An Efficient Deep Learning Approach To Predict Heart Failure From Image Data Using Ejection Fraction

by

Nazmul Karim Tanvir
18101054
MD. Nadim Yeasin
18101560
Sarker MD. Mahmuduzzaman
18301073
Jannat Ara
17301065

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Nazmul Karim Tanvir
18101054

MD. Nadim Yeasin
18101560

Sarker MD. Mahmuduzzaman
18301073

Jannat Ara
17301065

Approval

The thesis/project titled "An Efficient Deep Learning Approach To Predict Heart Failure From Image Data Using Ejection Fraction" submitted by

1. Nazmul Karim Tanvir(18101054)
2. MD. Nadim Yeasin(18101560)
3. Sarker MD. Mahmuduzzaman(18301073)
4. Jannat Ara(17301065)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 20, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Ashraful Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Md Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

____

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Heart is the core of human body. A normal heart beats almost 1,15,200 times in a day and 80 beats per second to make us live alive. But we often take it granted and do uncertain thinks which stops it to function perfectly. In today's world cardiovascular diseases(CVDs) almost kill 17-18 million life's each year worldwide which makes it the biggest disease of death. If early detection of heart malfunction or Heart failure(HF) can be detect millions of people will able to breath even longer than usual. In our research our main aim is to create an automated Deep Learning based model which will predict HF and the depth of the condition. Moreover, using which type of cardiac MRI image slice we can get better result will be consider to be our main research goal. For this we choose a cardiac MRI dataset which consists of 1100 different heart patients image having different slices in different pattern. Furthermore, with more observation and leveling different parameter with the help of Ejection Fraction(EF) values which depends on systole diastole value of heart we able to predict the heart failure with an efficient result. AI, ML & deep learning is the new trend for solving real life human problems. We used different Convolution Neural Network architecture and obtained accuracy are VGG-16(88.15%), VGG-19(87.93%), ResNet-50 (75.85%), ResNet-101 (79.53%) Inception-V3 (85.27%). Our model is being used to find the suitable result to detect the Heart Failure(HF) with Ejection Fraction(EF).

Keywords: Cardiovascular Disease(CVDs), heart failure(HF), ejection fraction(EF), Deep learning, CNN, Vgg-19, Vgg-16, Inception-V3, Cardiac MRI Data

Acknowledgement

Firstly, all praise to the Great Almighty for whom our thesis have been completed without any major interruption. Secondly, to our advisor Dr. Md. Ashraful Alam sir and co-advisor Md. Tanzim Reza sir for their kind support and advice in our work. They helped us whenever we needed help. And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Cardiovascular diseases (CVDs) is responsible for taking 17.9 million lives worldwide each year[1] and it causes 1 out of every 4 deaths worldwide. At present people are Eating foods high in fat which is increasing their cholesterol and sodium level in body. Moreover, not getting involve in enough physical activity, excessive alcohol intake is the reasons of HF (CADs in general). One of the main leading disease which cause death in the US and world wide is Cardiovascular disease. It includes coronary heart disease or heart attacks, cerebrovascular diseases or strokes, coronary artery blockages(CAB), and heart failure(HF).

Especially, heart failure happens when the heart is unable to pump enough blood to the body. Diabetes, excessive blood pressure, or other heart problems or diseases are typically the cause of HF. Premature deaths can be avoided by determining who is most vulnerable to HF and making sure they receive the right care. To ensure that those in need receive care and counseling, an early prediction of HF and identifying the disease condition is required.

Although heart failure is a serious medical condition, it does not indicate that the heart has stopped beating or that there are varying degrees of heart failure. Ejection fraction, the percentage of blood pumped by the heart during a single contraction, is physiologically measured between 50% and 75%, and is used by the medical community to classify patients with heart failure into one of two categories. In the former case, an ejection fraction of 40% or less denotes heart failure with reduced ejection fraction (HFrEF), also known as systolic heart failure or heart failure with left ventricular (LV) systolic dysfunction. Diastolic heart failure, also known as heart failure with normal ejection fraction (HFpEF), is the second type of heart failure. Here, the left ventricle has a normal systolic contraction. However, diastole ventricle rigidity and irregular relaxation impair filling.

Heart disease is characterized by cardiac dysfunction. Doctors evaluate cardiac health by measuring a patient's end-systolic and end-diastolic volumes, or the size of a single heart chamber at the beginning and center of each heartbeat. The ejection fraction is derived from these volumes (EF). The EF measures the volume of blood pumped out by the left ventricle during each cardiac cycle. In a nutshell, the ejection fraction is a reflection of the amount of blood that is pumped out of the heart with each contraction. Predicting cardiac disease using the volumes and ejection fraction. Even though there are a number of methods for measuring volumes or EF, the gold standard for calculating the exact force with which the heart contracts is magnetic resonance imaging (MRI)

In modern era scientists are creating new inventions and different image processing instrument MRI, X-Ray produces good quality image to identify Heart Diseases. However, complexity of detecting Heart Failure is to distinguish Heart Failure in different level and getting higher accuracy in leveling patient condition as it helps the doctors take proper necessities on time. But collecting information from image data manually is time consuming and human made errors cannot prevent fully.

Considering the fact that human cannot face the challenge of identifying HF alone different AI, Deep Learning and Machine learning approach can be helpful creating automated technique to detect the HF. Convolutional Neural Network(CNN) is helping in this regard. Different deep learning based approach are playing a crucial role in automated decision making and helps obtaining higher accuracy in CADs, HF detection.

1.2 Research Problem

Identifying CADs is complex and time-consuming when utilizing MRI to assess heart volumes and determine ejection fraction. MRI scans must be analyzed by a qualified cardiologist to determine EF. The procedure can take up to 20 minutes to finish, during which the cardiologist could continue seeing patients. Making this assessment technique more effective has significant ramifications for improving the science of treating heart disease and will improve clinicians' capacity to detect cardiac issues early. Moreover, finding accurate systole and diastole values from personal observation which is prerequisite of calculating EF is also difficult let alone finding HF exact levels.

Meanwhile, heart disease is the main cause of death among humans. Thus, researchers have started paying more attention to heart diseases[2]. Considerable cardiac diseases include myocardial hypertrophy, myocardial ischemia, heart failure, and so on have major indication value for the left ventricle (LV), the largest chamber of the heart. The LV has been the subject of a lot of research as of late, especially in the field of computer-assisted medicine. End-diastolic volumes (EDV), end-systolic volumes (ESV), and ejection fraction are all indices of left ventricular (LV) function that can be estimated with a cardiovascular magnetic resonance (CMR) scan (EF).

Magnetic resonance imaging (MRI) which is also known as NMRI or MRT. It is advance imaging technique which can produce high quality inside body images. In today's world, MRI gives much better results than any other diagnostic test. But doctors find it difficult and time-consuming to extract information from it, so they don't order MRIs usually and suggest other methods. This is particularly more difficult for cardiac MRI, as it is necessary to extract the MRI from different parts and observe which parts provide the best information. It is not possible to arrive at a final decision from a single specific view as to how much return has taken place, so it has to be observed from different positions. Observing through various AI, ML, NN techniques we can make a decision as to what stage of HF in. For this testing different angle data can be a good solution as from every part and every view, we won't know which position of the cardiac muscle it actually works best.

The main challenges of doing research on CVDs is the unavailability of data. Since, most of the doctors yet find it difficult to use MRI data they don't suggest for doing MRI in primary health condition. On the other hand, cardiac MRI is complex and researchers also find it difficult to understand it thoroughly. However, doing research on CVDs is increasing day by day and hopefully there will be more data available for the researchers which will motivate analyst to work for the biggest disease in the world.

After observing closely, to the issues raised above, our research seeks to provide a solution to the following:
How to Efficiently Predict Heart Failure from different positioned cardiac MRI Data Using Ejection Fraction with Deep neural network architecture?

1.3 Research Objective

The goal of our research is to find the suitable slice of a heart MRI so that the deep learning models can find a pattern to predict the systolic and diastolic volume of a heart. Therefore, we will preprocess the dataset into multiple steps which will help us find out the desired slice. Moreover, different architecture CNN models will be prepared to determine and get better result finding the volume of the heart, Ejection Fraction from our used Cardiac MRI data.

Chapter 2

Related Work

2.1 Literature Review

Using nave bayes and image processing[3], predicted the presence of heart disease. These days, heart disease is the leading cause of death. body mass index and cholesterol levels are the primary predictors of internal illness[3]. An expected system collects and stores patient data in a dataset, which is then compared to an inverted dataset containing patient data. This is a classic example of the overuse of square units when combining pathology data with graphical reporting like electrocardiogram and x-ray photography. First, there is the processing of pathology results, and second, there is the production of images for use in the processing of visual reports. Several reliable cardiovascular disease prediction algorithms employ these strategies. The training step involved the use of a naïve bayes classifier, and the recommended model performed adequately on 77 of the input examples. It had an average recall of 79.1% throughout testing and a recall of 85.0% during validation and regression testing.

In 2011, Michelle A. Borkin [4] created a model for the analysis of corridor representations for recognized confirmation of heart disease. They have developed a replacement second two-dimensional figure representation of blood vessel vein trees in addition to an undertaking scientific classification for the blood stream image. The results of their quantitative study show that the second outline is less sensitive to increased complexity within the project and that clients are much more accurate and practical at identifying areas of interest for an exact moment representation than a 3D delineation, where the rainbow coloring guide will significantly reduce a person's accuracy and strength.

In 2016, Chavan Krushna D, Kale Abhijeet A, Kulkarni Swapnil P, and Sayyed Ajmeer D set out to create a Hidden example examination for cardiovascular sickness order. The prediction of cardiovascular disease information handling system communicates with a collection of techniques for locating hidden instances that may be used to create a bring in aid organization. They concentrated on how information mining knowledge was organized and applied for information discovery. Information characterization and learning extraction utilizing different grouping strategies of information mining have advantages and disadvantages. Even though mining is a significant industry, using the photo technique improperly can reduce accuracy.

In expected systems, they frequently misuse computations like call trees, neural networks, and consequently Naive Thomas Bayes when managing information, and they frequently misuse well-known calculations like local binary patterns when processing images. The study's findings suggest that information preparation changes the health sector's ability to predict patterns inside datasets with an expected precision of 99%. Here, they frequently use image processing to examine the results of tests such as ECGs, CT scans, X-ray photos, and others in order to obtain more accurate results [5].

A methodology for locating and detecting blood vessel vein contracted segments abuse was developed in 2016. Nayana Mohan and Vishnukumar S explain the image-processing method. For the purpose of tracking down someone abusing DICOM images, an electronic infrastructure has been put in place. Fluctuating visual processes are interconnected to produce the result. The tasks are related to inconsistent DICOM images of awful courses. By applying morphological tasks adjacent to the vessel augmentation dissemination channel, the Conduits are divided from the main image. In order to determine how close a stricture is, vessel focus line extraction and vessel distance across estimation are used. A sign of stricture is thought to be an unexpected decrease in the vessel's width. If there is a five hundredth drop in vessel size, the method confirms the proximity of stricture. The framework indicates the surrounding area to clearly identify the district of stricture.

Different image processing techniques was proposed by [6] with artificial neural network for early recognition of coronary illness. According to medical science, the average person has a circulatory strain of 120/90 and a heart rate of 72 [7]. The grouping method used to predict a patient's risk of developing a cardiac illness based on factors such age, sex, circulatory strain, cholesterol level, and heart rate. There are many different characterization systems that are used to predict cardiac sickness, such as grouping procedures that were evaluated for their ability to do so. For example, you may use a fake neural network, a decision tree, a gullible Bayesian network, a super vector machine, or a hereditary calculator [8].

Since the Middle Ages, cardiovascular disease has been recognized as a major health problem, and it continues to be one of the world's leading killers today. Heart illness is very common, but coronary artery disease stands out because of how often it leads to death. Although angiograms are often regarded as the gold standard for identifying coronary artery disease, they come with a hefty price tag and some very serious risks. There has been extensive research using machine learning and data mining to identify potential other modes. Therefore, they[9] present a very accurate hybrid method of diagnosing coronary artery disease. This is because the evolutionary algorithm, which offers better weights for the neural network, is applied to the neural network's starting weights in the proposed method, hence increasing the network's performance by around 10%. On the Z-Alizadeh Sani dataset, they achieved 93.8 percent accuracy, 97.3 percent sensitivity, and 97.5 percent specificity using this approach. employing naive bayes and image processing[8] predicted the presence of cardiac disease. These days, heart disease is the leading cause of death. cholesterol, and excess body fat are the main indicators of internal illness[9]. The expected system collects and stores patient information in a dataset, and then compares this dataset to an inverted dataset containing information about the patients. This is a classic case of the overuse of square units when comparing pathology results with graphical information like electrocardiograms and X-ray photographs. It is expected that the system will make use of two methods: pathology findings management and image processing for graphical reports. Several reliable cardiovascular disease prediction algorithms employ these strategies. In the training phase, they utilized a naive bayes classifier; the recommended naive bayes model correctly classified 77 test cases. A recall of 85% was seen during validation, 85% during the regression phase, and 79% on average throughout testing. Deep learning algorithms like Convolutional Neural Network have been used in many Researches for the study of coronary artery blockage detection with higher accuracy than 90%. Tim et al. (2019) [10] proposed a 3d dilated CNN method to extract the Coronary artery centerline in cardiac CT angiography (CCTA) and found an average of 92% accuracy. In research cases of coronary artery blockage detection, deep learning has significant success to achieve higher accuracy. Here, CNN is largely used because it can take images as input then analyzing all The edges it is used for image classification and recognition because of their high accuracy.

Majd et al. (2018)[11] claim that they are the first to propose an improvement to a 3D convolutional neural network to detect coronary artery plaque and stenosis. They used experimental imaged data of 169 patients. MRI images of the arteries with a diameter of more than 1.5nm were taken to maintain the dataset's quality. First, it extracts the necessary objects from the image. After the preprocessing is done then the pre-processed data is used to analyze using an RCNN method automatically. It uses two multiclass simulations to detect and characterize the type of coronary artery plaque, which gives an accuracy of 77%. The other is to see and determine the anatomical significance of the coronary artery stenosis, which also achieved an accuracy of 80%. However, using a large dataset, applying an advanced scanner can improve the result.

The paper is about heart coronary artery segmentation and disease risk prediction using an advanced 3-d u-net CNN deep learning Algorithm. They used classified medical data set to identify the coronary plaque and coronary artery stenosis using a Machine learning method, which can scan the coronary angiography CT images and divide the coronary artery to detect any coronary plaque or stenosis. It eventually can help the physician to visualize the affected area. CAN et al. (2020)[12] stated that for data segmentation using a fully convolutional neural network, FCN has two central dominances; the Algorithm can take any sized image data. Then the results are better than other available methods. However, applying the deep belief networks and training regression networks provides excellent accuracy. As the research used two types of data, both centerline and without centerline, the experiment shows that the centerline pre-processing model training effect is above the first data. Therefore, the best impact reaches the Dice coefficient of 0.8491. The research also says that the data set has some shortcomings: it can't provide great images' details. Therefore, improving the data set can help the model perform more accurately and help the doctors improve the quality of diagnosis and treatment.

Nowadays, significant medical problems are worried to heart issues call cardiovascular disease. Plaque is the most common cause of respiratory failure. The formation of a smooth material inside the coronary arteries is known as cardiac plaque. Plaque causes occlusions in the coronary arteries[13]. Coronary angiography images were acquired using an X-ray projector and scanning camera setup. The acquired images are fed into the morphological administrator unit to prepare information picture coefficients and storage in the disk. The Edge locator handles the input image factors in the same way as the Watershed calculation block handles the Watershed calculation block for the primary processing of the preprocessed input picture. The accuracy depends on the image quality

2.2 Used Architecture

For this research we will use KERAS as it supports almost all the models of the neural networks. Moreover, keras is efficient to write and read while it focusses on simplicity.

2.2.1 ResNet

The fundamental component of a ResNet is known as a residual block. This method makes considerable use of batch normalization and is predicated on the idea of ”skip connections,” which enables the successful training of hundreds of layers without sacrificing training speed. The first thing that jumps out at us from the graphic that was just presented is that there is a straight relationship that goes through several of the model’s tiers. The so-called ”skip connection” is the driving force for residual blockages. As a result of skipping a connection, the output is not the same. In the absence of the skip link, the weights of the layer are first multiplied by the input ’X, and then a bias term is applied.

The idea behind this was that the deeper layers shouldn’t make any more mistakes in training than their counterparts who are in the shallower layers. For the purpose of putting this idea into practice, skip-connections were developed. The creators of this network used a pre-activation variation of the residual block that reduces the risk of ”vanishing gradients” by allowing gradients to pass through a shortcut connection to the layers that came before them.[14].

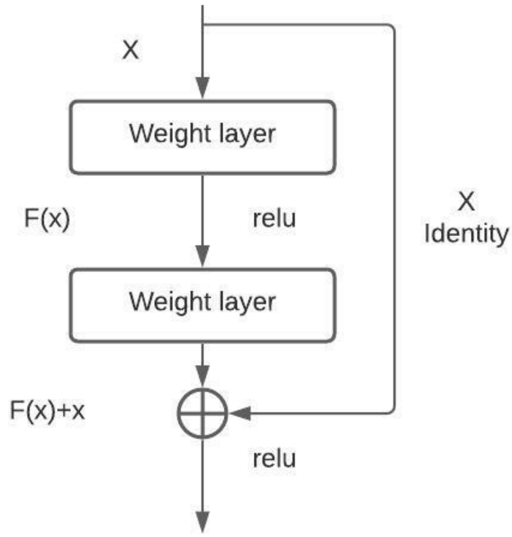

Figure 2.1: Shows How Input X Is Added To Output

ResNet50

There are several different iterations of ResNet, all of which are based on the same concept but have a different number of layers. The name ”ResNet-50” refers to a convolutional neural network that contains fifty hidden layers. A convolution and an identity block can be found in each of the five stages that make up the ResNet-50 model. The identity block, the convolution block, and each of its respective convolution layers each have three convolution layers. More than 23 million trainable parameters can be found in the ResNet-50[15].

ResNet101

With 101 hidden layers, ResNet-101 is a deep convolutional neural network. This network can be loaded in its pretrained state. As many as a thousand distinct types of objects, including as computers, mouse, pens, and animals, can be identified by the pretrained network. Due to this, the network has acquired the ability to learn complex feature representations across a wide variety of image types[16]. The maximum input size for the network is a 224x224 picture.

2.2.2 VGG-16

The VGG-16 network has 16 neural layers, making it a very deep and complicated neural network. It is possible to load a network from the ImageNet database that has been pretrained using more than a million images[17]. With 92.7% accuracy[18], the trained network can sort images into one thousand distinct categories, such as various animals, a mouse, a keyboard, and a pencil. Because of its ease of use and widespread popularity, transfer learning has become a powerful tool for classifying images.

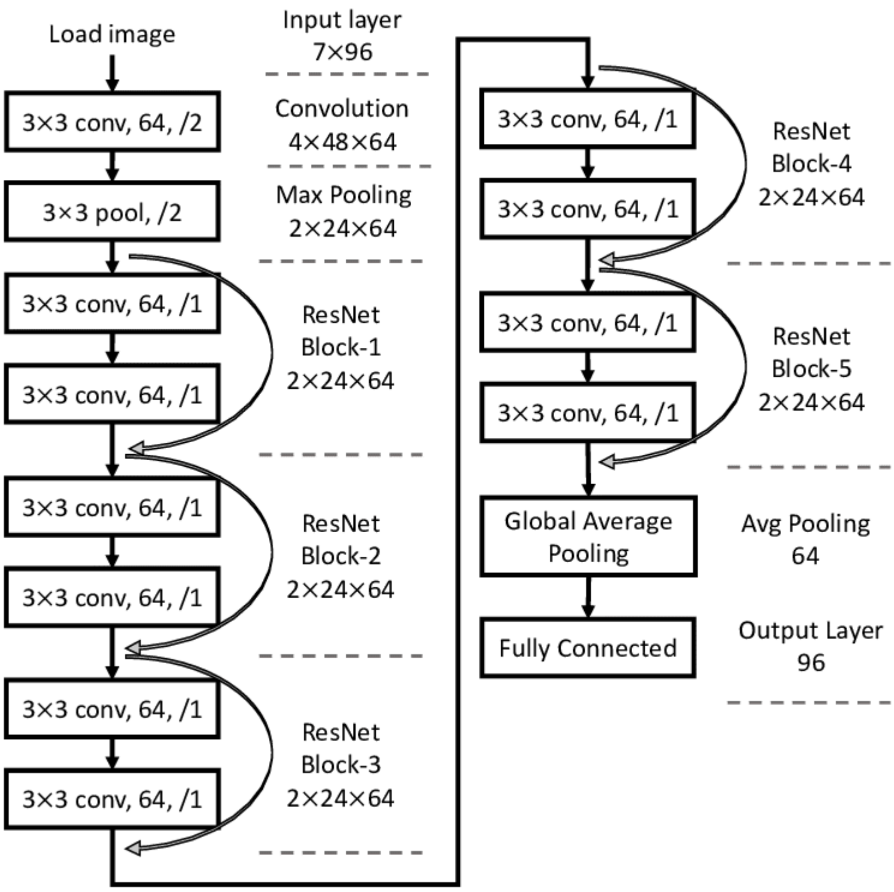

Figure 2.2: VGG-16 Architecture

2.2.3 VGG-19

VGG-19 is a powerful architecture, to properly implement this model, an RGB image of exactly 224 by 224 pixels is required[19]. A stack of convolutional neural networks (CNN) should be used to process the image, after which it would be adjusted using filters in a 3x3 pixel receptive field. Both parameters and nonlinearities are included in the 3*3. The input convolution stride and spatial padding must be set to 1 pixel for the 3*3 layer. To implement the Stack, we need to make use of three Fully Connected (FC) layers, the first two of which must have 4096 channels and the third 1000 channels. Finally, 5 MaxPool layers and 1 SoftMax layer are required. VGG-19's final layer is a SoftMax model[20].

2.2.4 Inception-V3

The Inception V3 convolutional neural layer uses multiple convolutions and a max pooling operation. Prior to the 3*3 and 5*5 convolutions, this model used to generate two 1*1 convolutions to save on processing time. When this is done, the computational cost is reduced, and the latest version does this even more efficiently. Factorization, smaller convolutions, Asymmetric convolutions, Auxiliary classifier, Grid size reduction, and so on are just few of the features. The input data is partitioned into three or four smaller 3D spaces via Inception-1x1 V3's convolutional operation, and then mapped using 9 conventional (3x3 or 5x5) convolutions. To create Inception-ResNet, Szegedy et al. fused the efficacy of residual learning with the inception block. The resulting concatenation of connections served as a filter, thanks to the surviving link[21].

Figure 2.3: Inception V3 Asymmetric Convolutions

2.3 Convolutional Layer

As the name implies, the ”convolutional layer” is crucial to the functioning of any CNN. This network works well with two-dimensional information. In order to convert images from 2D to 3D, it makes use of convolutional filters. Speedy learning is possible in a model. CNNs with a biological slant are typically multi-layer perceptrons. They have an innate ability to spot patterns in raw image data. These deep networks use multi-layer neurons and shared weights in each convolutional layer to analyze localized regions of the input image called receptive fields.

2.3.1 Activation Function

Different types of activation function is used to convert input data to desirable output data. It is possible for a network layer’s output values to lie in the negative infinity to positive infinity range.

Rectified Linear Unit

The rectified linear unit, or ReLU, is a non-linear or piecewise linear function widely used in deep learning models. If the input is negative, the function will return zero, and if it is positive, it will return the positive number. The same true result is returned by the function[22]. The formula for the function is:

$$f(x) = max(0, x) \tag{2.1}$$

Since using Sigmoid or Tanh in the hidden layers leads to the dreaded ”Vanishing Gradient” problem, ReLU is used instead. The ”Vanishing Gradient” prevents the network’s earlier layers from learning new information while backpropagating. Because sigmoid functions produce values between 0 and 1, they are best used at the output layer when solving problems involving regression or binary classification.

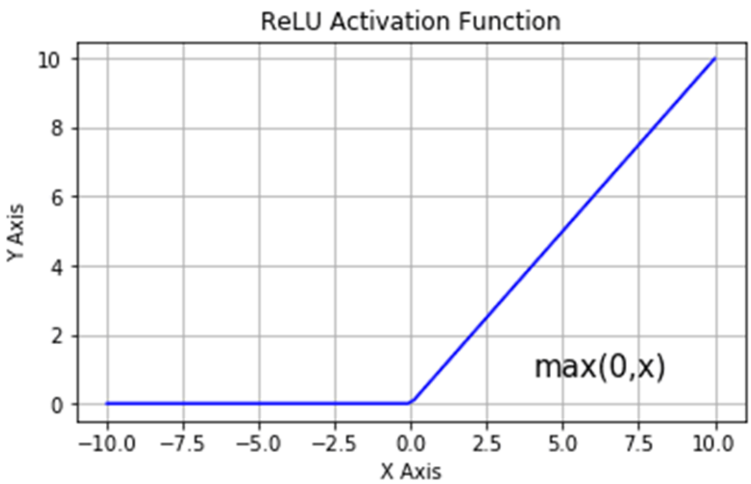

Figure 2.4: Plotted Graph of ReLU Function

SoftMax

The interpretation of the neural network's output is simplified by the softmax activation function. The softmax activation function takes the raw outputs of the neural network and transforms them into a vector of probabilities, which is essentially a probability distribution over the input classes. An N-class multiclass classification task with softmax activation yields an N-element output vector, with the entry at index I representing the probability that the corresponding input belongs to class i[23].
The softmax function is,

$$S(y)_i = \frac{exp(y_i)}{\sum_{j=1}^{n} exp(y_j)} \tag{2.2}$$

Chapter 3

Work Plan

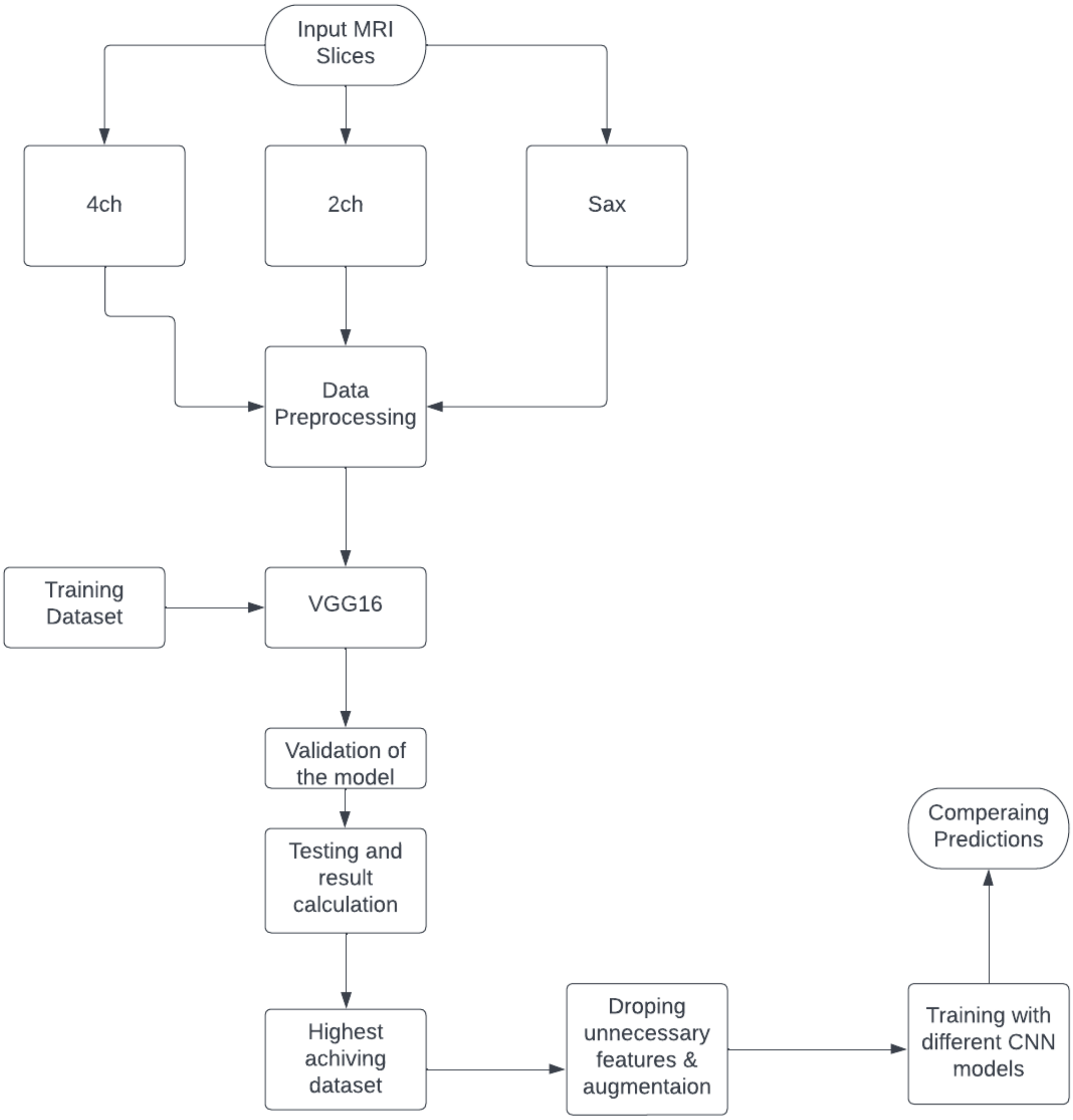

Figure 3.1: Flow chart of our proposed work plan

The core objective of our research is find out the slices which can predict the volume of the heart more accurately.Initially have prepared 3 separate dataset according to the views.Then we will train those dataset using VGG16 model with 20 epochs.After that we will compare the predicted accuracy.Therefore the highest achieved accuracy will be selected for the next implementation.Furthermore, we will drop out some extra features of the slices which does not required to be learn for our objective.Then different types of augmentation will be applied to get better result.The updated dataset then will be used in different CNN architectures like VGG19,VGG16,ResNet50 etc.Finally we will compare the models to evaluate the dataset.

Chapter 4

Implementation

4.1 Source

Data Science Bowl Cardiac Challenge Data [24]

4.2 Data Sample

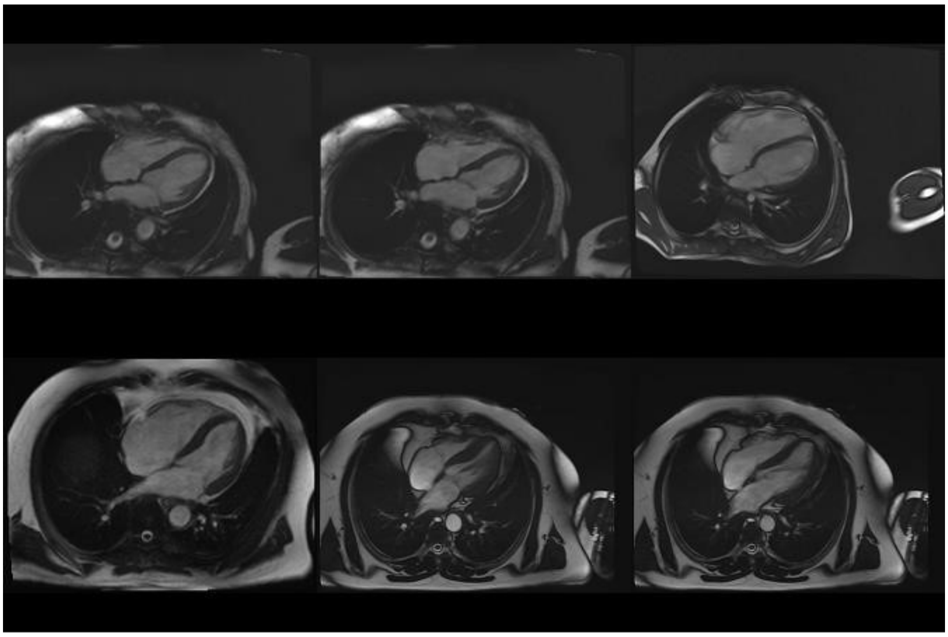

Figure 4.1: Sample Data of the Dataset

In the figure , we can see some data samples of the MRI of the heart

4.3 Data Classification

We have splitted our dataset into 7:3:1 for our research.700 patient of our data was used in the training dataset, 300 patient of the data as testing set and 100 patient of the data as validation set respectively.

4.3.1 Training Set

It is a set of data which will be used to train the model in every epoch.Eventually the model will learn new features and predict in future to completely unknown data.

4.3.2 Validation Set

Validation dataset is used to avoid overfitting.During each epoch it will classify the input.These data is separate from the training set so the samples are not already trained in the model.

4.3.3 Testing Set

The test set is also separate from the training and validation set. We will test the model after it is trained and validated.The tasting set will not be labeled like the other two sets.

4.4 Data Pre-processing

4.4.1 Data Labeling

Our selected data set has 1100 patients heart's MRI data. The dataset is in DICOM format.This format is being used to collect medical image data.It also stores the patients personal details like age,sex,patient name, patient id etc.The dataset also comes with a CSV file which contains the patient ID and it's correspondence Stytolic and Dyastolic volume value.

ID	Systole	Diastole
1	**108.3**	**246.7**
2	**54**	**137**
3	**32.7**	**99.3**
4	**57.7**	**154.5**
5	**83.3**	**235.5**
6	**225.3**	**317.9**
7	**64.9**	**138**
8	**158.3**	**305.5**
9	**61.4**	**152.2**
10	**105.2**	**219.3**
11	**45.4**	**121.7**

Table 4.1: Sample CSV file of the Dataset)

We will calculate the ejection fraction using the formula and classify the heart condition in four sections.

$$EF = (1 - \frac{Vs}{Vd}) \times 100 \tag{4.1}$$

Here, Vs means Systolic volume and Vd means Diastolic volume. According to the Left Venticular Ejection Function, the dataset has been divided into four categories:

I absence (EF of 53%-73%), (ii) mild (EF of 41%-52%), (iii) moderate (EF of 30%-40%), or (iv) severe (EF 30%). [2][25].

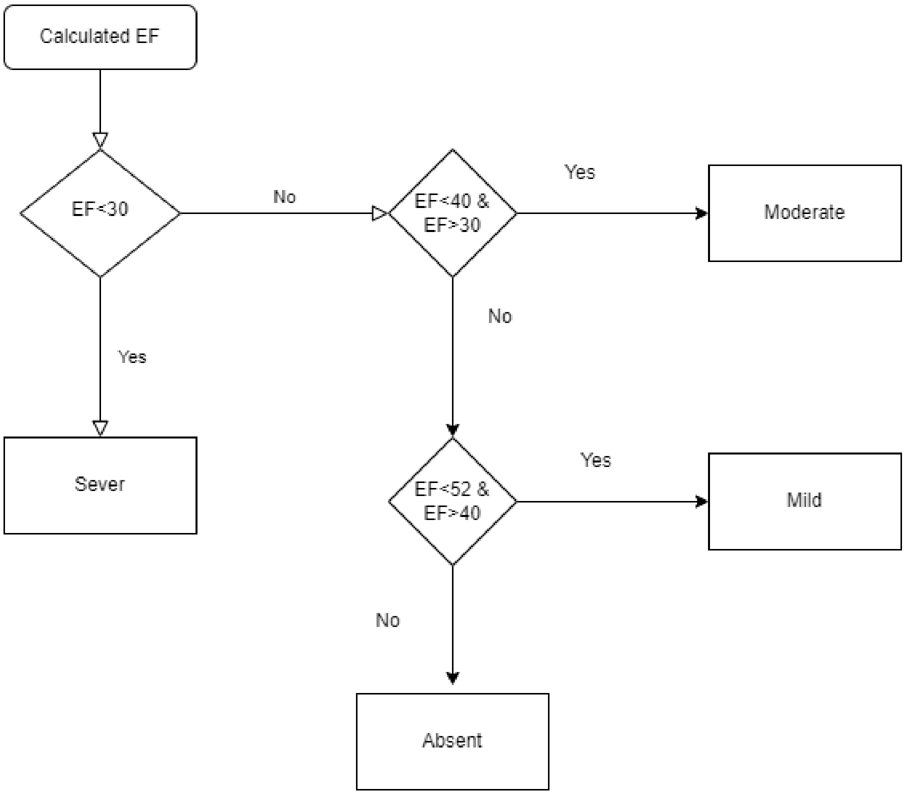

Figure 4.2: Data leveling Flowchart

4.4.2 Data Categorization

To reach our objective we prepared 3 different dataset according to the graphical view.
The short axis stack is the primary view for evaluating ventricle size, as it consists of pictures taken in a plane that is perpendicular to the long axis of the left ventricle. The "sax" prefix denotes these. Therefore, there are two more views given in the dataset such as the 4 chamber view and the 2 chamber view which denote in prefix "4ch" and "2ch" respectively.The most frequent aspect for identifying structural cardiac abnormalities during standard prenatal ultrasonography is the four-chamber MRI scan of the fetal heart[26]. Furthermore, these datasets will be used in various CNN models to check which dataset is suitable for our purpose.

4.4.3 Image Resize and Conversion

The primary goal of our approach is to provide the most accurate slice possible for MRI scan. Because of this, as we have seen, we separated out the categorization and training of various models. Both patients with and without a HF are represented in this dataset, which consists of MRI scan slices. When training the network, all the DICOM formatted files are converted into JPG format images. After that all images are scaled down to a uniform 224 by 224 pixels. As a result, we'll be making use of the Scikit Image [27], TensorFlow[28], and Caffe[29] frameworks. ImageDataGenerator[30][31] is a class in Keras that we use to normalize the pixel values in our image dataset before we begin modeling. When our image dataset is being created, verified, or evaluated, this class collects all of the images into a single location, restores them to the algorithm in batches, and does any necessary scaling operations. It provides a robust and rational method for scaling visual input when utilizing neural networks for modeling. The Image Data Generator supports numerous feature selection techniques, in addition to percentage of pixel scaling methods. The Image Data Generator class permits a reference to leveling because it predominantly employs the mean calculated upon that training dataset as feature-wise centering. Regression on the training sample is not possible until the statistics are calibrated.

4.4.4 Normalization and Scaling Images

Principal component analysis (pca)[32] is a multivariate method for examining a data table where observations are described by many quantitative dependent variables that are interconnected. The principal component analysis is used to determine the eigen flat fields of a collection of 16 flat fields (PCA). The projections of MRI scan images are then normalized by linearly combining the most important eigen flat fields. The proposed dynamic flat field correction has been shown experimentally to significantly minimize systematic errors in projection intensity normalization compared to the baseline flat field correction. For this, we'll use Keras's Image Data Generator class. The term "normalization" refers to the procedure of reducing data to a scale of 0-1 in order to better analyze it. To accomplish this, the input from the re-scale operation must be transformed into a multiplicative ratio that can be applied to each individual pixel. In order to achieve this, the re-scale parameter can be transformed into a ratio that can be applied to each pixel to obtain the appropriate range.The 2ch view, which is a long axis image and is perpendicular to the four chamber view, only shows the right ventricle and atrium.

4.4.5 Data Augmentation

Improving the Quality of Dataset We enhance our data by making a number of random adjustments to the images in an effort to boost our model's accuracy. All of the pictures are flipped horizontally and vertically as well as rotated by 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330 and 360 degrees. This will be accomplished with the help of Keras's ImageDataGenerator class. Keras, a platform for deep learning neural networks, has a class called mageDataGenerator that can augment picture data to better fit models. Enhancing the size of the training dataset and the model's ability to generalize requires the addition of image data. The dataset's picture selection is not done analytically. The model has no access to standard quality images. Training photos can be either modified or near-copies, depending on the parameters of the random image enhancement. Datasets for both validation and testing are often defined by a Data Generator. Data augmentation normally makes use of a different ImageDataGenerator instance than the one used for training data, however the two may share the same pixel scale option. To improve model performance on an unsegmented dataset, data augmentation is solely employed to artificially extend the training data set. Shifting an image is to change its orientation while maintaining the same width and height. Shift amounts in both directions can be sequentially adjusted using the width and height shift range parameters of the Image Data Generator constructor.

4.5 Architecture Training

The very first step in training a Neural Network is to determine what values should be assigned to the Neural Connections' weights. Our dataset was split into three parts for training neural networks. It all starts with the word "train" in the opening track. 700 patient's of available data is being used to optimize the model. It was used to test out the VGG16, VGG19, ResNet50, ResNet101, and InceptionV3 models. We tested these neural network architectures with a 32-batch size and a 35 epoch. The remaining parts of the dataset are split evenly between "test" (300 patients) and "valid" (100 patients). We processed the data using a system with an AMD Ryzen 9 5950x Processor, 64 GB of RAM, and an NVIDIA RTX3080Ti 12 GB graphics processing unit.

Chapter 5

Result and analysis

5.1 For Individual Dataset

Dataset	Validation Accuracy of VGG-16	val_loss
sax	**45.78%**	**1.6573**
2ch	**51.53%**	**0.9115**
4ch	**78.44%**	**0.6523**

Table 5.1: Val categorical accuracy of 3 different Dataset

We have implemented the VGG16 model on the mentioned 3 dataset with 20 epochs. From the val catagorical accuracy in table 4.2 we can clearly see that the sax and 2ch prefix data set cannot predict the pattern according to our leveling in preprocessing.Therefore, those dataset has other features which we do not want our model to learn.However the 4ch dataset can learn the features to predict the volume of the heart.As a result we will do further implementation our the 4ch dataset.

5.2 Training 4ch Dataset for Individual Models

After That, we have manually inspected the MRI slices and dropped the slices which contain features that are no use of predicting the volume.Then the remaining dataset were augmented by 30 degree rotation and flipped vertically to increase the accuracy.We have used used VGG19,VGG16,ResNet50,ResNet101 and InceptionV3 to check which model is more suitable for our dataset.We only took the last layer of the models to train our dataset so that we can get all the convolution.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 model (Functional)          (None, 1, 1, 512)         14714688

 flatten (Flatten)           (None, 512)               0

 dropout (Dropout)           (None, 512)               0

 dense (Dense)               (None, 512)               262656

 dense_1 (Dense)             (None, 4)                 2052

=================================================================
Total params: 14,979,396
Trainable params: 14,979,396
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 5.1: Model summaryl

5.2.1 InceptionV3

We have splitted the dataset into 7:2:1 ratio and run 30 epochs.The summary of the training model is given below.

```
Epoch 2/30
40/40 [==============================] - 11s 283ms/step - loss: 1.3445 - categorical_accuracy: 0.4023 - val_loss: 1.3355 - val_categorical_accuracy: 0.3675
Epoch 3/30
40/40 [==============================] - 11s 279ms/step - loss: 1.1643 - categorical_accuracy: 0.4984 - val_loss: 1.1010 - val_categorical_accuracy: 0.5277
Epoch 4/30
40/40 [==============================] - 11s 281ms/step - loss: 1.0745 - categorical_accuracy: 0.5586 - val_loss: 1.0199 - val_categorical_accuracy: 0.5938
Epoch 5/30
40/40 [==============================] - 11s 277ms/step - loss: 0.9189 - categorical_accuracy: 0.6273 - val_loss: 0.9379 - val_categorical_accuracy: 0.6507
Epoch 6/30
40/40 [==============================] - 11s 281ms/step - loss: 0.8636 - categorical_accuracy: 0.6617 - val_loss: 0.9174 - val_categorical_accuracy: 0.6705
Epoch 7/30
40/40 [==============================] - 11s 276ms/step - loss: 0.7614 - categorical_accuracy: 0.6953 - val_loss: 0.8865 - val_categorical_accuracy: 0.6887
Epoch 8/30
40/40 [==============================] - 11s 281ms/step - loss: 0.7201 - categorical_accuracy: 0.7078 - val_loss: 0.8742 - val_categorical_accuracy: 0.6902
Epoch 9/30
40/40 [==============================] - 11s 277ms/step - loss: 0.6979 - categorical_accuracy: 0.7344 - val_loss: 0.8369 - val_categorical_accuracy: 0.7342
Epoch 10/30
40/40 [==============================] - 11s 282ms/step - loss: 0.6378 - categorical_accuracy: 0.7648 - val_loss: 0.8532 - val_categorical_accuracy: 0.7077
Epoch 11/30
40/40 [==============================] - 11s 282ms/step - loss: 0.6022 - categorical_accuracy: 0.7602 - val_loss: 0.8199 - val_categorical_accuracy: 0.7570
Epoch 12/30
40/40 [==============================] - 11s 285ms/step - loss: 0.5728 - categorical_accuracy: 0.7930 - val_loss: 0.8296 - val_categorical_accuracy: 0.7509
Epoch 13/30
40/40 [==============================] - 11s 279ms/step - loss: 0.5537 - categorical_accuracy: 0.7930 - val_loss: 0.8142 - val_categorical_accuracy: 0.7654
Epoch 14/30
40/40 [==============================] - 11s 285ms/step - loss: 0.4874 - categorical_accuracy: 0.8258 - val_loss: 0.8216 - val_categorical_accuracy: 0.7677
Epoch 15/30
40/40 [==============================] - 11s 277ms/step - loss: 0.4850 - categorical_accuracy: 0.8305 - val_loss: 0.8245 - val_categorical_accuracy: 0.7722
Epoch 16/30
40/40 [==============================] - 11s 280ms/step - loss: 0.4693 - categorical_accuracy: 0.8242 - val_loss: 0.8169 - val_categorical_accuracy: 0.7752
Epoch 17/30
40/40 [==============================] - 11s 280ms/step - loss: 0.4240 - categorical_accuracy: 0.8594 - val_loss: 0.7952 - val_categorical_accuracy: 0.8079
Epoch 18/30
40/40 [==============================] - 11s 282ms/step - loss: 0.4155 - categorical_accuracy: 0.8625 - val_loss: 0.8118 - val_categorical_accuracy: 0.7957
Epoch 19/30
40/40 [==============================] - 11s 278ms/step - loss: 0.4030 - categorical_accuracy: 0.8594 - val_loss: 0.8055 - val_categorical_accuracy: 0.8003
Epoch 20/30
40/40 [==============================] - 11s 280ms/step - loss: 0.3725 - categorical_accuracy: 0.8773 - val_loss: 0.8161 - val_categorical_accuracy: 0.7980
Epoch 21/30
40/40 [==============================] - 11s 281ms/step - loss: 0.3695 - categorical_accuracy: 0.8813 - val_loss: 0.7931 - val_categorical_accuracy: 0.8185
Epoch 22/30
40/40 [==============================] - 11s 280ms/step - loss: 0.3541 - categorical_accuracy: 0.8805 - val_loss: 0.8066 - val_categorical_accuracy: 0.8178
Epoch 23/30
40/40 [==============================] - 11s 280ms/step - loss: 0.3398 - categorical_accuracy: 0.8828 - val_loss: 0.8061 - val_categorical_accuracy: 0.8299
Epoch 24/30
40/40 [==============================] - 11s 283ms/step - loss: 0.3110 - categorical_accuracy: 0.8930 - val_loss: 0.8152 - val_categorical_accuracy: 0.8193
Epoch 25/30
40/40 [==============================] - 11s 283ms/step - loss: 0.3107 - categorical_accuracy: 0.9078 - val_loss: 0.8227 - val_categorical_accuracy: 0.8193
Epoch 26/30
40/40 [==============================] - 11s 281ms/step - loss: 0.2985 - categorical_accuracy: 0.9055 - val_loss: 0.8305 - val_categorical_accuracy: 0.8193
Epoch 27/30
40/40 [==============================] - 11s 277ms/step - loss: 0.2982 - categorical_accuracy: 0.9031 - val_loss: 0.8277 - val_categorical_accuracy: 0.8276
Epoch 28/30
40/40 [==============================] - 11s 278ms/step - loss: 0.2660 - categorical_accuracy: 0.9039 - val_loss: 0.8243 - val_categorical_accuracy: 0.8413
Epoch 29/30
40/40 [==============================] - 11s 279ms/step - loss: 0.2546 - categorical_accuracy: 0.9203 - val_loss: 0.8248 - val_categorical_accuracy: 0.8504
Epoch 30/30
40/40 [==============================] - 11s 281ms/step - loss: 0.2351 - categorical_accuracy: 0.9242 - val_loss: 0.8393 - val_categorical_accuracy: 0.8527
```

Figure 5.2: Summary of training model

The accuracy we received from this model is far better than our first phase. The val catagorical accuracy and the val loss is given below.

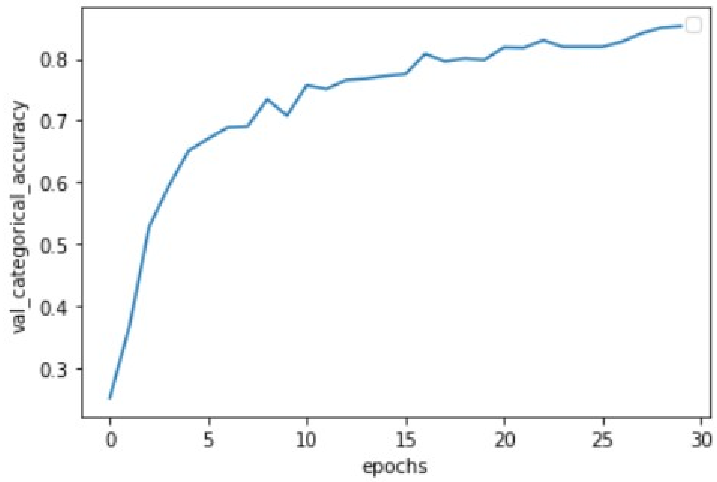

Figure 5.3: Validation accuracy curve of InceptionV3

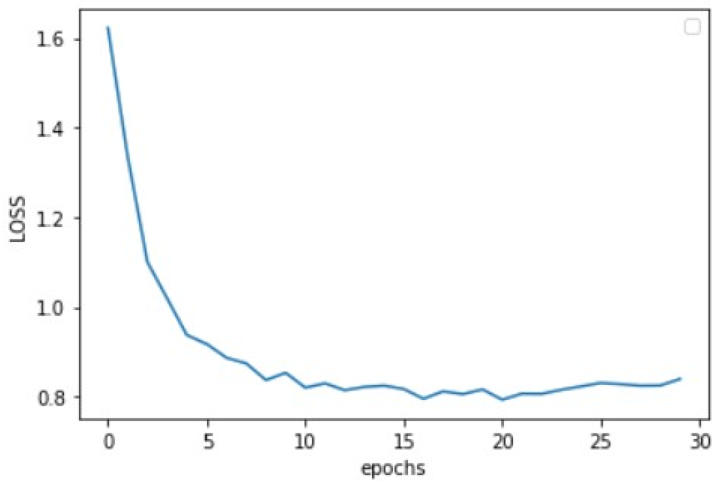

Figure 5.4: Validation loss curve of InceptionV3

From the learning curve and summary we can see that we have achieved 85.27% accuracy.

5.2.2 VGG-16

For VGG-16 implementation we have also kept the same train, test and validation ratio.After 20 epochs we got 88.15% accuracy which is even better than the previous model.

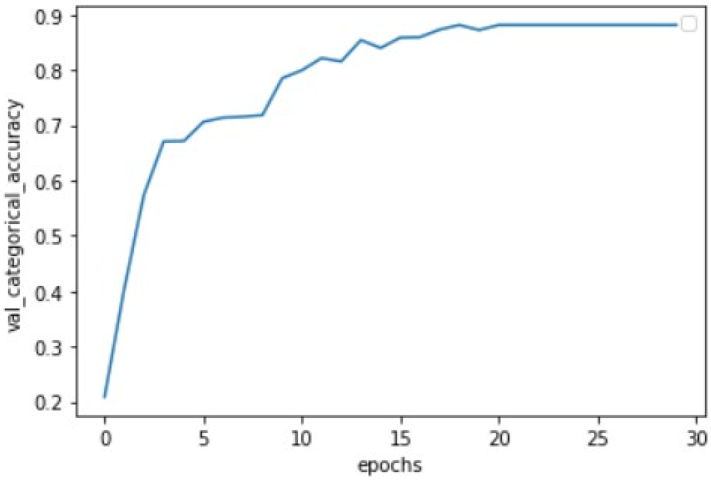

Figure 5.5: Validation accuracy curve of VGG16

The classification report is also generated from the confusion matrix to deeply understand the result.

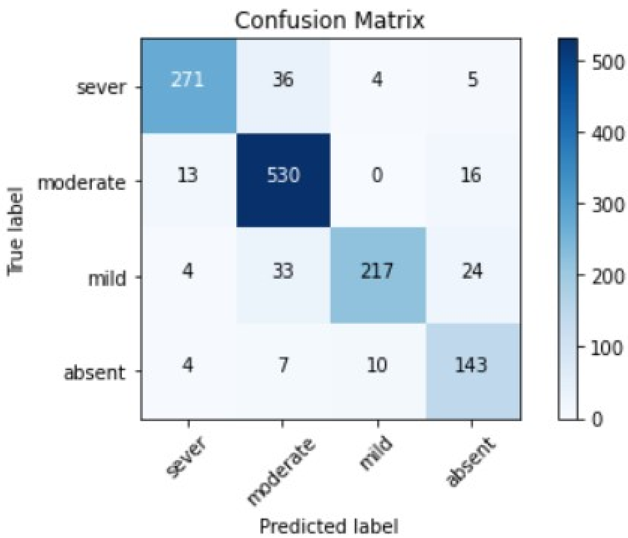

Figure 5.6: Confusion matrix of VGG16 model

	precision	recall	f1-score	support
0	0.93	0.86	0.89	316
1	0.87	0.95	0.91	559
2	0.94	0.78	0.85	278
3	0.76	0.87	0.81	164
accuracy			0.88	1317
macro avg	0.88	0.86	0.87	1317
weighted avg	0.89	0.88	0.88	1317

Figure 5.7: Categorical report of VGG16

5.2.3 VGG-19

After the considerably good performance of VGG16, the performance of VGG19 upset us a little. As the VGG19 has 3 extra layers we were hoping for a better performance of this model than the previous.After 30 epochs it gave 84.89 percent val catagorical accuracy.

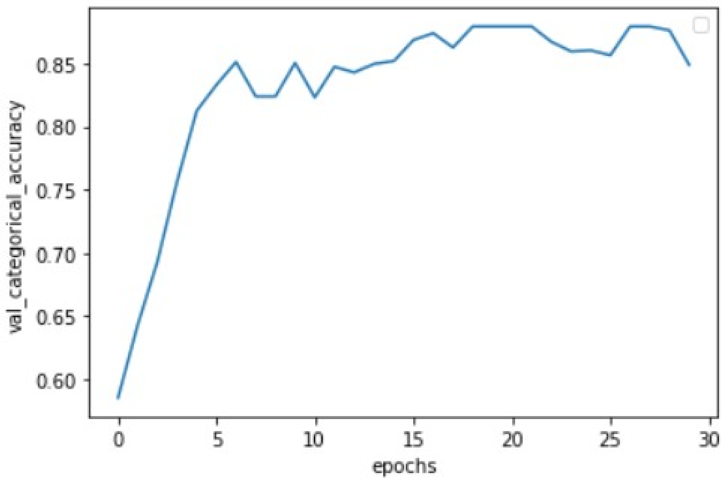

Figure 5.8: Validation accuracy of VGG19

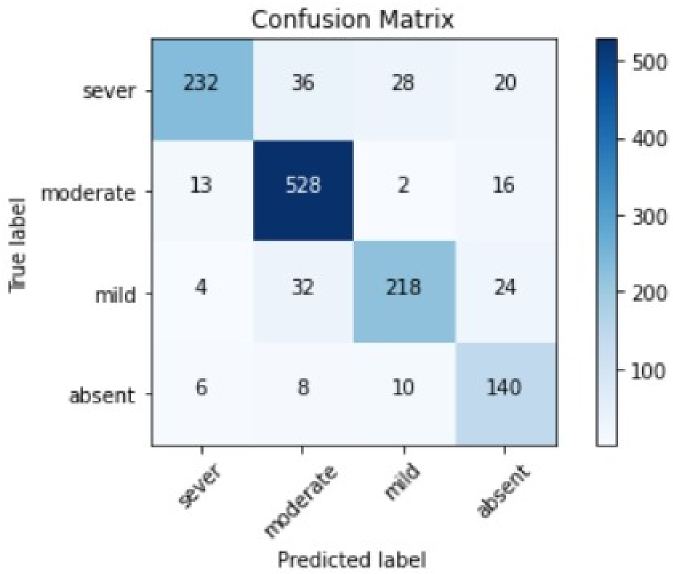

Figure 5.9: Confusion matrix of VGG19

	precision	recall	f1-score	support
0	0.91	0.73	0.81	316
1	0.87	0.94	0.91	559
2	0.84	0.78	0.81	278
3	0.70	0.85	0.77	164
accuracy			0.85	1317
macro avg	0.83	0.83	0.83	1317
weighted avg	0.85	0.85	0.85	1317

Figure 5.10: Categorical report of VGG19

Furthermore, we have also implemented ResNet50 and ResNet101 for our dataset.However, the result was not as good as the previous 3.

5.2.4 Comparative Analysis of the different models

From this finding we are able to know how different CNN models perform in our selected slices. The performance of VGG16 and InceptionV3 was almost identical.However the VGG16 architecture is able to predict the pattern quite well.

Architecture	Val Categorical Accuracy(%)	Val Loss
VGG-19	**84.89**	**0.7030**
VGG-16	**88.15**	**0.8810**
ResNet50	**75.85**	**1.2014**
ResNet101	**79.53**	**1.2337**
Inception V3	**85.27**	**0.8393**

Table 5.2: Comparison between Used Models

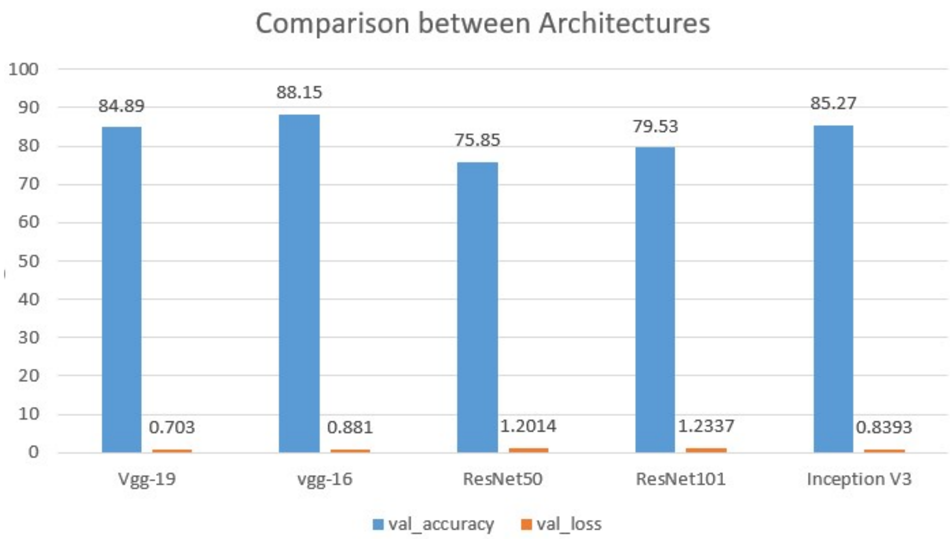

Figure 5.11: Comparison between different architecture

Chapter 6

Conclusion & Future Work

6.1 Conclusion

In our research we created an automated deep learning based model which can predict Heart failure and its death. One having Heart Failure(HF) does not mean that his heart is completely stopped functioning. Moreover, after different steps our heart can completely stop functioning. In our research we able to detect HF in four level using Ejection Fraction(EF). Cardiac MRI Image with different pattern slice data is being used to identify which slice gives the best result detecting HF using EF. Different CNN model was considered to identify the best possible result. At first we run CNN model to the 4ch, sax, 2ch slice of MRI data and after validation and result analysis we able to identify that 4ch slice data gives the best result with EF. Furthermore, after more data processing and running more CNN models we get even better result detecting Heart Failure(HF). VGG-16(88.15%), VGG-19(87.93%), ResNet-50 (75.85%), ResNet-101 (79.53%) Inception-V3 (85.27%). This research can be conducted for future work with different deep learning methodology. Further process of data and combining different methodology will help to predict new features as well as the accuracy with our research we tried to fulfill our motivation to save human life.

6.2 Future Work

Heart MRI dataset is very rare to find as doctors suggests cardiac patients to test angiogram. Our selected cardiac dataset has 1100 patients and very patients has upto 330 2D slices. We can merge all the 2D slices and simulate a 3D view of the heart which will help deep learning models to learn features more accurately. CNN architecture maintains two things, one is weight tying and another is local receptive. The local receptive field is used for tiny and localized area with particular images so that each unit of the network can give a more efficient result than work in the whole area with less efficiency. Weight tying also works for the first layer unit for sharing the exact weights, which minimizes the pressure to learn complex parameters and hard codes. "Tiled Convolutional Neural Network'' is an extension of CNN, which works on unsupervised pertaining and weight tiling. It works on k separate convolution kernels in the same layer. In tiled CNN, the tiled layers can provide scale invariance, rotational invariance and the translational invariance through the pooling operation. In this tiled CNN, each convolution operation uses training data that is previously learning for further procedure.

Bibliography

[1] K. G. Dinesh, K. Arumugaraj, K. D. Santhosh, and V. Mareeswari, "Prediction of cardiovascular disease using machine learning algorithms," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, IEEE, 2018, pp. 1–7.

[2] Y. Li, T. P. Fitzgibbons, D. D. McManus, R. P. Goddeau, B. Silver, and N. Henninger, "Left ventricular ejection fraction and clinically defined heart failure to predict 90-day functional outcome after ischemic stroke," *Journal of Stroke and Cerebrovascular Diseases*, vol. 28, no. 2, pp. 371–380, 2019, ISSN: 1052-3057. DOI: https://doi.org/10.1016/j.jstrokecerebrovasdis.2018.10.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1052305718305810.

[3] M. Rani, A. Bakshi, and A. Gupta, "Prediction of heart disease using naïve bayes and image processing," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2020, pp. 215–219. DOI: 10.1109/ESCI48226.2020.9167537.

[4] T. M. Chieng, Z. Omar, and S. Kadiman, "Efficient morphological-based edge detection algorithm for computed tomography cardiac images," in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2015, pp. 423–428. DOI: 10.1109/ICSIPA.2015.7412228.

[5] N. Mohan and S. Vishnukumar, "Detection and localization of coronary artery stenotic segments using image processing," in *2016 International Conference on Emerging Technological Trends (ICETT)*, 2016, pp. 1–5. DOI: 10.1109/ICETT.2016.7873727.

[6] M. Rani, A. Bakshi, K. Munjal, and A. S. Tomar, "Heart disease prediction using artificial neural network and image processing," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2020, pp. 203–208. DOI: 10.1109/ESCI48226.2020.9167669.

[7] J. T. T.P.R, "Human heart disease prognostication system utilizingstatistics mining techniques," 2016.

[8] C. Sowmiya and S. Pitchai, "Analytical study of heart disease diagnosis using classification techniques," Mar. 2017, pp. 1–5. DOI: 10.1109/ITCOSP.2017.8303115.

[9] Z. Arabasadi, R. Alizadehsani, M. Roshanzamir, H. Moosaei, and A. A. Yarifard, "Computer aided decision making for heart disease detection using hybrid neural network-genetic algorithm," *Computer Methods and Programs in Biomedicine*, vol. 141, pp. 19–26, 2017, ISSN: 0169-2607. DOI: https://doi.org/

10.1016/j.cmpb.2017.01.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260716309695.

[10] J. M. Wolterink, R. W. van Hamersvelt, M. A. Viergever, T. Leiner, and I. Išgum, "Coronary artery centerline extraction in cardiac ct angiography using a cnn-based orientation classifier," *Medical Image Analysis*, vol. 51, pp. 46–60, 2019, ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2018.10.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841518308491.

[11] M. Zreik, R. W. van Hamersvelt, J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "A recurrent cnn for automatic detection and classification of coronary artery plaque and stenosis in coronary ct angiography," *IEEE Transactions on Medical Imaging*, vol. 38, no. 7, pp. 1588–1598, 2019. DOI: 10.1109/TMI.2018.2883807.

[12] C. Xiao, Y. Li, and Y. Jiang, "Heart coronary artery segmentation and disease risk warning based on a deep learning algorithm," *IEEE Access*, vol. 8, pp. 140 108–140 121, 2020. DOI: 10.1109/ACCESS.2020.3010800.

[13] S. B. Mudigoudar and A. I. Rasheed, "Design and implementation of image processing algorithms for cardiac blockage detection on fpga," in *2016 IEEE Annual India Conference (INDICON)*, 2016, pp. 1–5. DOI: 10.1109/INDICON.2016.7839159.

[14] C. Jianfei and Z. Changming, "Research on image recognition based on improved resnet," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 1422–1426. DOI: 10.1109/ICCC51575.2020.9345181.

[15] G. Boesch, *Deep residual networks (resnet, resnet50) - 2022 guide*, en-US, Jun. 2022. [Online]. Available: https://viso.ai/deep-learning/resnet-residual-neural-network/.

[16] P. Ghosal, L. Nandanwar, S. Kanchan, A. Bhadra, J. Chakraborty, and D. Nandi, "Brain tumor classification using resnet-101 based squeeze and excitation deep neural network," in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 2019, pp. 1–6. DOI: 10.1109/ICACCP.2019.8882973.

[17] G. L. Team, *Introduction to vgg16 | what is vgg16?* en-US, Oct. 2021. [Online]. Available: https://www.mygreatlearning.com/blog/introduction-to-vgg16/.

[18] O. Genus, *VGG16 architecture — iq.opengenus.org*, https://iq.opengenus.org/vgg16/, [Accessed 20-Sep-2022], 2021.

[19] T. Carvalho, E. R. De Rezende, M. T. Alves, F. K. Balieiro, and R. B. Sovat, "Exposing computer generated images by eye's region classification via transfer learning of vgg19 cnn," in *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, IEEE, 2017, pp. 866–870.

[20] D. I. Sec, *Vgg-19 convolutional neural network*, en-US, Mar. 2021. [Online]. Available: https://blog.techcraft.org/vgg-19-convolutional-neural-network/.

[21] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, 2017, pp. 783–787. DOI: 10.1109/ICIVC.2017.7984661.

[22] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, IEEE, 2020, pp. 281–286.

[23] S. Gold, A. Rangarajan, *et al.*, "Softmax to softassign: Neural network algorithms for combinatorial optimization," *Journal of Artificial Neural Networks*, vol. 2, no. 4, pp. 381–399, 1996.

[24] *Second annual data science bowl.* [Online]. Available: https://www.kaggle.com/competitions/second-annual-data-science-bowl/data.

[25] W.-J. Kim, H.-W. Nah, D.-H. Kim, and J.-K. Cha, "Association between left ventricular dysfunction and functional outcomes at three months in acute ischemic stroke," *Journal of Stroke and Cerebrovascular Diseases*, vol. 25, no. 9, pp. 2247–2252, 2016, ISSN: 1052-3057. DOI: https://doi.org/10.1016/j.jstrokecerebrovasdis.2016.05.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1052305716300593.

[26] S. M. Shultz, D. H. Pretorius, and N. E. Budorick, "Four-chamber view of the fetal heart: Demonstration related to menstrual age.," en, *Journal of Ultrasound in Medicine*, vol. 13, no. 4, pp. 285–289, Apr. 1994, ISSN: 02784297. DOI: 10.7863/jum.1994.13.4.285. [Online]. Available: http://doi.wiley.com/10.7863/jum.1994.13.4.285.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "{tensorflow}: A system for {large-scale} machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[30] A. Gulli and S. Pal, *Deep learning with Keras.* Packt Publishing Ltd, 2017.

[31] N. K. Manaswi, N. K. Manaswi, and S. John, *Deep learning with applications using python.* Springer, 2018.

[32] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37–52, 1987, Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, ISSN: 0169-7439. DOI: https://doi.org/10.1016/0169-7439(87)80084-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0169743987800849.