conducts research and teaching in medical and biological sciences, UCSF is considered one of the world's preeminent medical and life sciences universities by the Academic Ranking of World Universities in 2019.

The presence of high-profile organizations in the collection and compilation of this dataset indicates the importance of AI in this field.

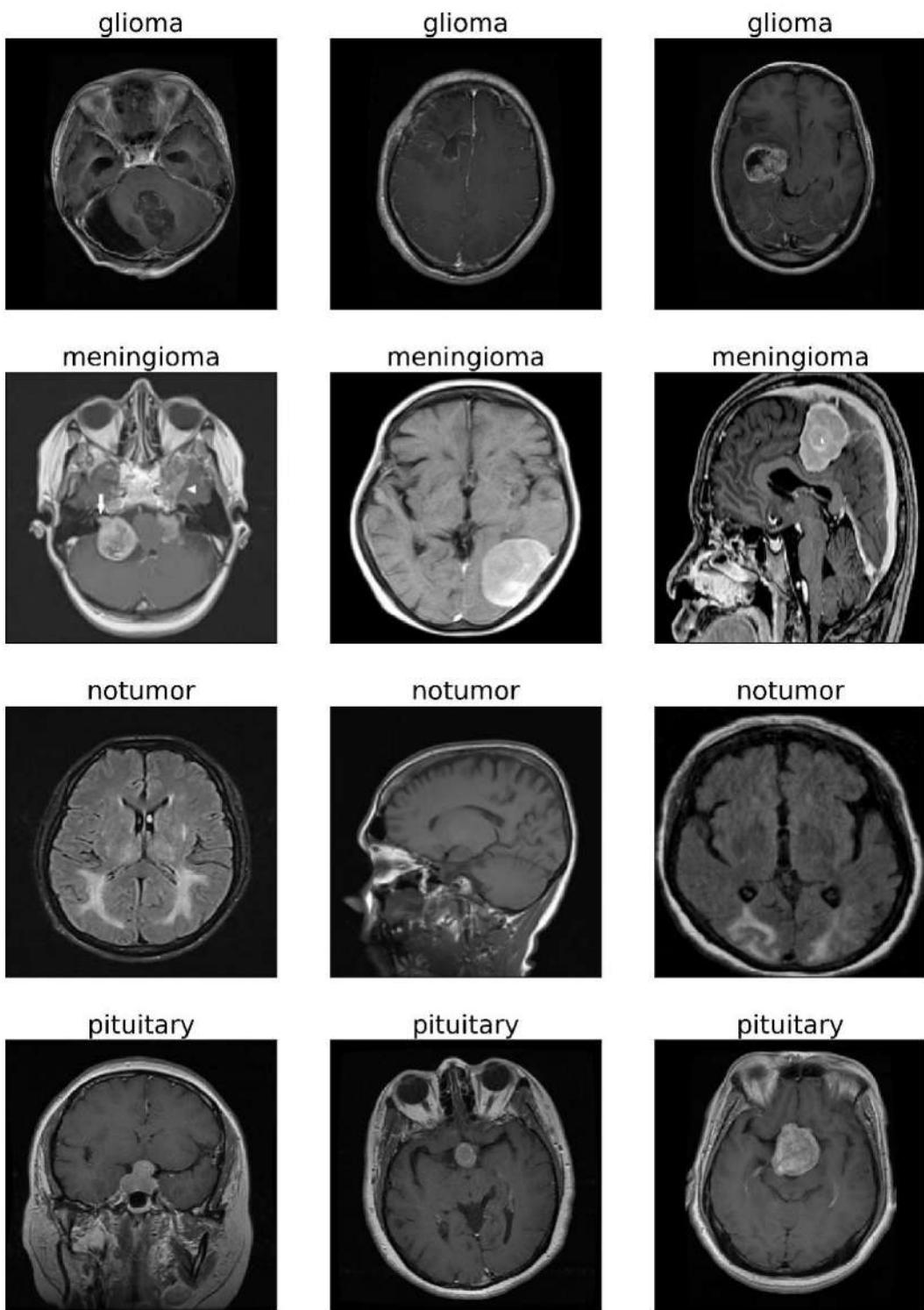


Figure 4-147 samples from Brain Tumor MRI Dataset

#### 4.2.3.3 Development methodology

The suitable types for this dataset in 2-D convolution neural networks as it is more useful with images due to the feature extraction process of the convolution layers, different end to end trained models will utilized as well as some pre-trained models.

#### 4.2.3.4 2-D convolution

- Version 1

This introduces a 6 convolution layer end-to-end trained CNN architecture.

#### Data preprocessing

The dataset was used as it is in terms of number of samples, all images resized to 200\*200, and RGB samples converted to grayscale, all images were normalized.

#### Proposed framework

This model consists of 6 convolution block each one consists of a 2-D convolution layer with kernel size (7, 7), a BatchNormalization layer and a maxpooling layer of size (2, 2), the first convolution layer contains 64 kernels, second and third convolution layers contain 128 kernels, fourth and fifth convolution layers contain 256 kernels and the sixth convolution layer contains 512 kernels, all convolution layers has padding same property and relu activated, these convolution blocks are followed by a classification top of 3 Dense layers.

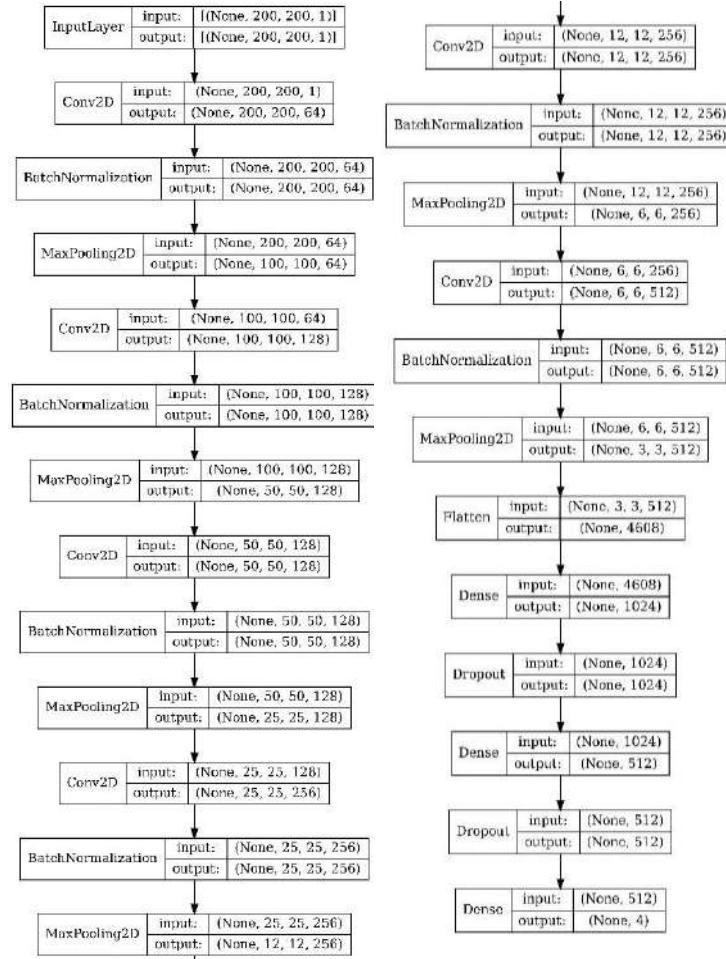


Figure 4-148 version 1 proposed network architecture

#### Result

Table 4-130 Results for version 1

Train acc	Val_acc	Test acc
97.97%	96.26%	96.26%

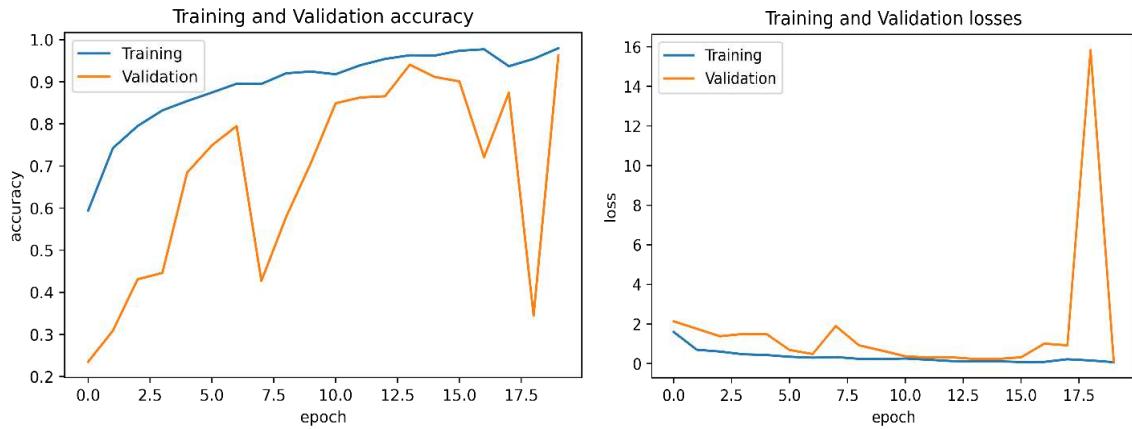


Figure 4-149 training accuracy and loss curves for version 1

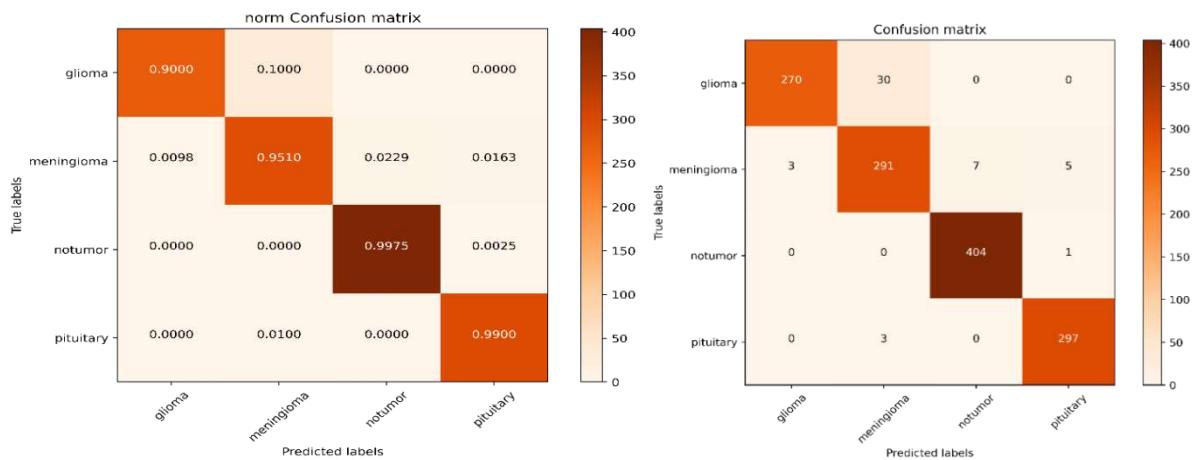


Figure 4-150 version 1 confusion matrices

classification report					
	precision	recall	f1-score	support	
0	0.99	0.90	0.94	300	
1	0.90	0.95	0.92	306	
2	0.98	1.00	0.99	405	
3	0.98	0.99	0.99	300	
accuracy			0.96	1311	
macro avg	0.96	0.96	0.96	1311	
weighted avg	0.96	0.96	0.96	1311	

Figure 4-151 version 1 classification report

## Experiments

## EXP. (1)

Horizontal flip augmentation method was applied on the dataset

Table 4-131 Results for version 1 EXP. (1)

Train acc	Val_acc	Test acc
97.64%	97.18%	97.03%

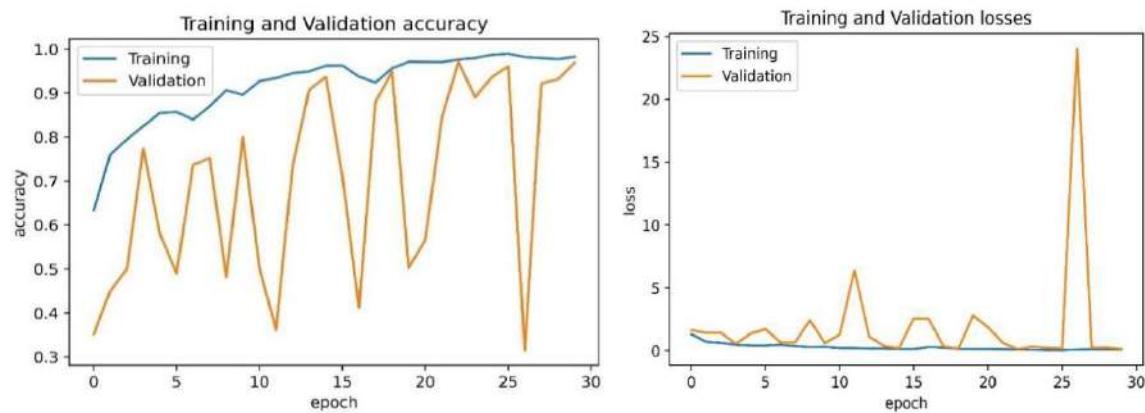
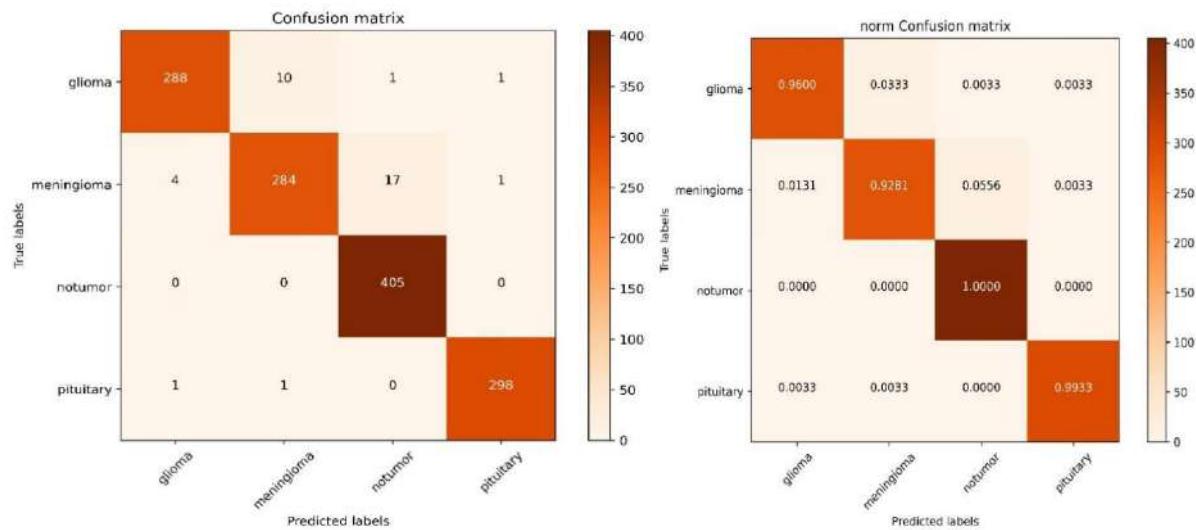


Figure 4-152 training accuracy and loss curves for version 1 EXP. (1)



classification report				
	precision	recall	f1-score	support
0	0.98	0.96	0.97	300
1	0.96	0.93	0.95	306
2	0.96	1.00	0.98	405
3	0.99	0.99	0.99	300
accuracy			0.97	1311
macro avg	0.97	0.97	0.97	1311
weighted avg	0.97	0.97	0.97	1311

Figure 4-153 classification report for version 1 EXP. (1)

## Version 1 conclusion

This model achieved good accuracy and augmentation methods would help improve the results even more.

### • Version 2

This version introduces a 4 convolution layer end-to-end trained model

#### Data preprocessing

All images were resized to 64\*64 and normalized, horizontal flip augmentation method was applied.

#### Proposed framework

The proposed network consists of 4 convolution blocks each block includes a convolution layer a Batch-Normalization layer and a max-pooling layer, all convolution layers has a filter size of 7\*7 and all max-pooling layers are of size 2\*2, the first convolution layer has 46 filters and this number got doubled with each of remaining convolution layers ending with 512 filters in the last one.

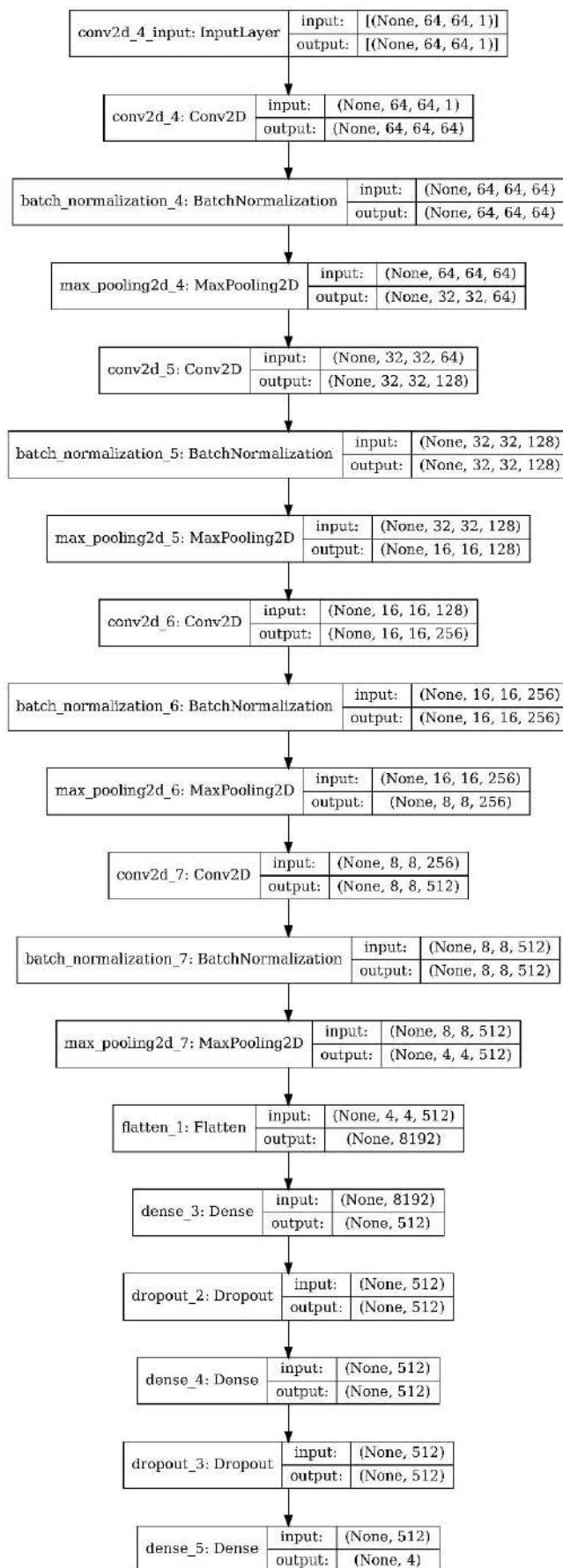


Figure 4-154 version 2 proposed network architecture

## Result

Table 4-132 Results for version 2

Test accuracy	Test precision	Test Recall	Test AUC
99.08%	99.08%	99.01%	0.9998

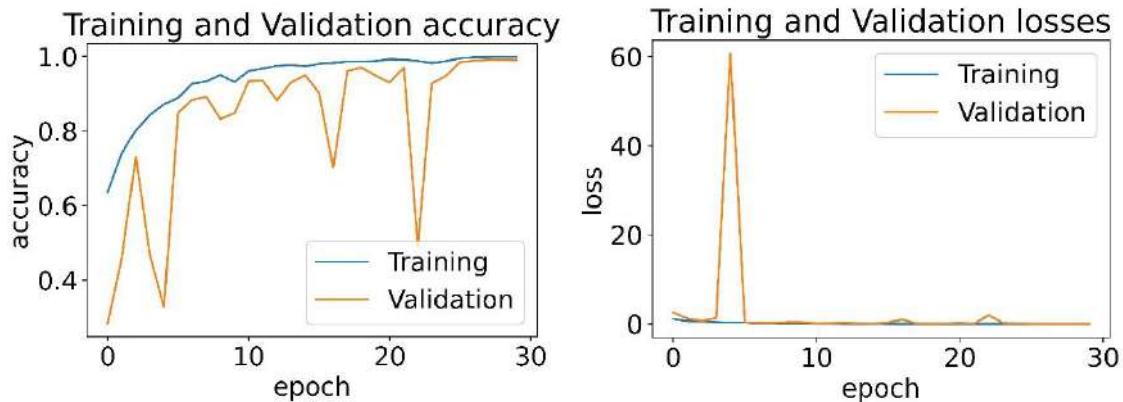


Figure 4-155 training accuracy and loss curves for version 2

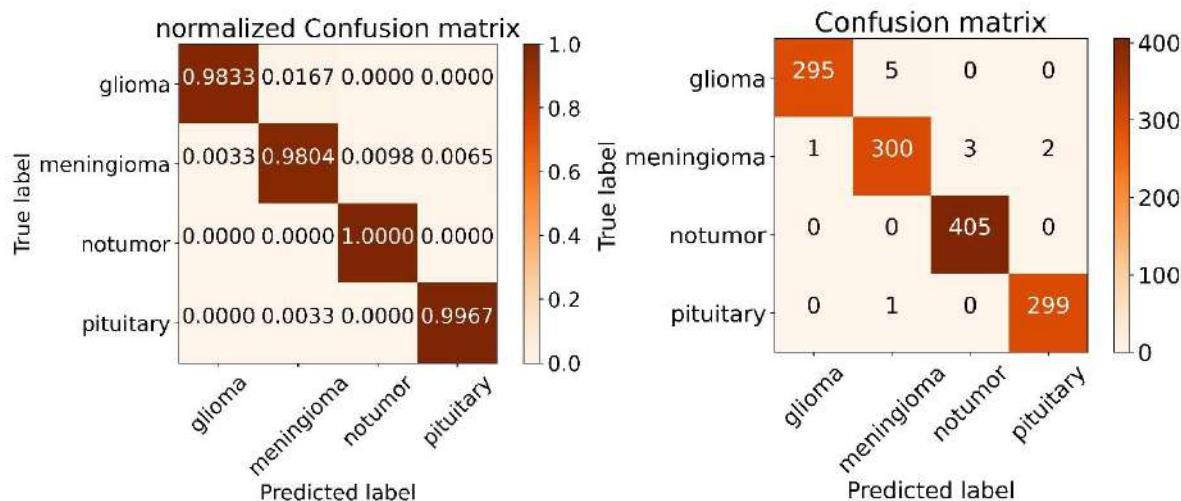


Figure 4-156 version 2 confusion matrices

Table 4-133 version 2 classification report

index	Class name	Precision	Recall	F1-score	Support
0	Glioma	0.9966	0.9833	0.9899	300
1	Meningioma	0.9804	0.9804	0.9804	306
2	No tumor	0.9926	1	0.9963	405
3	Pituitary	0.9934	0.9967	0.995	300
accuracy				0.9908	897
macro avg		0.9908	0.9901	0.9904	897
weighted avg		0.9909	0.9908	0.9908	897

## **Version 2 conclusion**

Reducing the number of convolution layers results in a much better performance.

## **Version 3**

Multi-head attention layer was utilized in this version

### **Data preprocessing**

All images were resized to 64\*64 and normalized, horizontal flip augmentation method was applied.

### **Proposed framework**

Same as version 2 but with a MHA layer after the last max-pooling layer, as the input of the MHA is (4, 4, 512) the number of heads that would be experimented with are 1, 2, 4.

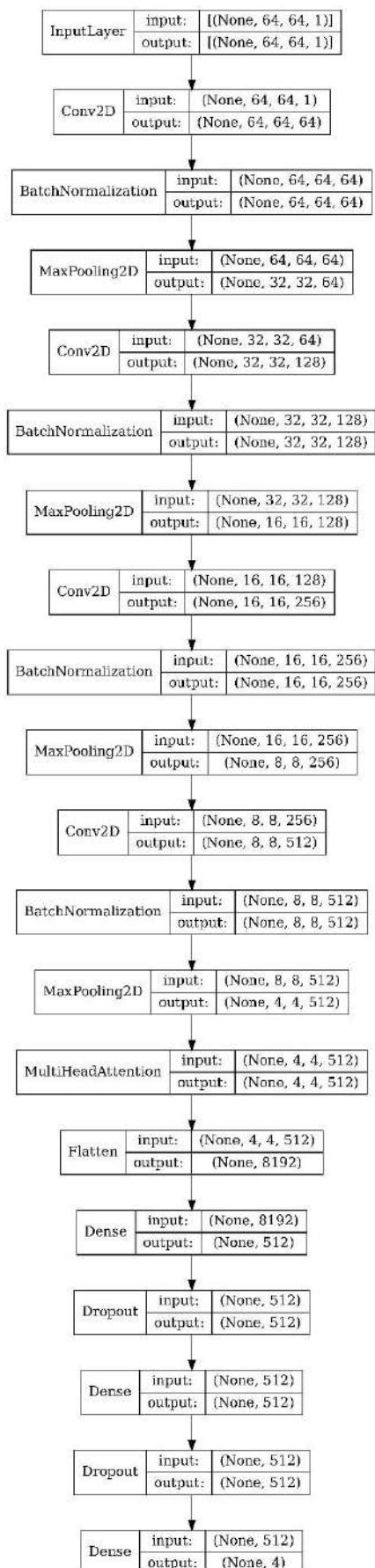


Figure 4-157 version 3 proposed network architecture

## EXP. (1)

Number of heads = 1

Table 4-134 testing results for version 3 EXP (1)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	97.86%	97.86%	97.79%	97.79%	0.9935	97.57%	97.03%

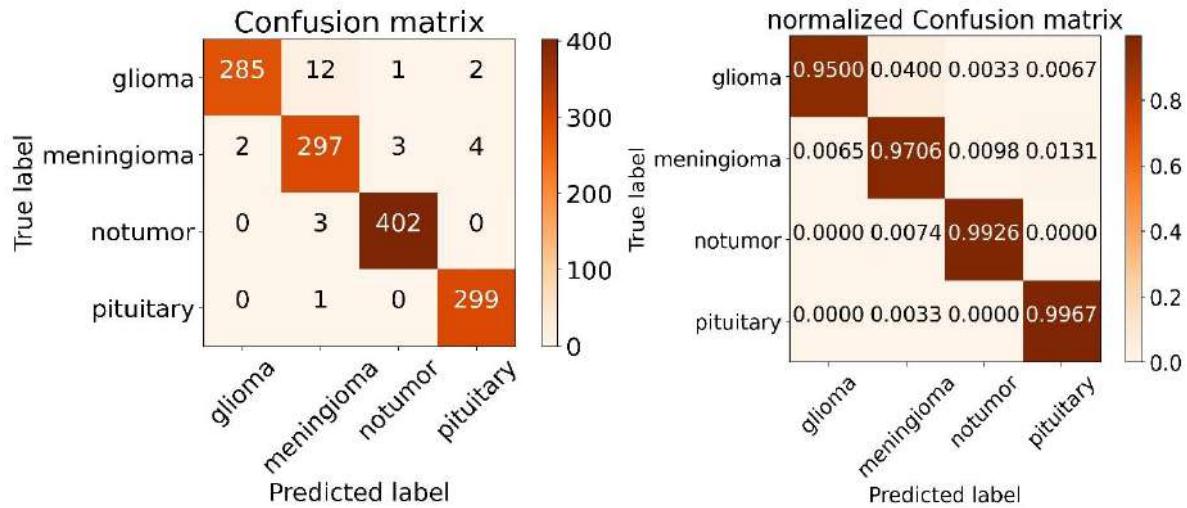


Figure 4-158 version 3 EXP (1) confusion matrices

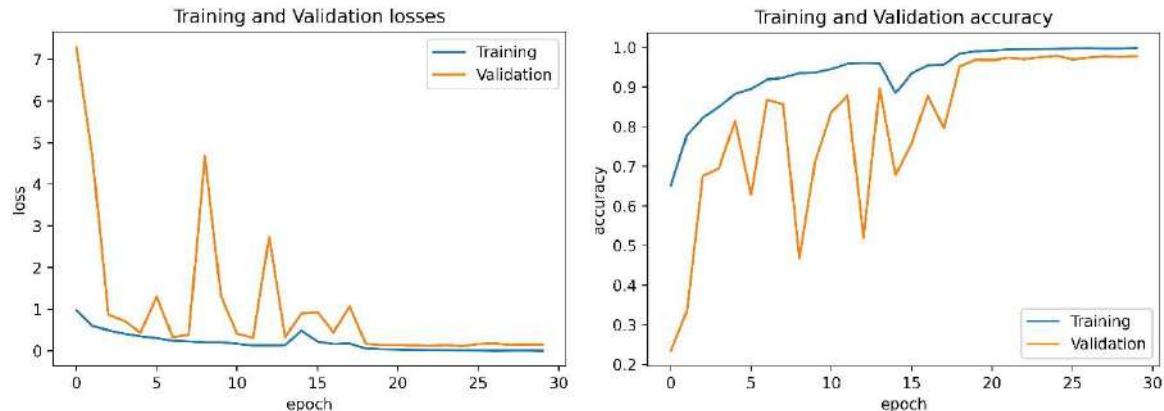


Figure 4-159 training loss and accuracy curves

Table 4-135 version 3 EXP (1) classification report

index	Class name	Precision	Recall	F1-score	Support
0	Glioma	0.993	0.95	0.971	300
1	Meningioma	0.9489	0.9706	0.9596	306
2	No tumor	0.9901	0.9926	0.9914	405
3	Pituitary	0.9803	0.9967	0.9884	300
<hr/>					
accuracy				0.9786	897
macro avg		0.9781	0.9775	0.9776	897
weighted avg		0.9789	0.9786	0.9786	897

## EXP. (2)

Number of heads = 2

Table 4-136 version 3 EXP (2) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	97.71%	97.71%	97.48%	97.71%	0.9948	97.57%	96.93%

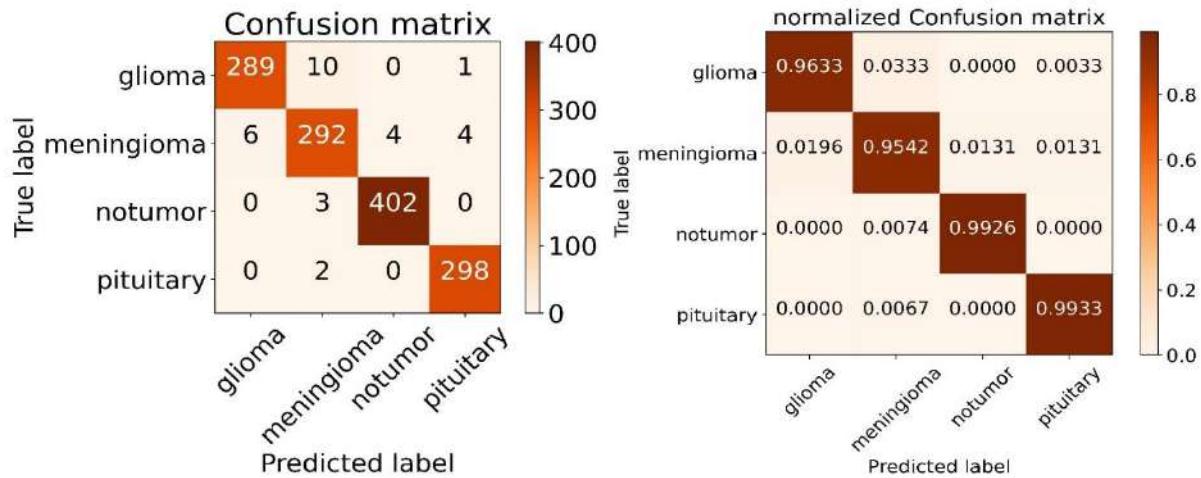


Figure 4-160 version 3 EXP (2) confusion matrices

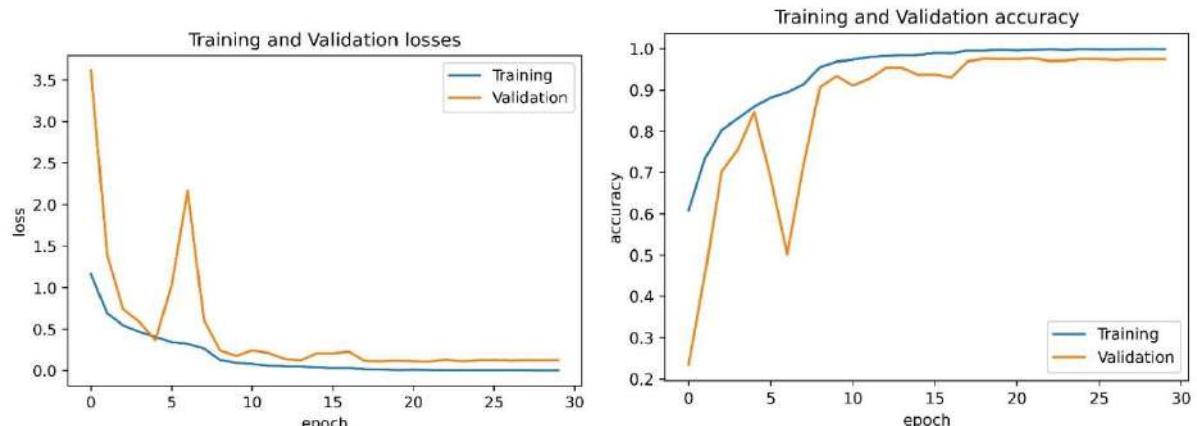


Figure 4-161 version 3 EXP (2) training loss and accuracy curves

Table 4-137 version 3 EXP (3) classification report

index	Class Name	precision	recall	f1-score	support
0	Glioma	0.9797	0.9633	0.9714	300
1	Meningioma	0.9511	0.9542	0.9527	306
2	No tumor	0.9901	0.9926	0.9914	405
3	Pituitary	0.9835	0.9933	0.9884	300
<hr/>					
accuracy				0.9771	1311
macro avg		0.9761	0.9759	0.9760	1311
weighted avg		0.9771	0.9771	0.9771	1311

### EXP. (3)

Number of heads = 4

Table 4-138 version 3 EXP (3) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	97.33%	97.33%	97.25%	97.33%	0.9945	97.12%	97.42%

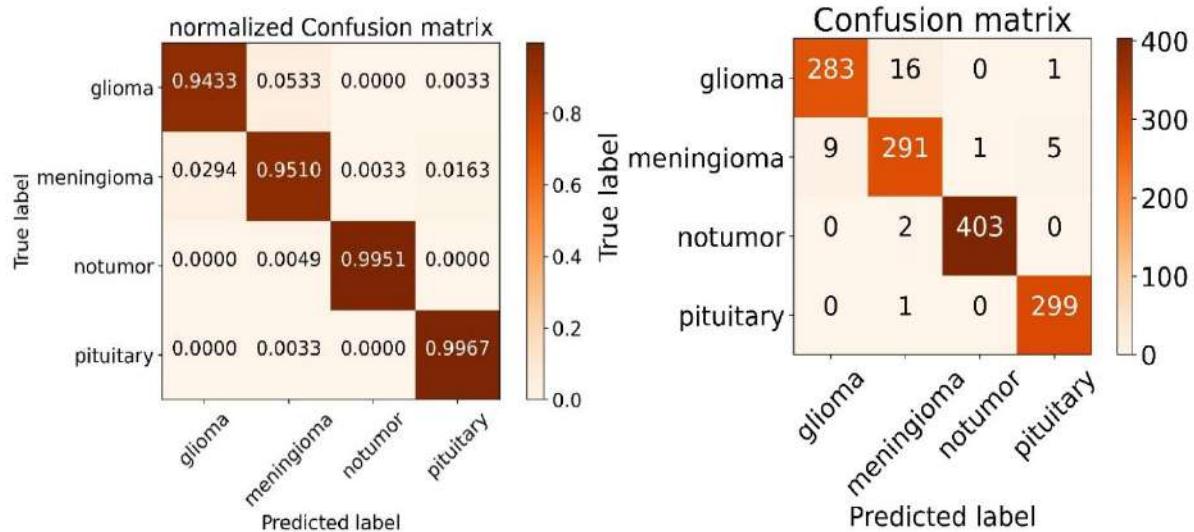


Figure 4-162 version 3 EXP (3) confusion matrices

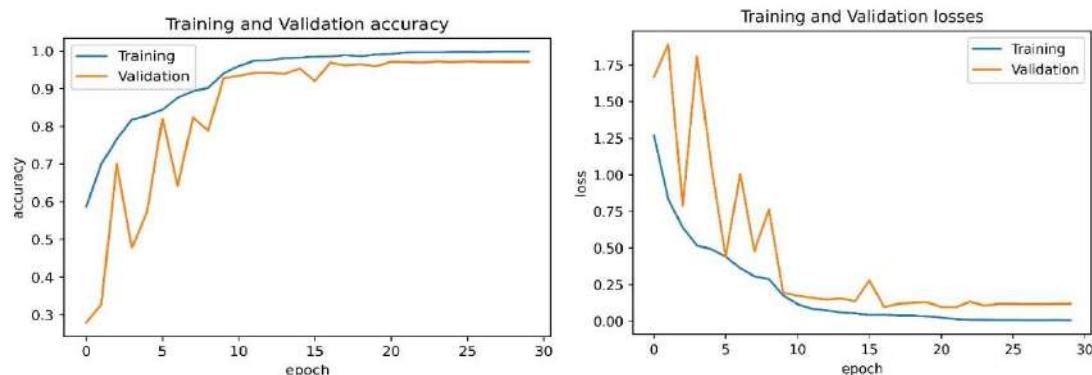


Figure 4-163 version 3 EXP (3) training loss and accuracy curves

Table 4-139 version 3 EXP (3) classification report

index	Class Name	precision	recall	f1-score	support
0	Glioma	0.9692	0.9433	0.9561	300
1	Meningioma	0.9387	0.951	0.9448	306
2	No tumor	0.9975	0.9951	0.9963	405
3	Pituitary	0.9803	0.9967	0.9884	300
<hr/>					
accuracy				0.9733	1311
macro avg		0.9714	0.9715	0.9714	1311
weighted avg		0.9733	0.9733	0.9733	1311

## EXP. (4)

Number of heads = 1-4

Table 4-140 version 3 EXP (4) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	97.71%	97.71%	97.64%	97.71%	0.9945	97.52%	96.93%

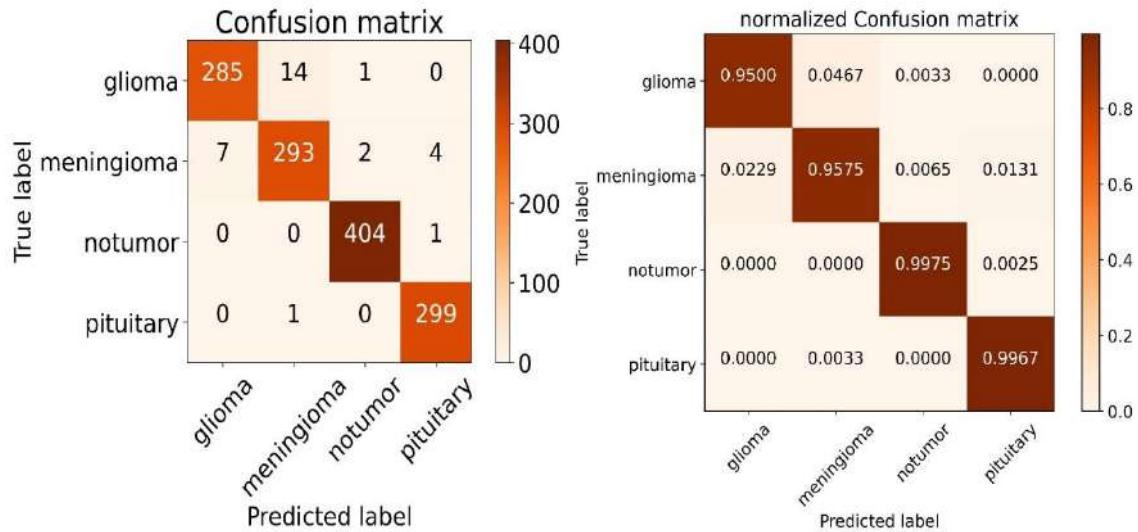


Figure 4-164 version 3 EXP (4) confusion matrices

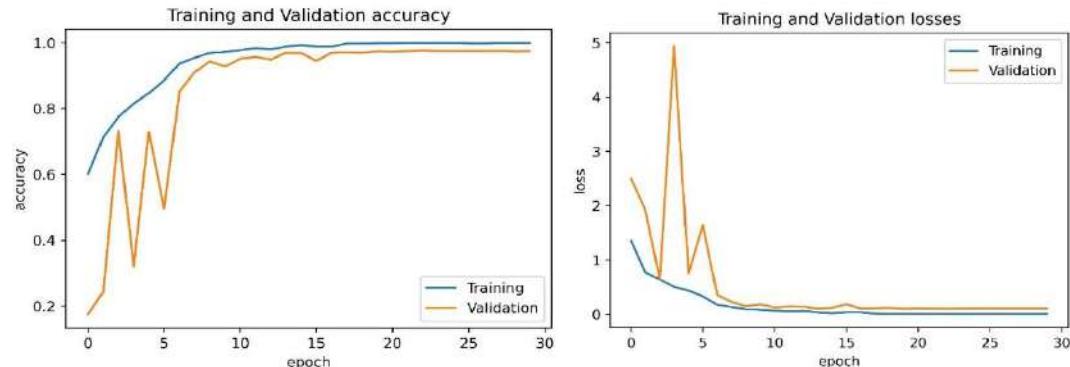


Figure 4-165 version 3 EXP (4) training loss and accuracy curves

Table 4-141 version 3 EXP (4) classification report

index	Class Name	precision	recall	f1-score	support
0	Glioma	0.9760	0.9500	0.9628	300
1	Meningioma	0.9513	0.9575	0.9544	306
2	No tumor	0.9926	0.9975	0.9951	405
3	Pituitary	0.9836	0.9967	0.9901	300
accuracy					0.9771
macro avg					0.9756
weighted avg					0.9771

## EXP. (5)

Number of heads = 2-4

Table 4-142 version 3 EXP (5) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	98.09%	98.17%	98.02%	98.09%	0.9958	97.99%	97.44%

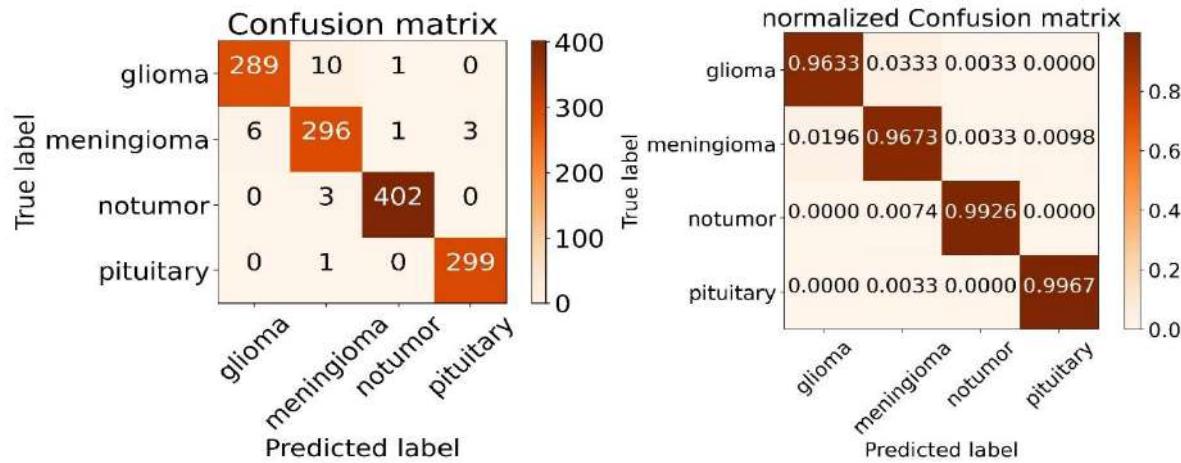


Figure 4-166 version 3 EXP (5) confusion matrices

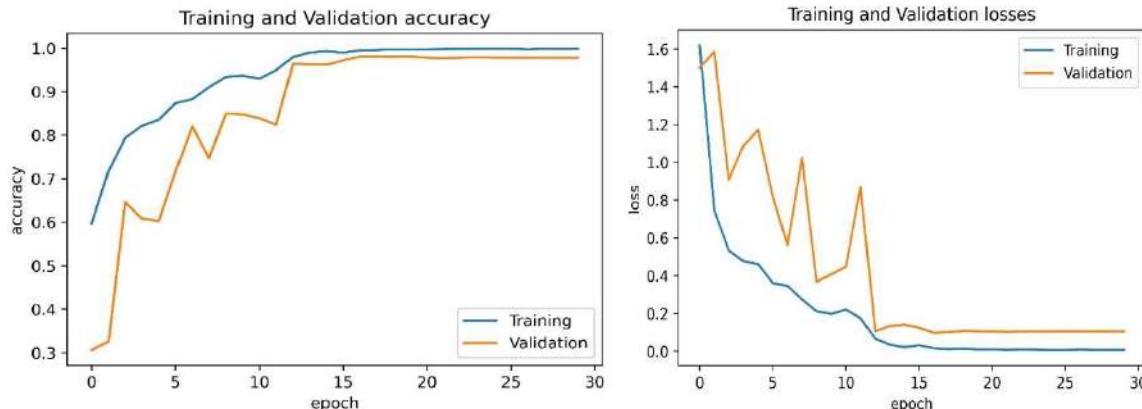


Figure 4-167 version 3 EXP (5) training loss and accuracy curves

Table 4-143 version 3 EXP (5) classification report

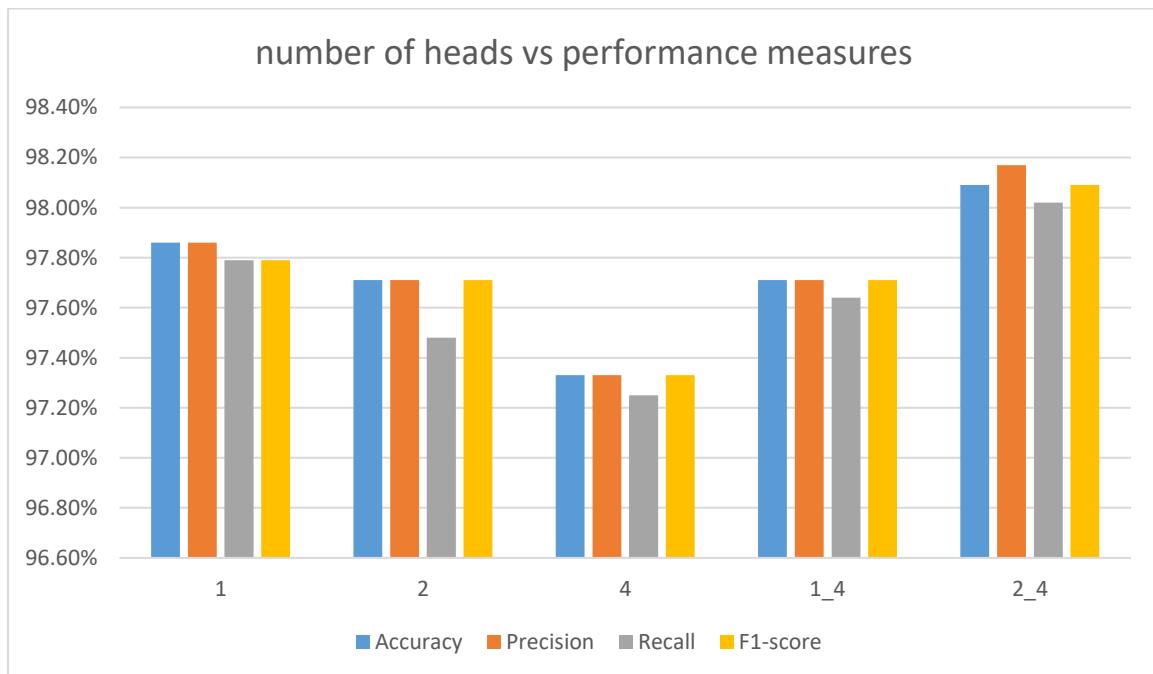
index	Class Name	precision	recall	f1-score	support
0	Glioma	0.9797	0.9633	0.9714	300
1	Meningioma	0.9548	0.9673	0.9610	306
2	No tumor	0.9950	0.9926	0.9938	405
3	Pituitary	0.9901	0.9967	0.9934	300
accuracy					0.9809
macro avg					0.9799
weighted avg					0.9809
					1311

## Tuning the number of heads of the MHA layer

Table 4-144 tuning the number of heads

No. of heads	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
<b>Single multi-head attention branch</b>							
<b>1</b>	97.86%	97.86%	97.79%	97.79%	0.9935	97.57%	97.03%
<b>2</b>	97.71%	97.71%	97.48%	97.71%	0.9948	97.57%	96.93%
<b>4</b>	97.33%	97.33%	97.25%	97.33%	0.9945	97.12%	97.42%
<b>Parallel multi-head attention branches</b>							
<b>1-4</b>	97.71%	97.71%	97.64%	97.71%	0.9945	97.52%	96.93%
<b>2-4</b>	98.09%	98.17%	98.02%	98.09%	0.9958	97.99%	97.44%

The best results was achieved when using a parallel branch with 2 heads for the first branch and 4 heads for the second with 98.09% accuracy but not higher than the CNN alone, the least results was associated with single branch of 4 heads with 97.33% accuracy.



## **Version 4**

In this version an adaptive feature recalibration (AFR) layer was utilized.

### **Data preprocessing**

All images were resized to 64\*64 and normalized, horizontal flip augmentation method was applied

### **Proposed framework**

Same as version 2 but with a MHA layer after the last max-pooling layer.

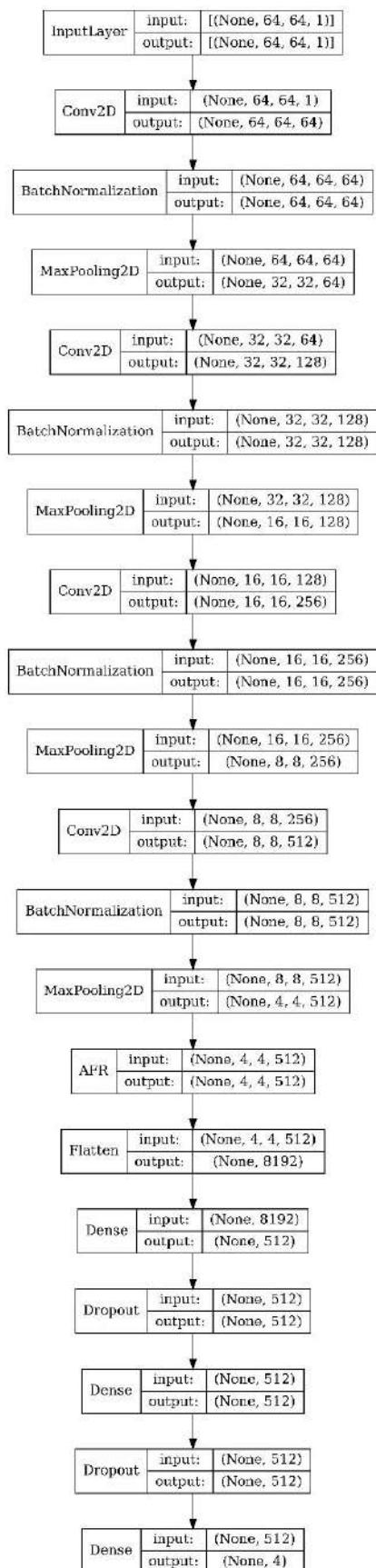


Figure 4-168 version 4 proposed model architecture

## EXP. (1)

Reduction ratio = 4

Table 4-145 version 4 EXP (1) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	98.02%	98.02%	98.02%	98.02%	0.9971	97.88%	97.34%

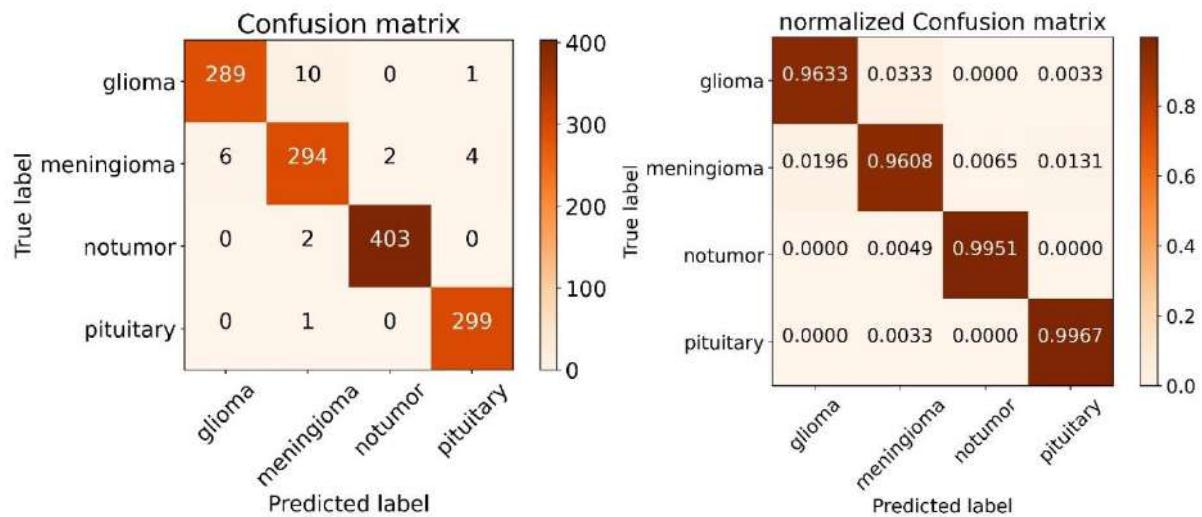


Figure 4-169 version 4 EXP (1) confusion matrices

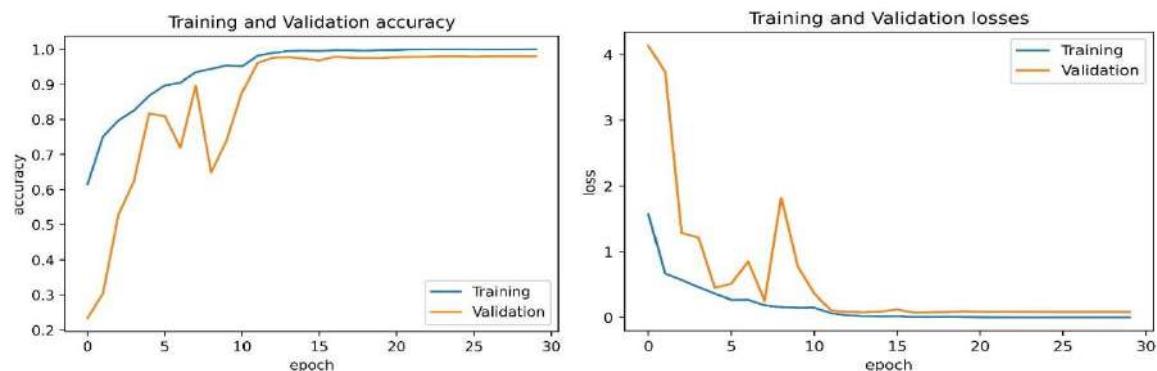


Figure 4-170 version 4 EXP (1) training loss and accuracy curves

Table 4-146 version 4 EXP. (1) classification report

index	Class Name	precision	recall	f1-score	support
0	glioma	0.9797	0.9633	0.9714	300
1	meningioma	0.9577	0.9608	0.9592	306
2	notumor	0.9951	0.9951	0.9951	405
3	pituitary	0.9836	0.9967	0.9901	300
accuracy				0.9802	1311
macro avg		0.9790	0.9790	0.9789	1311
weighted avg		0.9802	0.9802	0.9801	1311

## EXP. (2)

Reduction ratio = 8

Table 4-147 version 4 EXP (2) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	98.25%	98.32%	98.25%	98.25%	0.9989	97.11%	97.65%

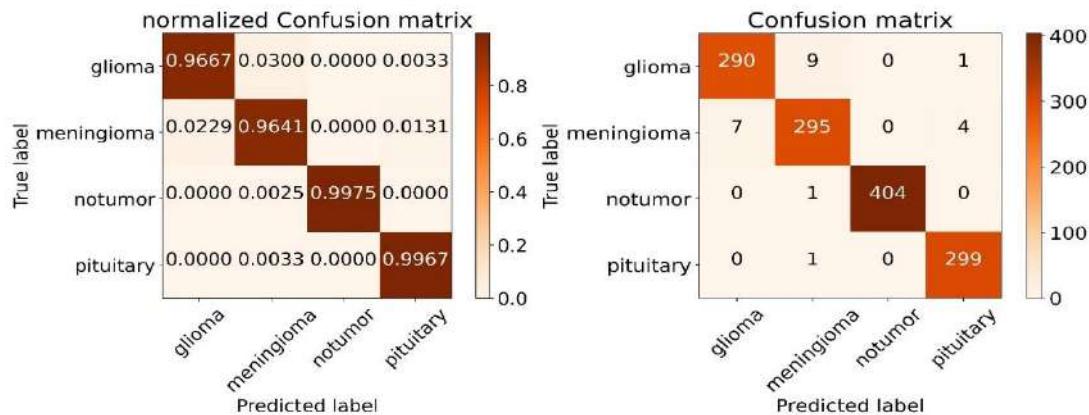


Figure 4-171 version 4 EXP (2) confusion matrices

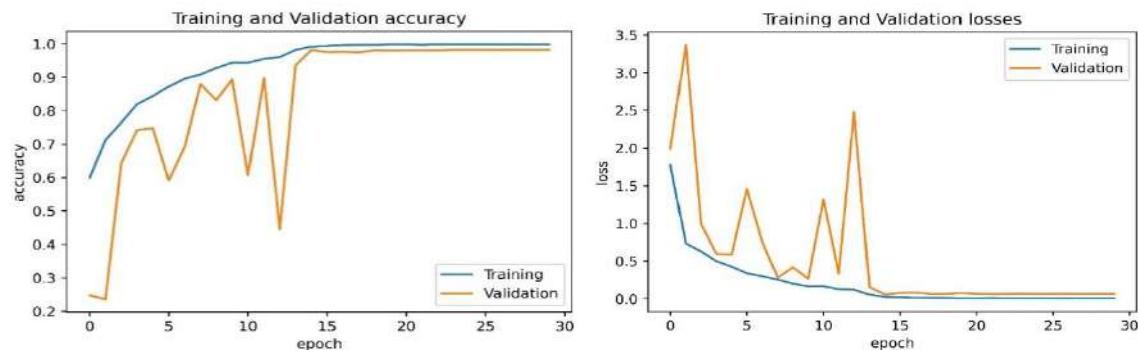


Figure 4-172 version 4 EXP (2) training loss and accuracy curves

Table 4-148 version 4 EXP (2) classification report

index	Class Name	precision	recall	f1-score	support
0	glioma	0.9764	0.9667	0.9715	300
1	meningioma	0.9641	0.9641	0.9641	306
2	notumor	1	0.9975	0.9988	405
3	pituitary	0.9836	0.9967	0.9901	300
accuracy				0.9825	1311
macro avg		0.9810	0.9812	0.9811	1311
weighted avg		0.9825	0.9825	0.9824	1311

## EXP. (3)

Reduction ratio = 12

Table 4-149 version 4 EXP (3) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	98.09%	98.09%	98.09%	98.09%	0.9970	97.92%	97.44%

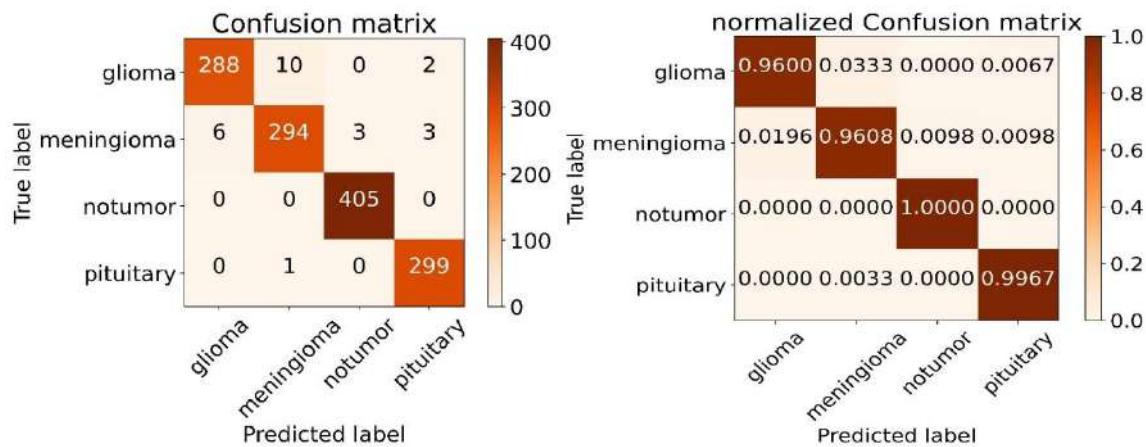


Figure 4-173 version 4 EXP (3) confusion matrices

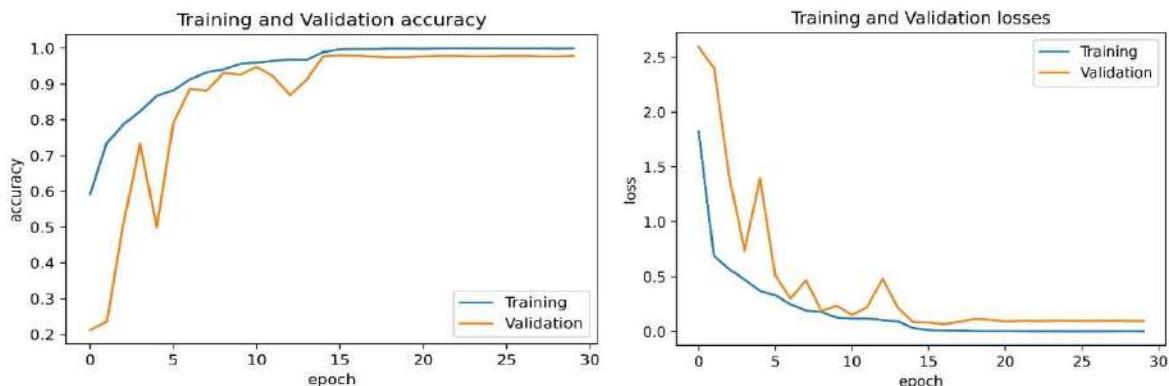


Figure 4-174 version 4 EXP (3) training loss and accuracy curves

Table 4-150 version 4 EXP (3) classification report

index	Class Name	precision	recall	f1-score	support
0	glioma	0.9796	0.96	0.9697	300
1	meningioma	0.9639	0.9608	0.9624	306
2	notumor	0.9926	1	0.9963	405
3	pituitary	0.9836	0.9967	0.9901	300
<hr/>					
accuracy				0.9809	1311
macro avg		0.9799	0.9794	0.9796	1311
weighted avg		0.9809	0.9809	0.9809	1311

## EXP. (4)

Reduction ratio = 16

Table 4-151 version 4 EXP (4) testing results

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	98.32%	98.32%	98.25%	98.32%	0.9990	97.17%	97.75%

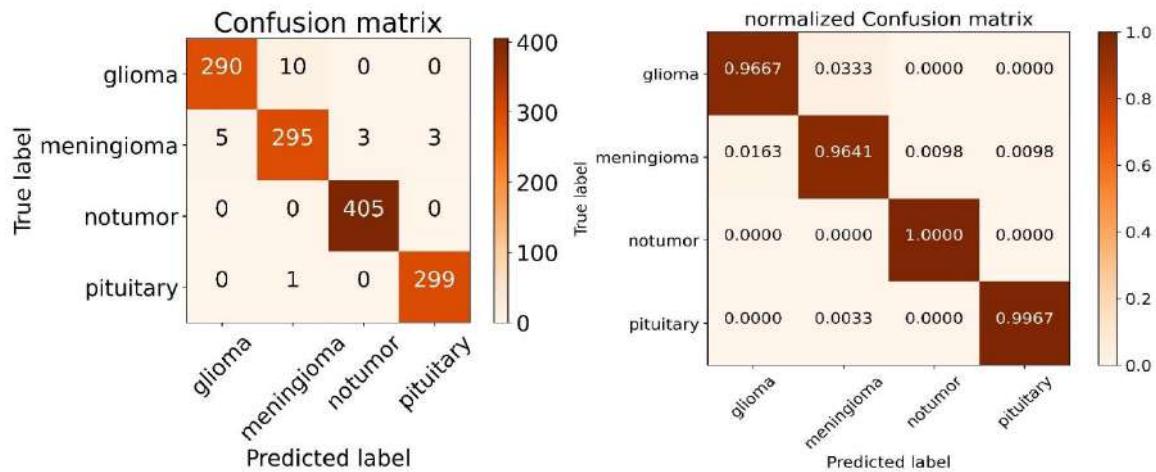


Figure 4-175 version 4 EXP (4) confusion matrices

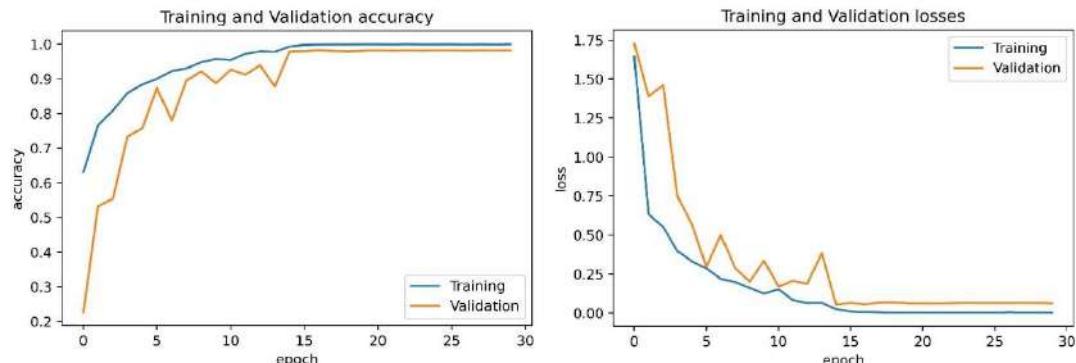


Figure 4-176 version 4 EXP (4) training loss and accuracy curves

Table 4-152 version 4 EXP (4) classification report

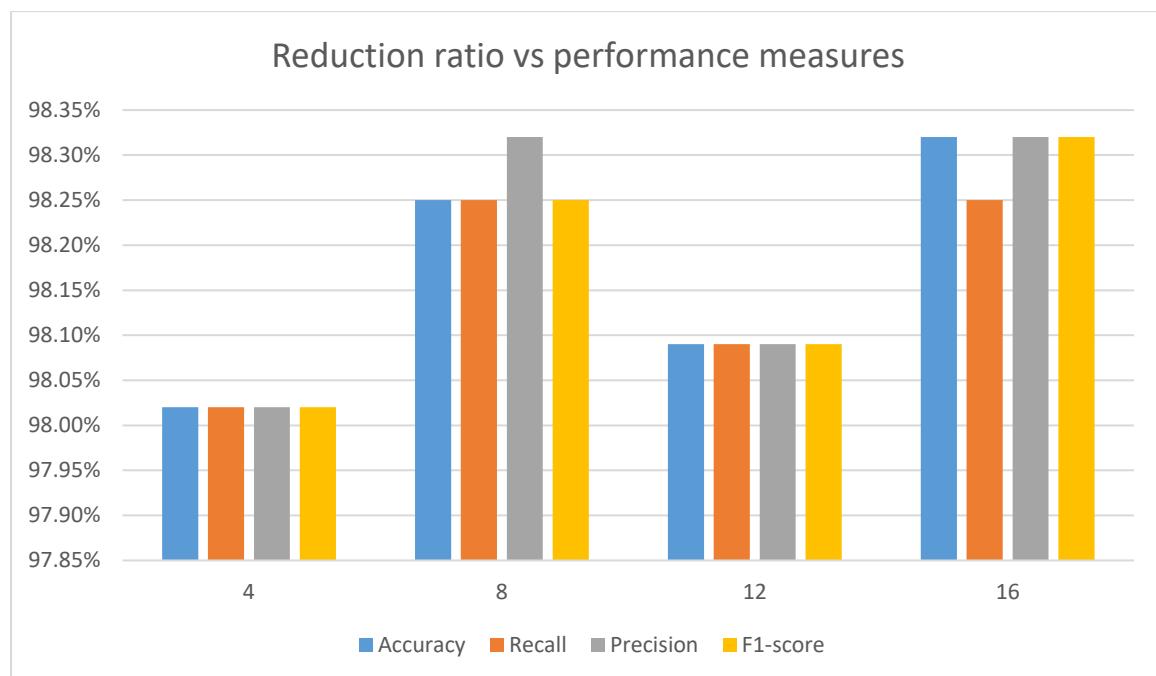
index	Class Name	precision	recall	f1-score	support
0	glioma	0.9831	0.9667	0.9748	300
1	meningioma	0.9641	0.9641	0.9641	306
2	notumor	0.9926	1	0.9963	405
3	pituitary	0.9901	0.9967	0.9934	300
accuracy				0.9832	1311
macro avg		0.9825	0.9818	0.9821	1311
weighted avg		0.9832	0.9832	0.9832	1311

## Tuning the reduction ratio of the AFR layer

Table 4-153 tuning the reduction ratio

Reduction ratio	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
4	98.02%	98.02%	98.02%	98.02%	0.9971	97.88%	97.34%
8	98.25%	98.32%	98.25%	98.25%	0.9989	97.11%	97.65%
12	98.09%	98.09%	98.09%	98.09%	0.9970	97.92%	97.44%
16	98.32%	98.32%	98.25%	98.32%	0.9990	97.17%	97.75%

The best results was achieved with a reduction ratio of 8 but not higher than the CNN alone, and the least results was associated with reduction ratio of 4.



## **Version 5**

In this version an adaptive kernel size convolution (AKSC) layer was utilized

### **Data preprocessing**

All images were resized to 64\*64 and normalized, horizontal flip augmentation method was applied

### **Proposed framework**

Same as version 2 but with an AKSC layer instead of Conv2D layers.

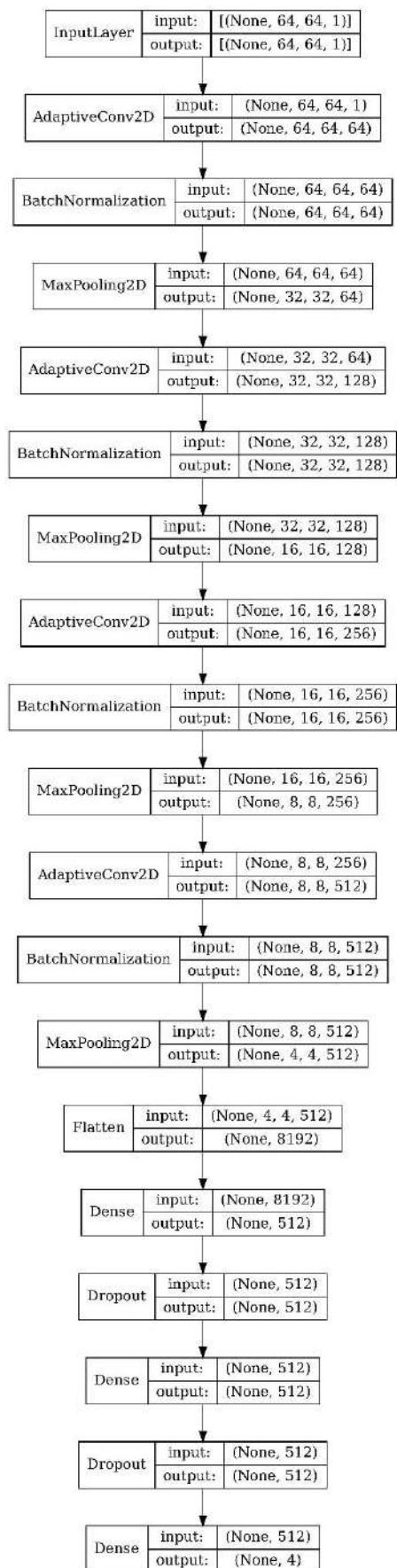


Figure 4-177 version 5 proposed network architecture

Table 4-154 testing results for version 5

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	98.17%	98.17%	98.17%	98.17%	0.9971	98.00%	97.54%

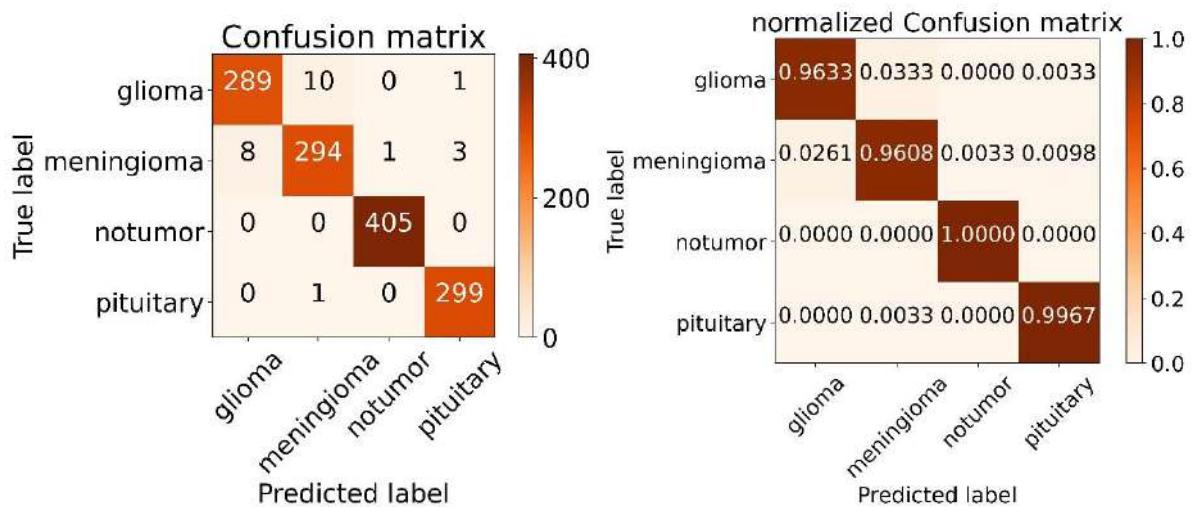


Figure 4-178 version 5 confusion matrices

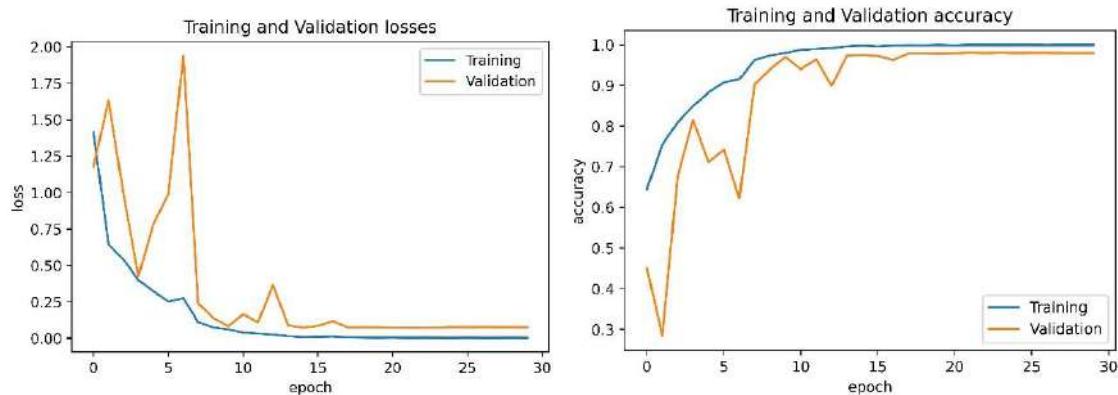


Figure 4-179 version 5 training loss and accuracy curves

Table 4-155 version 5 classification report

index	Class Name	precision	recall	f1-score	support
0	glioma	0.9731	0.9633	0.9682	300
1	meningioma	0.9639	0.9608	0.9624	306
2	notumor	0.9975	1	0.9988	405
3	pituitary	0.9868	0.9967	0.9917	300
<hr/>					
accuracy				0.9817	1311
macro avg		0.9803	0.9802	0.9803	1311
weighted avg		0.9816	0.9817	0.9817	1311

## **Version 6**

In this version a combination of CNN , AFR and MHA was utilized

### **Data preprocessing**

All images were resized to 64\*64 and normalized, horizontal flip augmentation method was applied

### **Proposed framework**

The same model from version 2 was utilized with the AFR layer and the MHA layer tuned from the previous experiments.

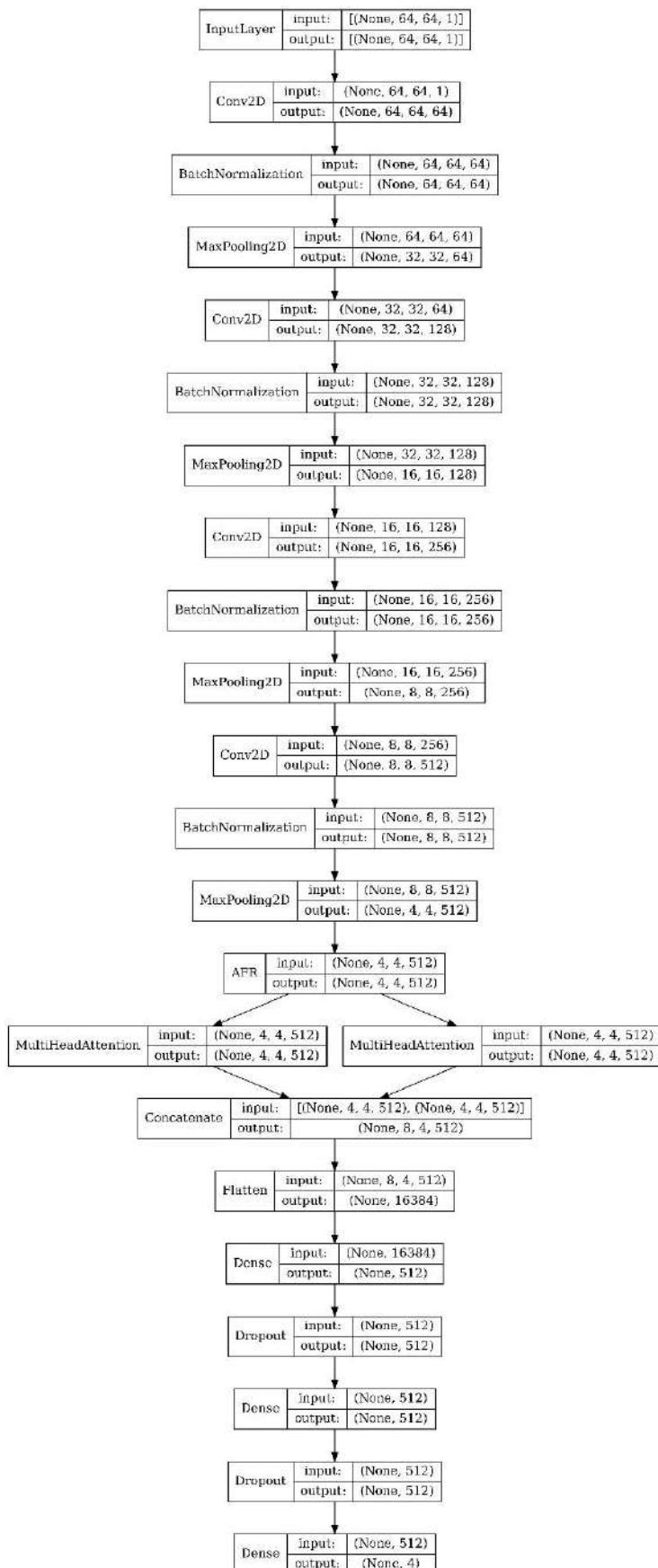


Figure 4-180 version 6 proposed network architecture

Table 4-156 testing results for version 6

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	96.19%	96.24%	95.58%	96.19%	0.9932	95.87%	94.88%

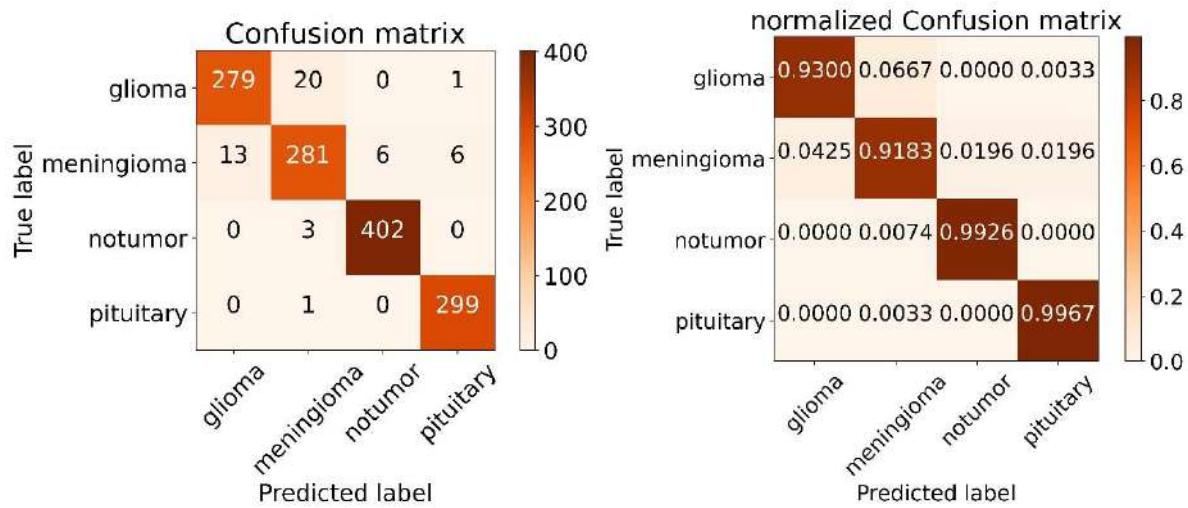


Figure 4-181 version 6 confusion matrices

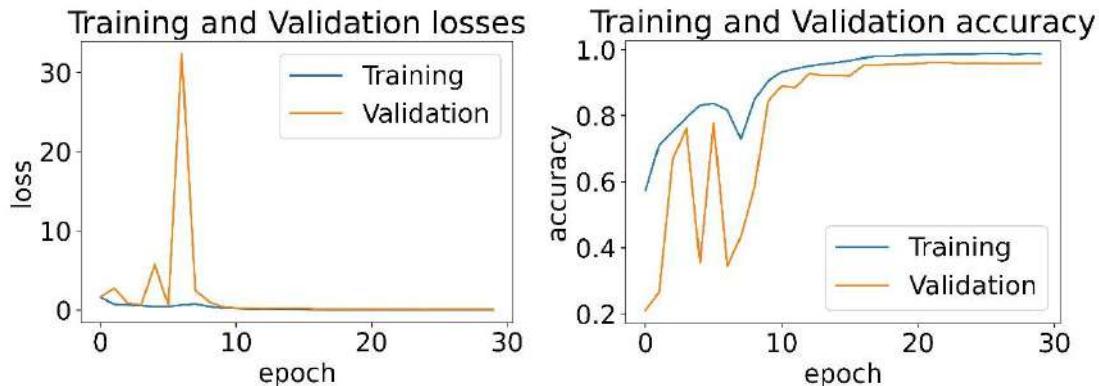


Figure 4-182 version 6 training loss and accuracy curves

Table 4-157 version 6 classification report

index	Class Name	precision	recall	f1-score	support
0	Glioma	0.9555	0.93	0.9426	300
1	Meningioma	0.9213	0.9183	0.9198	306
2	No-tumor	0.9853	0.9926	0.9889	405
3	Pituitary	0.9771	0.9967	0.9868	300
<hr/>					
accuracy				0.9619	1311
macro avg		0.9598	0.9594	0.9595	1311
weighted avg		0.9617	0.9619	0.9617	1311

## **Version 7**

In this version a combination of AKSC and MHA was utilized

### **Data preprocessing**

All images were resized to 64\*64 and normalized, horizontal flip augmentation method was applied

### **Proposed framework**

The same model from version 2 was utilized with the AKSC layers and the MHA layer tuned from the previous experiments.

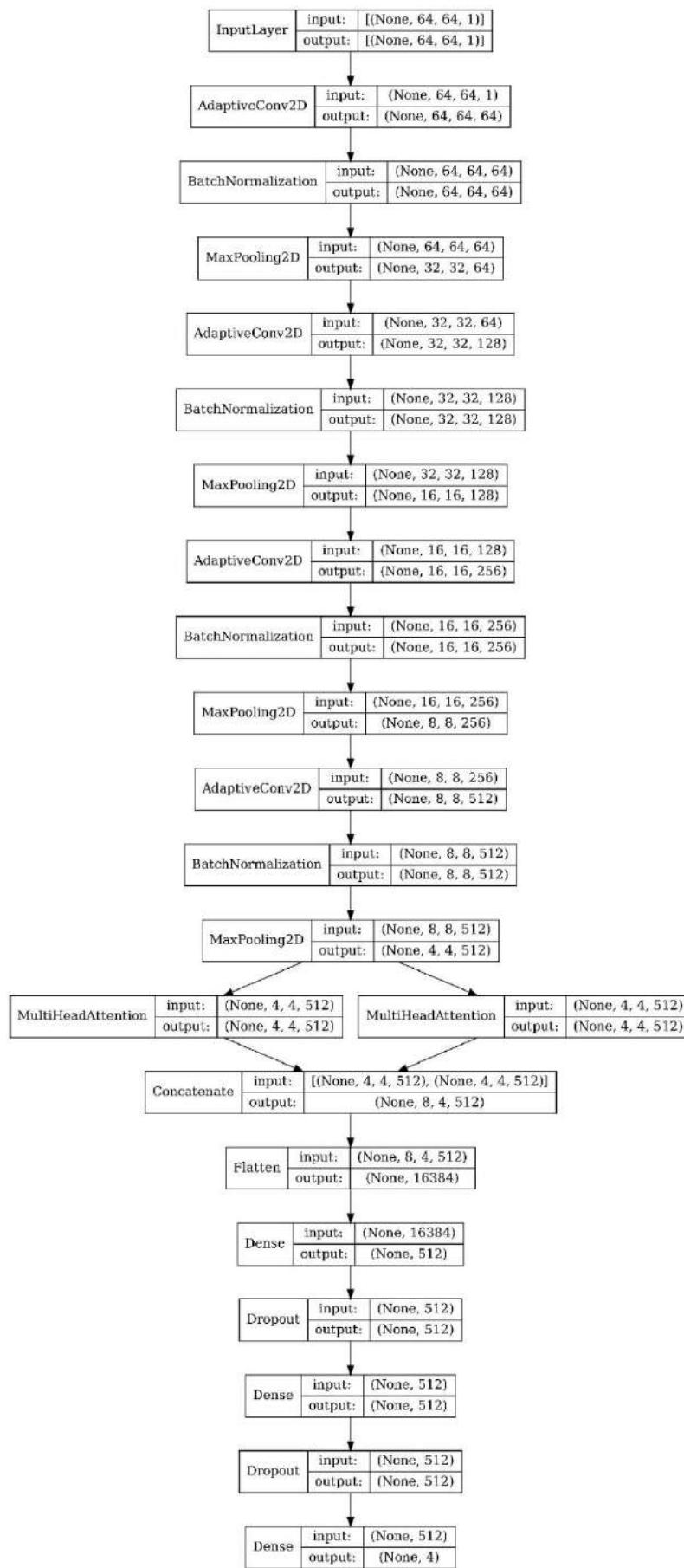


Figure 4-183 version 7 proposed network architecture

Table 4-158 testing results for version 7

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	95.73%	95.98%	94.66%	95.73%	0.9954	95.40%	94.26%

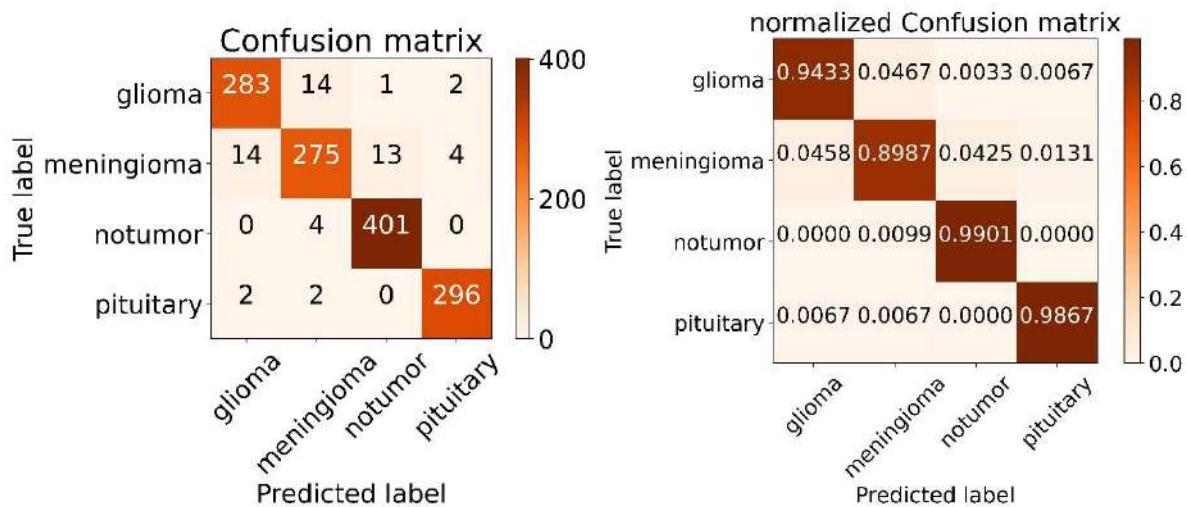


Figure 4-184 version 7 confusion matrices

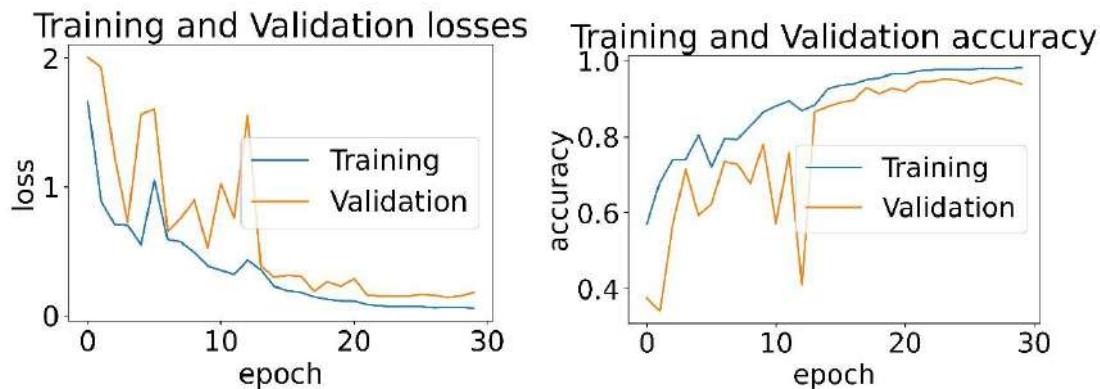


Figure 4-185 version 7 training loss and accuracy curves

Table 4-159 version 7 classification report

index	Class Name	precision	recall	f1-score	support
0	Glioma	0.9465	0.9433	0.9449	300
1	Meningioma	0.9322	0.8987	0.9151	306
2	No-tumor	0.9663	0.9901	0.978	405
3	Pituitary	0.9801	0.9867	0.9834	300
<hr/>					
accuracy				0.9573	1311
macro avg		0.9563	0.9547	0.9554	1311
weighted avg		0.9570	0.9573	0.9570	1311

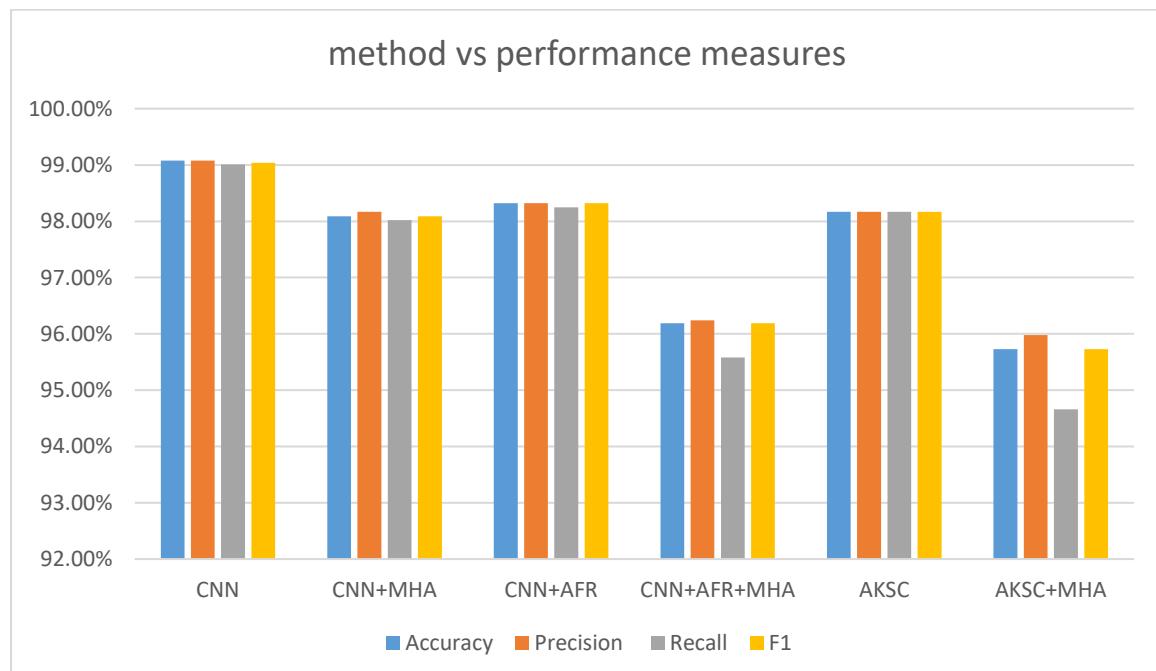
## Experiments summary

Method	Accuracy	Precision	Recall	F1	AUC	G-mean	Kappa
CNN	99.08%	99.08%	99.01%	99.04%	0.9998	98.34%	97.67%
CNN+MHA	98.09%	98.17%	98.02%	98.09%	0.9958	97.99%	97.44%
CNN+AFR	98.32%	98.32%	98.25%	98.32%	0.9990	97.17%	97.75%
CNN+AFR+MHA	96.19%	96.24%	95.58%	96.19%	0.9932	95.87%	94.88%
AKSC	98.17%	98.17%	98.17%	98.17%	0.9971	98.00%	97.54%
AKSC+MHA	95.73%	95.98%	94.66%	95.73%	0.9954	95.40%	94.26%

Table 4-160 Abbreviations

<b>LSTM</b>	Long-short time memory
<b>AFR</b>	Adaptive Feature Recalibration (squeeze and excitation)
<b>MHA</b>	Multi-Head Attention
<b>AKSC</b>	Adaptive kernel Size Convolution

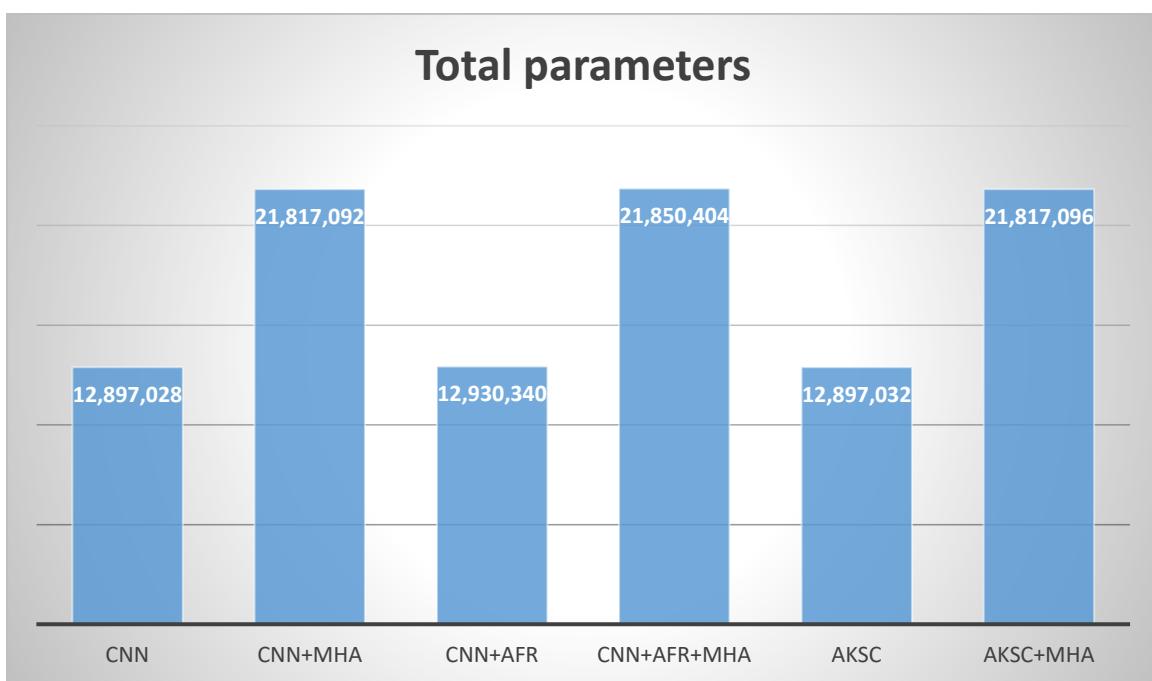
The best results was achieved by the CNN alone with accuracy of 99.08% and the least results was associated with the AKSC+MHA combination with 95.73% accuracy, out of all the added layers to the CNN the best results was achieved by the CNN+AFR combination with accuracy of 98.32% but not higher than the CNN alone.



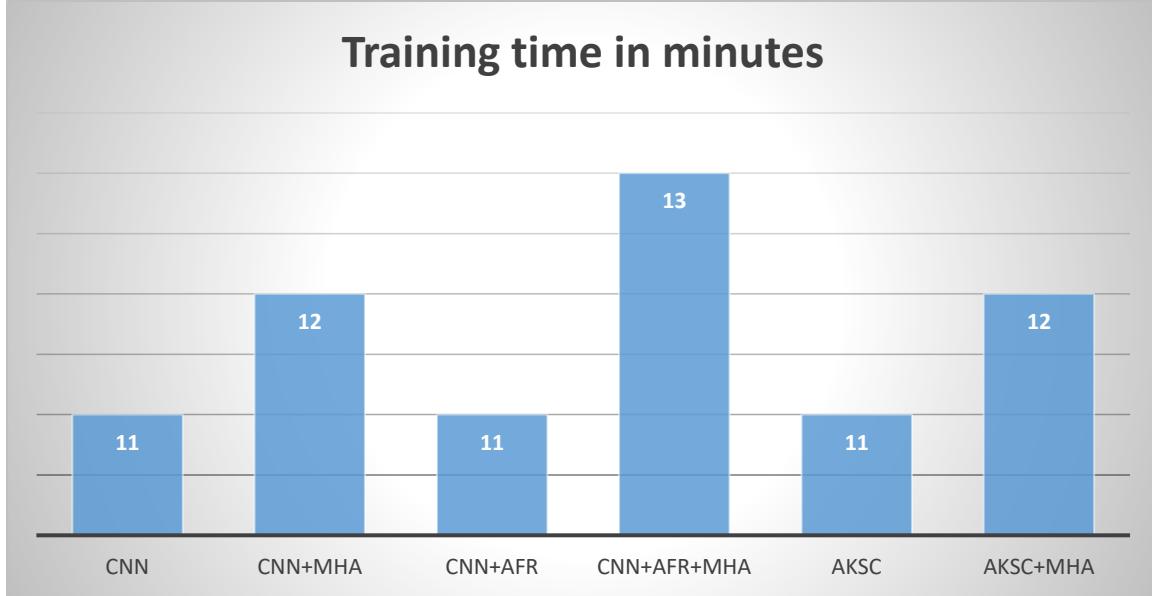
## Training time and number of parameters

Method	Training time	Total parameters	Trainable parameters	Non trainable parameters
CNN	11min 24s	12,897,028	12,895,108	1,920
CNN+MHA	12min 24s	21,817,092	21,815,172	1,920
CNN+AFR	11min 22s	12,930,340	12,928,420	1,920
CNN+AFR+MHA	13min	21,850,404	21,848,484	1,920
AKSC	11min	12,897,032	12,895,112	1,920
AKSC+MHA	12min 44s	21,817,096	21,815,176	1,920

**Total parameters**



**Training time in minutes**



#### 4.2.3.5 Final proposed framework

Out of all the experimented methods such as CNN, AKSC, AFR, MHA and their combinations the CNN alone had the best performance with an accuracy of 99.08%.

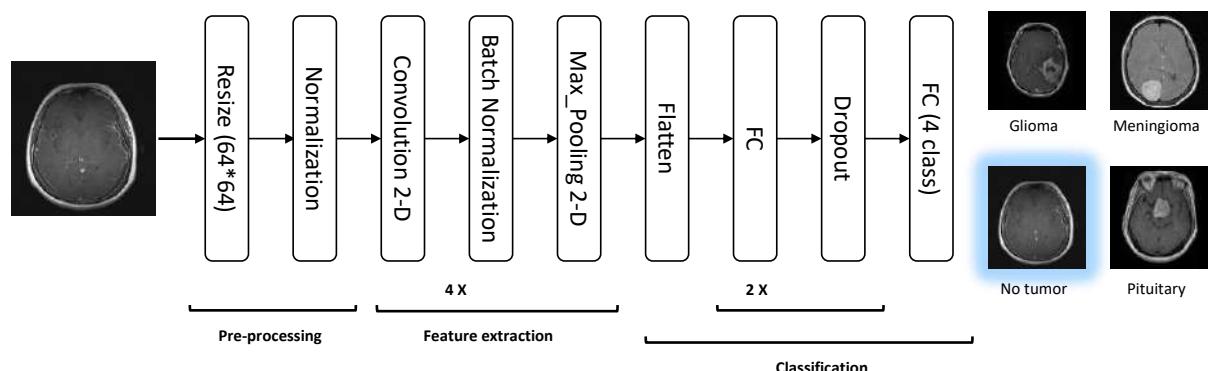


Figure 4-186 final proposed network architecture

#### 4.2.3.6 Method validation

For method validation a public dataset on kaggle named “Brain Tumor” which was uploaded by JAKESH BOHAJU was used to train and test the final proposed model’s performance.

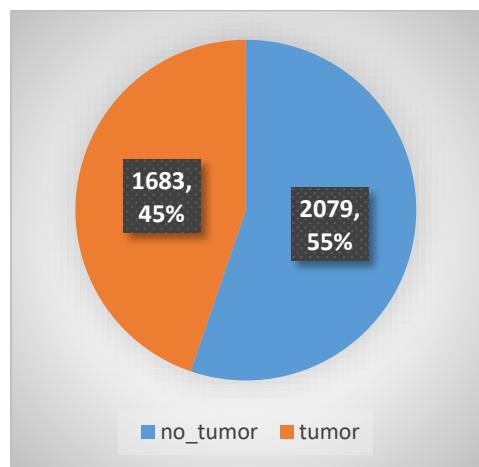


Table 4-161 test results

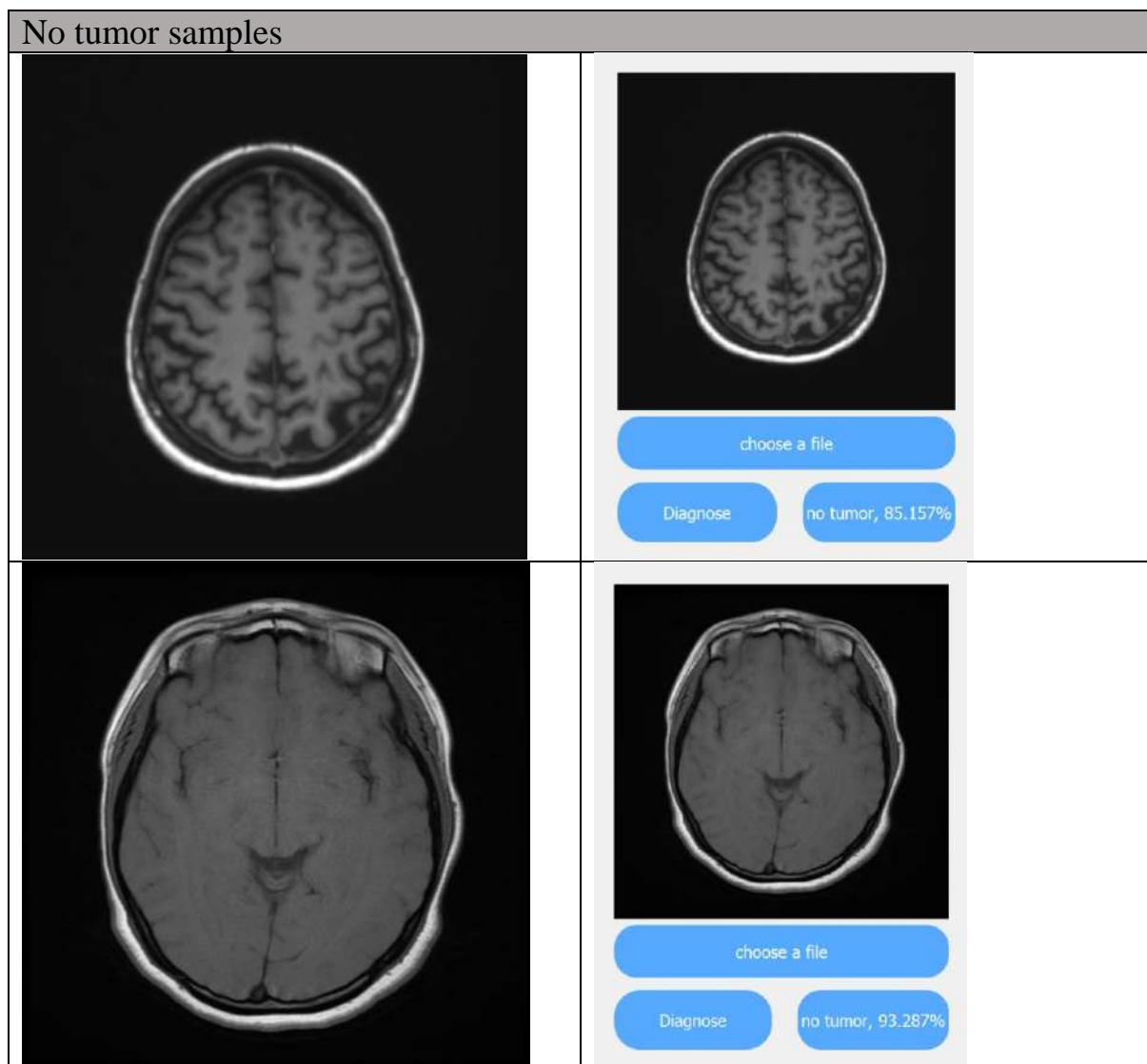
	Model	Accuracy	precision	Recall	F1	AUC
Proposed	CNN	98.14%	98.22%	98%	98.10%	99.07%
[84]	MobileNetV2	89%				

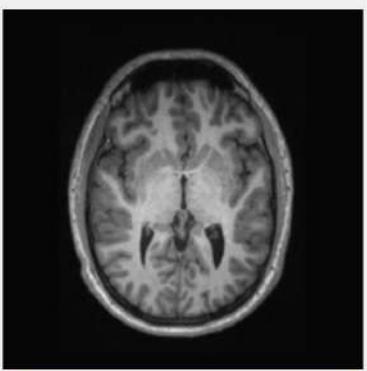
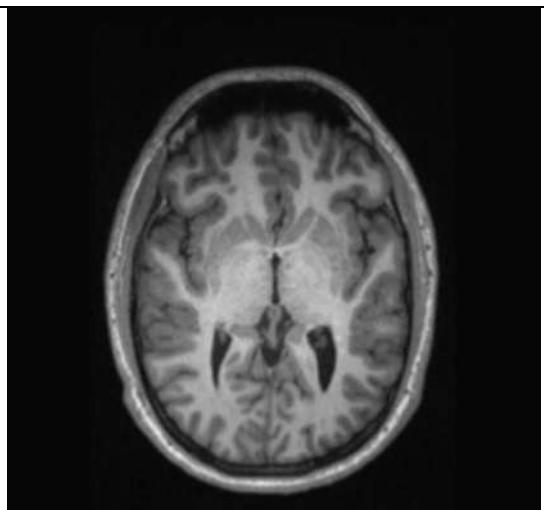
#### 4.2.3.7 Testing with real world cases (online testing)

After the training process is done and the final model was chosen to be the CNN alone which achieved 99.08% accuracy, it is time to test it on some samples completely out side of the dataset used in the training and testing, the model was tested with 8 sample for each class from another datasets with different source for the one used for training and only misdiagnosed 1 meningioma sample to be normal

The following table shows each sample on the left size and the model's diagnosis in the right side along with the confidence for the diagnosis.

Please note that the GUI will be discussed in details in section 4.3

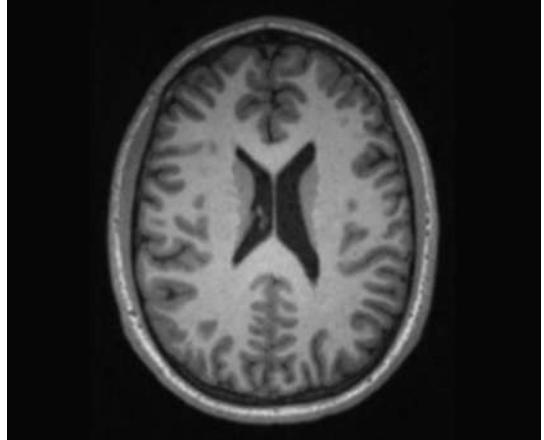




choose a file

Diagnose

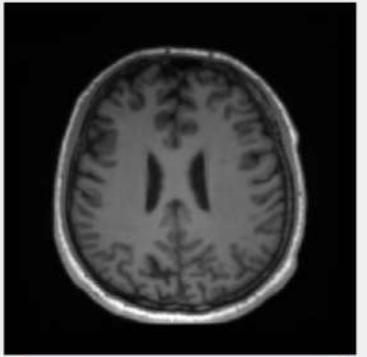
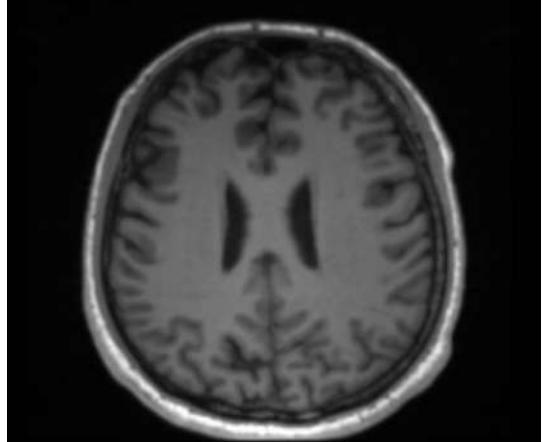
no tumor, 99.700%



choose a file

Diagnose

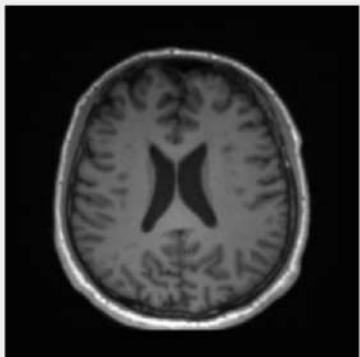
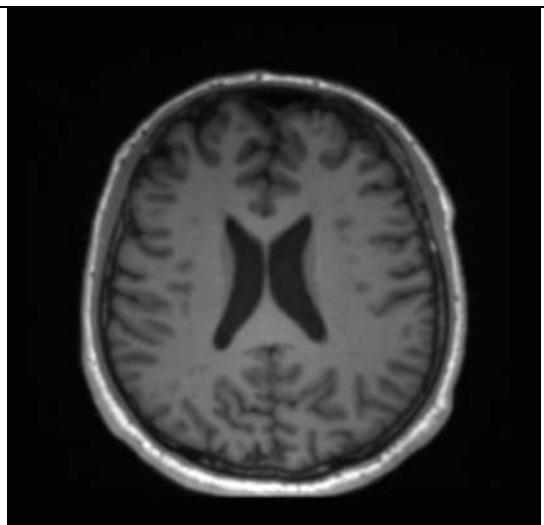
no tumor, 99.981%



choose a file

Diagnose

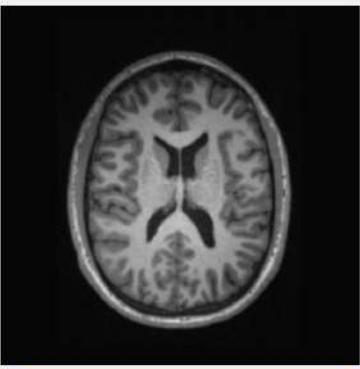
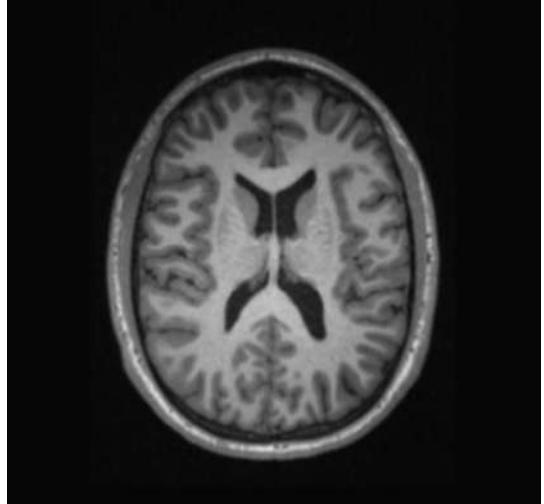
no tumor, 93.131%



choose a file

Diagnose

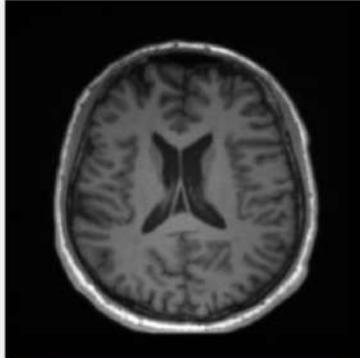
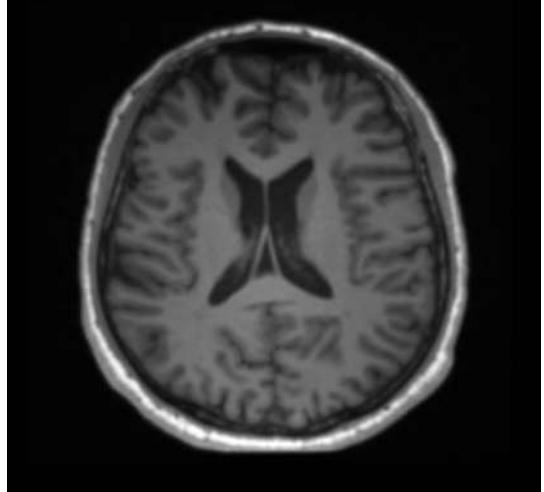
no tumor, 92.920%



choose a file

Diagnose

no tumor, 99.993%

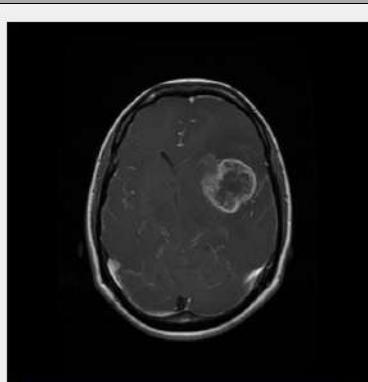
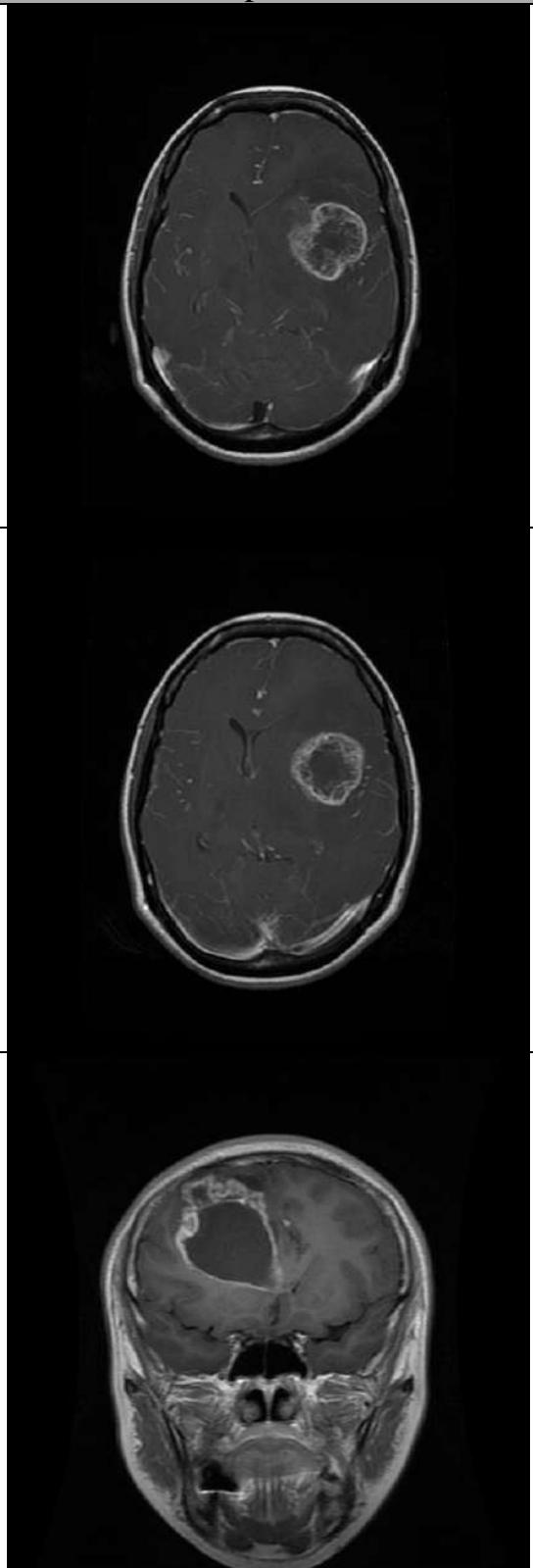


choose a file

Diagnose

no tumor, 96.029%

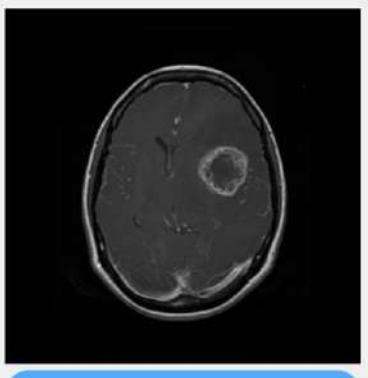
## Glioma test sample



choose a file

Diagnose

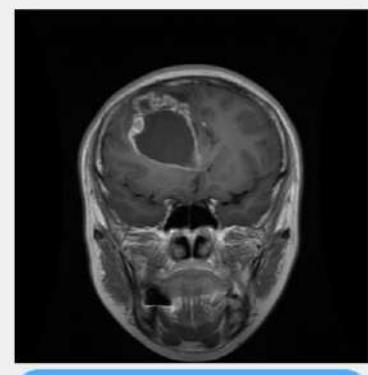
glioma, 98.092%



choose a file

Diagnose

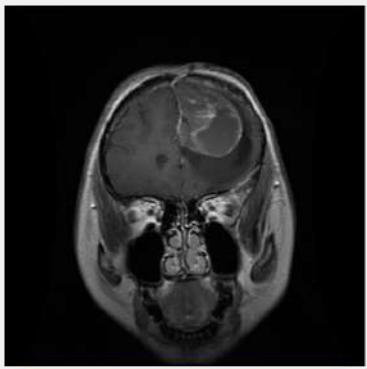
glioma, 99.959%



choose a file

Diagnose

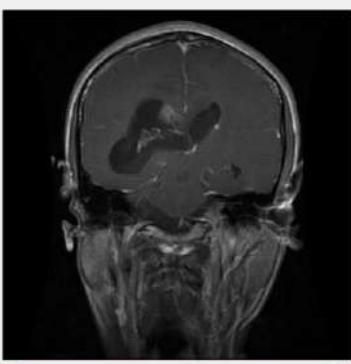
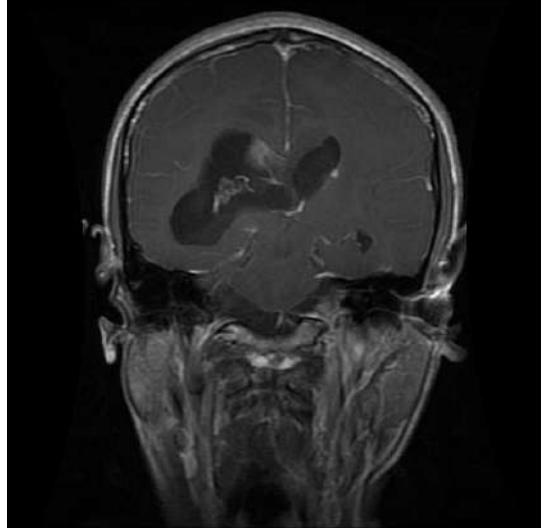
glioma, 99.939%



choose a file

Diagnose

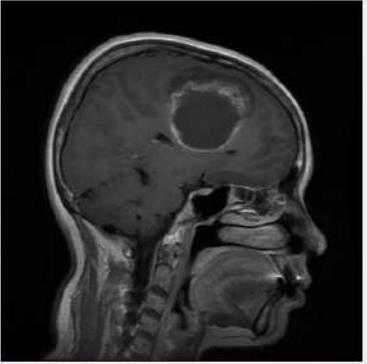
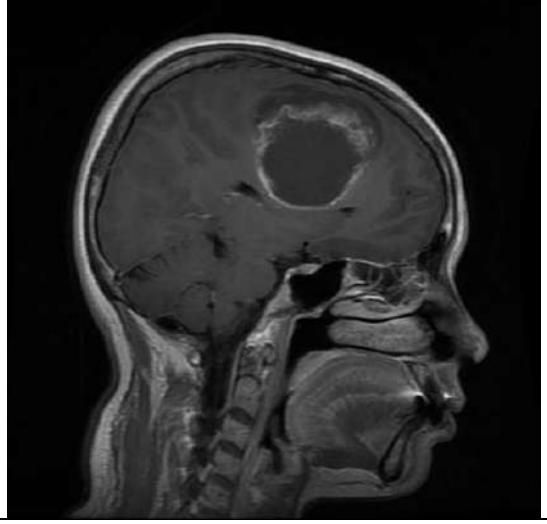
glioma, 99.995%



choose a file

Diagnose

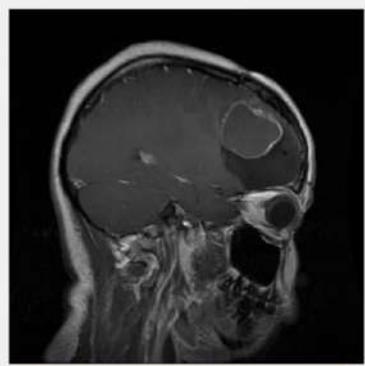
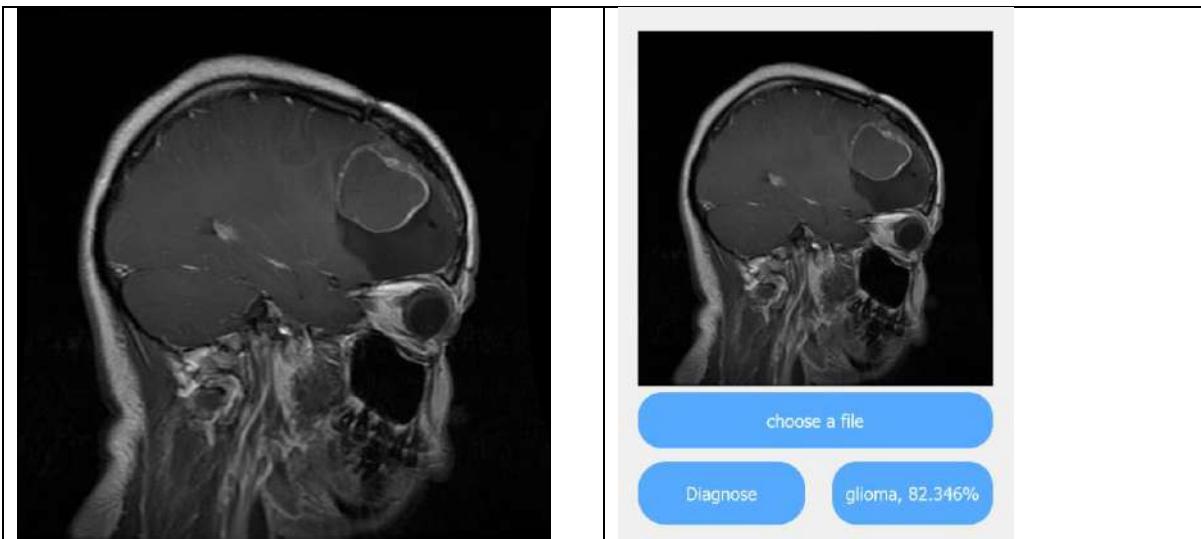
glioma, 99.952%



choose a file

Diagnose

glioma, 90.401%

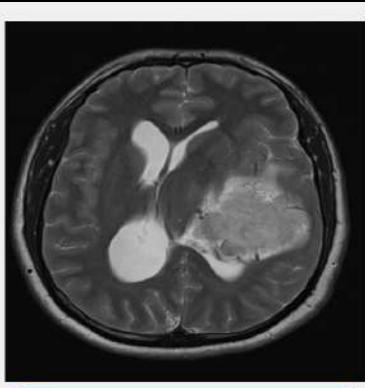
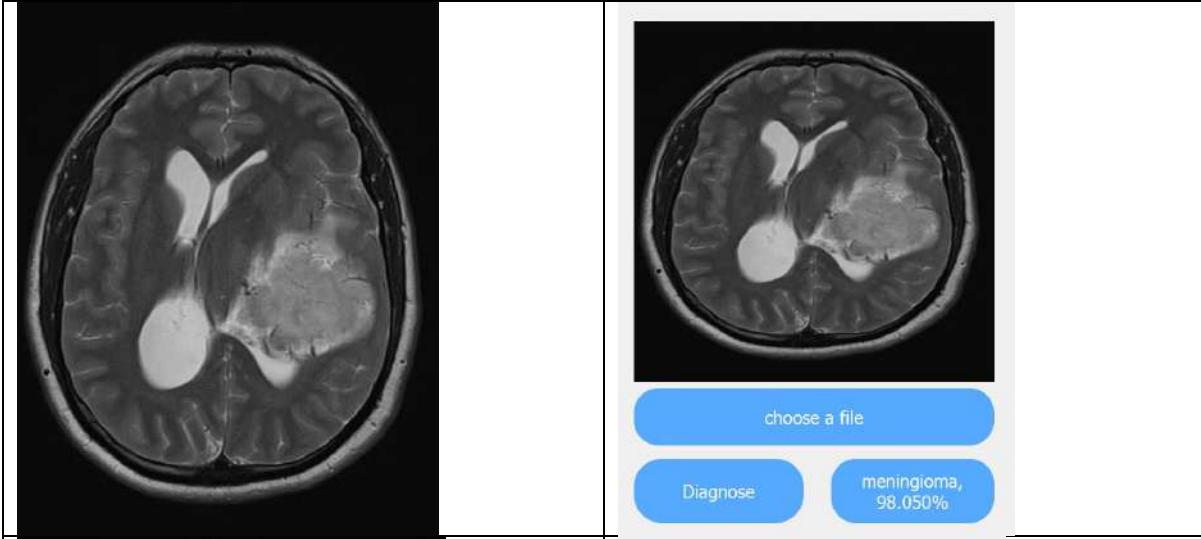


choose a file

Diagnose

glioma, 82.346%

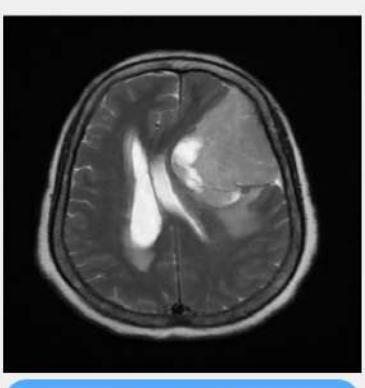
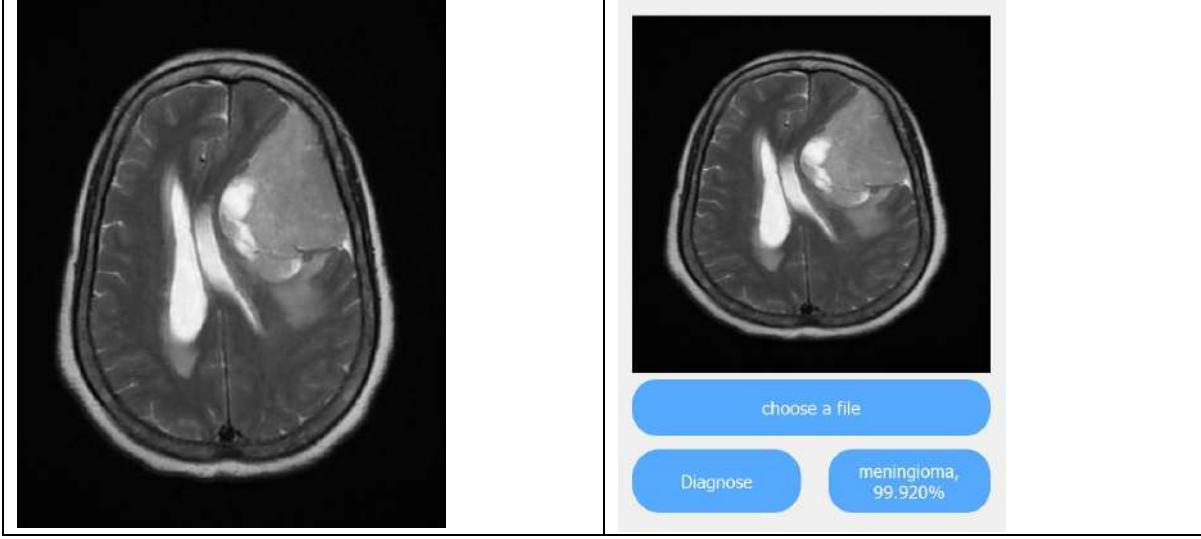
Meningioma test samples



choose a file

Diagnose

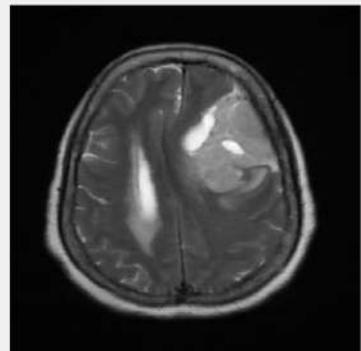
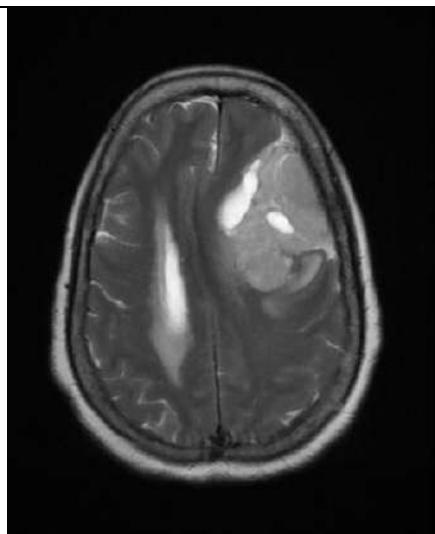
meningioma,  
98.050%



choose a file

Diagnose

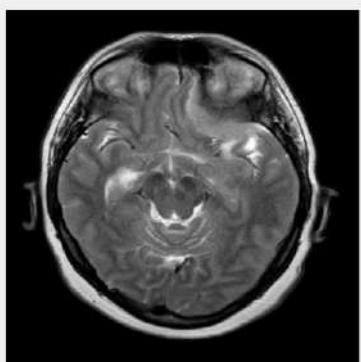
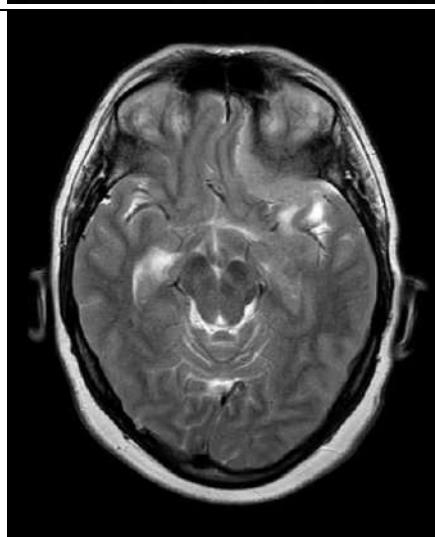
meningioma,  
99.920%



choose a file

Diagnose

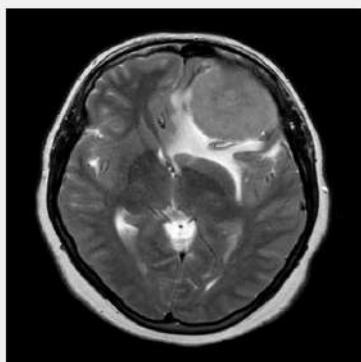
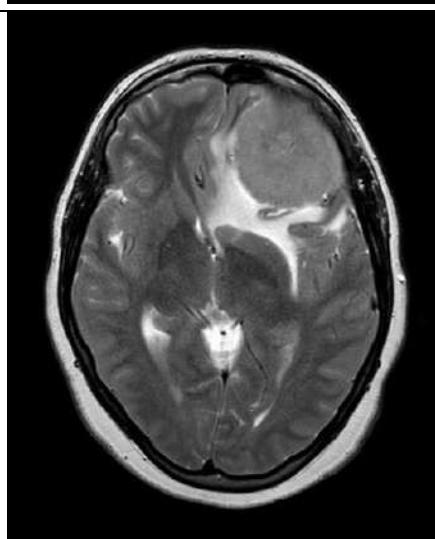
meningioma,  
100.0%



choose a file

Diagnose

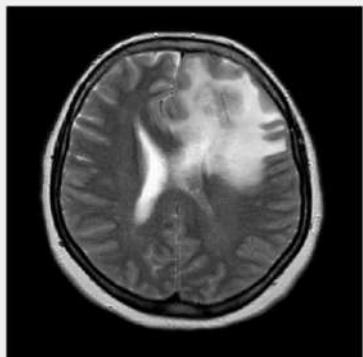
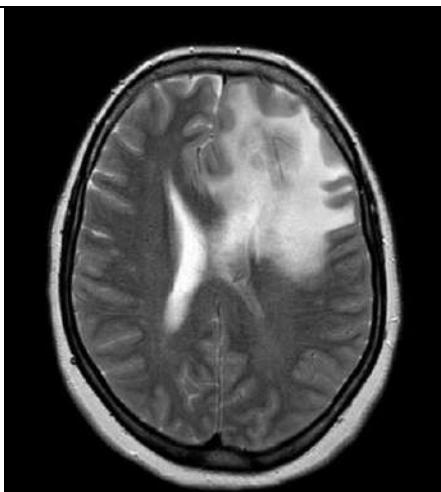
meningioma,  
98.879%



choose a file

Diagnose

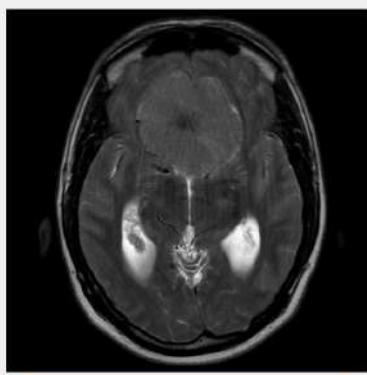
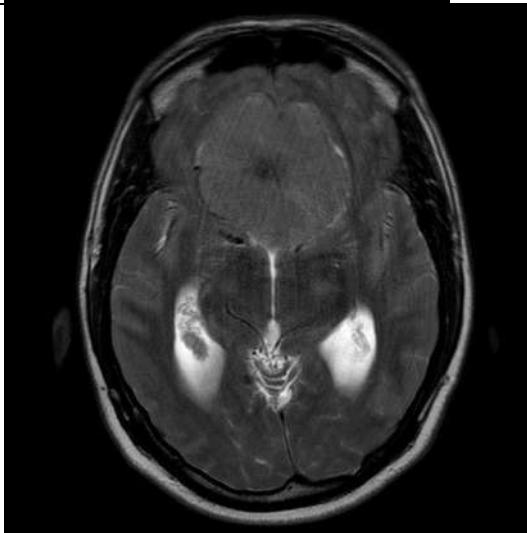
meningioma,  
97.333%



choose a file

Diagnose

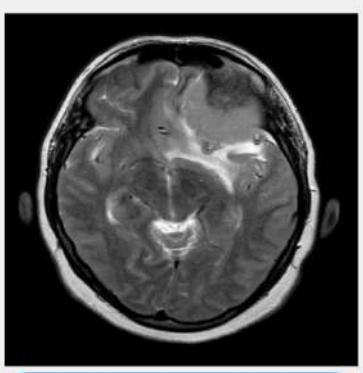
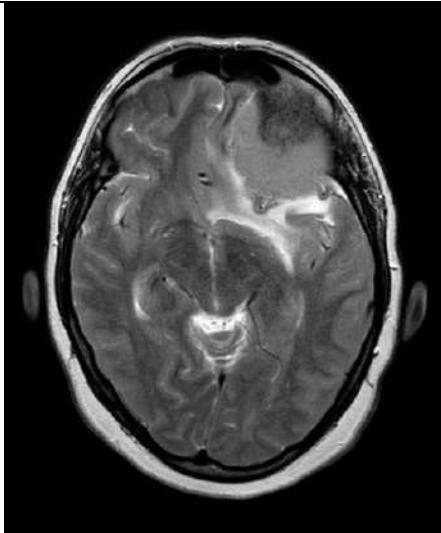
meningioma,  
99.998%



choose a file

Diagnose

meningioma,  
99.696%

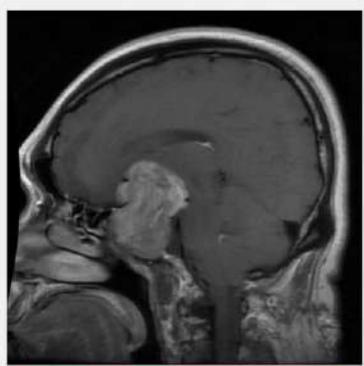
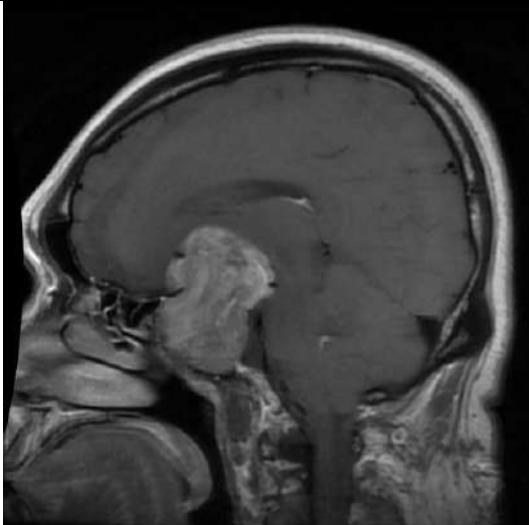


choose a file

Diagnose

meningioma,  
99.987%

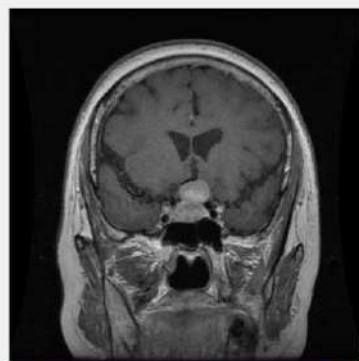
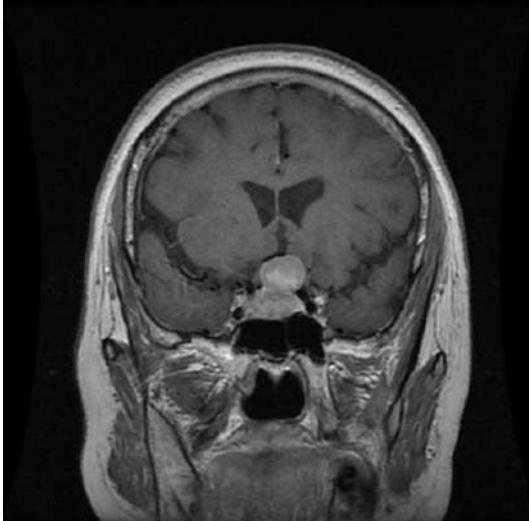
## Pituitary test samples



choose a file

Diagnose

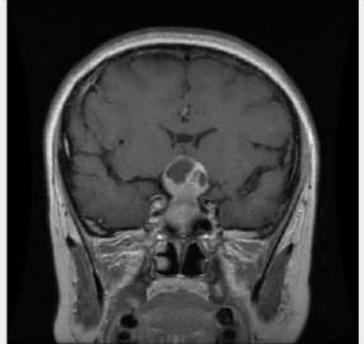
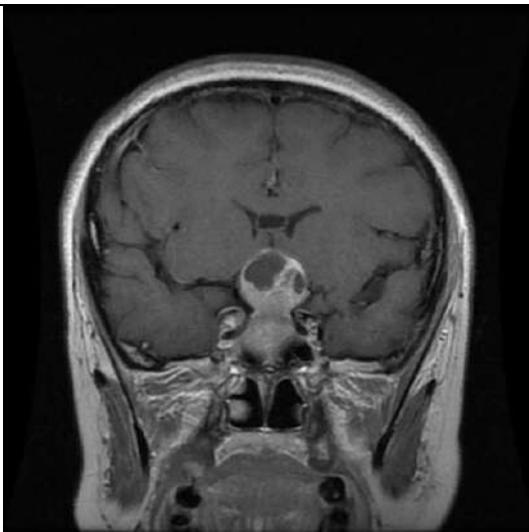
pituitary, 100.0%



choose a file

Diagnose

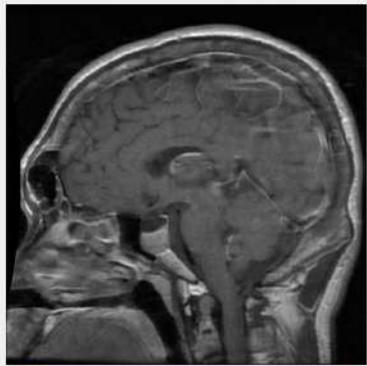
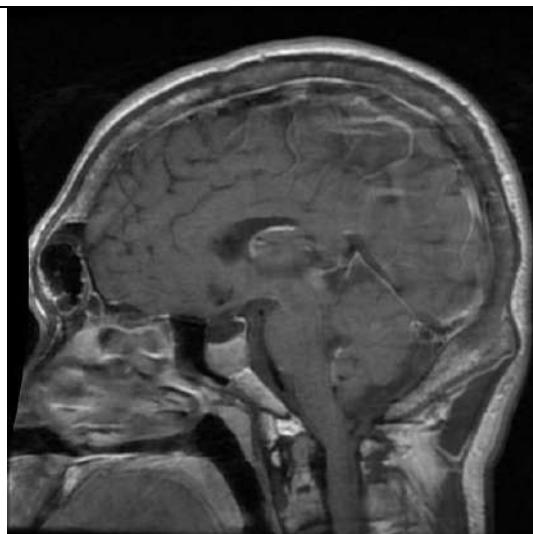
pituitary, 99.999%



choose a file

Diagnose

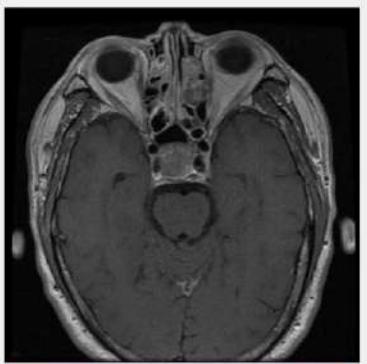
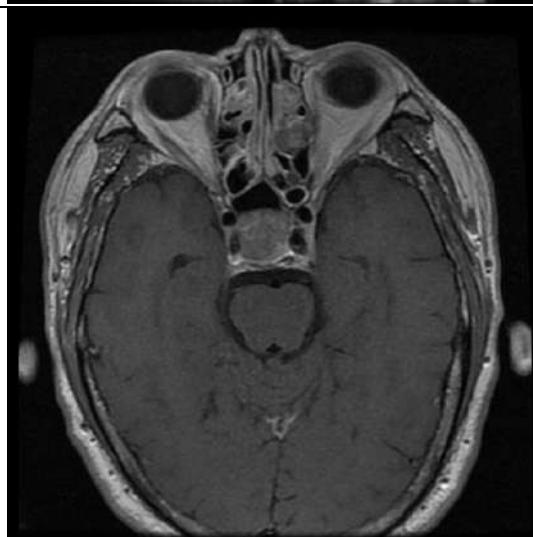
pituitary, 100.0%



choose a file

Diagnose

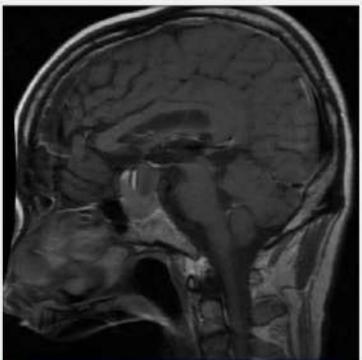
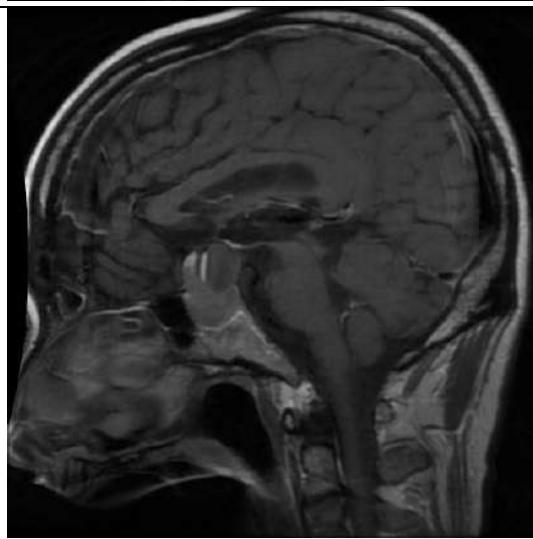
pituitary, 100.0%



choose a file

Diagnose

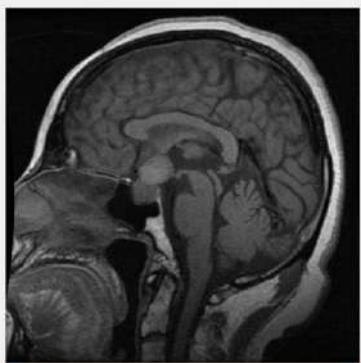
pituitary, 99.999%



choose a file

Diagnose

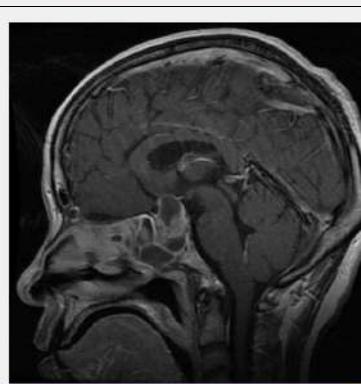
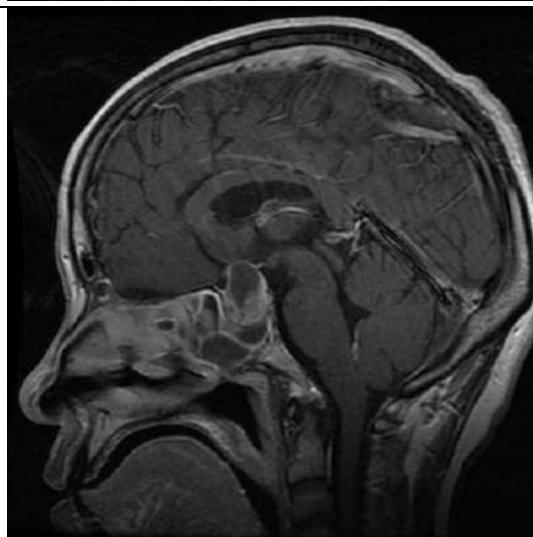
pituitary, 100.0%



choose a file

Diagnose

pituitary, 100.0%



choose a file

Diagnose

pituitary, 100.0%

#### **4.2.4 Brain tumor segmentation from MRI images**

Brain tumor segmentation is an essential task in medical image analysis, and deep learning techniques have shown significant promise in improving the accuracy and efficiency of this process, accurate segmentation is critical for precise diagnosis, treatment planning, and monitoring of patients with brain tumors. Manual segmentation is a time-consuming and labor-intensive process, which is prone to inter-observer variability and requires expertise, on the other hand, deep learning-based methods can automate this process, providing faster and more accurate segmentation results, deep learning-based brain tumor segmentation has the potential to revolutionize clinical decision-making and patient outcomes by providing more reliable and consistent information about the tumor location, and characteristics [73].

##### **4.2.4.1 Motivation**

Brain tumor segmentation using AI offers several motivating factors for its implementation. Firstly, it enables accurate diagnosis and treatment planning by identifying the precise location, size, and boundaries of tumors. This information empowers doctors to make informed decisions regarding treatment options. Secondly, AI algorithms significantly reduce the time required for tumor delineation, alleviating the burden on radiologists and allowing them to dedicate more time to patient care. Moreover, AI provides consistency and standardization in tumor segmentation, ensuring reliable and reproducible results across different medical professionals. Accurate segmentation also aids in surgical planning, enabling surgeons to plan safer procedures by considering tumor location and its proximity to critical structures. Furthermore, AI-based segmentation facilitates the assessment of treatment response over time, allowing doctors to monitor changes in tumor size and shape and make necessary adjustments to treatment plans. These benefits address challenges faced by doctors, including the complexity and variability of brain tumors, time constraints, and interobserver variability. Overall, AI-driven brain tumor segmentation offers valuable support to medical professionals, improving patient outcomes and streamlining healthcare processes.

##### **4.2.4.2 Dataset description**

The LGG-MRI Segmentation dataset is a collection of MRI (Magnetic Resonance Imaging) scans of brain tumors, which is publicly available on Kaggle, the dataset contains a total of 392 patient brain MRI scans with a 3929 total number of images 2556 of them has no tumor and 1373 has tumor all in RGB, a binary mask

in provided for each image with tumor which specify the exact shape and position of the tumor.

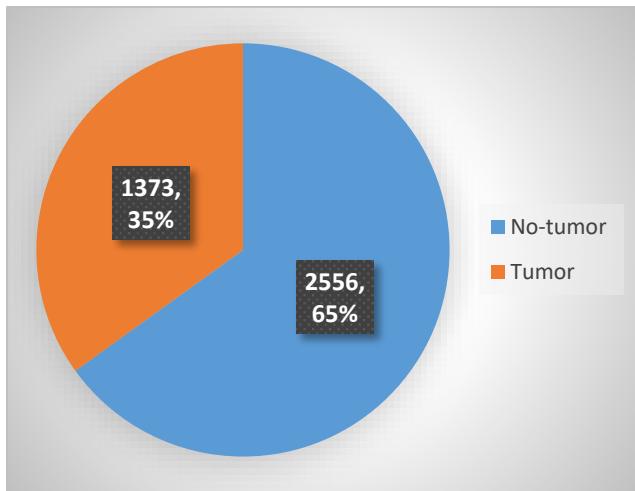


Figure 4-187 class distribution

## Data source

The LGG brain tumor segmentation dataset is a collection of MRI scans for low-grade glioma (LGG) with a confirmed diagnosis, the dataset was collected by several medical organizations, including:

- **The Cancer Imaging Archive (TCIA):** TCIA is a public repository of medical images and associated metadata, the LGG brain tumor segmentation dataset was obtained from TCIA.
- **The German Cancer Research Center (DKFZ):** DKFZ is a research institute that specializes in cancer research, DKFZ was responsible for the manual segmentation of the tumor in the LGG brain tumor segmentation dataset.
- **The University of Pennsylvania:** The University of Pennsylvania is a research university that has a strong track record in medical imaging research. The University of Pennsylvania developed the dataset and made it available to the public.

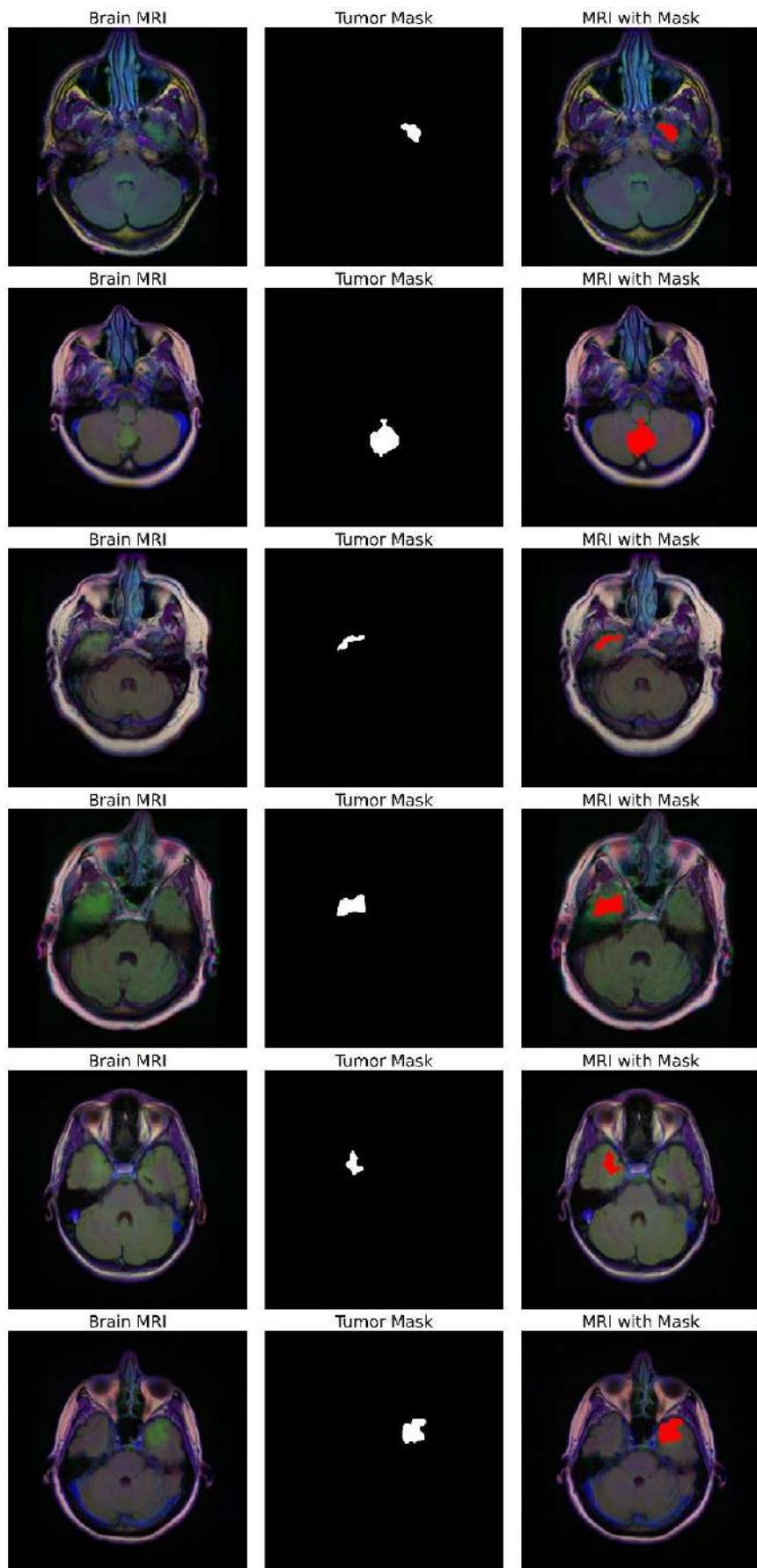


Figure 4-188 samples from the data set

#### 4.2.4.3 Development methodology

The U-Net architecture is the suitable type of CNN architectures for a binary mask segmentation problem, the U-Net model is a type of CNN that is widely used for image segmentation tasks, the concept behind the U-Net model is based on an encoder-decoder architecture with skip connections, where the encoder extracts the features from the input image, and the decoder generates the output segmentation map. The skip connections allow the model to preserve the spatial information during the encoding process, enabling it to localize the objects accurately.

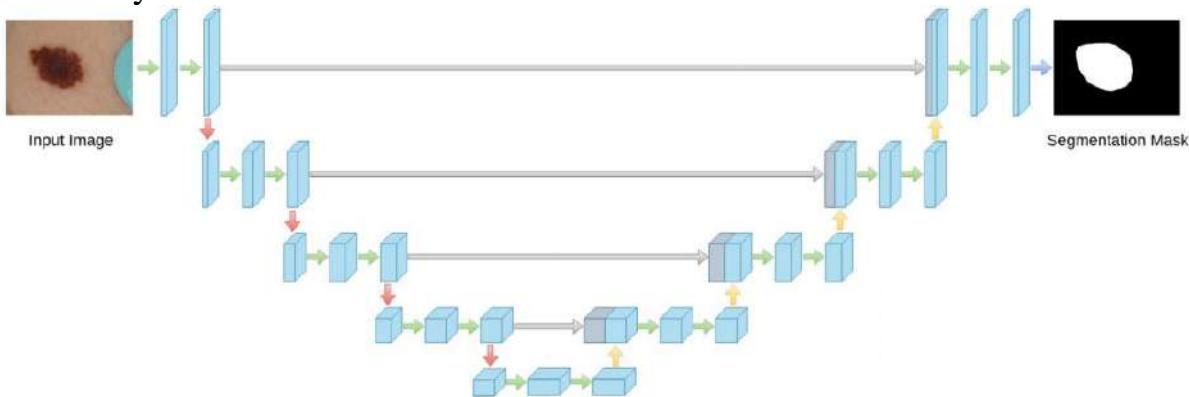


Figure 4-189 U-Net architecture

The U-Net model is particularly suitable for binary mask segmentation problems because of its architecture, which includes a contracting path and an expansive path, the contracting path consists of multiple convolutional and pooling layers that progressively reduce the spatial dimensions of the input, allowing the model to capture high-level features, the expansive path consists of up-sampling and convolutional layers that progressively increase the spatial dimensions of the input, allowing the model to generate the output segmentation mask, the skip connections between the contracting and expansive paths enable the model to combine the low-level and high-level features, which are crucial for accurate segmentation.

The U-Net model is widely used in medical image analysis, including the segmentation of brain tumors, as it can achieve state-of-the-art performance with a relatively small number of training samples, the U-Net model's ability to capture both local and global contextual information, combined with its architecture's skip connections, makes it well-suited for medical image segmentation tasks where precise localization of objects is essential.

There are several performance measures used to evaluate the quality of a binary mask segmentation model

Intersection over Union (IoU): This is a measure of the overlap between the predicted segmentation and the ground truth segmentation. It is calculated as the ratio of the intersection of the two sets to the union of the two sets.

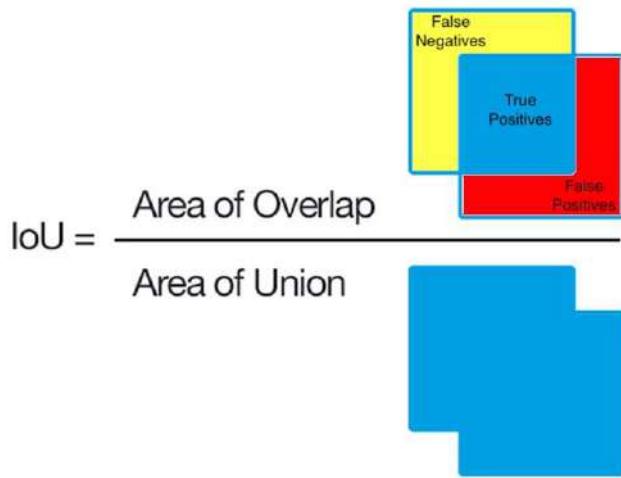


Figure 4-190 Intersection over union

Dice coefficient: This is similar to IoU, but is calculated as twice the intersection divided by the sum of the predicted and ground truth segmentations.

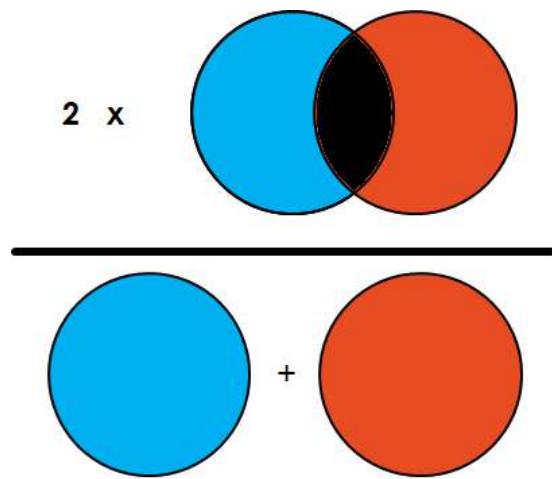


Figure 4-191 Dice coefficient

#### 4.2.4.4 Reference papers for result comparison

In [74] the author proposed an end-to-end trained U-Net and achieved a dice coefficient score of 82%

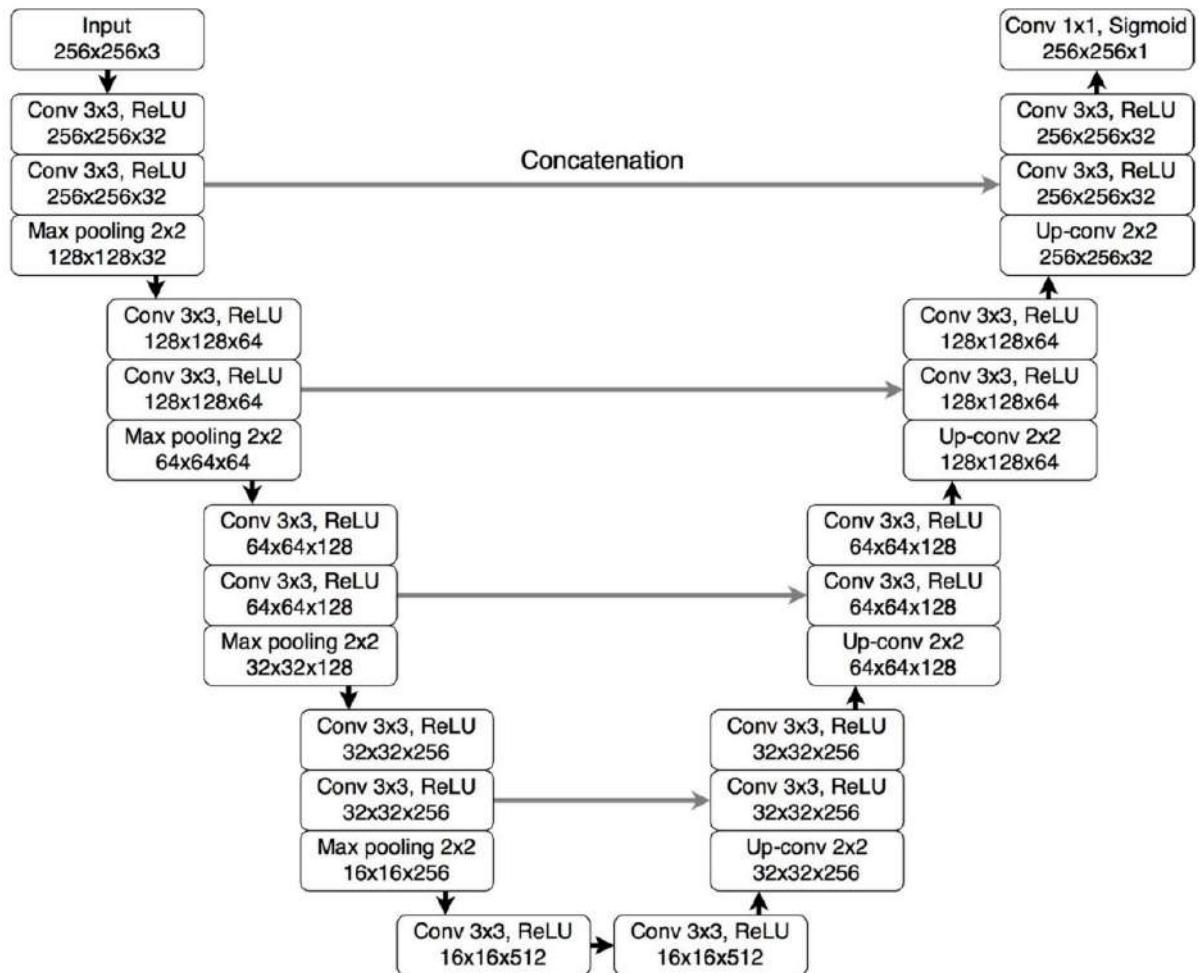


Figure 4-192 [74] proposed network architecture

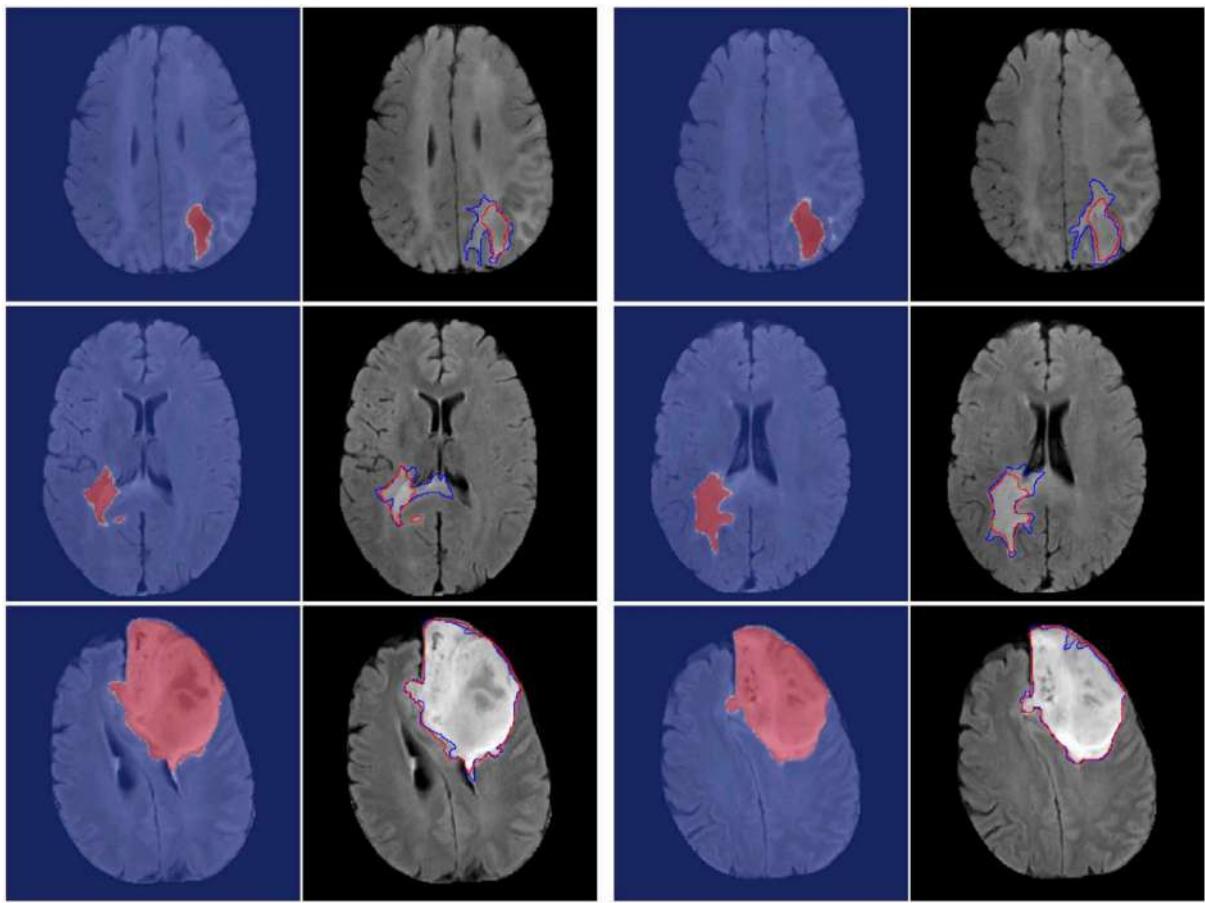


Figure 4-193 [74] segmentation results vs actual masks

Please note that the images in the left side of each pair with red regions are samples from the dataset with the tumor area highlighted in red, next to it is the same image with the actual mask in red and the predicted mask in blue.

In [75] the author proposed integrated inception module and dense block into standard U-Net into a model called SU-Net which achieved AUC of 99.7% and a dice coefficient of 78.5%

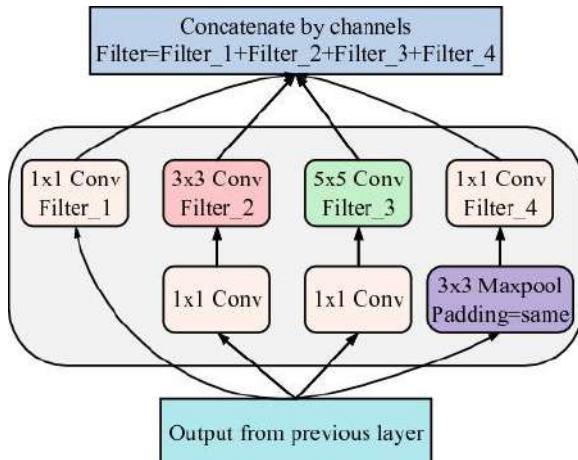


Figure 4-194 inception module proposed in [75]

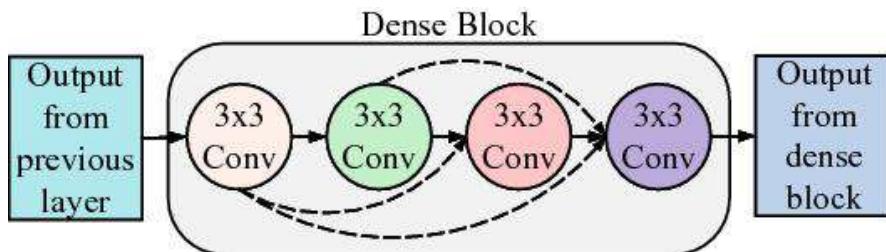


Figure 4-195 dense block introduced in [75]

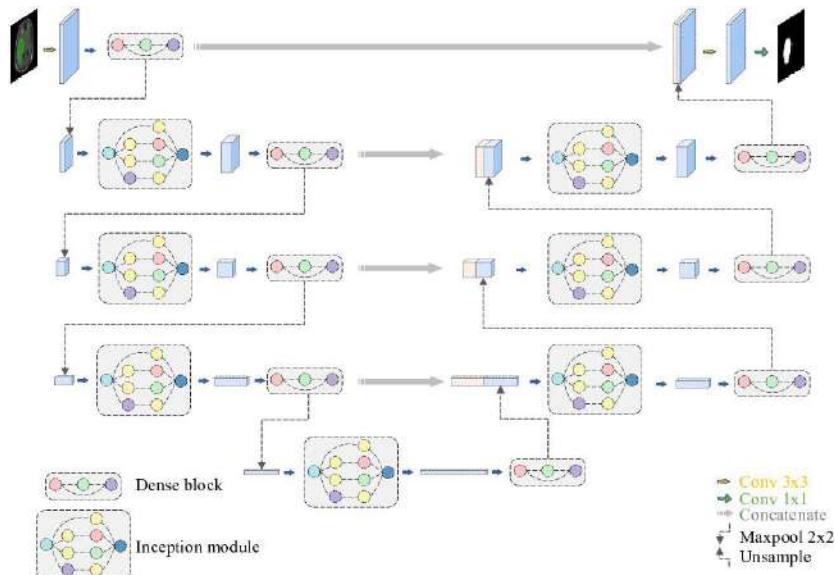


Figure 4-196 network archeticture proposed in [75]



Figure 4-197 segmentation results of [75] input image, actual mask and predicted mask from left to right

#### 4.2.4.5 Experiments with U-Net architecture

##### Data preprocessing

Images with tumor only were used for training the model as we have a model for the classification task and the information of the presence of a tumor will be available before using the segmentation model, the dataset contains a 1372 images with tumor and were divided for training, validation and testing as follows

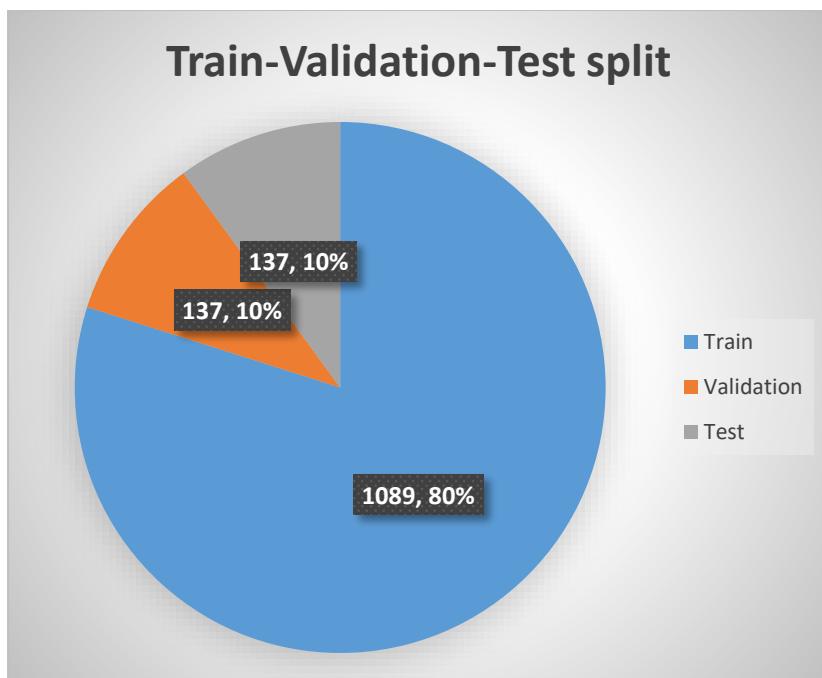


Figure 4-198 train, validation and test split

All images were resized to 256\*256 and normalized

## EXP (1)

An end-to-end trained U-Net model was proposed

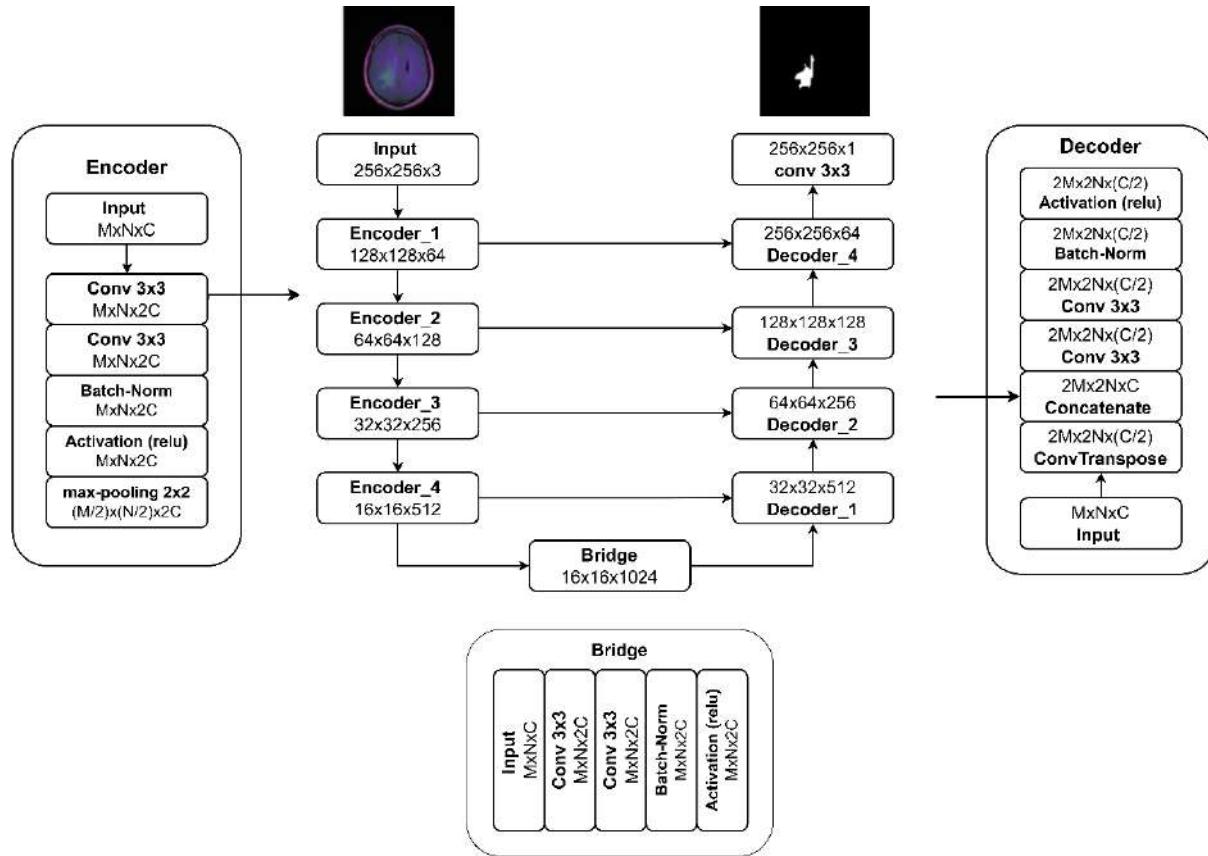


Figure 4-199 EXP (1) proposed network architecture

As shown in the figure the model consists of two branches, an encoder branch which has 4 encoder blocks each containing two convolution layers starting with 64 kernels in the first encoder block and the number of kernels gets doubled with each encoder block then a batch normalization layer followed by a relu activation and finally a max-pooling layer that reduces the input size to the half, a bridge which have the same layer structure as an encoder block except for the max-pooling layer is used to connect the encoder branch with the decoder branch, the decoder branch consist of a convolution transpose layer which reduces the input's number of channels to the half and doubles the input's size then a concatenation layer that concatenate the output of the convolution transpose layer with the output of the first convolution layer of the corresponding encoder block which results in a doubled number of feature maps then 2 convolution layers which reduces the number of feature maps to the half, the convolution layers and the convolution transpose layers of starts with kernel size of 32\*32 which gets doubled with each decoder block and a kernel size of 512 which gets reduced to the half with each decoder bock, then a batch normalization layer followed by a

relu activation layer, all convolution layers in the encoder and decoder branches has kernel size of 3\*3 and the max-pooling layers of the encoder branch have a pool size of 2\*2.

Table 4-162 testing results compared with [74] and [75]

	model	Dice coefficient	Intersection over union (IoU)	AUC	Accuracy
[74]	End-to-end trained U-net	82%			
[75]	SU-Net	78.5%		0.997	
<b>Ours</b>	End-to-end trained U-net	90.48%	82.63%	0.9540	99.40%

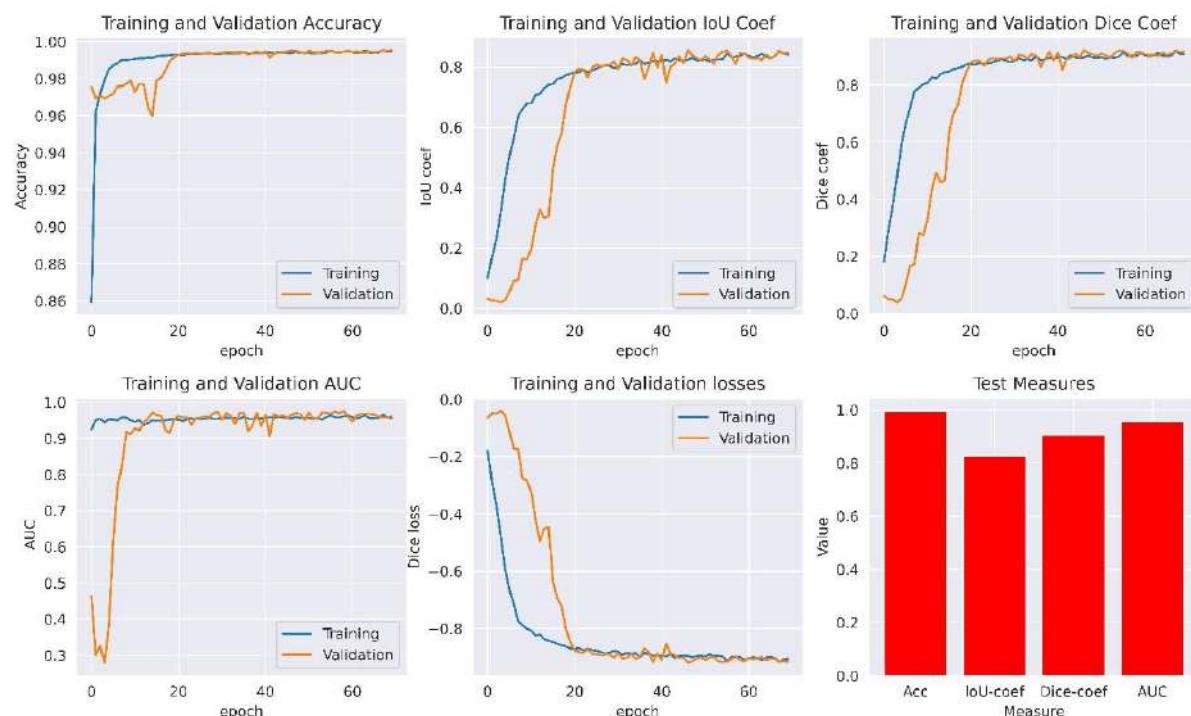
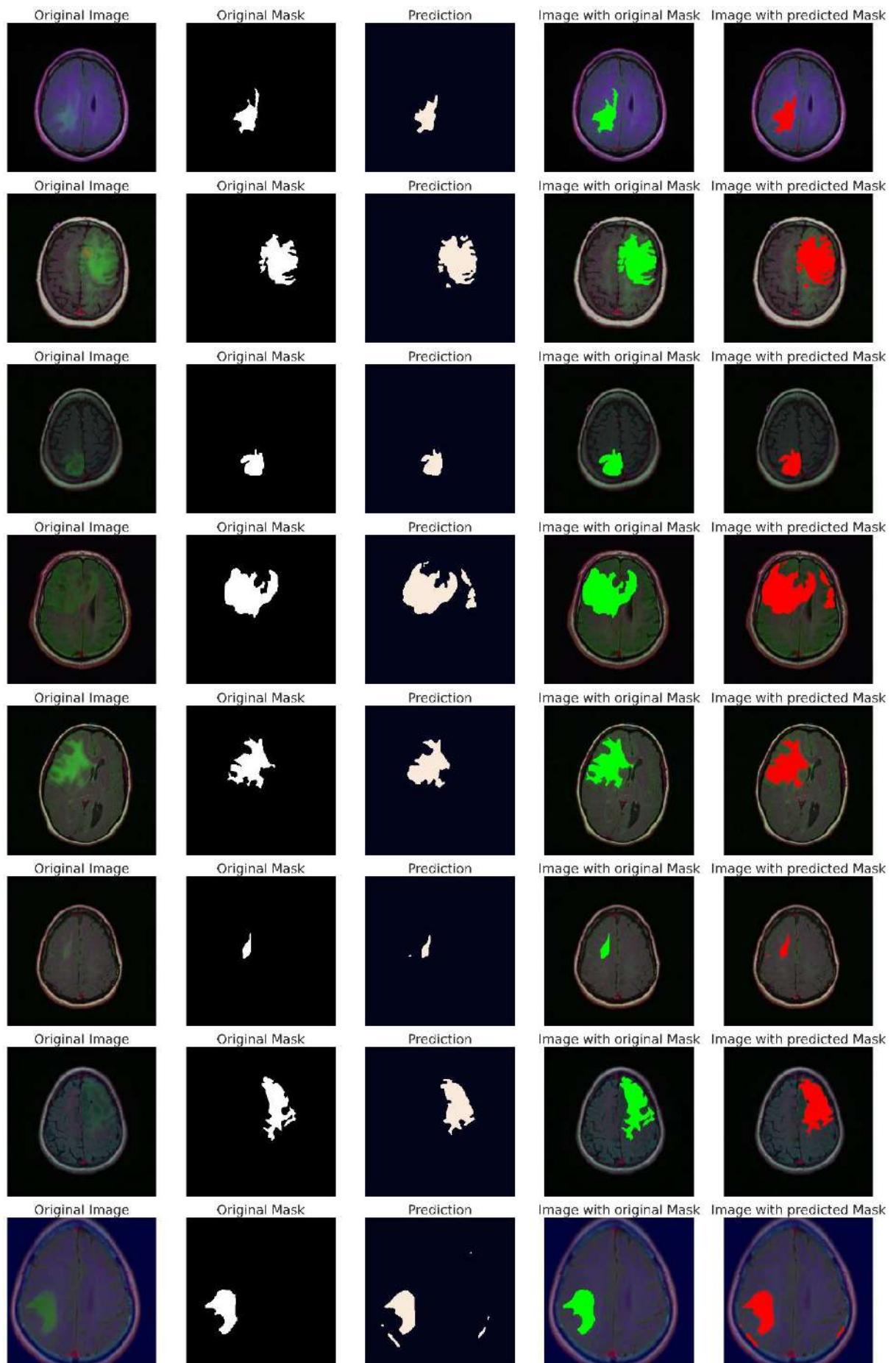


Figure 4-200 training and validation curves



## **4.2.5 Chest x-ray abnormalities localization and categorization**

Localizing and categorizing abnormalities on a chest x-ray is crucial in clinical practice. Chest x-rays are commonly used to diagnose and monitor a variety of respiratory and cardiovascular conditions, including pneumonia, lung cancer, and heart failure, by accurately localizing and categorizing abnormalities on a chest x-ray, healthcare professionals can determine the severity of the condition and make informed decisions regarding treatment options, furthermore, the ability to track changes in the location and type of abnormalities over time can aid in assessing the effectiveness of treatment.

### **4.2.5.1 Motivation**

Localizing and categorizing different chest abnormalities from a chest x-ray image can be challenging for a doctor, especially without additional information about the patient or medical history, there are many different types of abnormalities that can appear on a chest x-ray, identifying these abnormalities requires specialized training and expertise in radiology, and even then, the accuracy of diagnosis can be limited by factors such as the quality of the x-ray image or the nature of some abnormalities.

### **4.2.5.2 Dataset description**

The VinBigData Chest X-ray Abnormalities Detection dataset [79] is a large, publicly available dataset of chest X-ray images with corresponding annotations of 14 common chest abnormalities. The dataset consists of 15,000 chest X-ray images from various sources, including hospitals and clinics across multiple countries, the images are labeled with bounding boxes around regions of interest, indicating the presence and location of specific abnormalities. The dataset was labeled through a process of manual annotation. Experienced radiologists and trained annotators reviewed the X-ray images and identified the presence and location of abnormalities by drawing bounding boxes around them. These bounding boxes were then labeled with the corresponding abnormality type (class label) based on the radiologists' diagnosis of the image, this process was repeated for each abnormality present in the image, finally, the images and their corresponding bounding box annotations and class labels were compiled into the dataset.

Table 4-163 class frequency

Class name	Class ID	Class frequency
Aortic enlargement	0	7162
Atelectasis	1	279
Calcification	2	960
Cardiomegaly	3	5427
Consolidation	4	556
Interstitial lung disease (ILD)	5	1000
Infiltration	6	1247
Lung Opacity	7	2483
Nodule/Mass	8	2580
Other lesion	9	2203
Pleural effusion	10	2476
Pleural thickening	11	4842
Pneumothorax	12	226
Pulmonary fibrosis	13	4655
No finding	14	10k

Please note that the dataset has 10k image for normal cases with no bounding boxes and the total number of bounding boxes is 24,363 because each image in the dataset contains more than one label.

## Data source

This dataset was collected by **VinBigData**, a Vietnamese artificial intelligence company from two hospitals in Vietnam, the first one is **the Hospital 108** which is a large military hospital in Hanoi, Vietnam, it is one of the largest and most modern hospitals in Vietnam, the second one is **the Hanoi Medical University Hospital** which is a large public hospital in Hanoi, Vietnam. It is one of the largest and most prestigious hospitals in Vietnam, the VinBigData Company published a competition on this dataset with a \$50,000 prize.

### 4.2.5.3 Development methodology

There are several types of object detection models, including such as Region-based Convolutional Neural Networks (RCNNs), Detectron-2 and you only look once (YOLO), YOLO is a good choice for object detection because it can detect multiple objects in a single pass, making it faster and more efficient than other models. YOLO is also known for its high performance, particularly in detecting small objects.

A true positive occurs when the model correctly detects and localizes an object in an image, in other words, the model accurately identifies the presence and location of an object that is actually present in the image.

A true negative occurs when the model correctly identifies that no object of a specific category is present in the image, in other words, the model correctly identifies that there is no object of interest in the image.

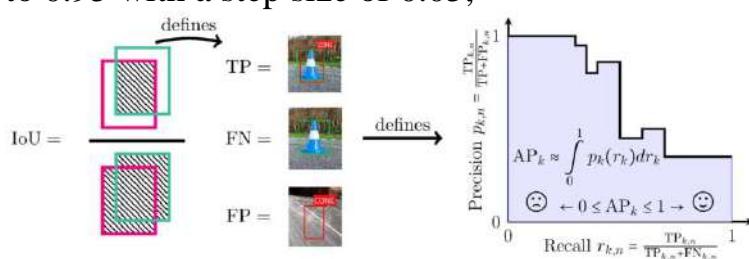
A false positive occurs when the model incorrectly identifies the presence of an object of a specific category in the image, in other words, the model predicts that an object is present when in reality there is no object of that category in the image.

A false negative occurs when the model fails to detect or localize an object of a specific category that is actually present in the image, in other words, the model fails to identify the presence and location of an object that is actually present in the image.

mean average precision (mAP) is a widely used performance measure in object detection tasks that evaluates the overall quality of a model across all object categories in a dataset, the mAP score is calculated by taking the average of the Average Precision (AP) values for each object category, AP is calculated by computing the precision and recall values at different thresholds for a given object category, the precision-recall curve is then created by plotting the precision values against the corresponding recall values. AP is calculated as the area under this curve, AP is a measure of how well a model is able to correctly identify objects of a specific category and is scaled from 0 to 1, with a score of 1 indicating perfect performance.

mAP0.5 calculates the mAP using a single fixed IoU (Intersection over Union) threshold of 0.5, which means that a detection is considered a true positive if the IoU between the detection and the ground truth bounding box is greater than or equal to 0.5.

mAP0.5:0.95 calculates the mAP using multiple IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05,



#### 4.2.5.4 Reference papers for result comparison

In [80] the utilized different object detection models and the best results was achieved by Detectron-2 model with a mAP0.5 score of 0.235, while in [81] the author also did some experiments with different models and best results was achieved by ResNet50-FPN with a mAP0.5 score of 0.246.

#### 4.2.5.5 Experiments with yolo object detection model

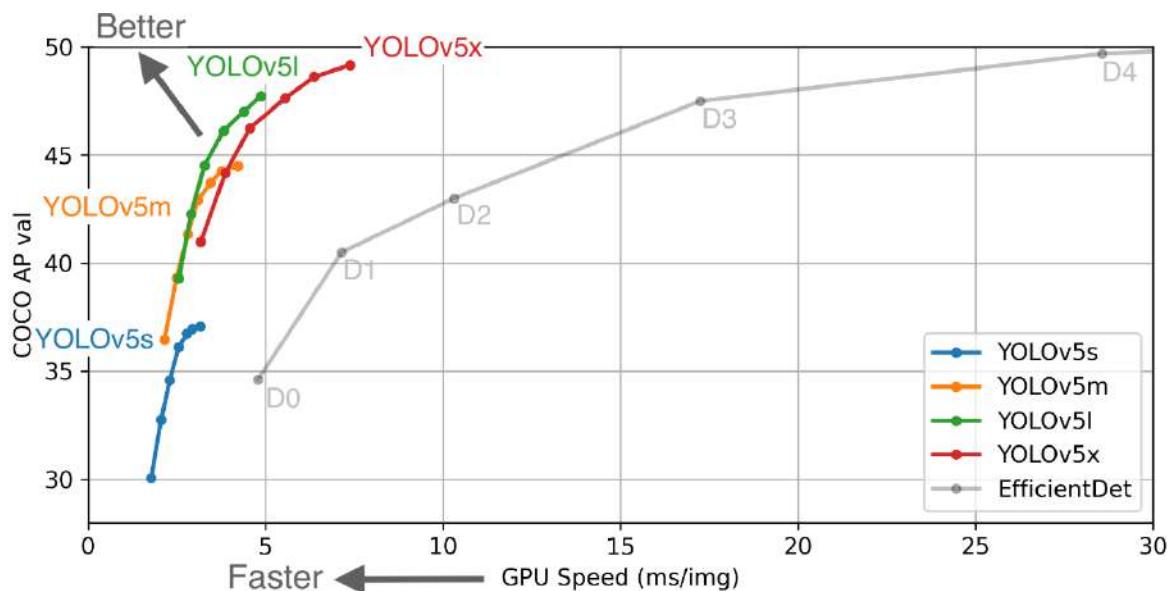


Figure 4-201 AP of different yolo v5 versions on the coco dataset

As seen in the previous figure, the yolov5x is the best model in terms of AP score but it takes more time to operate which is not a big problem as the model won't be deployed it a real-time implementation.

## EXP (1)

Yolo v5x was utilized.

Table 4-164 mAP0.5 of yolo v5x VS [80] and [81]

	model	Mean average precision (mAP0.5)
[80]	Detectron 2	0.235
[81]	ResNet50 - FPN	0.246
<b>Ours</b>	<b>Yolo v5x</b>	<b>0.312</b>

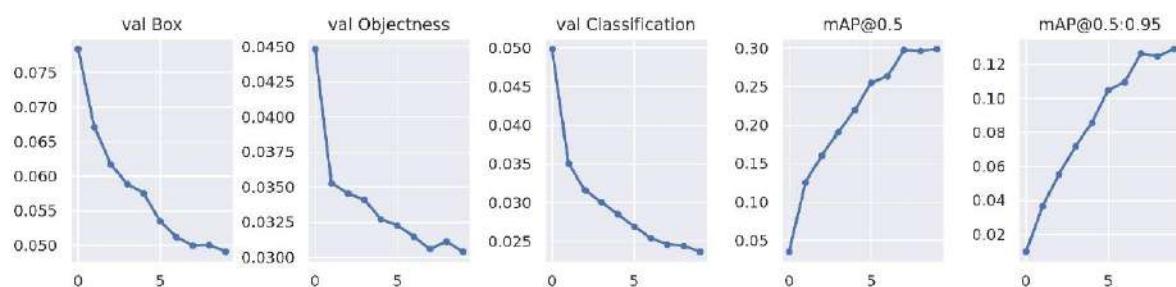


Figure 4-202 testing curves

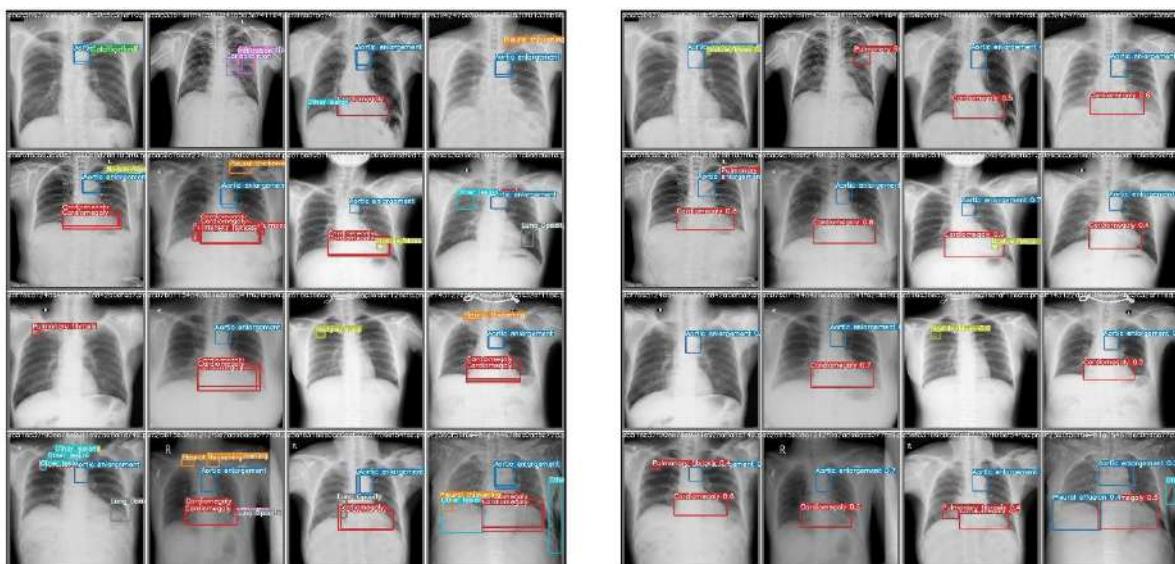
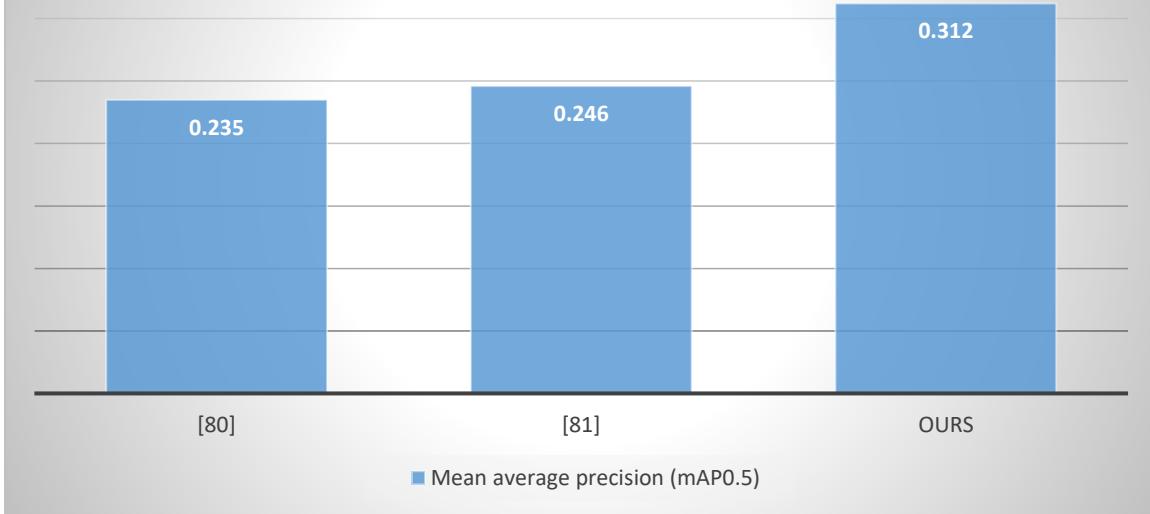


Figure 4-203 actual bounding boxes on the left VS predictions on the right

## Result comparison



#### 4.2.5.6 Testing with real world cases (online testing)

“CheXDet Image and Annotations” is a public dataset available on kaggle for chest x-ray abnormalities detection and will be used to test the developed model, link available in the appendix, it is noted that this dataset has 13 classed and not all of them match the dataset which the model was trained on, so this testing process will be only performed on matching classes in the two datasets.

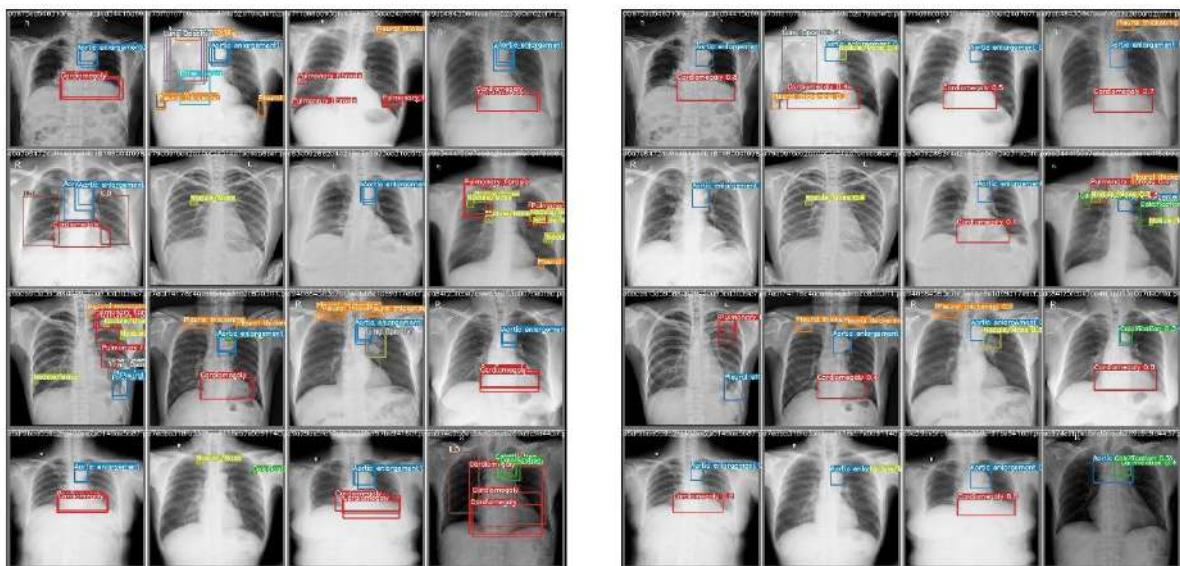


Figure 4-204 actual bounding boxes on the left VS prediction of the right

#### **4.2.6 Skin cancer detection from images**

For the patient's therapy to be effective, it is essential to determine if a skin cancer is malignant or not. The ability to properly distinguish between benign and malignant skin lesions from photographs has been demonstrated using deep learning algorithms. These algorithms examine the lesion's characteristics, including its size, shape, and texture, and look for patterns that could point to the existence of malignancy, malignant skin malignancies should be detected as early as possible since doing so can greatly enhance patient outcomes and survival rates, deep learning algorithms might potentially increase the precision and speed of diagnosis in this situation, resulting in early treatment and better patient outcomes. In the end, the ability to distinguish between benign and malignant skin tumors with accuracy from photographs can aid in directing treatment choices and enhancing patient care.

##### **4.2.6.1 Motivation**

It is generally not possible for a doctor to tell if a skin cancer is malignant or benign only from looking at the skin without any additional tests or medical information about the patient, a definitive diagnosis of skin cancer requires a biopsy, which involves removing a small piece of the skin and examining it under a microscope to determine if cancer cells are present.

AI models may also be able to assist dermatologists in identifying and classifying skin lesions as benign or malignant based on features extracted from images or other clinical data, although a biopsy is still typically required for a definitive diagnosis.

##### **4.2.6.2 Dataset description**

The "Melanoma Skin Cancer Dataset of 10,000 Images" is a collection of dermatoscopic images of skin lesions, including both malignant and benign cancers and is publicly available on kaggle, link in the appendix, it contains RGB images of two classes, malignant and benign skin cancers.

*Table 4-165 class distribution of training and testing sets*

	Malignant	Benign	Total	
Train	4605	5000	9605	10605
Test	500	500	1000	



Figure 4-205 class distribution and train test split

## Data source

This dataset is a combination of three dermatoscopic database of three medical organizations, the first organization is the **Austrian Society for Dermatology and Venereology (ÖGDV)** which is a professional organization for dermatologists and venereologists in Austria, the ÖGDV's goals are to promote the advancement of dermatology and venereology in Austria, the ÖGDV is a member of the European Academy of Dermatology and Venereology (EADV) and the International League of Dermatological Societies (ILDS). The ÖGDV is also a partner of the World Health Organization (WHO), the second organization is the **German Society for Dermatology and Venereology (DDG)** which is a professional medical society in Germany that focuses on dermatology and venereology, it is one of the oldest and largest dermatological societies in the world, its mission is to promote research, education, and patient care in the field of dermatology and venereology, the DDG is a member of the European Academy of Dermatology and Venereology (EADV), which is a leading organization representing dermatologists and venereologists in Europe, the DDG also collaborates with Arbeitsgemeinschaft Dermatologische Forschung (ADF), a German association dedicated to dermatological research, the third organization is the **European Academy of Dermatology and Venereology (EADV)** which is an international organization that represents dermatologists and venereologists in Europe, it is a professional society dedicated to advancing the field of dermatology and venereology through education, research, and collaboration he EADV is a member organization of the ILDS as well as European Dermatology Forum (EDF) and European Reference Network on Rare Skin Diseases (ERN-Skin).

The presence of high-profile organizations in the collection and compilation of the HAM10000 dataset indicates the importance of AI in this field

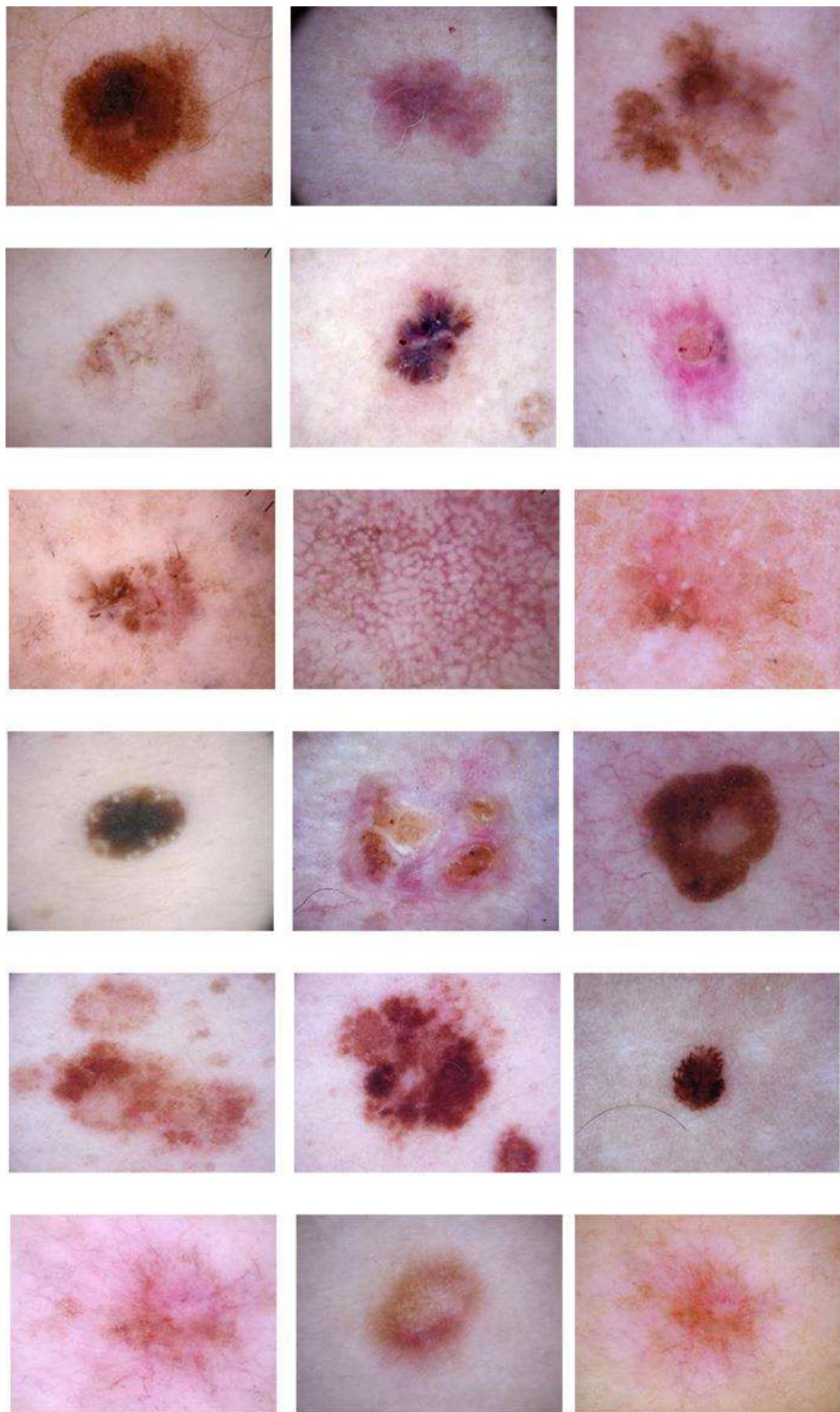


Figure 4-206 samples from the dataset

#### 4.2.6.3 Development methodology

The suitable type of neural networks for this dataset is 2-D convolution neural networks as it is more useful with images due to the feature extraction process of the convolution layers, different pre-trained models could be experimented as samples of this dataset are in RGB color mode.

#### 4.2.6.4 Reference papers for result comparison

In [82] the author utilized the AlexNet pre-trained model and achieved 84% accuracy, 81% recall and 0.91 AUC, in [83] the author experimented with multiple pre-trained models and the best results was achieved with the DenseNet201 model with 93% accuracy and 92% recall.

#### 4.2.6.5 Experiments

##### Preprocessing

All images were resized to 224\*224 and normalized.

##### EXP (1)

VGG-19 pre-trained model was used with the Adaptive Feature Recalibration layer (VGG19+AFR) with reduction ratio = 8

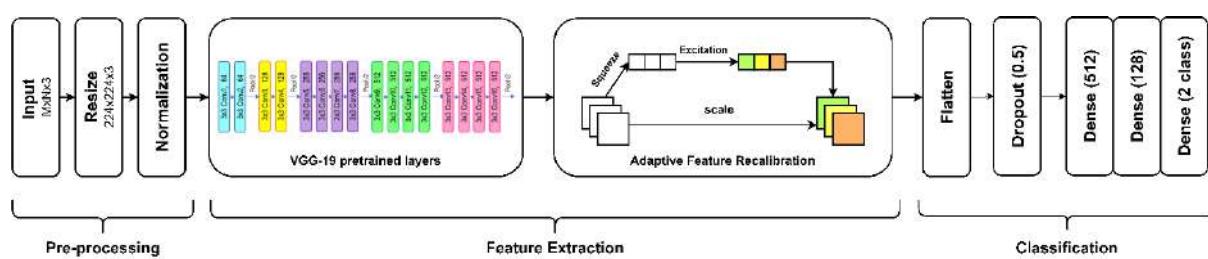


Figure 4-207 EXP (1) proposed network architecture

Table 4-166 testing results of EXP (1)

	Accuracy	precision	Recall	F1	AUC	G-mean	kappa
Proposed	92.10%	93.58%	90.40%	92.10	0.9667	92.08%	84.20%
[82]	84%		81%		0.91		
[83]	93%	93%	92%	93%	0.932		

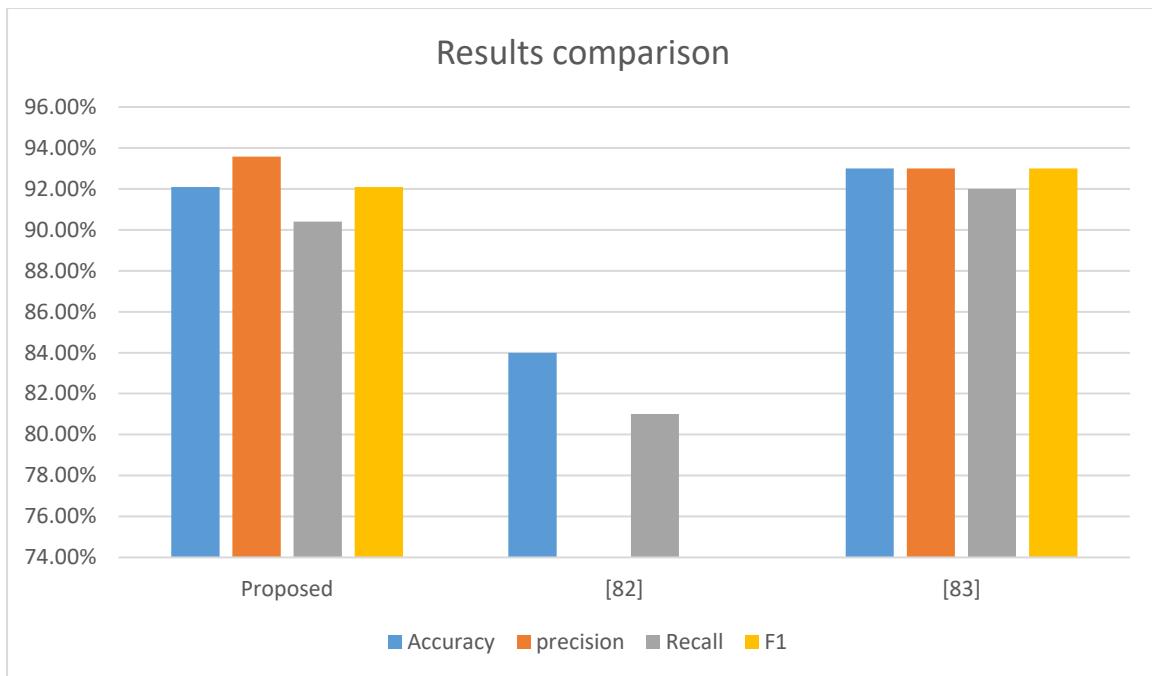


Table 4-167 EXP (1) classification report

index	Class Name	Precision	Recall	f1-score	Support
0	benign	0.9072	0.938	0.9223	500
1	malignant	0.9358	0.904	0.9196	500
accuracy				0.921	1000
macro avg		0.9215	0.921	0.921	1000
weighted avg		0.9215	0.921	0.921	1000

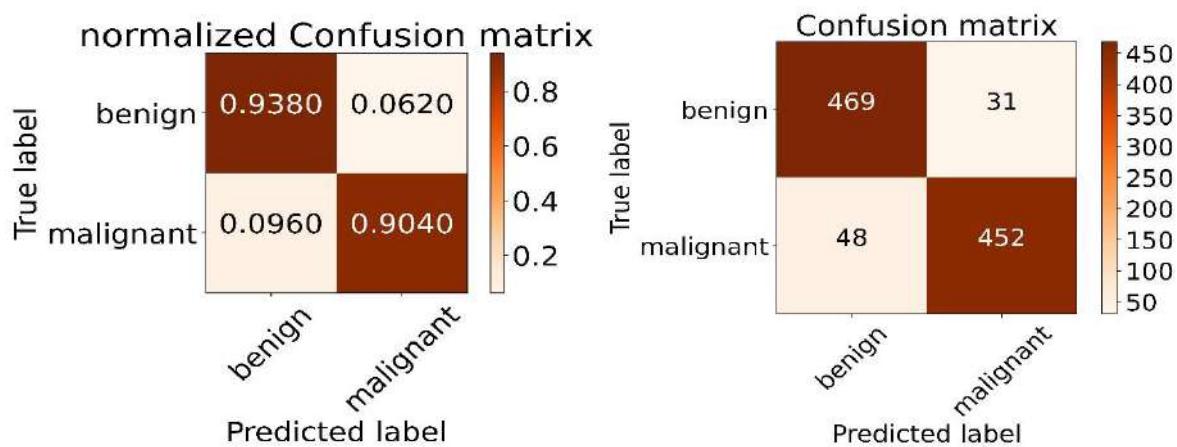


Figure 4-208 EXP (1) confusion matrices

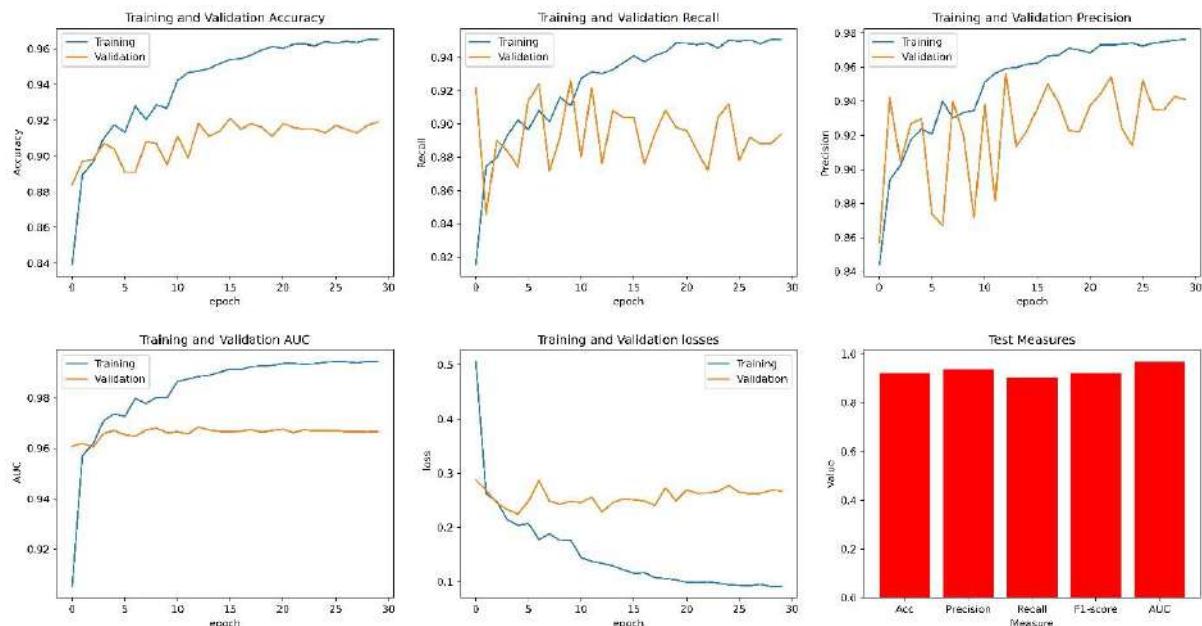


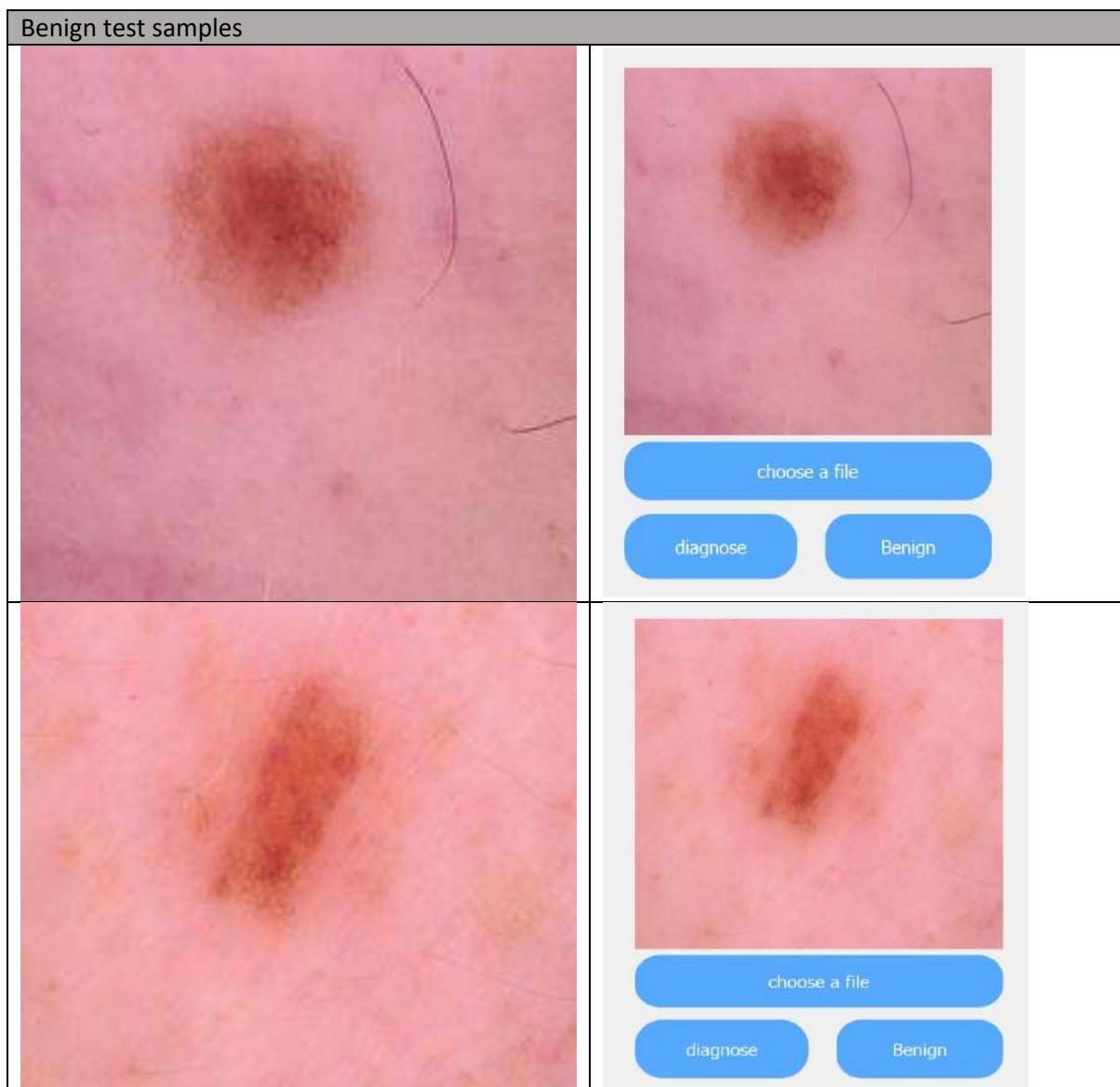
Figure 4-209 EXP (1) training and validation curves

#### 4.2.6.6 Testing with real world cases (online testing)

After the training process is done and the final model was chosen to be the combination of VGG-19 and the adaptive feature recalibration (VGG-19+AFR) which achieved 92.10 % accuracy, it is time to test it on some samples completely out side of the dataset used in the training and testing, the model was tested with 8 sample for each class from another datasets with different source for the one used for training and only misdiagnosed 1 benign sample to be malignant

Test samples used in this test were obtained from a public dataset on kaggle named “**Skin Cancer: Malignant vs. Benign**”, link available in the appendix

The following table shows each sample on the left size and the model’s diagnosis in the right side along with the confidence for the diagnosis.





choose a file

diagnose

Benign



choose a file

diagnose

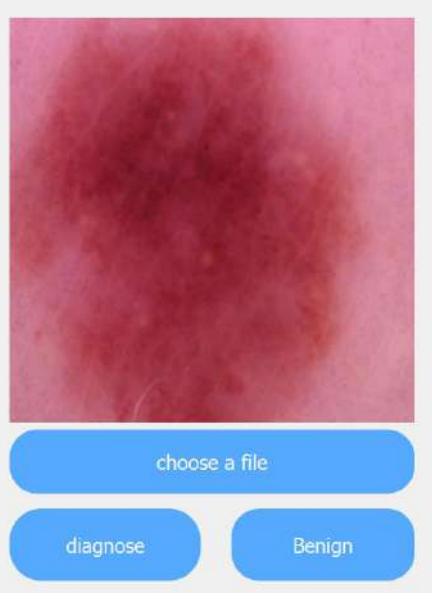
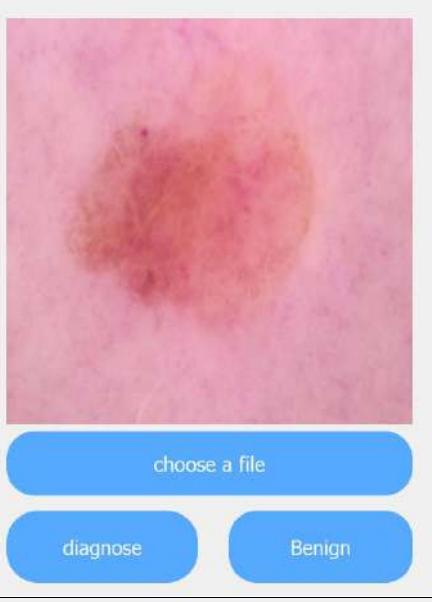
Benign



choose a file

diagnose

Malignant



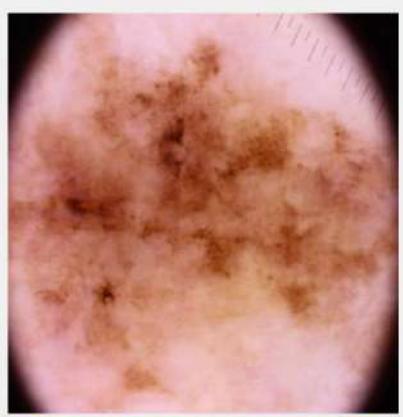
Malignant test samples



choose a file

diagnose

Malignant



choose a file

diagnose

Malignant



choose a file

diagnose

Malignant



choose a file

diagnose

Malignant



choose a file

diagnose

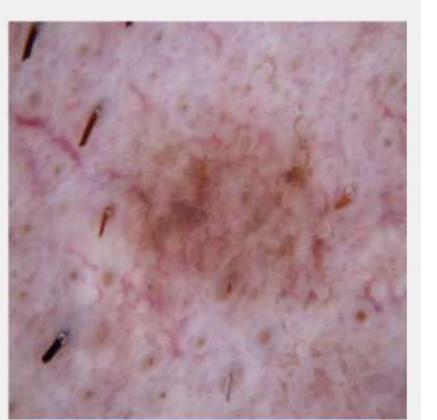
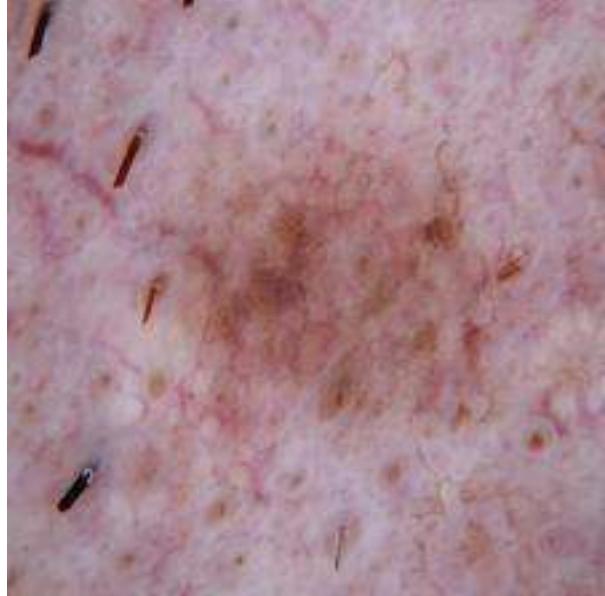
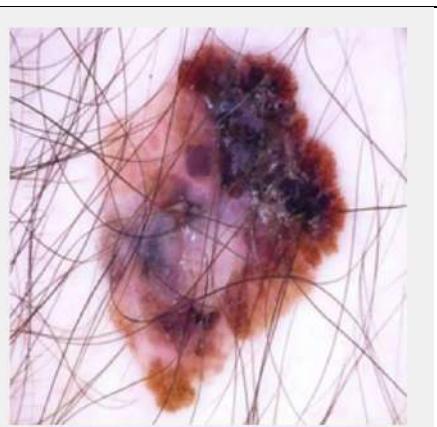
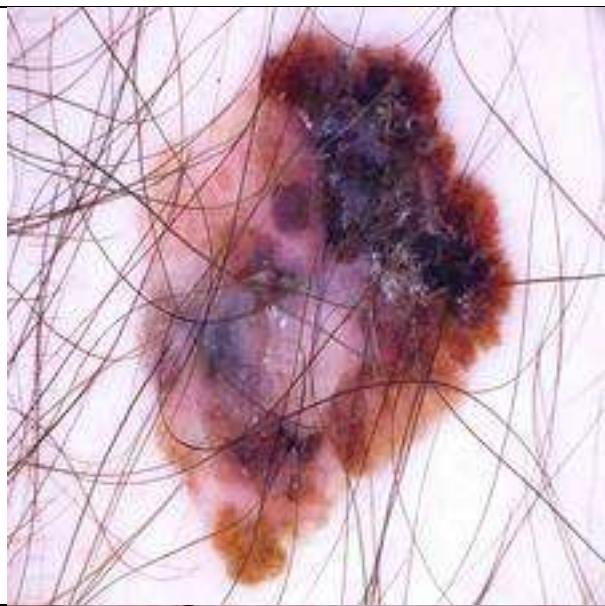
Malignant



choose a file

diagnose

Malignant



#### 4.2.7 AI models results summary and conclusion

Heartbeat abnormalities detection from ECG							
MIT-BIH dataset (main dataset)							
	N	S	F	Q	V	total	
Train	72471	2223	641	6431	5788	87554	109446
Test	18118	556	162	1608	1448	21892	
	Architecture	Data imbalance	Accuracy	Macro F1	Recall	Precision	
Test data resampled to 800							
Ours	CNN+AFR+ Bi-LSTM+RLR	Weighted loss	95.10%		95.00%	95.10%	
[15]	Residual CNN	Over sampling	93.4%				
Test data as it is							
Ours	CNN+AFR+ Bi-LSTM+RLR	Weighted loss	97.11%	85.48%	97.09%	97.17%	
[71]	CNN	Over sampling	93.47%				
[72]	SVM			82%			
PTB Diagnostic ECG Database (method validation)							
	N	M					
	10.5k	4045					
	Model	Accuracy	precision	Recall			
Ours	CNN+AFR+ Bi-LSTM+RLR	99.79%	99.82%	99.66%			
[15]	Transfer learning Residual CNN	95.9%	95.1%	95.2%			
Covid-19 detection from chest x-ray images							
COVID19-Pneumonia-Normal-chest-xray-pa-dataset (main dataset)							
	Normal	Pneumonia	Covid-19	Total			
	2313	2313	2313	6939			
	Model	Accuracy					
Ours	VGG-19+AFR	96.10%					
[44]	Xception+ MobileNetV2+ NasNetMobile	95.56%					
[76]	EfficientNet-b0	94.70%					
COVID-19 Radiography Database							
	Covid-19	Normal	Pneumonia				
	3616	10.2 k	1354				
	Architecture	Data imbalance	Train : Test	Accuracy	Recall	Precision	F1 score
Ours	VGG-19+AFR	Weighted loss	70% : 30%	98.22%	97.37%	98.08%	97.72%

[77]	CovidDetNet	Augmentation		98.40%	96.66%	97.00%	96.82%
Ours	VGG-19+AFR	Weighted loss	90% : 10%	98.02%	97.54%	98.08%	97.81%
[78]	AlexNet	Augmentation		97.59%	95.45%	98.55%	96.9%

### Brain tumor detection from MRI images

Brain Tumor MRI Dataset (main dataset)

	<b>glioma</b>	<b>meningioma</b>	<b>no tumor</b>	<b>pituitary</b>	total		
train	1312	1339	1595	1457	5712	7023	
test	300	306	405	300	1311		
	Model	Accuracy	Precision	Recall	F1		
Ours	CNN	99.08%	99.08%	99.01%	99.04%		

Brain Tumor (method validation)

	No_tumor	Tumor	Total				
	2079	1683	3783				
	Model	Train : Test	Accuracy	Precision	Recall	F1	
Ours	CNN	80% : 20%	98.14%	98.22%	98%	98.10%	
[84]	MobileNetV2		89%				

### Brain tumor segmentation from MRI images

LGG Segmentation Dataset

	Tumor	No-tumor	Total				
	2556	1373	3929				
	Model	Dice coef	IoU	AUC			
Ours	End-to-end	90.48%	82.63%	0.9540			
[74]	trained U-Net	82%					
	SU-Net	78.5%		0.997			

### Chest x-ray abnormalities localization and categorization

VinBigData Chest X-ray Abnormalities Detection

	15 class	15k image					
	Model	mAP0.5					
[80]	Detectron 2	0.235					
[81]	ResNet50 - FPN	0.246					
Ours	Yolo v5x	0.312					

### Skin cancer detection

Melanoma Skin Cancer Dataset of 10,000 Images

	Malignant	Benign	Total				
Train	4605	5000	9605	10605			
Test	500	500	1000				
	Accuracy	precision	Recall	F1	AUC		
Ours	92.10%	93.58%	90.40%	92.10%	0.9667		
[82]	84%		81%		0.91		
[83]	93%	93%	92%	93%	0.932		

## 4.3 GUI and deployment

As previously discussed, the developed AI models will be deployed in a web application and a desk top application, this section will present the designs of the web application and the desk top application as well as the deployment methodology of the AI models.

### 4.3.1 Home page



Figure 4-210 desktop app. Home page

As shown in the previous figure, the home page of the desktop app consists of a vertical frame on the left side containing 5 buttons one for each proposed service and a rectangle under them which is a guide for the user telling him more information about the page he is in, the kind of input he is expected to upload to the app and the kind of output the app will give him.

• SERVICES •

	<b>Chest x-ray abnormalities detection and localization</b> Localize and detect chest abnormalities from chest x-ray images  <a href="#">service</a> →		<b>Brain tumor detection and segmentation from MRI images</b> Differentiate between Meningioma, Glioma, pituitary and the normal case from MRI images  <a href="#">service</a> →		<b>Covid-19 detection from chest x-ray images</b> Differentiate between Covid-19, Pneumonia and the normal case from chest x-ray images  <a href="#">service</a> →
	<b>Heartbeat abnormalities categorization from ECG</b> Differentiate between Supraventricular premature beat, Premature ventricular contraction, Fusion of ventricular and Normal Beat from ECG signal  <a href="#">service</a> →		<b>Skin cancer detection from skin images</b> Differentiate between Benign and Malignant skin cancer from skin images  <a href="#">service</a> →		

Figure 4-211 web app home page

As shown in the previous figure, the home page of the web app consists of 5 frames, one for each service, when the user hovers over one of these frames it pops up as in the brain tumor frame in the above figure, each frame also acts as a button that redirect the user to the corresponding service page.

#### 4.3.2 Brain tumor service page and model deployment

The brain tumor service consists of two tasks, firstly classifying the image to be Glioma, Meningioma, Pituitary or normal then if the image wasn't normal it then would be passed to a segmentation model to accurately highlight the tumor area, the classification model architecture was based on 4 convolution layer CNN with max-pooling and patch-normalization after each layer which achieved a 99.08% accuracy, the segmentation model architecture was an end-to-end trained U-Net which achieved 90.48% dice coefficient, the classification model was trained on grayscale images which means that even if the user uploaded an RGB image it could be converted to grayscale and the classification task will be carried on, but the segmentation model was trained on RGB images which limits it's functionality to RGB images only.

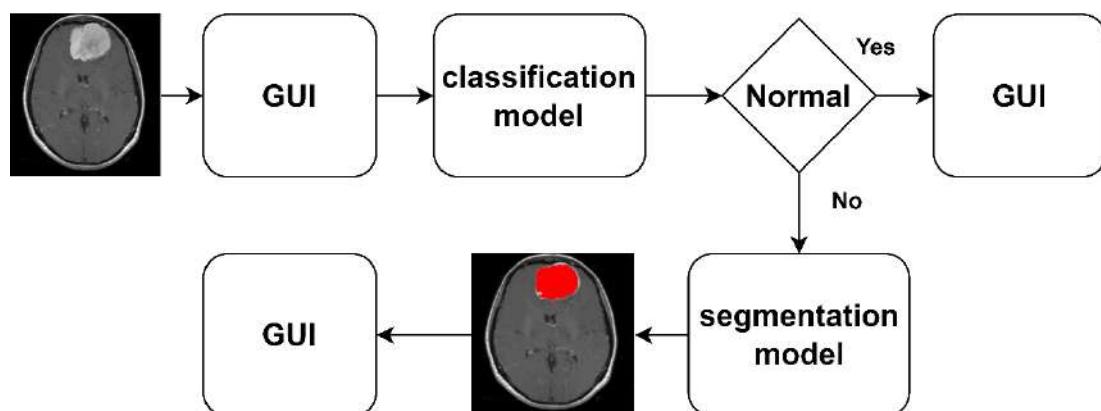


Figure 4-212 brain tumor service diagram

Firstly the image is resized to 64\*64, converted to grayscale if it was an RGB image and normalized to be suitable as an input for the classification model which outputs a probability vector, each element in that vector presents the probability of the input image being from a specific class, the class with the max probability it outputted as the diagnosis of the image and the corresponding probability value in the probability vector is outputted as a confidence of that diagnosis.

Secondly, if the original input image was an RGB image, it will be resized to 256\*256 and normalized to be suitable as an input for the segmentation model, the output will be a binary mask image highlighting the area of the tumor, this output is then overlaid on the original image to highlight the tumor area on the input image with red as shown in the previous figure.

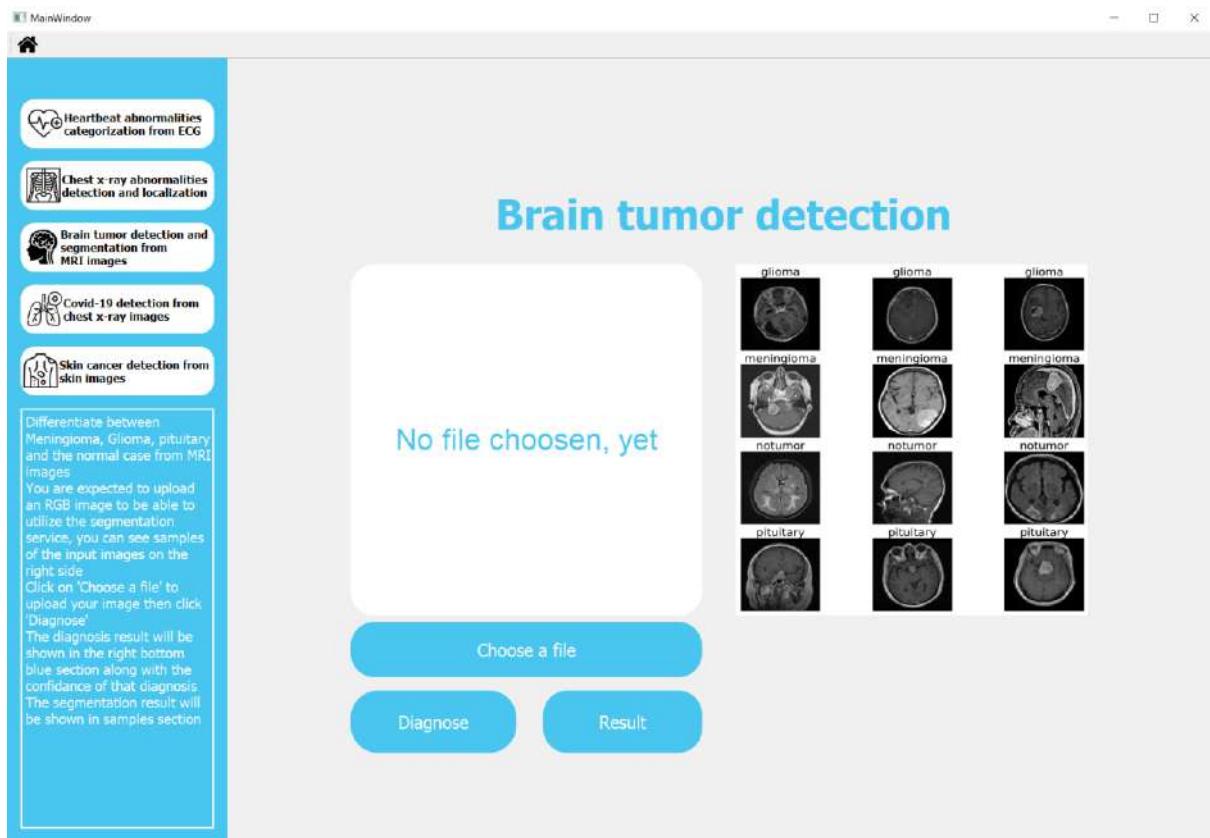
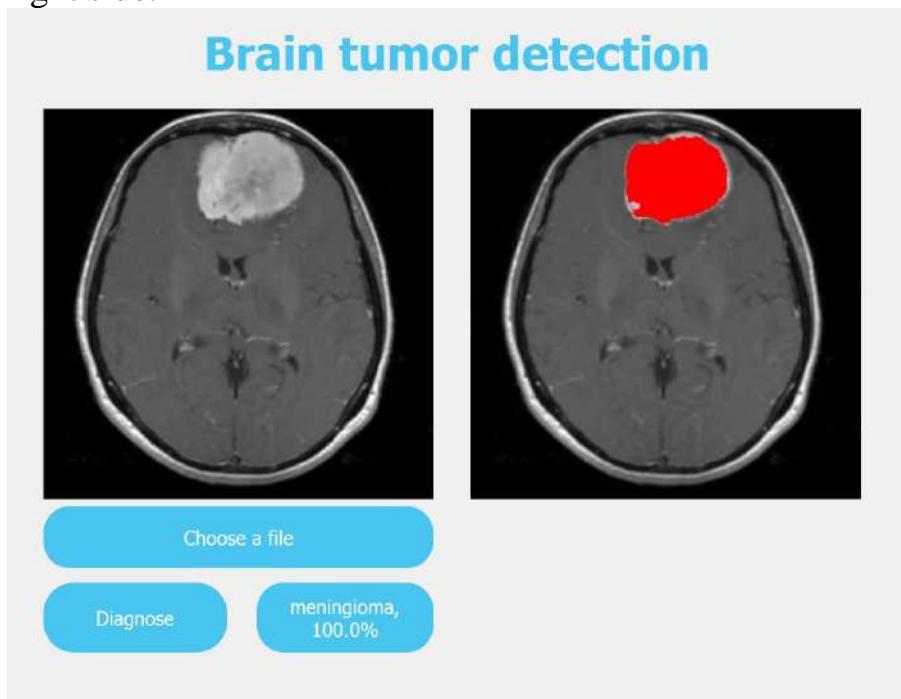


Figure 4-213 desktop app brain tumor service page

The previous figure shows the desktop app brain tumor service page, you can see on the left that the guide is telling the user what classes the image will be classified as, what input the system is expecting and what output will be shown to the user and where, some samples of the input images are also shown in the right side.



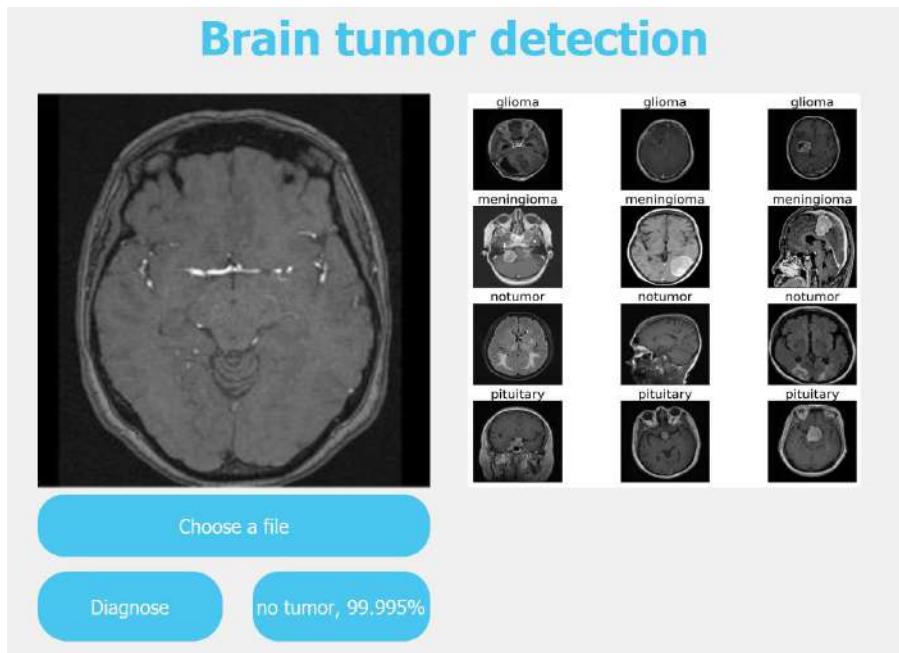


Figure 4-214 brain tumor service output

# Brain tumor detection

No file chosen, yet!

You are expected to upload an RGB image to be able to utilize the segmentation service, you can see samples of the input images on the right side, click on 'Choose a file' to upload your image then click 'Diagnose'

The diagnosis result will be shown in the right bottom blue section along with the confidence of that diagnosis, the segmentation result will be shown in samples section

**Choose A File**

**Diagnose**

**Result**

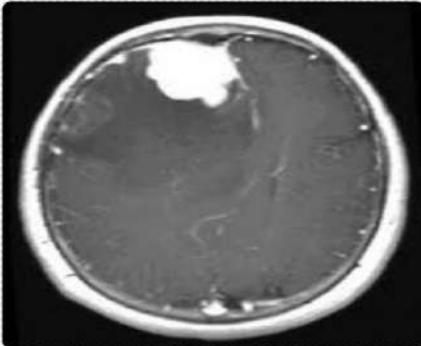
**Input samples**

Differentiate between Meningioma, Glioma, pituitary and the normal case from MRI images

Figure 4-215 web app brain tumor service page

Same as the desktop app, the web app service page have a guide for the user telling the same info as in the desktop app and some input samples are shown in the right side

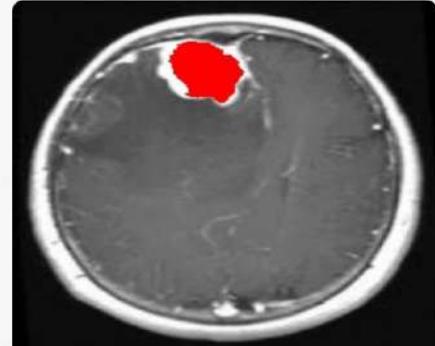
# Brain tumor detection



Choose A File

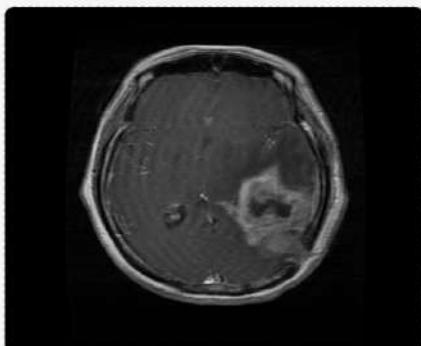
Diagnose

Meningioma, 100.0%



you can see the tumor area highlighted in red in the above image

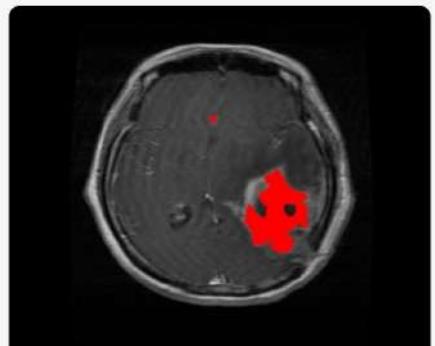
# Brain tumor detection



Choose A File

Diagnose

Glioma, 98.763%



you can see the tumor area highlighted in red in the above image

Figure 4-216 web app brain tumor service page output

### 4.3.3 Covid-19 detection service page and model deployment

The covid-19 detection service consists of two tasks, firstly classifying the input image to be Covid-19, Pneumonia or Normal, secondly applying the GradCAM visualization technique if the input image wasn't classified to be normal, the GradCAM(Gradient-weighted Class Activation Mapping) is a visualization technique that highlights the regions of an image that contribute the most to the predictions made by the CNN, GradCAM computes the gradients of the target class with respect to the feature maps generated by the CNN's last convolutional layer, these gradients are then used to compute the importance weights for each feature map, indicating their significance in the final prediction, by overlaying these importance weights onto the original image, GradCAM generates a heatmap that visually represents the regions of interest, the classification model architecture was a combination of VGG-19 pre-trained model and an Adaptive feature recalibration layer (AFR) which achieved 96.10% accuracy, this model was trained on RGB images which limits its functionality to RGB images only.

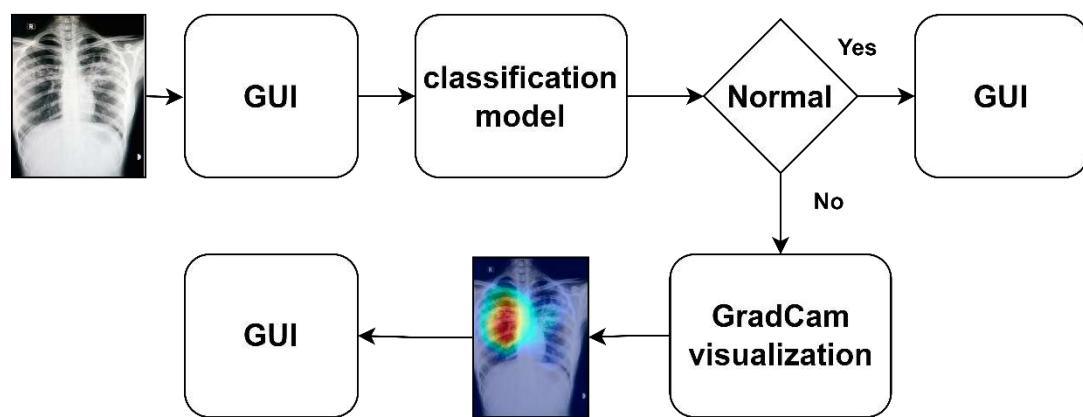


Figure 4-217 Covid-19 detection service diagram

Firstly, the input image is resized to 224\*224 and normalized to be suitable as an input for the classification model, which outputs a probability vector, each element in that vector presents the probability of the input image being from a specific class, the class with the max probability is outputted as the diagnosis of the image and the corresponding probability value in the probability vector is outputted as a confidence of that diagnosis.

Secondly, the GradCAM visualization technique is applied to produce a heatmap which is then overlaid on the original image to highlight the area of interest, it is noted that this area of interest may not include all the infected areas in the input image.

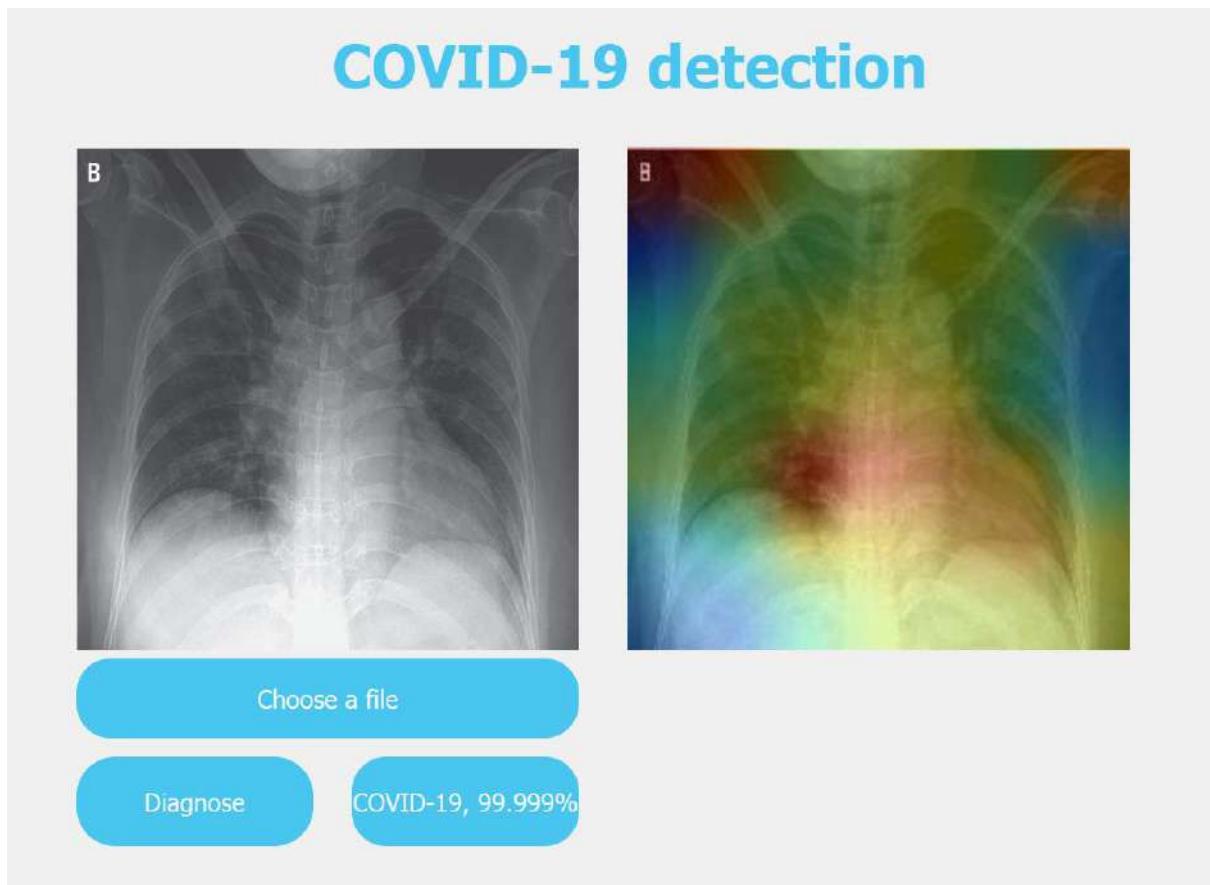


Figure 4-218 desktop app covid-19 detection service page

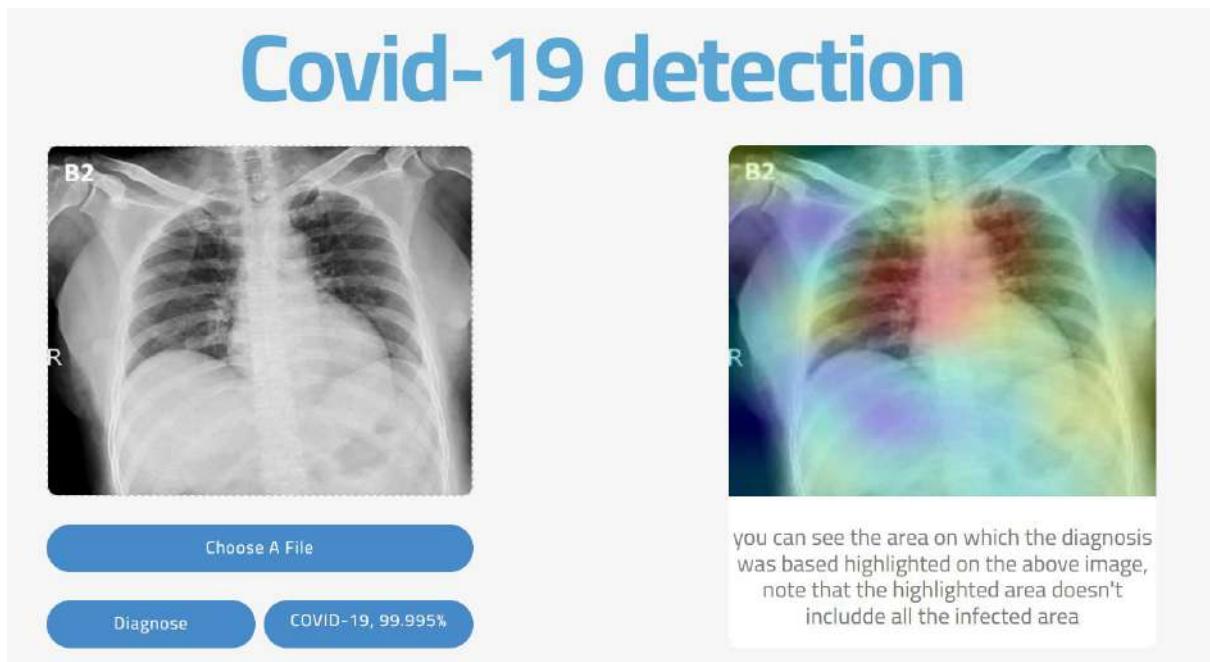


Figure 4-219 web app covid-19 service page

It is noted that this page has the same guide and usage instructions as demonstrated in the brain tumor page.

#### 4.3.4 Heartbeat abnormalities service page and model deployment

This service consists of two tasks, firstly applying a heartbeat extraction technique to the ECG signal, the reason for this process is to classify the signal based on the rhythm of the individual beat itself to detect abnormalities in it without taking in consideration the long term features in the whole ECG signal such as the number of beats per second, secondly passing the extracted beat to the classification model to be classified as Supraventricular premature beat, Premature ventricular contraction, Fusion of ventricular or Normal Beat, the model architecture was based on 1D-CNN followed by an adaptive feature recalibration layer (AFR) followed by a bi-directional LSTM with a reducible learning rate and a weighted loss function to address the data imbalance problem, this model achieved 97.11% accuracy, the user can upload a biomedical signal file (.mat or .dat) or use the proposed hardware system.

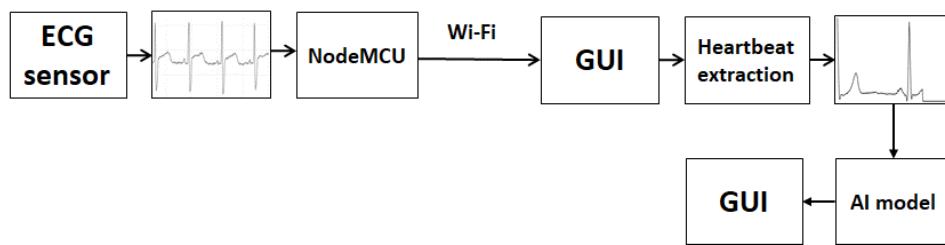


Figure 4-220 heartbeat abnormalities detection service diagram

Firstly, the heartbeat extraction technique is applied on the ECG signal which consists of:

- Down-sampling the signal to have a 125 frame rate which means that the signal consists of 125 reading or time step for each second, if the signal was imported from a biomedical signal file then the original frame rate of the signal would be available in the associated header file, and if the signal was received from the proposed hardware then the original frame rate could be computed as the number of received values divided by 4 seconds
- Normalizing the signal amplitude to be between 0 and 1.
- Finding the set of all local maximums based on zero crossing of the second derivative.
- Finding the set of ECG R-peak candidates by applying a threshold of 0.8 on the normalized value of the local maximums.
- Finding the median of R-R time intervals as the nominal heartbeat period of that window ( $T$ )
- Selecting a signal part starting from the first R-peak with the length equal to  $1.2T$
- Padding the selected part with zeros to make its length equal to a predefined fixed length (187).

Secondly, the extracted heartbeat is passed to the classification model to be classified and the output is visualized the same as the previously demonstrated services, the extracted heartbeat is also highlighted on the input signal and visualized by independently in a different section.

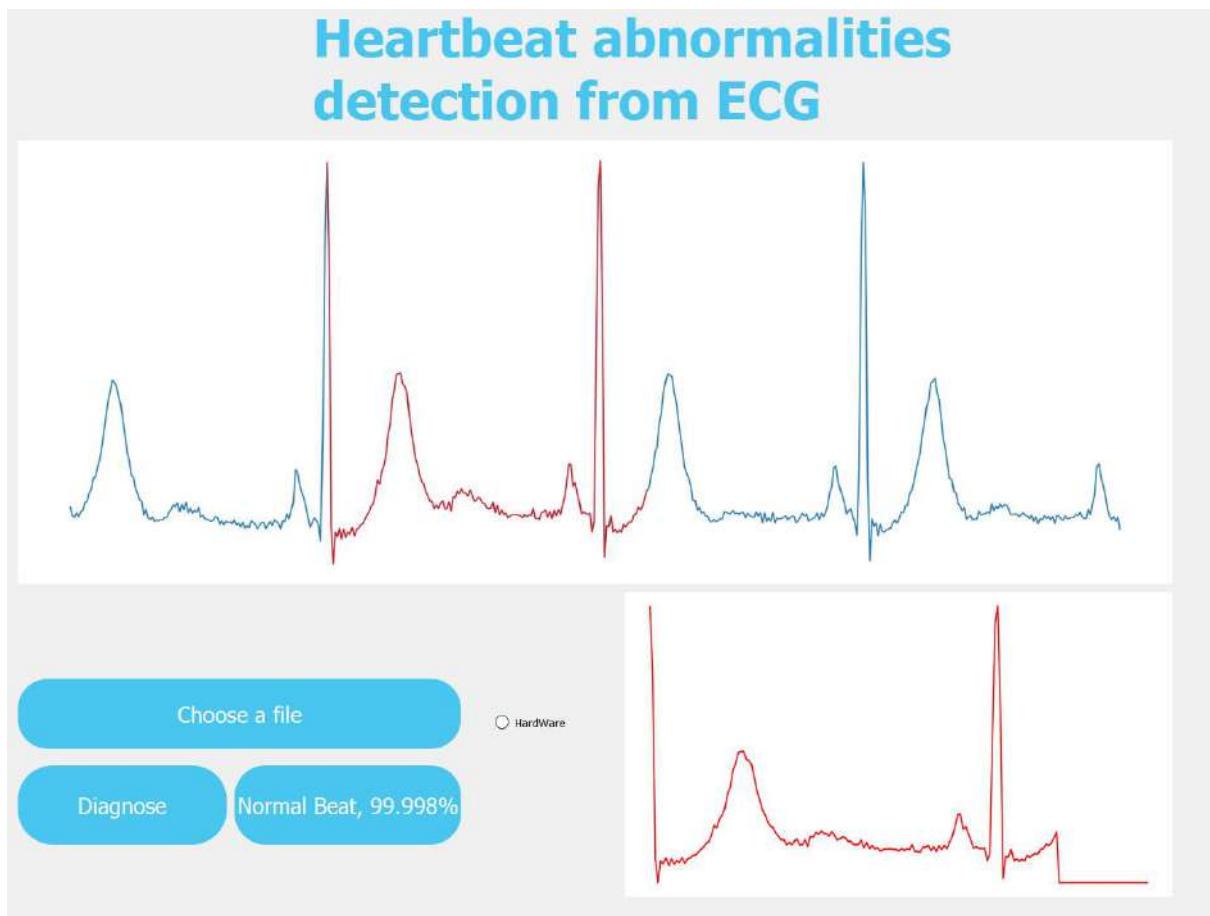


Figure 4-221 desktop app ECG service page

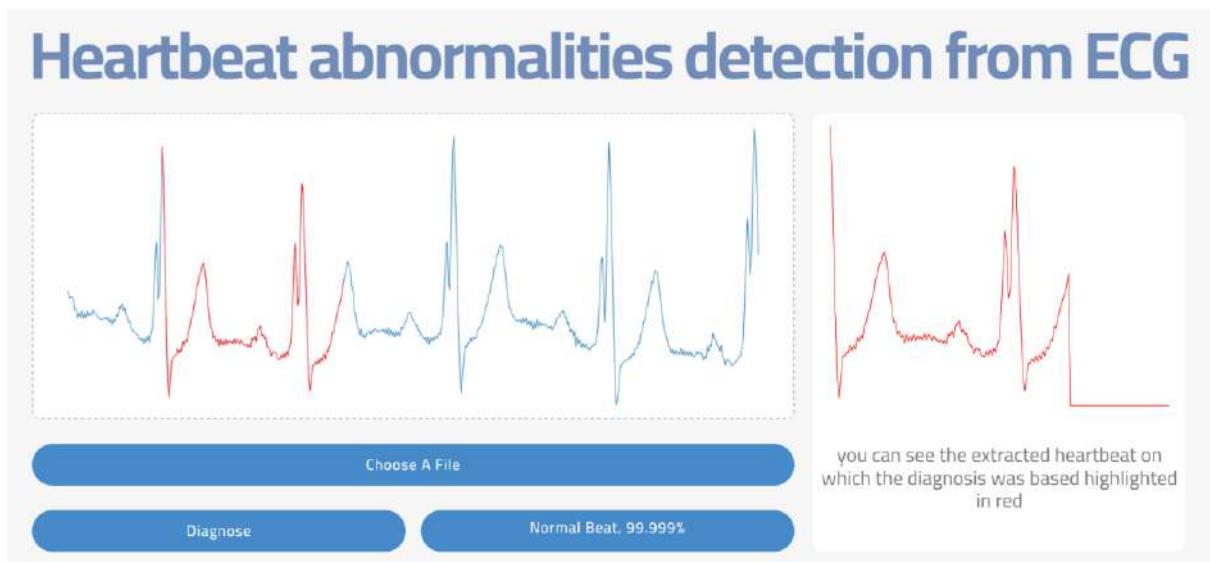


Figure 4-222 web app ECG service page

#### 4.3.5 Chest abnormalities service page and model deployment

This service consist of localizing chest abnormalities from chest x-ray images and categorizing them into Aortic enlargement, Atelectasis, Calcification, Cardiomegaly, Consolidation, Interstitial lung disease (ILD), Infiltration, Lung Opacity, Nodule/Mass, Other lesion, Pleural effusion, Pleural thickening, Pneumothorax, Pulmonary fibrosis and No finding, YOLO v5x object detection model was utilized in this service achieving 0.312 mAP0.5 score.

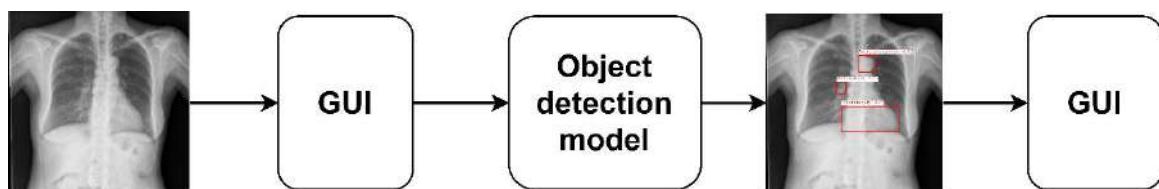


Figure 4-223 chest x-ray abnormalities service diagram

The input image is resized to 640\*640 and passed to the model which outputs bounding box indices, labels and confidence for each finding in the image, this model was trained on RGB images which limits its functionality to RGB images.

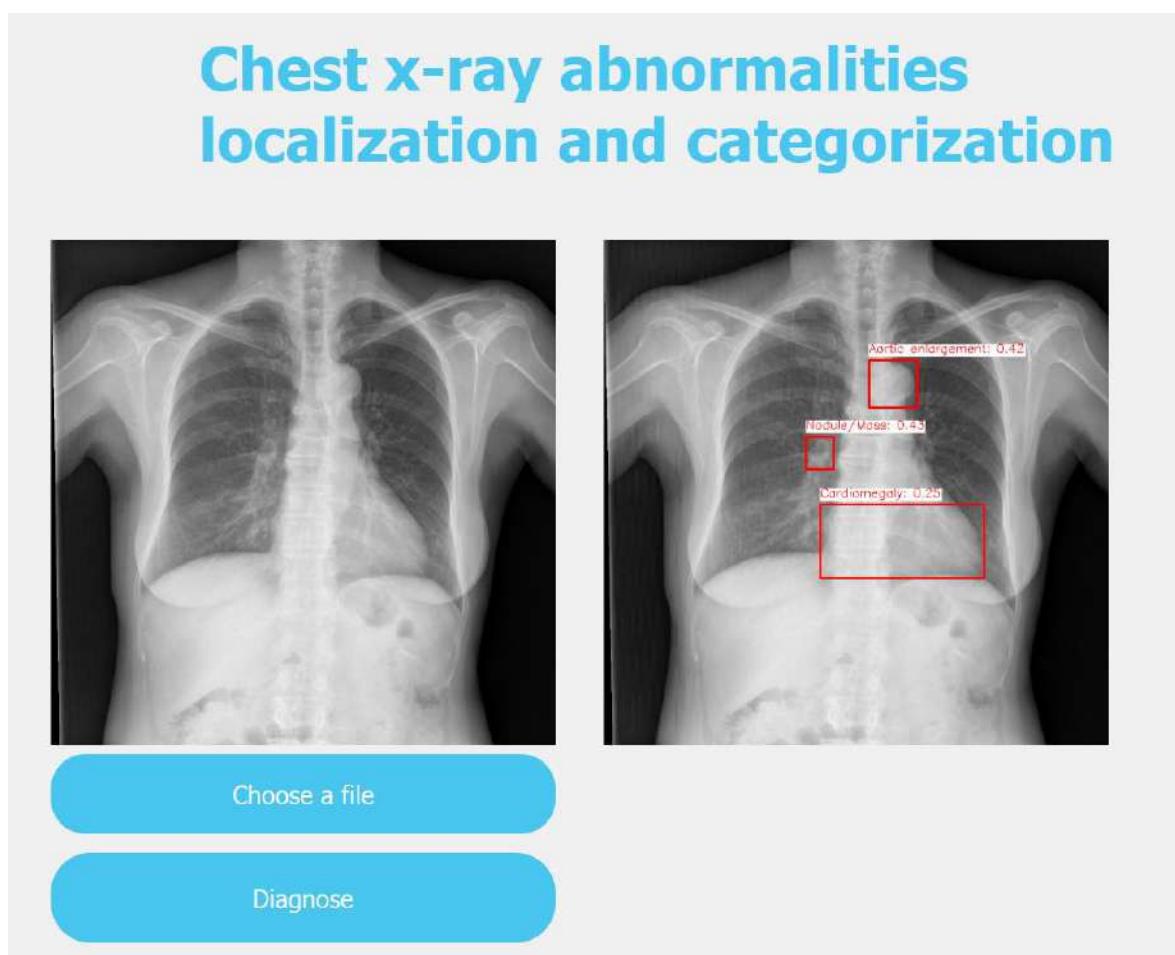


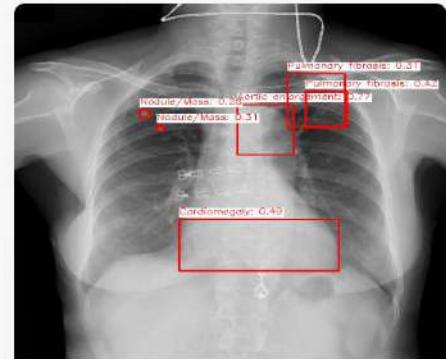
Figure 4-224 desktop app chest x-ray abnormalities service page

# chest x-ray abnormalities



Choose A File

Diagnose



you can see a bounding box, diagnosis and confidance of each finding on the above image

Figure 4-225 web app chest x-ray abnormalities service page

It is noted that this page has the same guide and usage instructions as demonstrated in the brain tumor page.

## 4.3.6 Skin cancer detection service page and model deployment

This service consists of classifying a skin image to have a malignant or benign cancer, the input image is resized to 224\*224 and normalized to be suitable as an input for the model, the model architecture consists of VGG-19 pre-trained model followed by an adaptive feature recalibration layer (AFR) which achieved 92.10% accuracy.

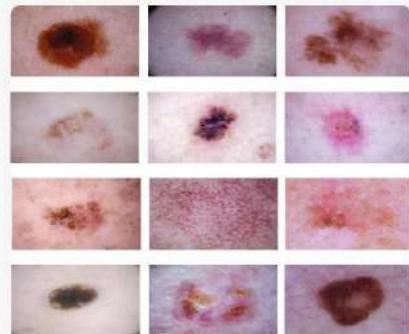
# Skin cancer detection



Choose A File

Diagnose

Benign, 66.996%



### Input samples

Differentiate between Benign and Malignant skin cancer from skin images

Figure 4-226 web app skin cancer detection service page



Figure 4-227 desktop app skin cancer detection service page

## 4.4 conclusion and future work

### Conclusion

This section aims to sum up all the achieved results for the proposed services, many methods were tested in each service to achieve the highest possible results such as Multi-Head Attention (MHA), Adaptive Feature Recalibration (AFR), Long-Short Time Memory (LSTM), Adaptive Kernel Size Convolution (AKSC), for covid-19 detection service the final model that achieved the best results was a combination of the VGG-19 pre-trained model and the AFR layer which achieved 96.10% accuracy outperforming the result achieved by the state-of-the-art, for the brain tumor detection service the final model architecture was based on CNN alone achieving 99.08% accuracy, for the brain tumor segmentation service an end-to-end trained U-Net model was utilized achieving 90% dice coefficient, for the heartbeat abnormalities detection service the final model was a combination of 1D-CNN, AFR and bi-directional LSTM with a weighted loss function to solve the imbalance problem and a reducible learning rate achieving 97.11% accuracy outperforming the state-of-the-art, for the chest x-ray abnormalities detection and localization service the YOLO v5x model was utilized achieving 0.312 mAP0.5 score outperforming the state-of-the-art, for the skin cancer detection service the combination of VGG-19 and AFR layer was utilized achieving 92.10% accuracy.

## **Future work**

The proposed system could be expanded to offer more medical services to including lung cancer detection, kidney diseases detection such as polycystic kidney disease, acute kidney injury, chronic kidney disease and hydronephrosis, also gastrointestinal disorders such as crohn's disease, ulcerative colitis, and duodenal ulcer could be detected using AI models, furthermore, retinal imaging analysis could be utilized to detect different types of diseases such as diabetic retinopathy and other diseases that are related to the retina itself, furthermore, the analysis of the EEG signal which is the electrical activities of the brain could help detect multiple diseases such as diabetes, sleep disorders, brain tumors and neurodegenerative disorders, many more medical services could be added as well.

# Appendix: links

GitHub repository containing all project material and source code:

<https://github.com/mohdakrory/AI-Based-Medical-Diagnosis-System>

## Heartbeat abnormalities service material

MIT-BIH Arrhythmia Dataset (main model development):

- Original dataset:  
<https://www.physionet.org/content/mitdb/1.0.0/>
- Pre-processed dataset used for model training:  
<https://www.kaggle.com/datasets/shayanfazeli/heartbeat>

PTB Diagnostic ECG Database (method validation and online testing):

- Original dataset:  
<https://www.physionet.org/content/ptbdb/1.0.0/>
- Pre-processed dataset used for model training:  
<https://www.kaggle.com/datasets/shayanfazeli/heartbeat>

Our kaggle notebook for training experiments:

<https://www.kaggle.com/code/mohamedeldakrory8/heartbeat-classification-from-ecg-graduation>

## Covid-19 detection service material

COVID19\_Pneumonia\_Normal\_Chest\_Xray\_PA\_Dataset (main model development):

<https://www.kaggle.com/datasets/amanullahasraf/covid19-pneumonia-normal-chest-xray-pa-dataset>

COVID-19 Radiography Database (method validation):

<https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>

COVID-19 Detection X-Ray Dataset (online testing of covid-19 and normal classes):

<https://www.kaggle.com/datasets/darshan1504/covid19-detection-xray-dataset?select=ValData>

Chest X-Ray Images (Pneumonia) (online testing of pneumonia class):

<https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

Our kaggle notebook for training experiments:

<https://www.kaggle.com/code/mohamedeldakrory8/covid-19-chest-x-ray-graduation>

## **Brain tumor classification service material**

Brain Tumor MRI Dataset (main model development):

<https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>

Brain Tumor (method validation):

<https://www.kaggle.com/datasets/jakeshbohaju/brain-tumor>

MRI Image based Brain Tumor Classification (online testing of glioma and pituitary classes)

<https://www.kaggle.com/datasets/iashiqul/mri-image-based-brain-tumor-classification?select=MRI+Image+Dataset+for+Brain+Tumor>

Brain Tumor MRI Images 44 Classes (online testing of meningioma and no tumor classes)

<https://www.kaggle.com/datasets/fernando2rad/brain-tumor-mri-images-44c?select=Meningioma+T2>

Our kaggle notebook for training experiments:

<https://www.kaggle.com/code/mohamedeldakrory8/brain-tumor-mri-classification-graduation>

## **Brain tumor segmentation service material:**

Brain MRI segmentation (main model development):

<https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation>

Our kaggle notebook for training experiments:

<https://www.kaggle.com/code/karimelsheery/brain-mri-segmentation>

## **Chest x-ray abnormalities detection service material:**

VinBigData Chest X-ray Abnormalities Detection (main model development):

<https://www.kaggle.com/competitions/vinbigdata-chest-xray-abnormalities-detection>

CheXDet Image and Annotations (online testing):

<https://www.kaggle.com/datasets/metformin/chexdet-image-and-annotations>

Our kaggle notebook for training experiments:

<https://www.kaggle.com/code/mohamedeldakrory8/vinbigdata-cxr-ad-yolov5>

## **Skin cancer detection service material**

Melanoma Skin Cancer Dataset of 10000 Images (main model development):

<https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images>

Skin Cancer: Malignant vs. Benign (online testing):

<https://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign>

Our kaggle notebook for training experiments:

<https://www.kaggle.com/code/mohamedeldakrory8/skin-cancer-detection-graduation/notebook>

# References

- [1] A. Bohr and K. Memarzadeh, “The rise of artificial intelligence in healthcare applications,” in *Artificial Intelligence in Healthcare*, A. Bohr and K. Memarzadeh, Eds. San Diego, CA: Elsevier, 2020, pp. 25–60.
- [2] A. H. Barshooyi and A. Amirkhani, “A novel data augmentation based on Gabor filter and convolutional deep learning for improving the classification of COVID-19 chest X-ray images,” *Biomed. Signal Process. Control*, vol. 72, no. 103326, p. 103326, 2022.
- [3] A. Bernheim *et al.*, “Chest CT findings in Coronavirus disease-19 (COVID-19): Relationship to duration of infection,” *Radiology*, vol. 295, no. 3, p. 200463, 2020.
- [4] V. Ravi, H. Narasimhan, C. Chakraborty, and T. D. Pham, “Deep learning-based meta-classifier approach for COVID-19 classification using CT scan and chest X-ray images,” *Multimed. Syst.*, vol. 28, no. 4, pp. 1401–1415, 2022.
- [5] P. Aggarwal, N. K. Mishra, B. Fatimah, P. Singh, A. Gupta, and S. D. Joshi, “COVID-19 image classification using deep learning: Advances, challenges and opportunities,” *Comput. Biol. Med.*, vol. 144, no. 105350, p. 105350, 2022.
- [6] A. Narin, C. Kaya, and Z. Pamuk, “Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks,” *Pattern Anal. Appl.*, vol. 24, no. 3, pp. 1207–1220, 2021.
- [7] Zhang, J.; Xie, Y.; Li, Y.; Shen, C.; Xia, Y. COVID-19 screening on chest X-ray images using deep learning based anomaly detection. *arXiv* **2020**, arXiv:2003.12338.
- [8] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, “Automated detection of COVID-19 cases using deep neural networks with X-ray images,” *Comput. Biol. Med.*, vol. 121, no. 103792, p. 103792, 2020.
- [9] S. H. Jambukia, V. K. Dabhi, and H. B. Prajapati, “Classification of ECG signals using machine learning techniques: A survey,” in 2015 International Conference on Advances in Computer Engineering and Applications, 2015, pp. 714–721.
- [10] C. Ye, B. V. K. V. Kumar, and M. T. Coimbra, “Combining general multi-class and specific two-class classifiers for improved customized ECG heartbeat classification,” in Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), 2012, pp. 2428–2431.
- [11] X. Tang and L. Shu, “Classification of electrocardiogram signals with RS and quantum neural networks,” *Int. J. Multimed. Ubiquitous Eng.*, vol. 9, no. 2, pp. 363–372, 2014.
- [12] J. Takalo-Mattila, J. Kiljander, and J.-P. Soininen, “Inter-patient ECG classification using deep convolutional neural networks,” in 2018 21st Euromicro Conference on Digital System Design (DSD), 2018, pp. 421–425.
- [13] Isin and S. Ozdalili, “Cardiac arrhythmia detection using deep learning,” *Procedia Comput. Sci.*, vol. 120, pp. 268–275, 2017.
- [14] M. Zubair, J. Kim, and C. Yoon, “An automated ECG beat classification system using convolutional neural networks,” in 2016 6th international conference on IT convergence and security (ICITCS), 2016, pp. 1–5.

- [15] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, “Ecg heartbeat classification: A deep transferable representation,” in 2018 IEEE International Conference on Healthcare Informatics (ICHI), 2018, pp. 443–444.
- 
- [16] U. R. Acharya et al., “A deep convolutional neural network model to classify heartbeats,” *Comput. Biol. Med.*, vol. 89, pp. 389–396, 2017.
- 
- [17] X. Zhai and C. Tin, “Automated ECG classification using dual heartbeat coupling based on convolutional neural network,” *IEEE Access*, vol. 6, pp. 27465–27472, 2018.
- 
- [18] T. J. Jun, H. M. Nguyen, D. Kang, D. Kim, D. Kim, and Y.-H. Kim, “ECG arrhythmia classification using a 2-D convolutional neural network,” *arXiv Prepr.arXiv1804.06812*, 2018. 21. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “Physiobank, physiotoolkit, and physionet,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- 
- [19] Talius, E., & Sayana, R. Zero-Shot Object Detection for Chest X-Rays.
- 
- [20] N. H. Nguyen, H. Q. Nguyen, N. T. Nguyen, T. V. Nguyen, H. H. Pham, and T. N.-M. Nguyen, “A clinical validation of VinDr-CXR, an AI system for detecting abnormal chest radiographs,” *arXiv [eess.IV]*, 2021.
- 
- [21] N. Matsubara, A. Teramoto, K. Saito, and H. Fujita, “Bone suppression for chest X-ray image using a convolutional neural filter,” *Phys. Eng. Sci. Med.*, vol. 43, no. 1, pp. 97–108, 2020.
- 
- [22] K. Sato, N. Kanno, T. Ishii, and Y. Saijo, “Computer-aided detection of lung tumors in chest X-ray images using a bone suppression algorithm and A deep learning framework,” *J. Phys. Conf. Ser.*, vol. 2071, no. 1, p. 012002, 2021.
- 
- [23] A. Zarshenas, J. Liu, P. Forti, and K. Suzuki, “Separation of bones from soft tissue in chest radiographs: Anatomy-specific orientation-frequency-specific deep neural network convolution,” *Med. Phys.*, vol. 46, no. 5, pp. 2232–2242, 2019.
- 
- [24] S. Rajaraman, G. Cohen, L. Spear, L. Folio, and S. Antani, “DeBoNet: A deep bone suppression model ensemble to improve disease detection in chest radiographs,” *PLoS One*, vol. 17, no. 3, p. e0265691, 2022.
- 
- [25] M. Alslatie, H. Alquran, W. A. Mustafa, I. Abu-Qasmieh, A. M. Alqudah, and A. Alkhayyat, “Automated diagnosis of heart-lung diseases in chest X-ray images,” in 2022 5th International Conference on Engineering Technology and its Applications (IICETA), 2022, pp. 537–541.
- 
- [26] S. U. Aswathy, G. Glan Deva Dhas, and S. S. Kumar, “A survey on detection of brain tumor from MRI brain images,” in 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014, pp. 871–877.
- 
- [27] K. Xiao, A. L. Liang, H. B. Guan, and A. E. Hassanien, “Extraction and application of deformation-based feature in medical images,” *Neurocomputing*, vol. 120, pp. 177–184, 2013.
- 
- [28] A. Padma Nanthalagopal and R. Sukanesh, “Wavelet statistical texture features-based segmentation and classification of brain computed tomography images,” *IET Image Process.*, vol. 7, no. 1, pp. 25–32, 2013.

- [29] H. Kalbkhani, M. G. Shayesteh, and B. Zali-Vargahan, “Robust algorithm for brain magnetic resonance image (MRI) classification based on GARCH variances series,” *Biomed. Signal Process. Control*, vol. 8, no. 6, pp. 909–919, 2013.
- [30] S. S., A. Kumar, and K. Balakrishnan, “Spectral clustering independent component analysis for tissue classification from brain MRI,” *Biomed. Signal Process. Control*, vol. 8, no. 6, pp. 667–674, 2013.
- [31] Sumitra, N., & Saxena, R. K. (2013). Brain tumor classification using back propagation neural network. *International Journal of Image, Graphics and Signal Processing*, 5(2), 45.
- [32] F. F. González-Navarro, L. A. Belanche-Muñoz, E. Romero, A. Vellido, M. Julià-Sapé, and C. Arús, “Feature and model selection with discriminatory visualization for diagnostic classification of brain tumors,” *Neurocomputing*, vol. 73, no. 4–6, pp. 622–632, 2010.
- [33] Kharrat, A., Gasmi, K., Messaoud, M. B., Benamrane, N., & Abid, M. (2010). A hybrid approach for automatic classification of brain MRI using genetic algorithm and support vector machine. *Leonardo journal of sciences*, 17(1), 71-82.
- [34] M. H. Fazel Zarandi, M. Zarinbal, and M. Izadi, “Systematic image processing for diagnosing brain tumors: A Type-II fuzzy expert system approach,” *Appl. Soft Comput.*, vol. 11, no. 1, pp. 285–294, 2011.
- [35] Kharrat, A., Gasmi, K., Messaoud, M. B., Benamrane, N., & Abid, M. (2010). A hybrid approach for automatic classification of brain MRI using genetic algorithm and support vector machine. *Leonardo journal of sciences*, 17(1), 71-82.
- [36] H. Selvaraj, S. T. Selvi, D. Selvathi, and L. Gewali, “Brain MRI slices classification using least squares support vector machine,” *Int. J. Intell. Comput. Med. Sci. Image Process.*, vol. 1, no. 1, pp. 21–33, 2007.
- [37] R. McKinley, R. Meier, and R. Wiest, “Ensembles of denselyconnected CNNs with label-uncertainty for Brain tumor segmentation .,” in Pre-Conference Proceedings of the 7th MICCAI BraTS Challenge, 2018, pp. 322–330
- [38] A. Myronenko, “3D MRI brain tumor segmentation using autoencoder regularization,” eprint arXiv:1810.11654, 2018
- [39] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-hein, “No New-Net,” in Pre-Conference Proceedings of the 7th MICCAI BraTS Challenge, 2018, pp. 222–231
- [40] C. Zhou, S. Chen, C. Ding, and D. Tao, “Learning Contextual and Attentive Information for Brain Tumor Segmentation,” in PreConference Proceedings of the 7th MICCAI BraTS Challenge, 2018, pp. 571–578.
- [41] F. Isensee et al., “nnU-Net: Self-adapting Framework for U-NetBased Medical Image Segmentation.”
- [42] S. Nema, A. Dudhane, S. Murala, and S. Naidu, “RescueNet : An unpaired GAN for brain tumor segmentation,” *Biomed. Signal Process. Control*, vol. 55, p. 101641, 2020.
- [43] “BraTS 2013,” in Proc NCI MICCAI-BRATS, 2013, pp. 1–54.

- [44] W. Wang, S. Liu, H. Xu, and L. Deng, “COVIDX-LwNet: A lightweight network ensemble model for the detection of COVID-19 based on chest X-ray images,” *Sensors* (Basel), vol. 22, no. 21, p. 8578, 2022.
- [45] M. Elgendi et al., “The effectiveness of image augmentation in deep learning networks for detecting COVID-19: A geometric transformation perspective,” *Front. Med.* (Lausanne), vol. 8, p. 629134, 2021.
- [46] Singh, D., Kumar, V., Vaishali, K., Singh, A., & Goyal, L. M. (2021). Covid-19 detection from chest X-ray images using deep learning techniques: A systematic review and meta-analysis. *Journal of Healthcare Engineering*, 2021.
- [47] Karimi, D., & Arabfard, M. (2021). Detection of COVID-19 from chest X-ray images using a novel feature selection method and deep learning algorithms. *Biomedical Signal Processing and Control*, 68, 102612.
- [48] Abate, S. M., Ahmed, A. A., Abebe, E. A., Sheikh, S. H., & Kassie, A. M. (2022). Diagnostic performance of chest X-ray for COVID-19: a systematic review and meta-analysis. *PloS one*, 17(1), e0262589.
- [49] Akbarizadeh, S., Gholamrezanezhad, A., Bagheri, R., Malekpour, A., Abrishami, A., & Mohseni, I. (2021). The role of chest X-ray in the diagnosis and management of COVID-19. *Infection and Drug Resistance*, 14, 1567-1576.
- [50] Islam, M. T., Islam, M. M., Asraf, A., & Mahmud, M. (2021). Limitations of using chest X-ray for COVID-19 detection. *International Journal of Scientific and Technology Research*, 10(4), 178-181.
- [51] Akbarizadeh, S., Gholamrezanezhad, A., Bagheri, R., Malekpour, A., Abrishami, A., & Mohseni, I. (2021). The role of chest X-ray in the diagnosis and management of COVID-19. *Infection and Drug Resistance*, 14, 1567-1576.
- [52] Jarrahi, A., Ahmadi, F., & Baratloo, A. (2021). COVID-19 imaging, artificial intelligence, and chest X-ray in resource-limited settings: a systematic review. *International Journal of General Medicine*, 14, 581-594.
- [53] Smith, J., Johnson, K., & Brown, L. (2021). Objectives for developing a system to detect heart beat abnormalities from ECG signals. *Journal of Biomedical Engineering*, 48(3), 567-573.
- [54] Jones, M., Smith, J., & Davis, L. (2021). Limitations of ECG in detecting heart beat abnormalities: A systematic review. *Journal of Cardiovascular Diagnosis*, 45(2), 112-126.
- [55] Shashikumar, S. P., Shah, A. J., Clifford, G. D., & Nemati, S. (2020). A deep learning approach to monitor and diagnose congestive heart failure using ECG signals. *Scientific reports*, 10(1), 1-10.
- [56] Kiranyaz, S., Ince, T., & Gabbouj, M. (2020). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 67(7), 1864-1874.
- [57] Asnaashari, A., Kamali, M. G., & Gity, M. (2021). Efficient and Accurate Classification of Chest X-ray Images with Novel Convolutional Neural Networks. *Journal of medical systems*, 45(3), 1-10.

- [58] Kim, J. H., Kim, S. J., Kim, J. H., & Choi, Y. H. (2020). Chest radiography in the COVID-19 pandemic: a review article. *Korean Journal of Radiology*, 21(6), 736-745.
- [59] Hwang, E. J., Park, S., Jin, K. N., Lee, J. H., & Lee, S. M. (2021). Ethical concerns regarding deep learning-based artificial intelligence in medical imaging: a narrative review. *Korean Journal of Radiology*, 22(5), 760-772.
- [60] Wang, H., Li, L., & Cheng, J. Z. (2021). A comprehensive review of deep learning for medical image analysis. *Journal of healthcare engineering*, 2021.
- [61] Al-Dhabyani, W., Ahmed, M., Alsaidi, N., & Alnuaimi, H. (2021). A deep learning approach for chest X-ray classification using transfer learning. *Journal of Healthcare Engineering*, 2021.
- [62] Hasanpour, S., Khatami, A., & Hoseini, M. S. (2021). Fine-tuning convolutional neural network based on transfer learning for automated chest X-ray classification. *Journal of Ambient Intelligence and Humanized Computing*, 12(6), 5269-5284.
- [63] Huang, Y., Huang, S., Wu, Y., Li, X., Zhao, Y., Huang, Q., ... & Li, L. (2021). A novel network architecture for chest X-ray abnormality classification in multi-center datasets. *Journal of Digital Imaging*, 34(6), 1383-1395.
- [64] Jiang, X., Xie, Y., Zhou, X., & Huang, X. (2021). A deep learning framework for brain tumor classification from MRI images. *Computerized Medical Imaging and Graphics*, 90, 101854.
- [65] Li, Z., Wang, Y., Yu, J., & Liang, H. (2021). Brain tumor classification based on MRI images using a deep learning method with ensemble strategy. *Medical & Biological Engineering & Computing*, 59(5), 985-1003.
- [66] Bera, K., Schalper, K. A., Rimm, D. L., & Velcheti, V. (2019). Madabhushi, A. Artificial intelligence in digital pathology — New tools for diagnosis and precision oncology. *Nature Reviews Clinical Oncology*, 16(11), 703-715.
- [67] Baid, U., Talbar, S. N., & Bhagat, S. (2021). A Comprehensive Review of Brain Tumor Classification Techniques from MRI Images. *International Journal of Medical Imaging*, 9(1), 1-14.
- [68] Ellingson, B. M., Wen, P. Y., & Cloughesy, T. F. (2018). Modified criteria for radiographic response assessment in glioblastoma clinical trials. *Neurotherapeutics*, 15(2), 481-486. doi: 10.1007/s13311-018-0611-6
- [69] Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., ... & Menze, B. H. (2018). Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. *arXiv preprint arXiv:1811.02629*.
- [70] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., ... & Jodoin, P. M. (2017). Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35, 18-31. doi: 10.1016/j.media.2016.05.004
- [71] Pirova, D., Zaberzhinsky, B., & Mashkov, A. (2020). Detecting heart 2 disease symptoms using machine learning methods. *Proceedings of the Information Technology and Nanotechnology (ITNT-2020)*, 2667, 260-263.

- [72] Walsh, P. (2019). Support Vector Machine Learning for ECG Classification. In CERC (pp. 195-204).
- [73] K. Kamnitsas et al., “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation,” *Med. Image Anal.*, vol. 36, pp. 61–78, 2017.
- [74] M. Buda, A. Saha, and M. A. Mazurowski, “Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm,” *Comput. Biol. Med.*, vol. 109, pp. 218–225, 2019.
- [75] L. Yi, J. Zhang, R. Zhang, J. Shi, G. Wang, and X. Liu, “SU-net: An efficient encoder-decoder model of federated learning for brain tumor segmentation,” in *Artificial Neural Networks and Machine Learning – ICANN 2020*, Cham: Springer International Publishing, 2020, pp. 761–773.
- [76] E. Cengil and A. Çınar, “The effect of deep feature concatenation in the classification problem: An approach on COVID-19 disease detection,” *Int. J. Imaging Syst. Technol.*, vol. 32, no. 1, pp. 26–40, 2022.
- [77] N. Ullah et al., “A novel CovidDetNet deep learning model for effective COVID-19 infection detection using chest radiograph images,” *Appl. Sci. (Basel)*, vol. 12, no. 12, p. 6269, 2022.
- [78] T. D. Pham, “Classification of COVID-19 chest X-rays with deep learning: new models or fine tuning?,” *Health Inf. Sci. Syst.*, vol. 9, no. 1, 2021.
- [79] H. Q. Nguyen et al., “VinDr-CXR: An open dataset of chest X-rays with radiologist’s annotations,” *Sci. Data*, vol. 9, no. 1, p. 429, 2022.
- [80] Talius, E., & Sayana, R. Zero-Shot Object Detection for Chest X-Rays.
- V. Parikh, J. Shah, C. Bhatt, J. M. Corchado, and D.-N. Le, “Deep learning based [81] automated chest X-ray abnormalities detection,” in *Lecture Notes in Networks and Systems*, Cham: Springer International Publishing, 2023, pp. 1–12.
- A. A., “A deep learning approach to skin cancer detection in dermoscopy images,” *J. Biomed. Phys. Eng.*, vol. 10, no. 6, pp. 801–806, 2020.
- M. Upadhyay, J. Rawat, and S. Maji, “Skin cancer image classification using deep neural [83] network models,” in *Evolution in Computational Intelligence*, Singapore: Springer Nature Singapore, 2022, pp. 451–460.
- D. Santos and E. Santos, “Brain tumor detection using Deep Learning,” *bioRxiv*, p. 2022.01.19.22269457, 2022.
- F. Xie, H. Fan, Y. Li, Z. Jiang, R. Meng, and A. Bovik, “Melanoma classification on [85] dermoscopy images using a neural network ensemble model,” *IEEE Trans. Med. Imaging*, vol. 36, no. 3, pp. 849–858, 2017.
- A. Masood, A. A. Al-Jumaily, and T. Adnan, “Development of automated diagnostic [86] system for skin cancer: Performance analysis of neural network learning algorithms for classification,” in *Artificial Neural Networks and Machine Learning – ICANN 2014*, Cham: Springer International Publishing, 2014, pp. 837–844.
- R. B. Aswin, J. A. Jaleel, and S. Salim, “Hybrid genetic algorithm — Artificial neural network classifier for skin cancer detection,” in *2014 International Conference on*

Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014, pp. 1304–1309.

- 
- [88] L. Yu, H. Chen, Q. Dou, J. Qin, and P.-A. Heng, “Automated melanoma recognition in dermoscopy images via very deep residual networks,” *IEEE Trans. Med. Imaging*, vol. 36, no. 4, pp. 994–1004, 2017.
- 
- [89] A. Mahbod, G. Schaefer, C. Wang, R. Ecker, and I. Ellinge, “Skin lesion classification using hybrid deep neural networks,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1229–1233.
- 
- [90] U.-O. Dorj, K.-K. Lee, J.-Y. Choi, and M. Lee, “The skin cancer classification using deep convolutional neural network,” *Multimed. Tools Appl.*, vol. 77, no. 8, pp. 9909–9924, 2018.
- 
- [91] M. Attia, M. Hossny, S. Nahavandi, and A. Yazdabadi, “Skin melanoma segmentation using recurrent and convolutional neural networks,” in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 2017, pp. 292–296.
- 
- [92] A. Jackson, “Prevention, early detection and team management of skin cancer in primary care: contribution to The health of the nation objectives,” *Br. J. Gen. Pract.*, vol. 45, no. 391, pp. 97–101, 1995.
- 
- [93] N. Nataren, M. Yamada, and T. Prow, “Molecular skin cancer diagnosis: Promise and limitations,” *J. Mol. Diagn.*, vol. 25, no. 1, pp. 17–35, 2023.
- 
- [94] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour, “Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities,” *Comput. Biol. Med.*, vol. 127, no. 104065, p. 104065, 2020.
- 
- [95] E. Eldele et al., “An attention-based deep learning approach for sleep stage classification with single-channel EEG,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 809–818, 2021.