DON BOSCO COLLEGE OF ENGINEERING

FATORDA, MARGAO, GOA – 403 602

DEPARTMENT OF COMPUTER ENGINEERING

2018 – 2019

"Automated Pneumonia Detection"

By

Mr. Mrunal Thakur

Miss. Shreya Pokle

Mr. Ronan Revadker

Miss. Swizel Rodrigues

Miss. Drushti Shah

Under the Guidance of

Mr. Amey Kerkar

Assistant Professor

BACHELOR OF ENGINEERING: GOA UNIVERSITY

DON BOSCO COLLEGE OF ENGINEERING

FATORDA, MARGAO, GOA- 403 602

2018 - 2019

CERTIFICATE

This is to certify that this dissertation entitled

"Automated Pneumonia Detection"

submitted in partial fulfillment of the requirements for Bachelor's Degree in Computer Engineering of Goa University is the bonafide work of

Mr. Mrunal Thakur

Miss. Shreya Pokle

Mr. Ronan Revadker

Miss. Swizel Rodrigues

Miss. Drushti Shah

Mr. Amey Kerkar	**Mr. Gaurang Patkar**
Assistant Professor	**Head of Department**

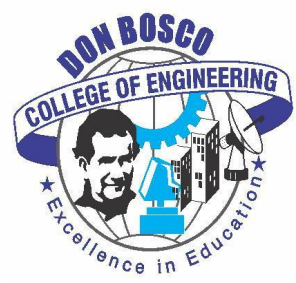
Dr. Neena S. P. Panandikar

Principal

DON BOSCO COLLEGE OF ENGINEERING

FATORDA, MARGAO, GOA- 403 602

2018 - 2019



CERTIFICATE

The dissertation entitled,

"Automated Pneumonia Detection"

submitted by,

Mr. Mrunal Thakur

Miss. Shreya Pokle

Mr. Ronan Revadker

Miss. Swizel Rodrigues

Miss. Drushti Shah

in partial fulfillment of the requirements of the Bachelor's Degree in Computer Engineering of Goa University is evaluated and found satisfactory.

DATE: _____ **EXAMINER 1: _____**

PLACE: _____ **EXAMINER 2: _____**

ABSTRACT

Challenge in detecting pneumonia using chest X-rays is that, pneumonia in most cases is not confined to one particular area. It is difficult to locate the consolidations on X-rays. Radiologists have to examine a lot of medical images and may miss some features that are essential for the diagnosis. This may happen mostly from human fatigue or strain from examining many patients. Our software can potentially overcome these drawbacks by using deep learning to determine if the patient is infected with pneumonia.

In this way, the software will use the chest X-ray images and conclude with the probability of pneumonia infection, which then will be examined by the radiologist. The major outcome of this project should be a successful setup to make the best use of the computerized system and its usage in medicine.

ACKNOWLEDGEMENT

First and foremost, we would like to thank our Director Fr. Kinley D' Cruz**,** Principal Dr. Neena S. P. Panandikar and Head of Department Mr. Gaurang Patkar for providing all the help we needed.

The team put in a lot of effort on the project. However, it would not have been possible to complete it without the help of many individuals.
We are thankful to Dr Gaurish Surlakar who provided expertise that helped us greatly. We would like to thank Professor and Head of Radiology Department at Goa Medical College Dr. Jeevan Vernekar, Resident doctor Dr. Miguel Furtado at GMC, Dr Paresh K Desai from Magnum Diagnostics, Dr. Jagruti Nadkarni and Dr Nalin Gosalia.
We would also like to thank Sampanna Barve, student at GMC.

We are highly indebted to Mr. Amey Kerkar (Asst. Professor) for his guidance and constant supervision as well as for providing necessary information regarding the project and to our co-guides Mr. Vivek Jog and Mr. Amey Tilve for their guidance.

We would like to take this opportunity to thank our institution DON BOSCO COLLEGE OF ENGINEERING and our faculty members of Computer Engineering Department who helped and supported us to go ahead with the project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

Chapter No. 1

INTRODUCTION

1.1. Introduction to Project

What is Pneumonia?

Pneumonia is defined as an acute respiratory illness associated with recently developed radiological pulmonary shadowing which may be segmental, lobar or multilobar. Pneumonias are classified as Community or Hospital acquired or those occurring in Immunocompromised hosts.

'Lobar Pneumonia' refers to homogenous consolidation of one or more lung lobes often associated with pleural inflammation.

'Bronchopneumonia' refers to more patchy alveolar consolidation associated with bronchial and bronchiolar inflammation often affecting both lower lobes[1].

Clinical Features

Community Acquired Pneumonia

Systemic Symptoms: fever, rigors, chest pain, pain in the shoulder or anterior abdominal wall, shivering & vomiting, loss of appetite and headache.

Pulmonary: breathlessness & cough.

Hospital Acquired Pneumonia

Purulent sputum, unexplained increase in Oxygen, core temperature exceeds 38.3℃ & leucocytosis.

Pneumonia in Immunocompromised Patients

cough, fever, breathlessness, profound degree of immunosuppression[1].

The goal of the project is to give the probability of infection of pneumonia in a patient's chest XRAY image so that the radiologist can examine the cases with high probability of infection. Our software will analyze the XRAY image and tell the probability of the patient being infected with pneumonia which can then be verified by a doctor.

1.2 Purpose of the Project

In the clinic, interpreting medical images has been performed by human experts, mostly radiologists and physicians. However, given the wide variations in pathology and the fatigue of human experts, doctors and researchers have begun to benefit from computer-assisted interventions.

The project aims to build software that will diagnose chest abnormalities with respect to pneumonia. This software will assist the radiologist in selecting the medical images that may have a higher probability of infection, and therefore need to be examined much closely. The major outcome of this project should be a successful setup to make the best use of technology in the medical field.

Problem Definition

Pneumonia is an infectious disease of the lung and major cause of mortality that affects nearly 10 million people in India every year. Currently, radiological examination is the essential method of diagnosing. Delayed detection of pneumonia may lead to severe consequences.

Radiologists have to diagnose many medical images (X-Rays, CT and MRI scans) at a time. However, they might miss certain parameters that could help make the diagnosis strong. Automating this detection task will greatly improve the efficiency of radiologists needing their expertise to analyze only questionable cases.

1.3.1 Existing System

A physical test is conducted to detect abnormal sounds during inhaling. A diagnostic test is required that includes blood test and chest x-ray.

Blood test of complete blood count (CBC) is taken to examine if your immune system is actively fighting an infection and if it has spread to your bloodstream.

A chest x-ray is taken to determine the extent and location of the infection. Chest radiography contains a large amount of information about a patient's health. A radiologist trained in interpretation of X-rays analyses the images, which is further read by your doctor who then specifies which treatment needs to be followed.

The doctor may recommend other test for severe cases that include Sputum test, Pleural fluid culture, Pulse oximetry, CT scan bronchoscopy.

1.3.2 Proposed System

A CNN motivated deep learning software to diagnose chest abnormalities, provides analysis of x-ray images as input. Software based automated interpretations to CXR reports leverage assistance to radiologists and doctors in identifying images that requires specific attention making use of computer aided software essential in medical field.

Scope of the Project

The scope of this software is to help mitigate the interpretability challenges faced when dealing with medical imagery to assist the radiologist with chest x-rays that need to be examined closely.

1.3 Report organization

The current introductory section provides a brief introduction about each chapter.

Chapter 1: Introduction

This section focusses on the purpose and scope of the proposed system of Detection of various life stages of Coconut using image processing. It also highlights the limitations of the existing system with regards to the proposed system.

Chapter 2: Literature Survey

This section describes the concepts and technologies used to develop the project.

Chapter 3: Software Requirement Specification

This section provides information about specific requirement of the proposed system.

Chapter 4: System Design

This section describes the software lifecycle model, which will be used in developing the software. It also includes the system design and detailed design.

Chapter 5: Implementation

This section deals with the implementation of the project where in the snapshots of each execution step are shown.

Chapter 6: Conclusion

This section deals with the conclusion that can be derived after implementing the final system.

Chapter No. 2

LITERATURE SURVEY

2.1 PNEUMONIA

Pneumonia is an infection in lungs caused by microbes, resulting in inflammation. Inflammation brings water into the lung tissue which makes it harder to breathe. While breathing, air travels through Trachea, Bronchi, Bronchioles and Alveoli. Alveoli are tiny air sacs which are wrapped in network of capillaries where most of the air exchange takes place. Oxygen leaves the alveoli and passes into the blood stream whereas Carbon dioxide leaves the blood stream and is exhaled out.

In addition to air we constantly breathe other substances like microbes. Human body is usually good at protecting itself from microbes. It does this by coughing, mucociliary escalator (lines the entire airway and moves out the larger bacteria) and macrophages that are situated deep inside the alveoli and destroys anything that lands there. But occasionally a particular microbe might succeed in colonizing the bronchioles or alveoli and when that occurs it causes Pneumonia. These microbes usually multiply and cross over from airways into the lung tissue generating an inflammatory response. In response to this inflammation the tissue quickly fills with White Blood Cells (WBCs) as well as proteins, fluids and even Red Blood Cells (RBCs) if any nearby capillaries get injured in the process.

Pneumonia is usually caused by viruses and bacteria, but it can also be triggered by fungi and a special class of bacteria called Mycobacteria. In adults the top common cause of Pneumonia is influenza (flu). The bacterial causes in adults include *Streptococcus pneumoniae*, *Haemophilus influenzae* and *Staphylococcus aureus*. There are other unusual bacteria with no cell walls and are well known or causing "Atypical or Walking Pneumonia" because they often cause vague symptoms like fatigue. In individuals with normal immune system fungi are a rare cause of Pneumonia and often regional. Mycobacteria are like slow growing fungi hence 'myco' even though they are bacteria.

Pneumonia can also be categorized by how it is acquired.

Community Acquired Pneumonia (CAP) is when a person gets ill outside the hospital or healthcare setting. Hospital Acquired Pneumonia (HAP) is when a person is already hospitalized for some illness, this happens to be more serious because they have already weakened immune system and microbes in hospital are more resistant to common antibiotics. Another class is ventilation associated pneumonia (pneumonia in immunocompromised patients), it is a subset of HAP and develops when ill individuals are connected to a ventilator. The microbes can move directly from the tube into lung and cause pneumonia. Another type is Aspiration Pneumonia which is caused when having food. You accidently breathe in the food item instead of swallowing it, that is you aspirated that particular food item. Normally you would gag and start coughing and work it out of your lungs but these gag reflexes can be compromised by drugs and alcohol abuse, brain injuries or swallowing issues. These food particles obviously aren't sterile and can infect the lungs and cause pneumonia. This can happen with drinks or gastric content.

Another way of categorizing pneumonia is by the areas affected by the infection. In Bronchopneumonia, the infection can be all over the lungs involving the bronchioles as well as alveoli. In atypical or interstitial, infection is mostly just outside the alveoli or interstitial. In lobar pneumonia, it causes complete consolidation of whole lobe of the lung, meaning the entire region is filled with fluid.

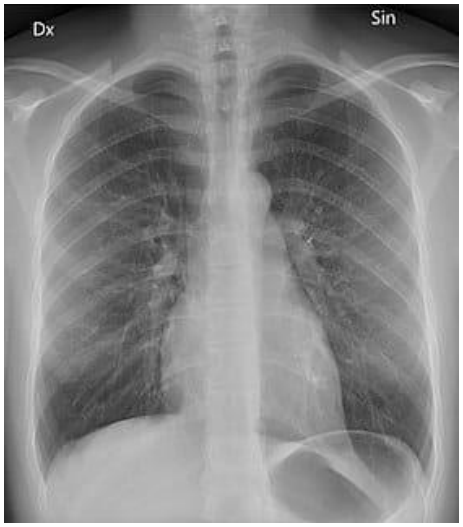


Fig. 2.1 CXR of normal patient

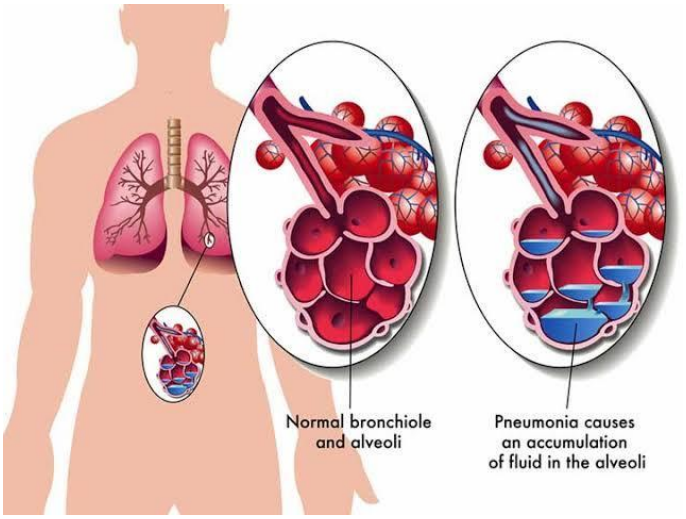


Fig. 2.2 Normal v/s infected alveoli and bronchiole

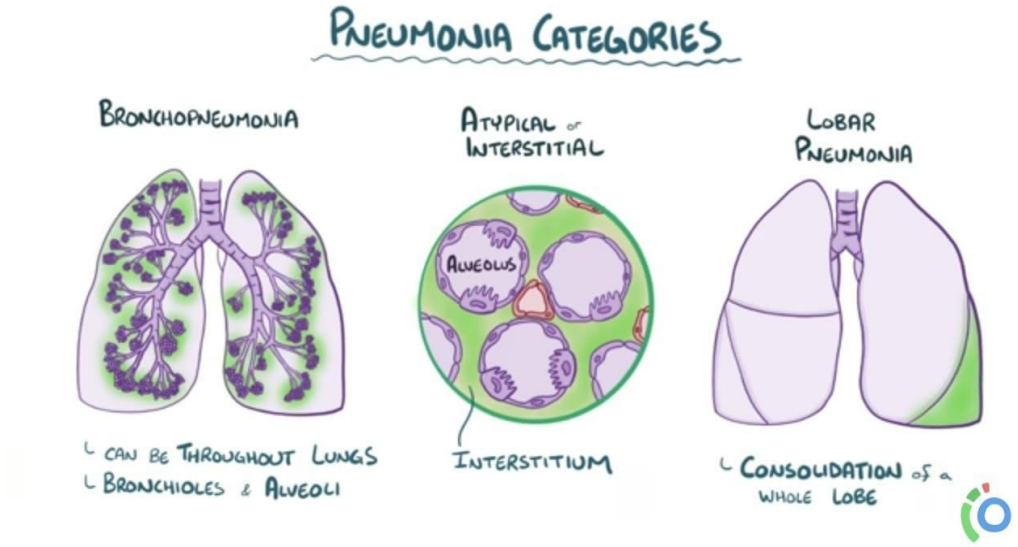


Fig. 2.3 Categories of Pneumonia

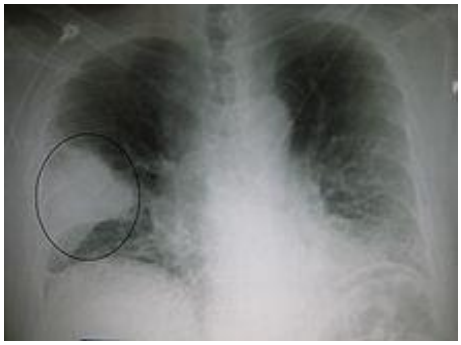


Fig. 2.4 CXR of infected patient

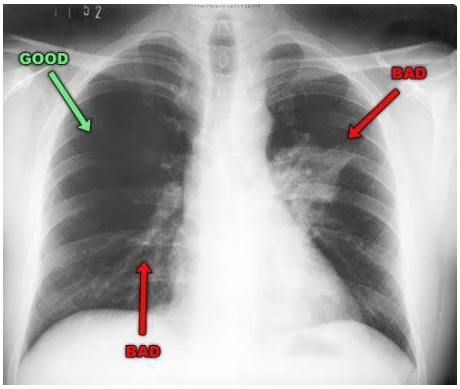


Fig. 2.5 Distinction of normal and infected regions in CXR

2.2 X-rays

X-ray is an imaging technique which uses small amounts of radiations to produce images of chest and internal organs. A type of x-ray is chest x-ray (CXR). Chest x-ray can determine the abnormalities or diseases in the airways, bones, heart and lungs. It can also spot any fluid buildup in the lungs. Doctor can order for chest x-ray when the symptoms suspected are related to chest abnormalities. These symptoms can include chest pain, shortness of breath, persistent cough, fever etc.

To get the CXR, the chest is exposed to the radiations from x-ray machine and an image is produced on a film or into a digital computer. Various organs and bones have different densities and depending on that they absorb varying amounts of radiations to produce shadows on the film. The ribs and vertebrae absorb more radiations as they are denser compared to lung tissue, which is mostly composed of air and hence allows most of the radiations to pass through. The heart will appear whitish but less bright than bones.

The procedure followed to get the CXR is as follows:

i. The patient is made to stand in front of a surface adjacent to the film that records the images. Front of the chest is closest to the surface.
ii. Another machine releases radiations which is placed about 6 feet away, behind the patient.
iii. The technician may advice to take deep breath and hold it for a while and takes the image by operating the device. Image is captured in few seconds.

Usually one image is taken from back to front (Posterior –Anterior or "PA" view) and a second image is taken using sideways view from side to side (Lateral). For some patients who are too weak to stand in front of the x-ray machine or children, the image is taken from front to back; it is called an Anterior –Posterior (AP) view. A CXR is an easy, quick and effective test hence it is used widely to view some most vital organs.

2.3 CONVOLUTIONAL NEURAL NETWORKS

2.3.1 DRAWBACKS OF IMAGE PROCESSING

Due to the widespread variation from patient to patient data, traditional learning methods are not trustworthy. Machine learning has developed over the last few years by its ability to move through complex and big data. Many image diagnosis tasks require a domain expert to analyze the images, identify a set of filters, transformations and pre-processing steps that would be best to uniquely identify the abnormalities. An algorithm would be devised that would do the same, but in deep learning some distinct and labelled images are given beforehand. Based on these images, classification rules are derived. Digital image processing has shown the inability to learn from new information and to overcome this deficiency deep learning approaches are introduced.

2.3.2 TYPES OF LEARNING ALGORITHMS

SUPERVISED LEARNING

Supervised learning is the process of learning through pre-labelled inputs, which act as targets. For each training example, there will be a collection of input values or vectors and one or more accompanying designated output values.

The goal of supervised learning is to reduce the model's overall classification error, through the correct computation of the output value of training example by training. [10]

UNSUPERVISED LEARNING

In unsupervised learning, the training set does not include any labels. Success is usually determined by whether the network is able to increase or reduce an associated cost function. Unsupervised learning is used for pattern detection or recognition and descriptive modeling. It cannot be directly applied to a regression or classification problem since you have no idea what the values for the output data might be. [10]

2.3.3 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) are computational processing systems which are heavily inspired by the way biological nervous systems (such as the human brain) operate. ANNs mainly comprises of a high number of interconnected computational nodes (referred to as neurons), of which work is entwined in a distributed fashion to collectively learn from the input in order to optimize its final output.

We load the input usually in the form of a multidimensional vector to the input layer which will circulate it to the hidden layers. The hidden layers will then make selections from the previous layer and weigh up how a stochastic change within itself detriments or improves the final output. This is denoted as the process of learning.

One of the greatest limitations of conventional forms of ANN is that they struggle with the computational complexity required to compute image data. [11]

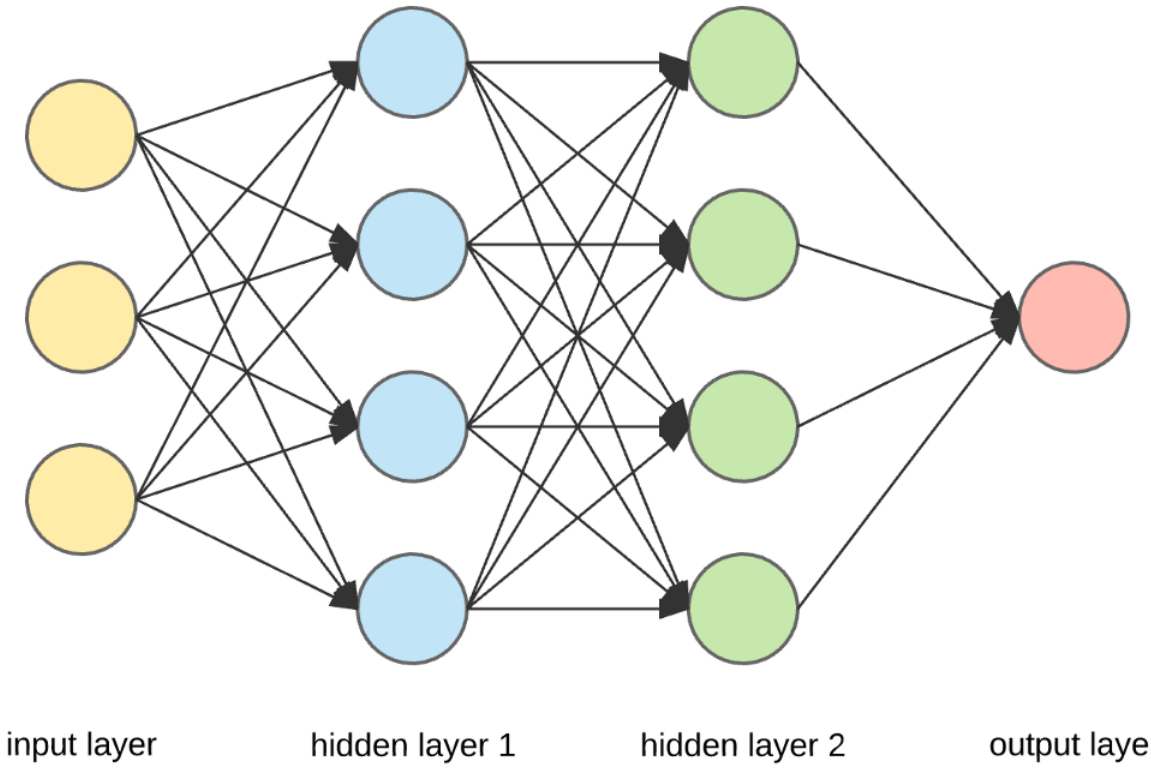


Fig. 2.6 Artificial neural network

FEED FORWARD

Artificial neural networks are models that imitate the structure of the neural system and learn patterns inbuilt in observations. The perceptron is the earliest neural network with a single layer architecture with an input layer and an output layer.

A perceptron with several output units is regarded as a linear model which prohibits its applications in problems with complicated data patterns. This can be overcome by using a hidden layer between the input and output layer. [2]

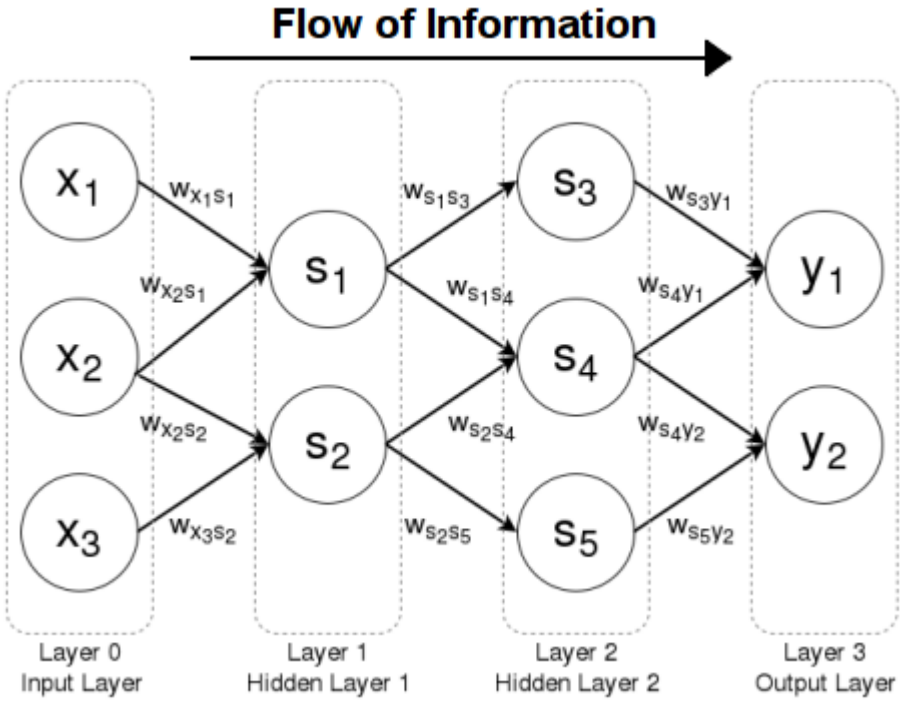


Fig. 2.7 feed forward network

BAYESIAN NETWORK

It is a graphical model representing the joint probability distribution of a set of random variables via directed acyclic graph and conditional probability distributions for each node in the graph.

Bayesian networks map the relationship between events demonstrating how occurrence of certain events influence the probability of other events occurring.

Sometimes we need to compute the probability of an unknown cause given some observed evidence. For example, doctors predict the probability of a specific disease based on certain observed symptoms in a patient. Such problems are complex as there are many inter-related

variables. There might be multiple symptoms, and even more potential causes of those symptoms.

Usually in such cases doctors obtain reversed conditional probability is obtained i.e. the probability of symptoms if the patient has a particular disease or probability of evidence given the disease.

In such cases, Bayesian approach is appropriate.

If there exists a causal probabilistic dependence between two random variables in the graph, the corresponding two nodes are linked by a directed edge. [12]

For example, a directed edge from a node A to a node B indicates that the random variable A causes the random variable B. Since the directed edges represent a static causal probabilistic dependence, cycles are not allowed in the graph.

Hence, Bayesian network could symbolize the probabilistic relationship between diseases and symptoms i.e. Given particular symptoms, the network can compute the probability of existence of numerous diseases. [5]

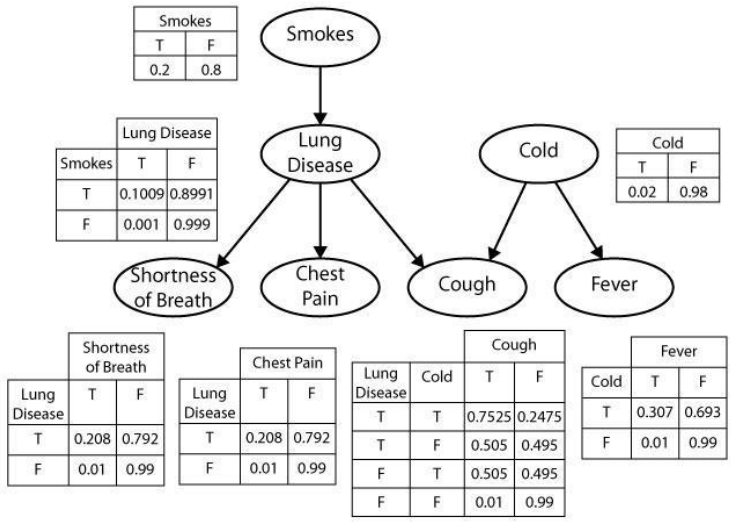


Fig. 2.8 Bayesian network

BACKPROPAGATION NEURAL NETWORK (BPNN)

Backpropagation neural network (BPNN) is a multi-layered feedforward neural network.

It uses a supervised learning algorithm known as error back-propagation algorithm.

Errors accumulate at the output layer which are propagated back into the network to adjust the weights. All the operations progress in the forward path during simulation. There isn't a backward pass of computation excluding the operations utilized in training.

The BPNN pseudocode algorithm given below:

(i) Network initialization: randomly choose the initial weights

(ii) Select first training pair

(iii) Forward computation that includes the following steps:

(a) Apply the inputs to the network

(b) Calculate the output for every neuron from the input layer, through the hidden layer(s), to the output layer

(c) Calculate the error at the outputs

(iv) Backward computation

(a) Use the output error to compute error signals for pre-output layers

(b) Weight adjustments are done using error signals

(c) Apply the weight adjustments

(v) Repeat computations for other training pairs.

(vi) Regularly evaluate the network performance.

Repeat computations until the network converges on the target output.

$$O_{pj}(\text{net}_j) = \frac{1}{1 + e^{-\lambda \text{net}_j}},$$

$$\text{where} \quad \text{net}_j = b_j + \sum_k O_{pk} W_{kj},$$

Eqn. 2.1 Output of network

where k ranges over the input indices, Wkj is the weight on the connection from k-th input to j-th neuron, and bj is the bias weight for the j-th output neuron.

The below equation calculates the error signal at the output

$$E = \frac{1}{2} \sum_{i=1}^{N} \left(T_{pj} - O_{pj}\right)^2,$$

Eqn. 2.2 Error signal at output

where Tpj is the target value of the j-th output neuron for pattern p and Opj is the actual output value of the j-th output neuron for pattern p.

The backpropagation algorithm is based on the gradient descent optimization method [13]. By determining the derivative of error, we can update the network parameters.[3]

COMPETITIVE NEURAL NETWORK

The competitive neural network is a simple neural network that consists of two layers: Input layer and Output layer and it uses unsupervised learning. The input layer is fully connected to the output layer and features of the image are given as input to the network and it outputs classes.

When the input features are applied to the input layer, the neurons in the output layer compete among themselves. This network relies on the Hebbian Learning Rule.

At a time, only one neuron can be fired. It uses the "winner takes all" strategy where only the weights connected to the winner neuron are revised in a particular epoch and the other weights remain the same. This strategy has the resulting effect of fortifying the correlation between the inputs and the corresponding winner neurons during learning. [3]

Output of the winner neuron k is given by

$$\Delta w_{kj} = \eta\left(x_j - w_{kj}\right).$$

Eqn. 2.3 Output at winner neuron

STACKED AUTOENCODER

An autoencoder encodes the input values *x* using a function, then decodes the encoded values *f(x)* using a function *g* to form output values identical to input values. It learns a compressed representation of the input by minimizing the reconstruction error between the input and output values of the network. The representative power of a single layer autoencoder is very limited. In the stacked autoencoder, multiple autoencoders are stacked to considerably improve the representational power by using the activation values of the hidden units of one auto-encoder as the input to the next higher auto-encoder. Deep networks trained this way perform worse than networks with a shallow architecture as they fall into a poor local optimum. [2]

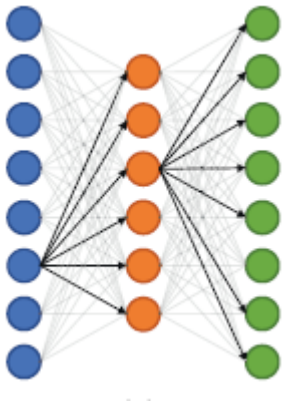


Fig. 2.9 Stacked autoencoder

DEEP BELIEF NETWORKS

A deep belief neural network can be seen as a multilayer construction of belief networks called Bayesian networks or Boltzmann machines. They are composed of multilayer of stochastic latent variables. The two top layers are undirected and the connections between the lower layers are directed. The deep belief network is pre-trained using a greedy algorithm. Then several steps of Gibbs sampling is implemented on the top two hidden layers. This stage is essentially used to obtain a sample from the RBM defined by the top two hidden layers. Then using a single pass of ancestral sampling through the rest of the model we obtain a sample from the visible units. Learning is inferred by a single bottom-up pass. Greedy pre-training is started with an observed data vector in the bottom layer. It then uses the generative weights in the reverse direction using fine-tuning.

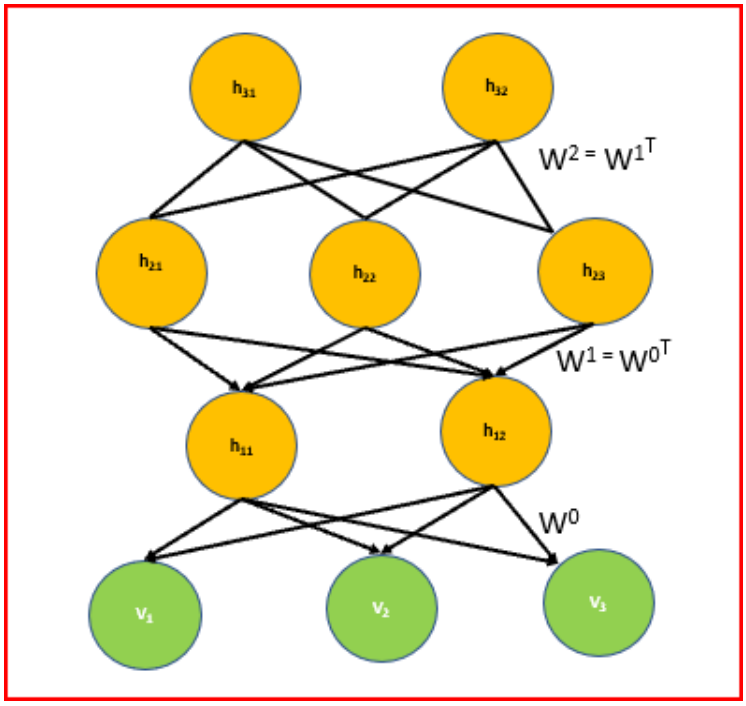


Fig. 2.10 deep belief network

DEEP LEARNING IN MEDICAL IMAGES

Recent advances in machine learning, especially in deep learning help in identifying, classifying and quantifying patterns in medical images. This is due to the ability to utilise hierarchical feature representations which are learned from data instead of features devised by hand specific to the domain.

In the clinic, interpretation of medical image has been performed mostly by human experts such as radiologists and physicians. However, given possible fatigue of human experts and wide variations in pathology, the doctors and researchers have begun to profit from computer-assisted interventions. [2]

The success of deep learning is due to :

● Advances in high tech central processing units (CPUs) and graphics processing units (GPUs)

● Availability of huge amounts of data

● Advancements in learning algorithms

CONVOLUTIONAL NEURAL NETWORKS FOR DETECTING LUNG CANCER [14]

The general method for cancer screening is CT or computerized tomography scanning which uses x-rays to generate a series of cross-sectional images. These scans are examined by a radiologist in order to determine whether they contain potentially cancerous lung nodules. To increase the accuracy of this process a computer system that flags images suspected to have lung cancer nodules is used as a second opinion. These systems are referred to as computer-aided diagnostics systems or CADs. CADs involve pre-processing, segmentation, candidate detection, feature extraction, and classification steps. Though they are effective in classifying the lung nodules their complexity makes them fragile. If one of the components of the system is disrupted, all of the subsequent steps suffer. Convolutional neural networks are one potential solution. Several different CNN architectures were trained and evaluated on the dataset from Kaggle.

Training:

Each batch contained eight positive and eight negative training images. Each model was trained on approximately one thousand randomly selected batches. The update step consisted of five iterations of stochastic gradient descent with a learning rate of .01 to minimize the in sample mean squared error. The best set of weights determined by the minimum mean squared error on the validation set was saved to prevent over fitting. Models were compared on the basis of their error on the validation set. The best model was evaluated with the held out test set.

Architecture:

Even though the number of layers changed in each experiment the implementation remained the same. The convolutional layers often were followed by max pooling. This downsized the images by using a 4x4 window. The fully connected layers were standard with their bias terms initialized to zero. Each layer used a tanh(hyperbolic tangent) activation function and batch normalization unless noted otherwise.

2.3.4 CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks recognize images by transforming the original image through layers to some class scores. They were inspired by the visual cortex ie every time we view an object, a series of layers of neurons are activated. Each layer then will detect a set of features such as lines and edges. The higher levels of layers detect more complex features so we can recognize what we have seen. In the stack autoencoders, deep belief networks the inputs are in vector form. The structural information among neighbouring pixels is important but vectorization damages such structural and configural information in images. Convolutional neural networks are better designed to utilize spatial and configural information since they take two and three-dimensional images as input.

CNN V/S REGULAR NN

Convolutional neural networks have a different architecture compared to regular neural networks. Regular neural networks change the input by passing it through a sequence of hidden layers. Every layer comprises of a set of neurons, where each layer is fully connected to all neurons in the previous layer. Lastly, there is a fully-connected layer which is the output layer that represents the predictions.

In convolutional neural networks, the layers are organized in three dimensions: width, height, and depth. Further, the neurons in the current layer do not connect to all the neurons in the next layer but only to a tiny region of it. Lastly, the last output is reduced to a single vector of probability scores.

ADVANTAGES OF USING CNN

● Convolutional neural networks use fewer parameters (weights) to learn than a fully connected network.

● They are intended to be invariant to distortion and object position in the scene.

● They automatically learn from the input domain and generalize features

● Regular neural networks don't scale well to full images i.e. full connectivity is wasteful and the huge no of parameters may instantly lead to overfitting

CONVOLUTION

It is a mathematical operation to merge two sets of information. The convolution is applied on the input data using a convolution filter which produces a feature map. The convolution is executed by sliding the filter on the input. At each location, we do element-wise multiplication and sum the result.

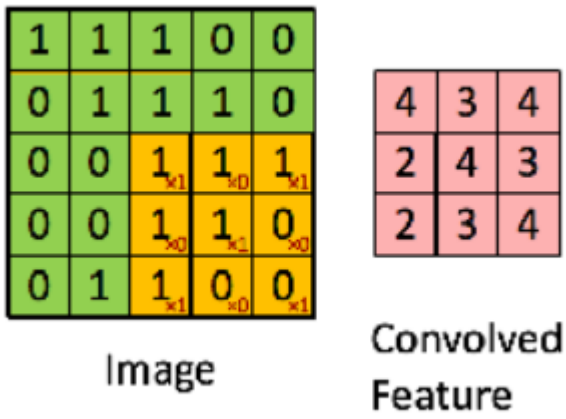


Fig. 2.11 Convolution

The main advantage of using convolution is sparsity of connections and parameter sharing. Parameter sharing helps by reducing the number of weight parameters in one layer without losing accuracy. The convolution operation breaks down the input features into a reduced feature space, each output value then depends on a small number of inputs and can be adjusted quickly.

FILTERS

The filter is a small matrix used for feature detection. A particular filter is designed so as to extract some particular features on the image. Filters are also known as kernel or feature detector. Features can be straight edges, simple colors, curves, etc.

FEATURE MAP

Feature map is the output volume formed by sliding the filter over the image and calculating the product. Feature maps are also known as convolved features or activation maps.

RECEPTIVE FIELD

The receptive field is a local area of the input volume that has the identical size as the filter.

STRIDE

Stride is defined as the number of pixels shifts over the input matrix. We move the filters to one pixel at a time when the stride is one. When the stride is two then we shift the filters to two pixels at a time and so on.

ZERO PADDING

To ensure that the convolutions finish with the same number of outputs as inputs, we add zeros around the outside of the input volume. This is known as zero padding. If we don't use padding the information at the border will be lost.

PARTS OF CNN:

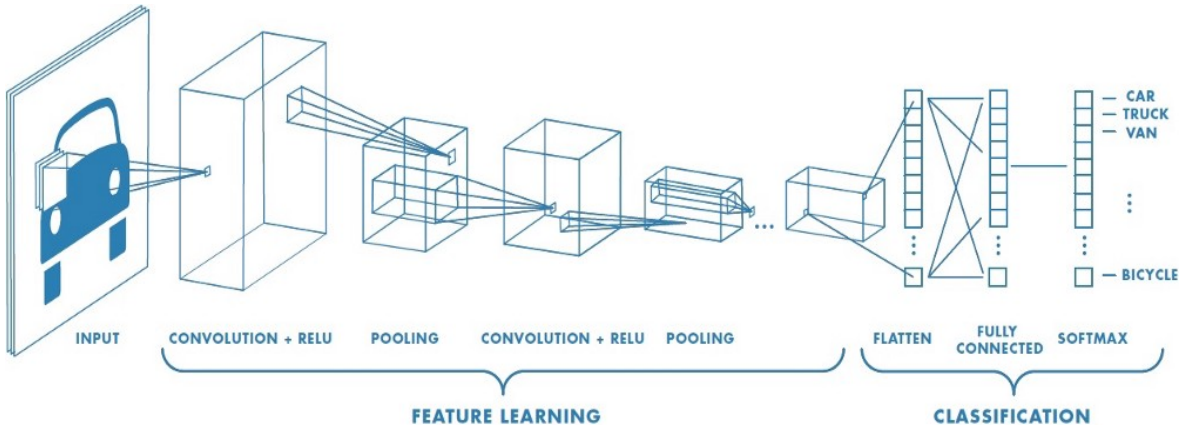


Fig. 2.12 Layers of CNN

FEATURE LEARNING:

Feature extraction starts with an initial set of data and builds features which are planned to be informative and non-redundant. Thus, easing the succeeding learning and generalization steps, and in some cases directing to better human interpretations. The selected features are expected to contain the relevant information from the input data. The desired task can then be performed by using the reduced representation instead of the complete initial data. The feature extraction process consists of the following:

● CONVOLUTION:

The kernel or the filter acts as the peephole which executes a mathematical operation on the image while scanning the image. The filter glides over the input image one pixel at a time beginning from the top left. The filter multiplies its own values with the corresponding values of the image and adds all of them up to yield a single value for each overlap. Convolution conserves the relationship between pixels by learning image features using small squares of input data.

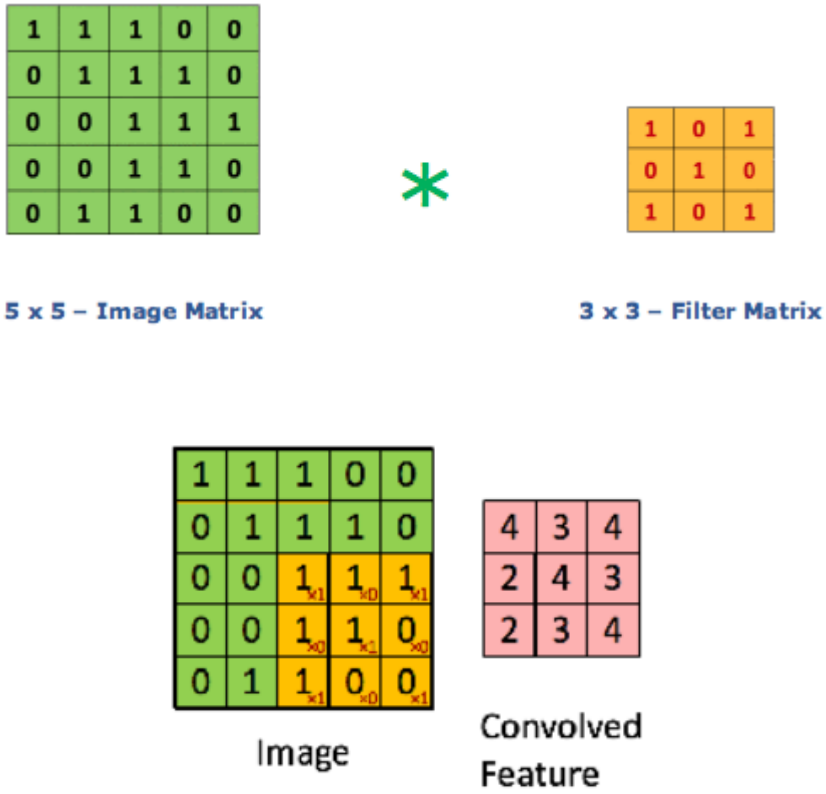


Fig. 2.13 convolution operation performed on a 5x5 matrix with a 3x3 filter matrix

● POOLING:

Pooling layer reduces the amount of computations and parameters in the network, thus controlling overfitting by progressively reducing the spatial size of the network. It is used to reduce the dimensionality which shortens the training time and combats overfitting. Max-pooling will take out only the maximum from a pool. This is done with the use of filters which slide through the input; and at every stride, the maximum value is taken out and the rest is dropped. This identifies the most important features. This leads to down sampling the network while keeping the significant information.

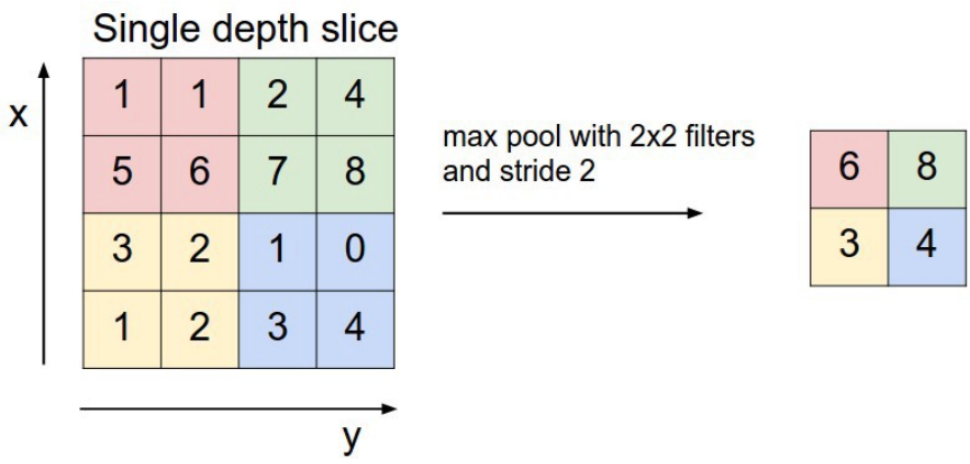


Fig. 2.14 Pooling operation

● OPTIMIZERS

Optimizers help to minimize the objective function (error function). Objective functions are mathematical functions that depend on the learnable parameters of the model.

Two major categories of optimization algorithms:

1) First order optimization algorithms

These algorithms use gradient values with regard to the parameters to minimize or maximize the loss function.

2) Second order optimization algorithms

These algorithms use the second order derivatives of the parameters also called the **Hessian** to minimize or maximize the loss function. Hessian is a matrix of second order partial derivatives. Second order derivative shows if first derivative is growing or declining which hints at the curvature of the function. But second order derivatives are costly to compute, so they are not used much.

Types of optimization algorithms:

1) Gradient Descent

It is used for to perform weight updates in the model so as to minimize the loss function. Backpropagation is used to train the neural networks, wherein we first propagate forward to calculate the dot product of the input signal and their corresponding weights and then apply activation function to those sums of products and also introduces non-linearities to the model which enables it to learn almost any arbitrary functional mapping. After this we back propagate in the network carrying error terms and updating weights using gradient descent. In this we calculate the gradient of error (E) function with respect to the weights (W) and update the weights in opposite direction of gradient of loss function.

$$\theta = \theta - \eta \cdot \nabla J(\theta)$$

Eqn. 2.4 Updated parameter for gradient descent

Where '**η**' is the learning rate and '**∇J(θ)**' gradient of loss function.

This calculates gradient for the whole data set but performs only one update, hence it can be slow and hard to control for data sets which are huge and don't fit into the memory.

Variants of gradient descent:

i) Stochastic Gradient Descent (SGD)

SGD performs parameter update for each training example. It is much faster and performs one update at a time.

$$\theta = \theta - \eta \cdot \nabla J(\theta; x(i); y(i))$$

Eqn. 2.5 Updated parameter for stochastic gradient descent

Due to these frequent updates, the updates have high variance and causes loss function to fluctuate to different intensities. This helps to discover new and better local minima but this also complicates the convergence to exact minimum and keeps overshooting.

ii) Mini Batch Gradient Descent

This performs an update for every batch with n training examples in each batch. It reduces variance in parameter updates. The batch size ranges from 50-256 samples.

Challenges faced by Gradient Descent and its variants:

- Learning rate can be difficult to choose. Too small learning rate leads to slow convergence while large learning rate can hinder convergence and loss function varies around minimum
- Same learning rate is applied to all parameter updates
- For non-convex error functions, the challenge is to avoid getting trapped in the numerous sub-optimal local minima and also from saddle points (points where one dimension slopes up and other slopes down).

2) Momentum

To overcome the high variance problem in SGD momentum was introduced which navigates along the relevant direction. It adds '*Y*' of the past update vector to current update vector.

$$V(t) = \gamma V(t-1) + \eta \nabla J(\theta)$$

Eqn. 2.6 Calculation of vector

$$\theta = \theta - V(t)$$

Eqn. 2.7 Updated parameter for momentum

The momentum is typically set to 0.9 or similar value. Momentum leads to faster and stable convergence, also reduces oscillations. The term '*Y*' increases for dimensions whose gradient points in same direction and decreases for those whose gradient changes directions. It does update for only relevant examples and leads to faster and stable convergence and reduced oscillations.

3) Nesterov Accelerated gradient

The problem with momentum was, when we reach the minima the momentum is high and we may entirely miss the minima and continue to move up. This problem is addressed in Nesterov Accelerated Gradient. Here, we first make a jump based on previous momentum and then calculate gradient and then make corrections which results in parameter update.

$$V(t) = \gamma V(t-1) + \eta \nabla J(\theta - \gamma V(t-1))$$

Eqn. 2.8 Calculation of vector

$$\theta = \theta - V(t)$$

Eqn. 2.9 Updated parameter for Nesterov gradient

4) Adagrad

Adagrad allows learning rate to alter based on parameters. It allows small updates for recurrent parameters and greater updates for infrequent parameters which makes it suitable to sparse data. It uses different learning rate for every parameter.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

Eqn. 2.10 Updated parameter for Adagrad

Due to buildup of each squared gradient in the denominator the learning rate starts to shrink and eventually becomes so small that model just stops learning and acquiring new additional knowledge. This Decaying Learning Rate problem is a disadvantage of Adagrad.

5) Adadelta

Adadelta overcomes the problem of decaying learning rate. It limits the window of collected past gradients to some fixed size **w** instead of accumulating all previous squared gradients. Instead of saving **w** previous squared gradients, the sum of gradients is recursively termed as decaying mean of all past squared gradients. Here we don't have to set default learning rate.

$$E[g^2](t) = \gamma . E[g^2](t-1) + (1-\gamma) . g^2(t)$$

Eqn. 2.11 Running average

$$\Delta\theta(t) = -\eta \cdot g(t,i)$$

$$\theta(t+1) = \theta(t) + \Delta\theta(t)$$

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

Eqn. 2.12 Updated parameter for Adadelta

6) Adam

Adaptive Moment Estimation calculates adaptive learning rates i.e. storing exponentially decaying average of past squared gradients; it also stores exponentially decaying average of past gradients similar to momentum.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}.$$

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

QUOTE is the first moment (mean) and QUOTE is the second ~~moment (variance). The~~ values for β_1 is 0.9, 0.999 for β_2

Eqn. 2.13 First and second moments of gradients

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t.$$

Eqn. 2.14 Parameter update for Adam

ACTIVATION FUNCTIONS

We need our neural network to not only work with linear functions but also with images, audio, video, speech etc. To work with such non-linear big data sets we need to apply activation function to generate non-linear mappings from input to output. Neural networks are Universal Function Approximators which means they can compute and learn any function at all. Activation function makes neural network powerful and adds the ability to learn complex data.

Types of activation functions:

1) Sigmoid or Logistic

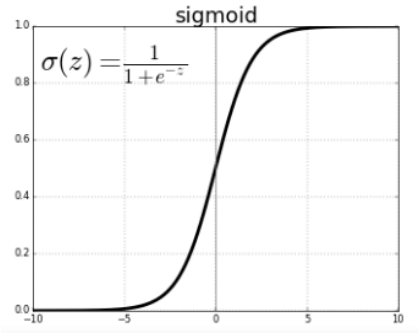


Fig. 2.15 Sigmoid function

It ranges between 0 and 1. It is easy to implement but faces the following problems:

- Vanishing Gradient problem
- Output isn't zero centered
- It saturates and kill gradients
- Slow convergence

2) Hyperbolic Tangent Function (Tanh)

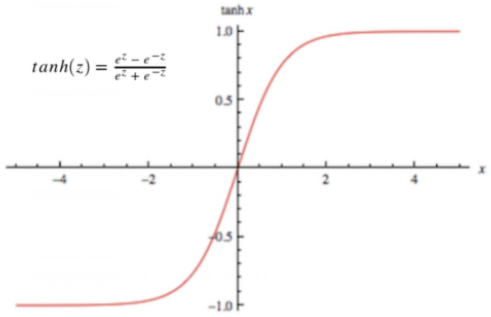


Fig. 2.16 Hyperbolic Tangent Function

The output is zero centered since it ranges from -1 to 1. Therefore optimization is easier but it still faces vanishing gradient problem.

3) Rectified Linear Units (ReLu)

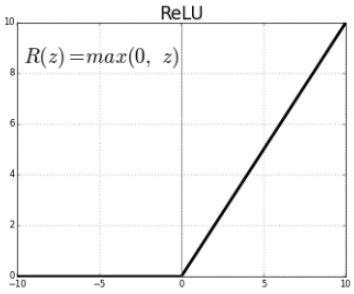


Fig. 2.17 Relu

ReLu overcomes all the problems faced by Sigmoid and Tanh. But ReLu can only be used within hidden layers of neural network. Therefore softmax function must be used for classification problem to compute probabilities. ReLu can result in dead neurons. To fix this Leaky ReLu was introduced which uses a small slope to keep updates alive.

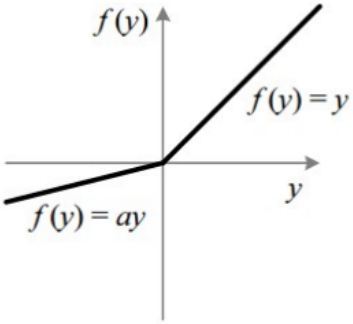


Fig. 2.18 Leaky ReLu

CLASSIFICATION:

After several convolutional and pooling layers, it is necessary to attach a fully connected layer. The three-dimensional volume is flattened into a one-dimensional vector which is given as input to the fully connected layer.

● FULLY CONNECTED:

In the fully connected layer, each neuron in the current layer is linked to every neuron in the subsequent layer. These layers learn to use the features generated by convolution in order to rightly classify the image.

The last fully-connected layer uses a softmax activation function to classify the produced features of the input image into different classes based on the training dataset. The final output will be reduced to a single vector of probability scores.

SOFTMAX:

This function is used to map non-normalised output into probability distribution over predicted output classes.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

Eqn. 2.15 Softmax equation

Chapter No. 3

Software Requirement Specification

3.1 Introduction

3.1.1Purpose

Purpose of the project is to analyse the XRAY image to diagnose the patient with respect to Pneumonia. We aim to make use of technology to provide medical care in areas that lack the resources.

3.1.2Scope

The scope of the project is to provide assistance with reading the XRAY and give an accurate diagnostic report with respect to Pneumonia.

3.1.3Overview

The software will provide the probability of pneumonia in a patient by analysing chest X-rays along with other clinical features and general body tests.

This software will save time in diagnosis of pneumonia and reduce the consequences due to late detection.

3.2 Overall Description

3.2.1Product Perspective

User Interface

- Frontend Software: Python
- Backend Software: Python

Hardware Interface

The software has been tested on the following hardware:

1. Intel core i5 8th Generation 2.3GHz
2. 8GB RAM

Software Interfaces

The software interfaces required for the project are:

1. Anaconda Distribution
2. Tkinter
3. Windows 10 Home Single Language
4. Libraries used: pydicom,Keras,h5py,NumPy, Matplotlib, OpenCV3.

Communication Interfaces

The communication between user and the application is Tkinter where the user can view different steps of processing the image and view intermediate and final results generated.

Chapter No. 4

Design

4.1 Software Development Model

Agile Software Development

Agile is a set of Values and Principles. It is a collection of beliefs that teams can use for making decisions of how to work while developing a software. Agile is very flexible and gives you a foundation that teams can use that result in better software development.

As per the agile manifesto, it is realised that we should value:

Individuals and interactions over processes and tools.

Working software over comprehensive documentation.

Customer collaboration over contract negotiation and responding to change over following a plan.

There are 12 principles that support these values.

1. Satisfy the customer by unceasing delivery of valuable software.
2. Welcome changing requirements even late in development phase. This makes agile more competitive and uses change for advantage of the customer.
3. Deliver working software regularly, in weeks or in a couple of months.
4. The developers and business people must work together daily throughout the project.
5. Build the project around motivated individuals. Give them the support they require and the environment and have faith in them to get the job done.
6. Face to face conversations.
7. Working software is the prime measure of progress.
8. Agile processes encourage sustainable development. The sponsors, users and developers should be able to maintain a continuous pace indefinitely.
9. Continuous attention to good design and technical excellence .
10. Simplicity- the art of maximizing the amount of work not done is essential.
11. The best architectures requirements and designs arise from self-organizing teams.

12. At regular intervals, the team suggests how to become more efficient, then tweaks and as a result adjusts its behaviour suitably.

Agile's real utility is giving people a common foundation for making decisions about the best way to develop software.

Scrum, Xtreme Programming and Kanban are Agile methodologies.

An agile team over time may change and refine its practices. A team that begins with Scrum may switch to Xtreme Programming later.

4.2 System Design

Flow Diagram

The following diagram represents the flow of the software.

The image dataset is divided into training and test sets. The images in the training set are processed into batches and given to the training phase of the model. The input image is given to the model and a probability of whether it is normal or infected is displayed as output.

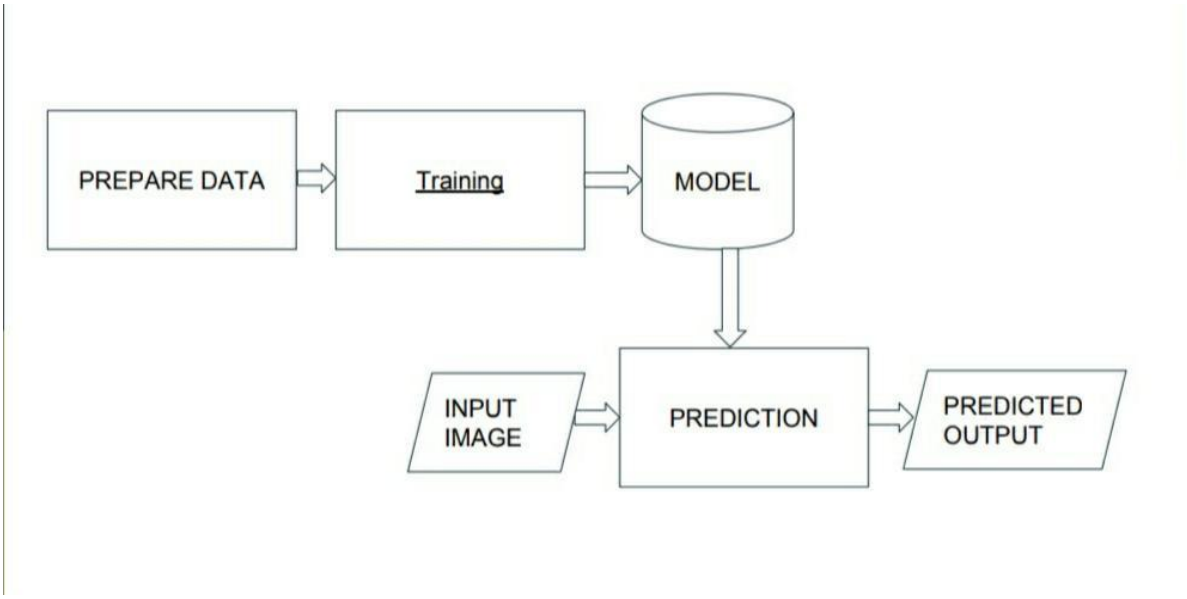


Fig. 4.1 Flow Diagram

4.3 Detailed Design

DICOM:

DICOM (Digital Imaging and Communications in Medicine) is a standard used to handle, print, store and transmit information in medical imaging. It includes a network communications protocol and file format definition. The communication protocol used to communicate between systems is an application protocol that uses TCP/IP. DICOM files can be exchanged between two entities. These entities are capable of receiving image and patient data in DICOM format. DICOM groups information into data sets. For example, the patient ID is contained in the file of a chest XRAY so that the image can never be separated from this information. This is similar to image formats like JPEG which also have embedded tags to identify and describe the image.

A data object of DICOM consists of a number of attributes which include name, ID, etc., and a special attribute which contains the image pixel data. A single DICOM object has only one attribute that contains the pixel data. For many modalities, this corresponds to a single image.

Prepare Data

A path list of all images and a patient ID list are created. The target labels, which classify the X-ray image as normal or infected, is copied from the training CSV file and rearranged according to the image path list. The image dataset is subdivided into two sets, the training set and the test set. The training set is then used to train the model so that it can give a probability that a given X-ray image is normal or infected. The test set is not shown to the model in the initial stage and is used to test the model for accuracy. The training set is then divided into batches. Pixel data required for diagnosing pneumonia is extracted from the DICOM images in each and this data is added to a numpy array. This image data is compressed and saved. A numpy array is also created to store the labels of the images in each batch. This process is repeated for all batches.

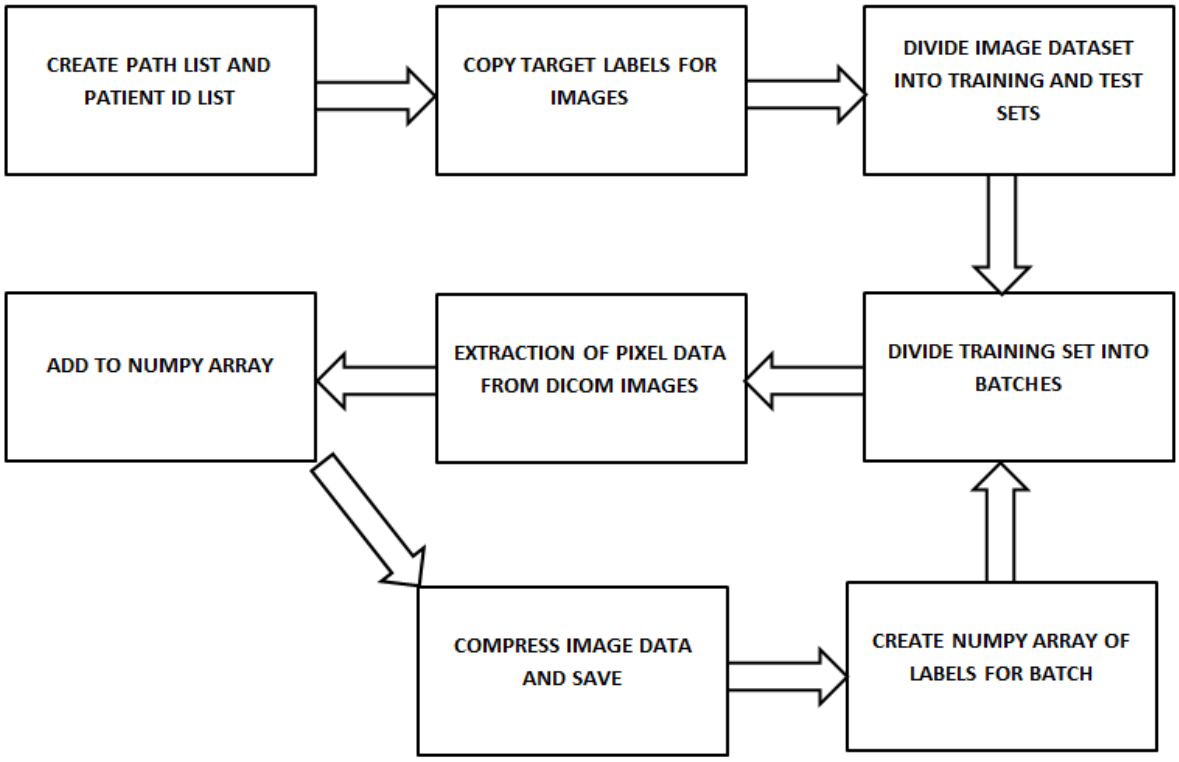


Fig. 4.2 Block Diagram for Prepare Data stage

Training

A total of 25,346 DICOM images were used in training. In the training phase, the model architecture is created. The model is a Convolutional Neural Network (CNN). It has the following layers:

1. 4 layers of Convolution along with Maxpooling
2. A Flatten layer
3. Three Dense layers with RELU activation functions
4. One Dense layer with Softmax activation function

The training set is sub divided into 19 batches, each with 1334 images. Each batch of images is then loaded from the numpy array that stores all the images and given for training in the model. The weights are saved and the model is evaluated for each batch. This process is repeated for all the batches. The final model is then saved for testing and prediction.

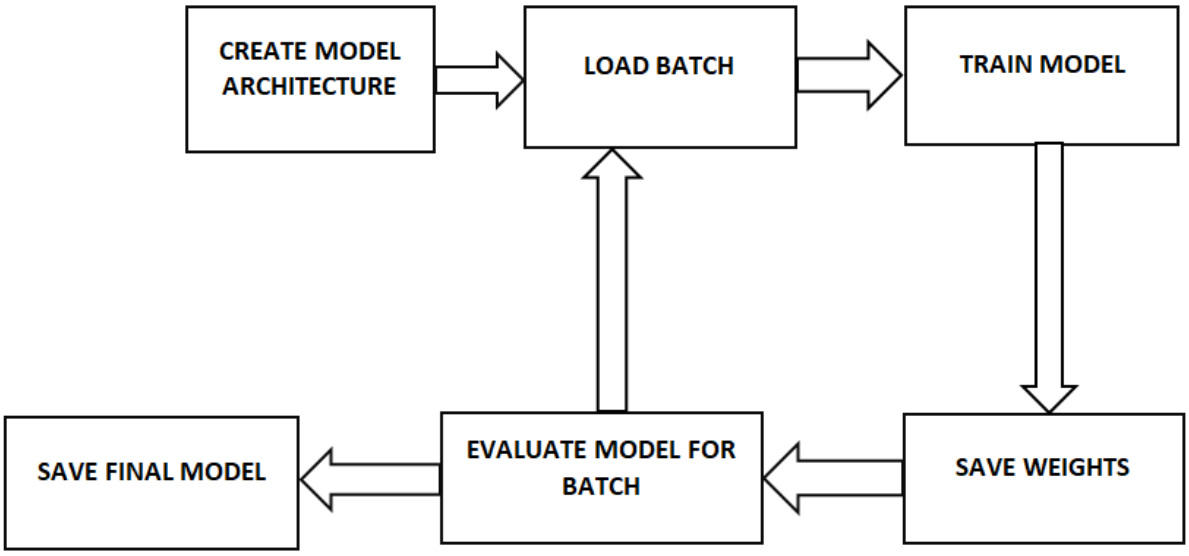


Fig. 4.3 Block Diagram for Training phase

Prediction

This is the working of the front-end interface. The user will give a digital copy of the chest radiograph (DICOM format) to be diagnosed as input to the software. The model created and evaluated in the training phase is loaded into the software. The image is then given as input to the model and the probability of whether it is normal or infected is displayed by the interface.

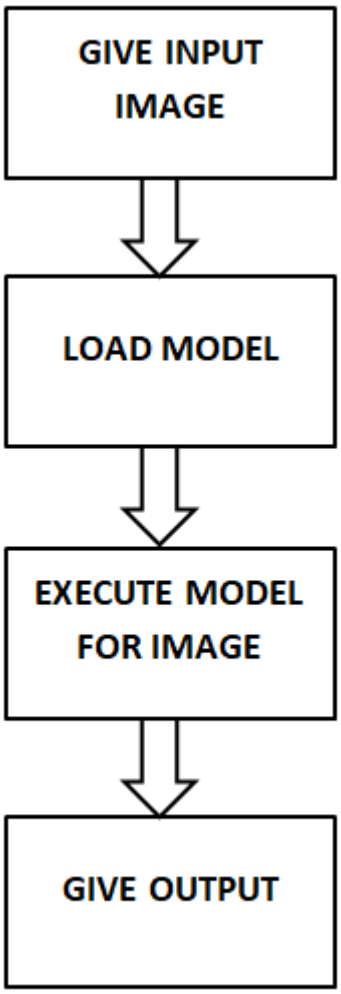


Fig. 4.4 Block Diagram for Prediction stage

Chapter No. 5

IMPLEMENTATION

5.1 Overview of the Technologies Used

Tensorflow

TensorFlow is a machine learning framework and is the most popular deep learning tool. It uses Python to provide a convenient front-end API for building applications with the framework.

Tkinter

Tkinter is a Python building to the Tk GUI toolkit. It is standard Python interface to the Tk GUI toolkit and is Python's conventional GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.

NumPy

NumPy is a library for Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions to operate on these arrays.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general purpose GUI toolkits like Tkinter.

OpenCV3

OpenCV is a library of programming functions mainly aimed at real time computer vision. It was originally developed by Intel. The library is cross-platform and free for use under the open-source BSD license.

Keras

Keras is a high-level neural networks API, which is written in Python and capable of running on top of TensorFlow. It allows fast and easy prototyping and supports convolutional networks and recurrent networks, as well as combinations of the two.

Pandas

Pandas makes importing and analyzing data much easier. It builds on packages like matplotlib and NumPy to give a convenient way to do data analysis and visualization work.

Pydicom

Pydicom is a python package for working with DICOM files such as medical images. Pydicom makes it easy to read complex files into natural pythonic structures that are easy to manipulate.

h5py

The h5py package is a pythonic interface to the HDF5 data format. It is used to store huge amounts of numerical data and is used to manipulate the data from NumPy.

Pydot

Pydot is an interface to Graphviz and can parse and dump into the DOT language used by GraphViz.

GraphViz

GraphViz is an open source graph visualization software. It has several main graph layout programs and it has web and interactive graphical interfaces, and auxiliary tools, libraries, and language bindings.

Graphical User Interface (GUI)

Fig. 5.1 GUI Welcome Screen

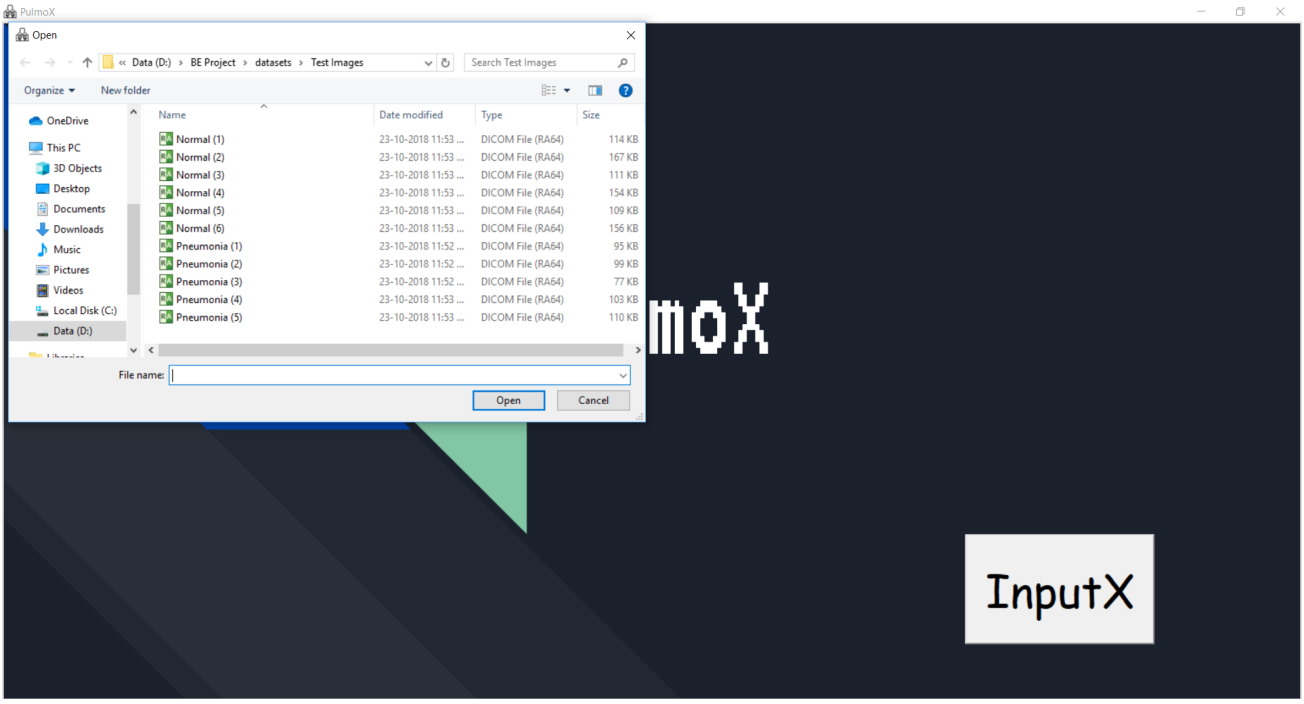


Fig. 5.2 GUI Browse Screen

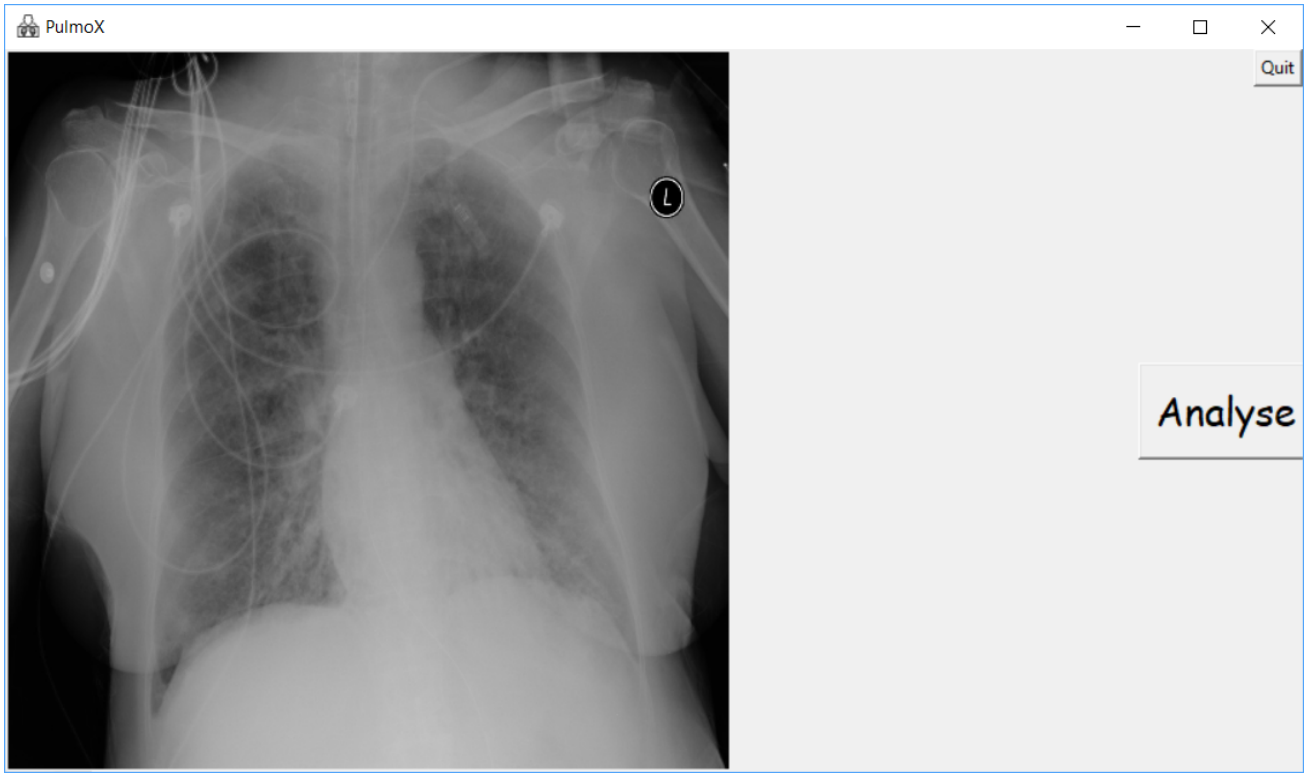


Fig. 5.3 GUI Loaded Image

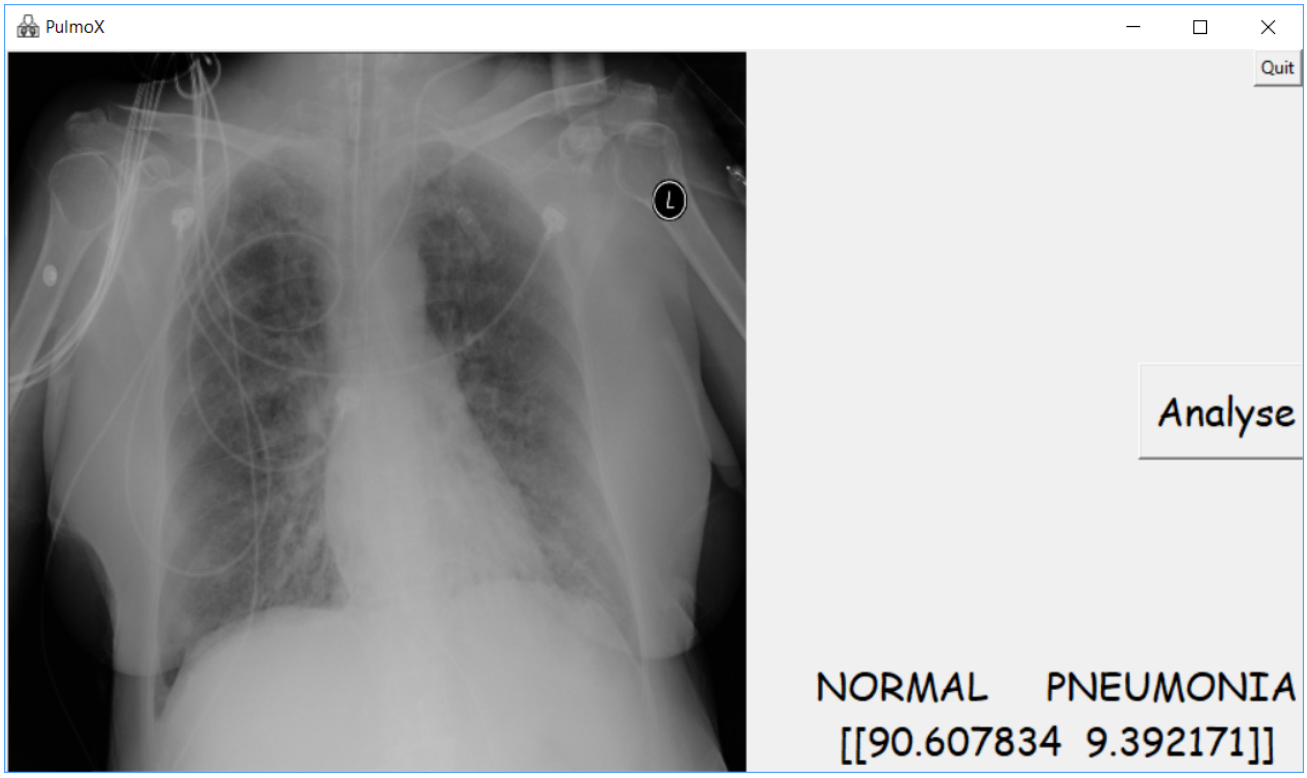


Fig. 5.4 GUI Analyzed Output for X-ray with No Pulmonary Abnormality

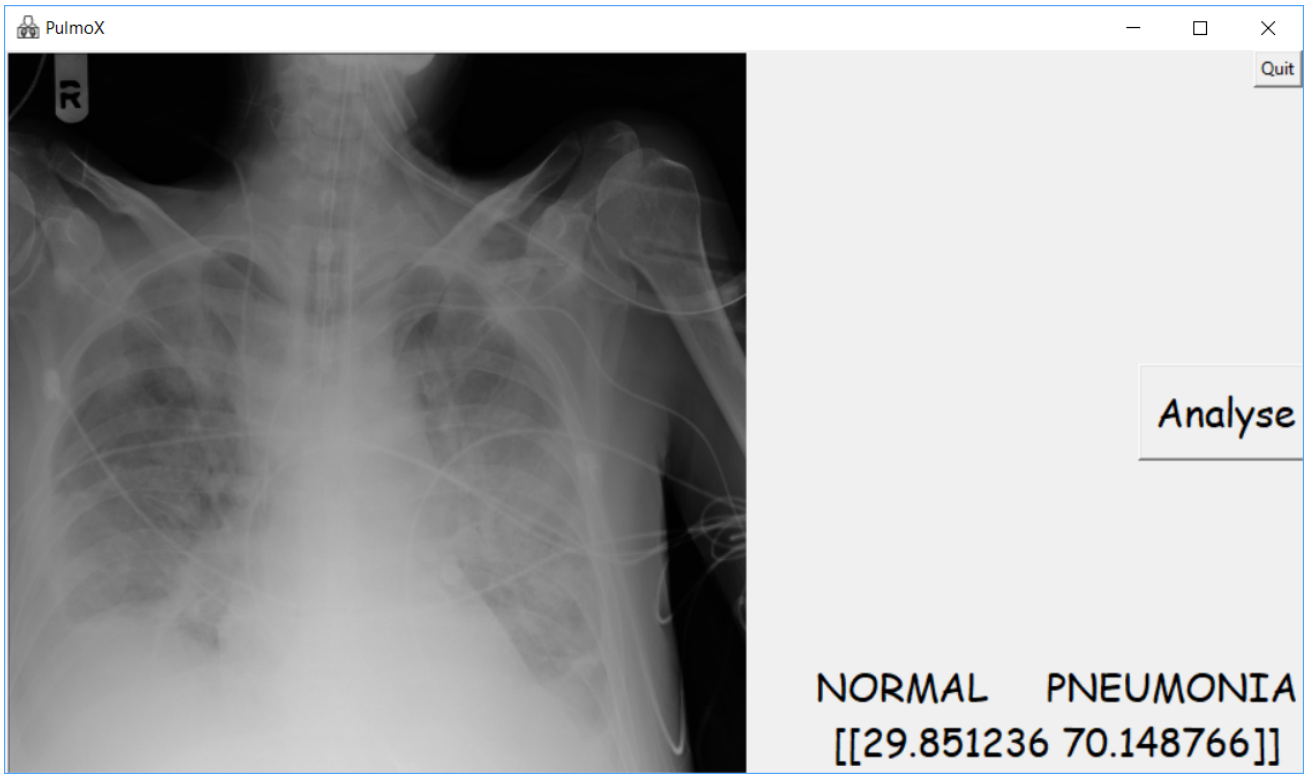


Fig. 5.5 GUI Analyzed output for X-ray with Pulmonary Abnormality

Chapter No. 6

TESTING

Testing was done on seven model structures with three different optimizer functions on 1334 DICOM Images.

Parameters used to analyze model performances are:

Accuracy:

Calculates how often the actual labels match the predictions of the model. It creates two variables total and count which computes the total predictions that match the label and returns accuracy as total divided by count where total is the number of correct predictions and count is the total number of predictions.

Loss:

Loss determines how far the predicted values deviate from the actual values in the training data.

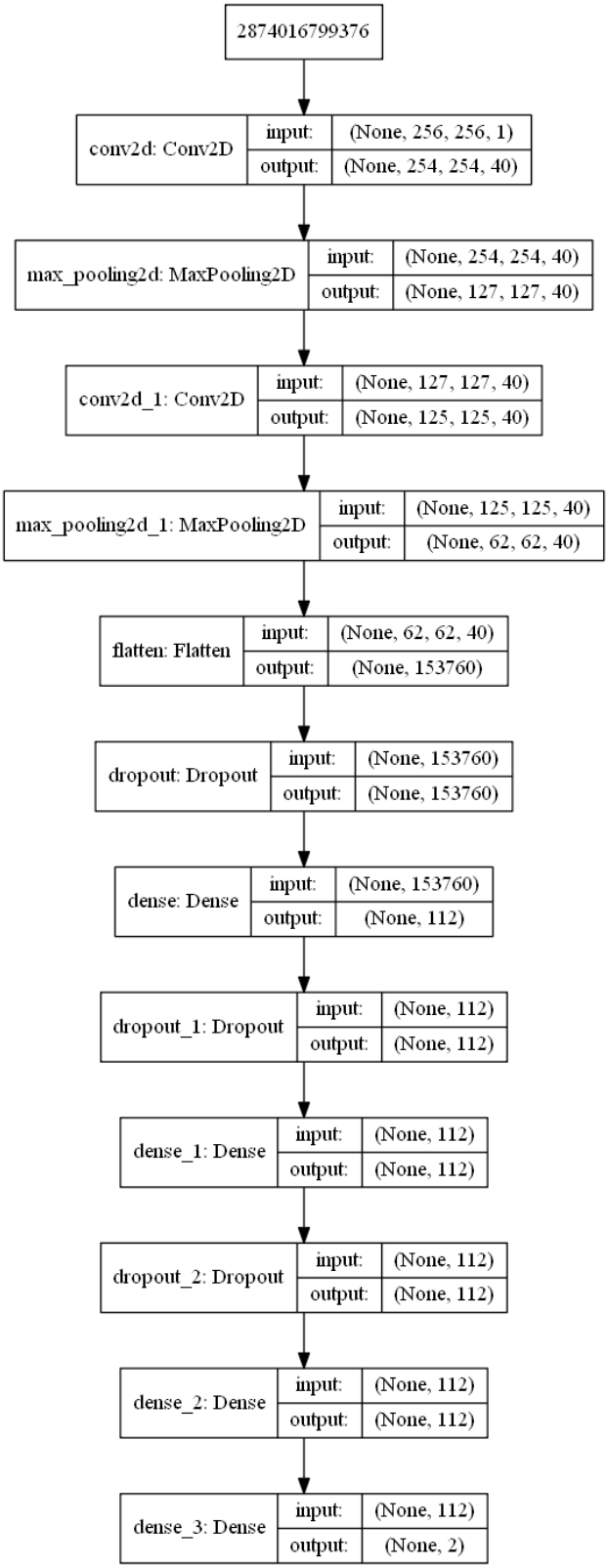


Fig. 6.1 Structure of Model 1

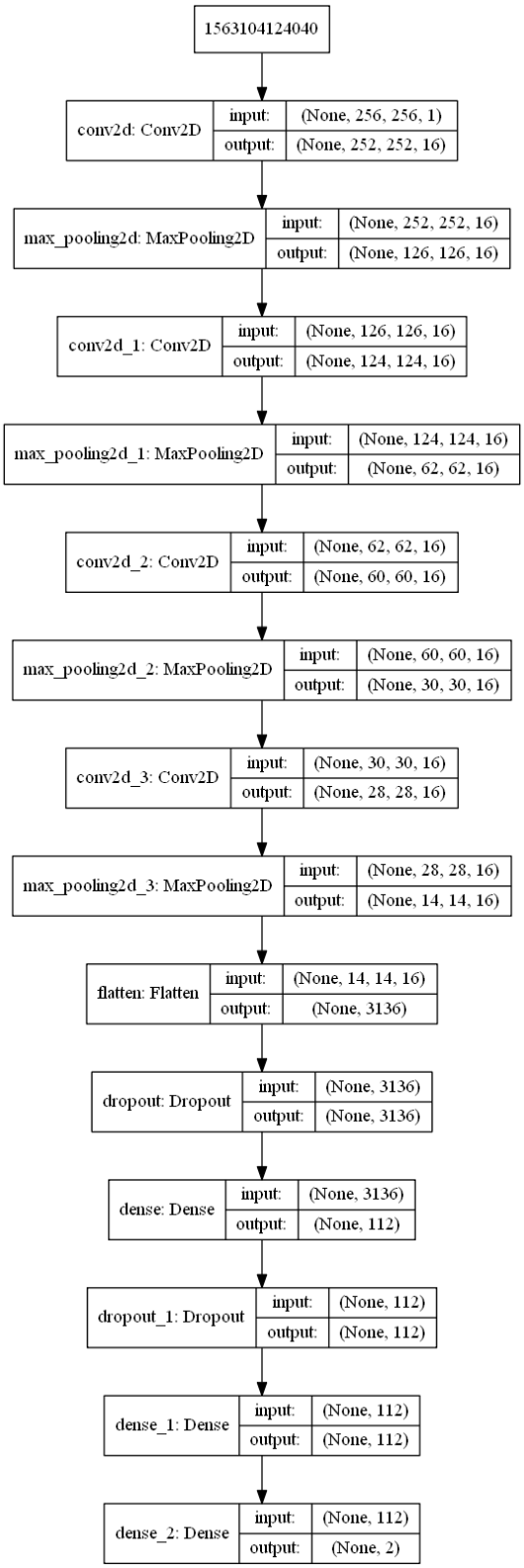


Fig. 6.2 Structure of Model 2

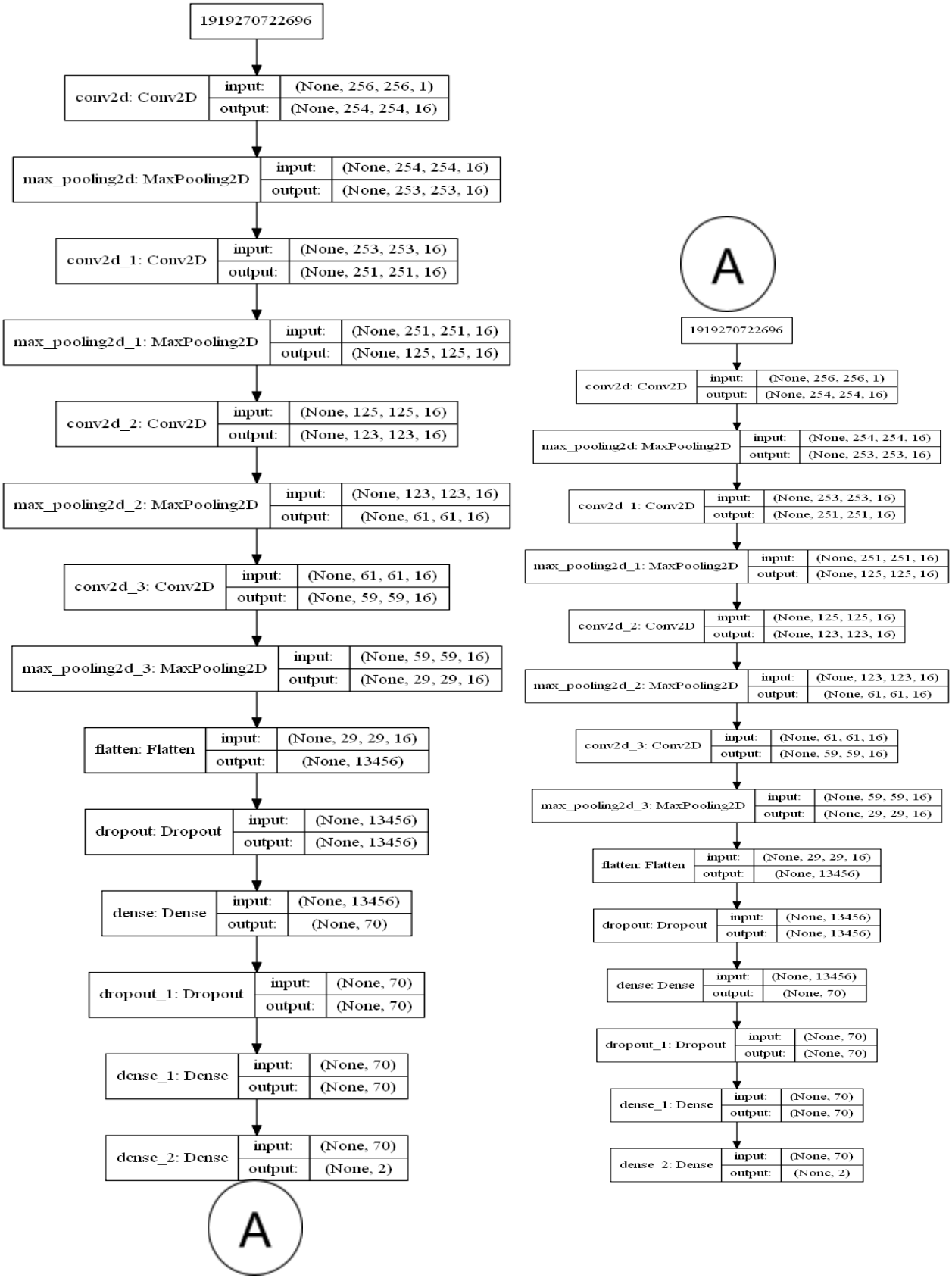


Fig. 6.3 Structure of Model 3

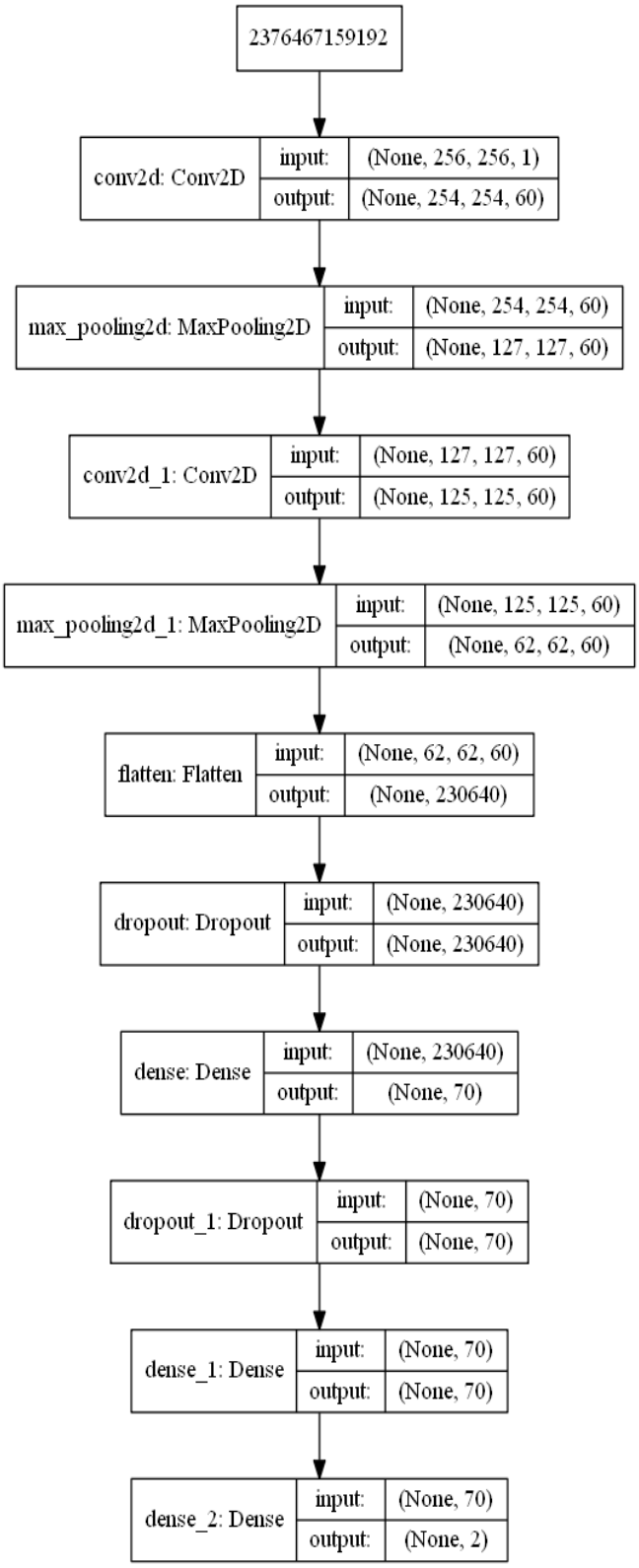
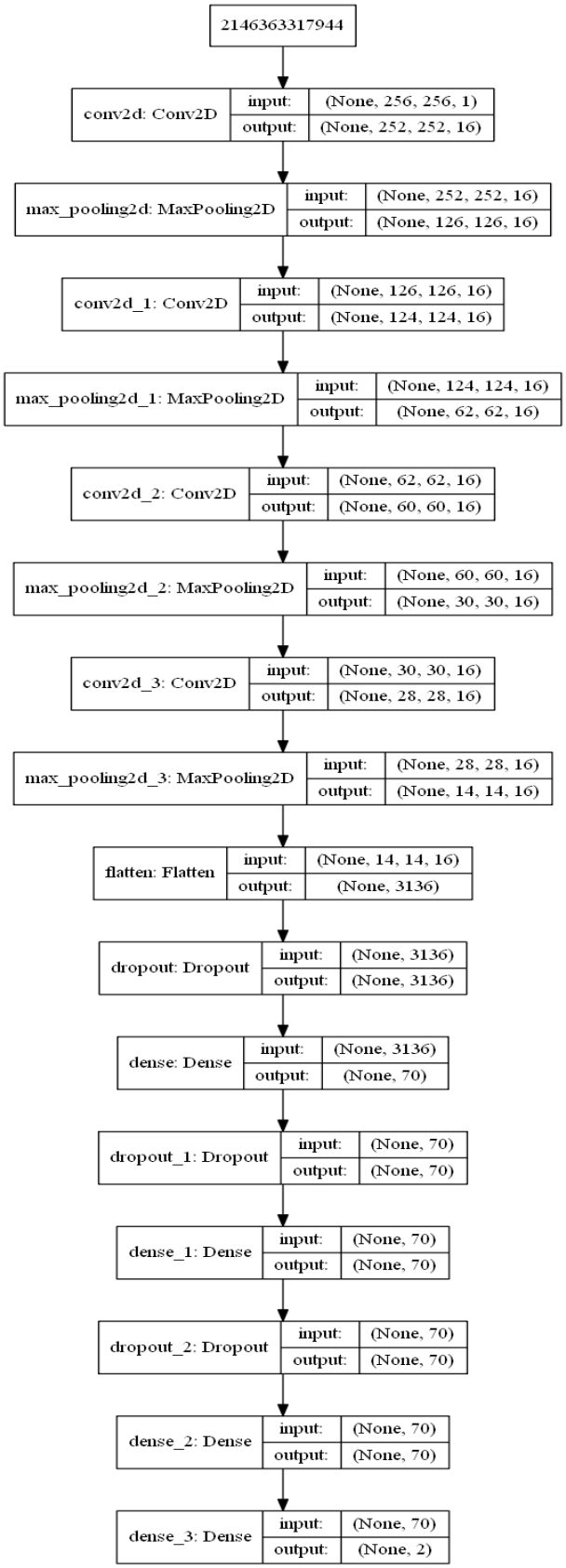


Fig. 6.4 Structure of Model 4

Fig. 6.5 Structure of Model 5


2146363317944
conv2d: Conv2D
input: (None, 256, 256, 1)
output: (None, 252, 252, 16)
max_pooling2d: MaxPooling2D
input: (None, 252, 252, 16)
output: (None, 126, 126, 16)
conv2d_1: Conv2D
input: (None, 126, 126, 16)
output: (None, 124, 124, 16)
max_pooling2d_1: MaxPooling2D
input: (None, 124, 124, 16)
output: (None, 62, 62, 16)
conv2d_2: Conv2D
input: (None, 62, 62, 16)
output: (None, 60, 60, 16)
max_pooling2d_2: MaxPooling2D
input: (None, 60, 60, 16)
output: (None, 30, 30, 16)
conv2d_3: Conv2D
input: (None, 30, 30, 16)
output: (None, 28, 28, 16)
max_pooling2d_3: MaxPooling2D
input: (None, 28, 28, 16)
output: (None, 14, 14, 16)
flatten: Flatten
input: (None, 14, 14, 16)
output: (None, 3136)
dropout: Dropout
input: (None, 3136)
output: (None, 3136)
dense: Dense
input: (None, 3136)
output: (None, 70)
dropout_1: Dropout
input: (None, 70)
output: (None, 70)
dense_1: Dense
input: (None, 70)
output: (None, 70)
dropout_2: Dropout
input: (None, 70)
output: (None, 70)
dense_2: Dense
input: (None, 70)
output: (None, 70)
dense_3: Dense
input: (None, 70)
output: (None, 2)

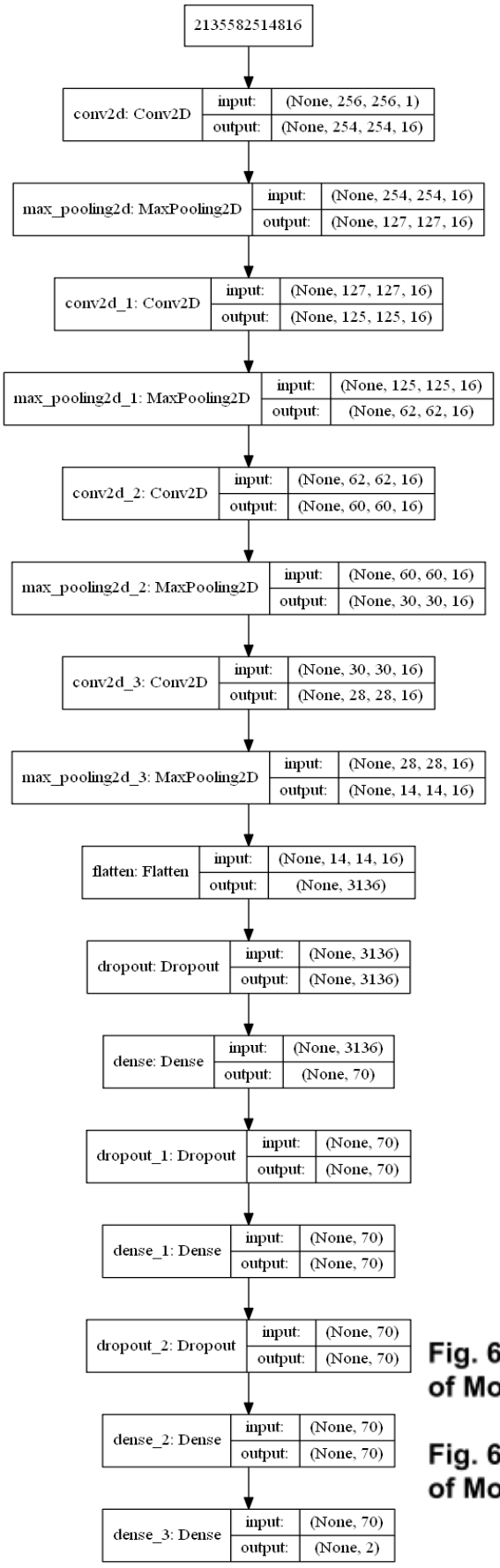
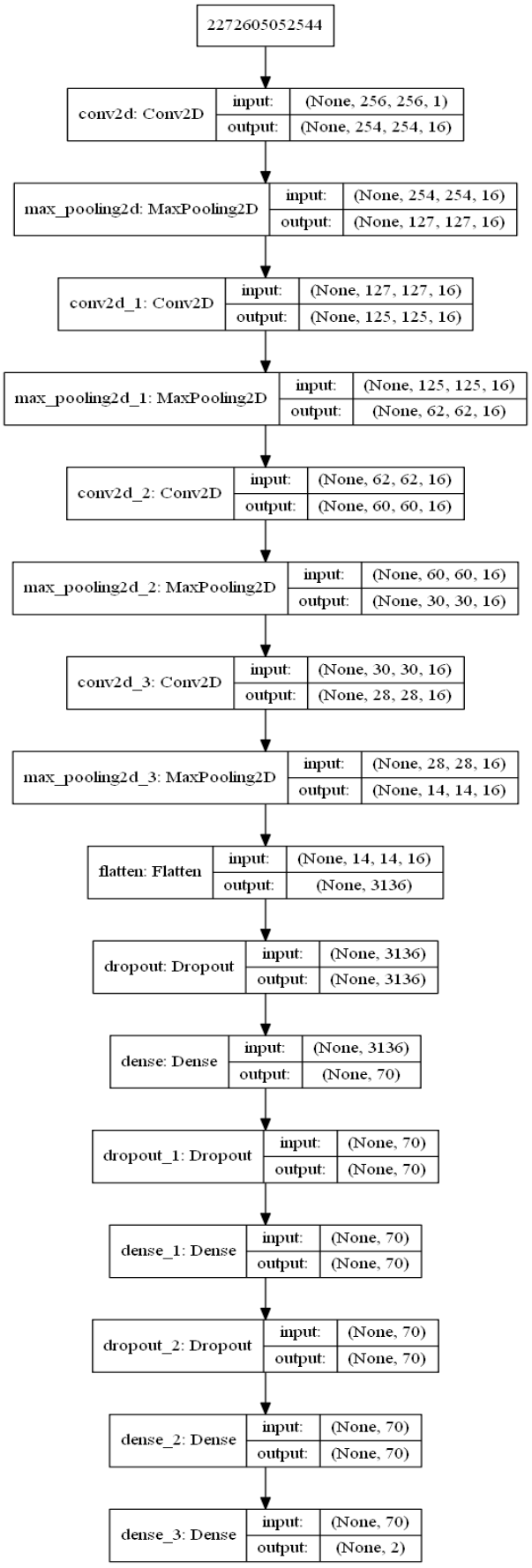


Fig. 6.6 Structure of Model 6

Fig. 6.7 Structure of Model 7


2272605052544
conv2d: Conv2D
input: (None, 256, 256, 1)
output: (None, 254, 254, 16)
max_pooling2d: MaxPooling2D
input: (None, 254, 254, 16)
output: (None, 127, 127, 16)
conv2d_1: Conv2D
input: (None, 127, 127, 16)
output: (None, 125, 125, 16)
max_pooling2d_1: MaxPooling2D
input: (None, 125, 125, 16)
output: (None, 62, 62, 16)
conv2d_2: Conv2D
input: (None, 62, 62, 16)
output: (None, 60, 60, 16)
max_pooling2d_2: MaxPooling2D
input: (None, 60, 60, 16)
output: (None, 30, 30, 16)
conv2d_3: Conv2D
input: (None, 30, 30, 16)
output: (None, 28, 28, 16)
max_pooling2d_3: MaxPooling2D
input: (None, 28, 28, 16)
output: (None, 14, 14, 16)
flatten: Flatten
input: (None, 14, 14, 16)
output: (None, 3136)
dropout: Dropout
input: (None, 3136)
output: (None, 3136)
dense: Dense
input: (None, 3136)
output: (None, 70)
dropout_1: Dropout
input: (None, 70)
output: (None, 70)
dense_1: Dense
input: (None, 70)
output: (None, 70)
dropout_2: Dropout
input: (None, 70)
output: (None, 70)
dense_2: Dense
input: (None, 70)
output: (None, 70)
dense_3: Dense
input: (None, 70)
output: (None, 2)

Observations:

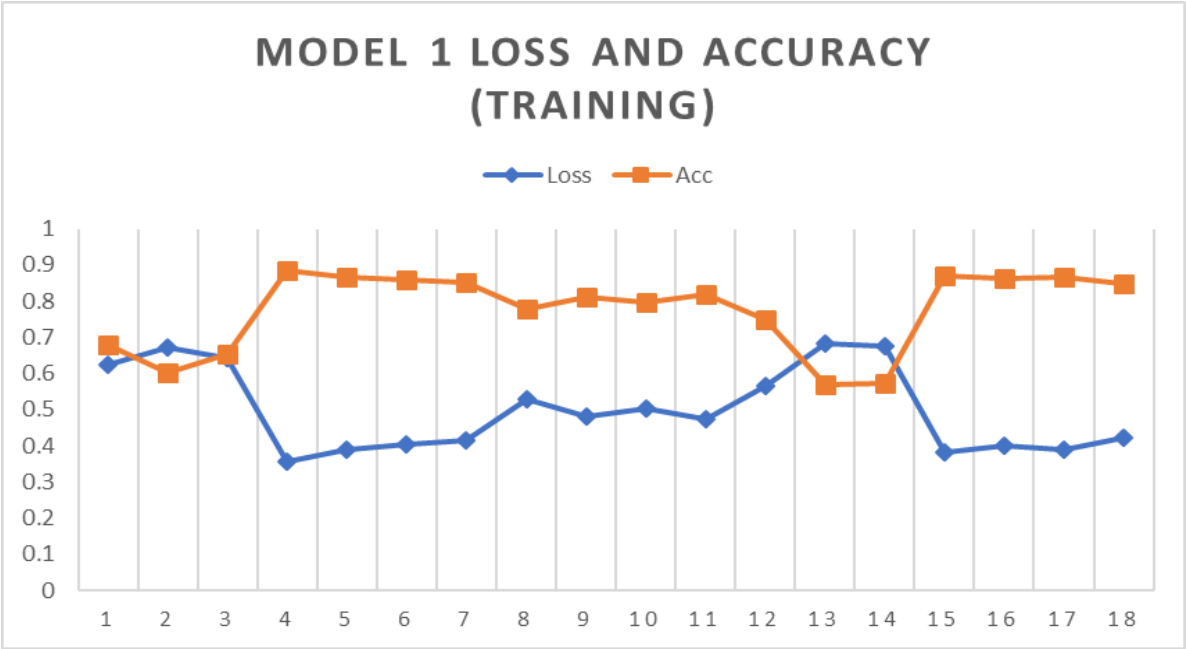


Fig. 6.8 Model 1 Training Loss And Accuracy

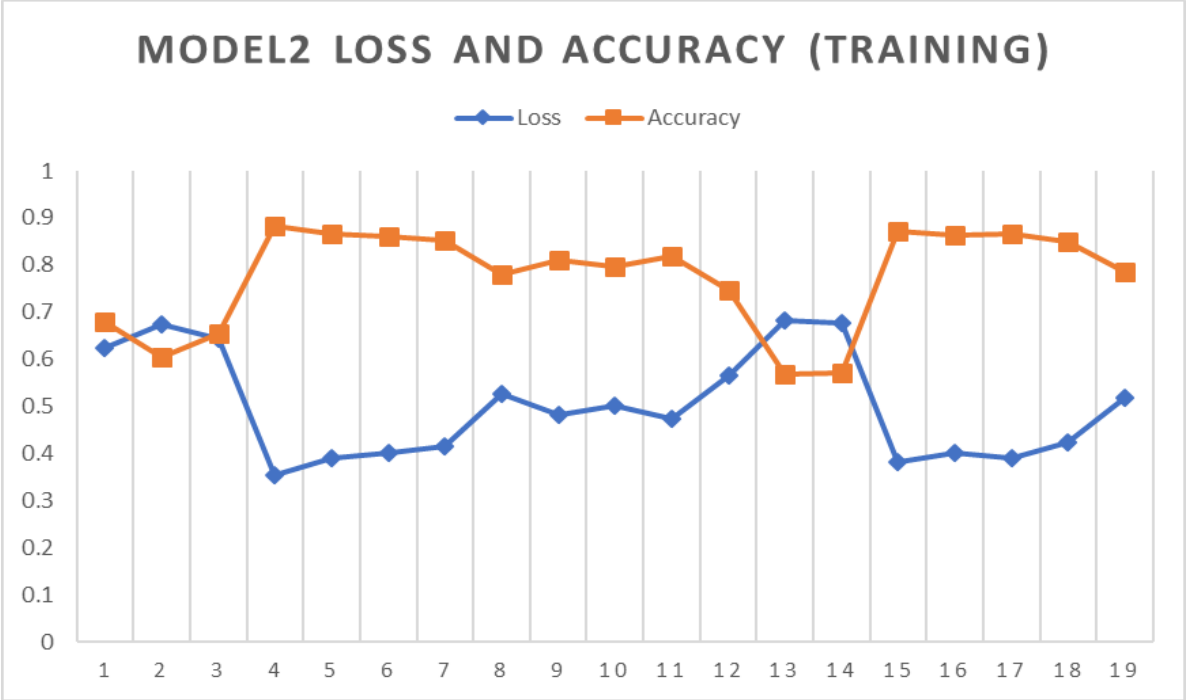


Fig. 6.9 Model 2 Training Loss and Accuracy

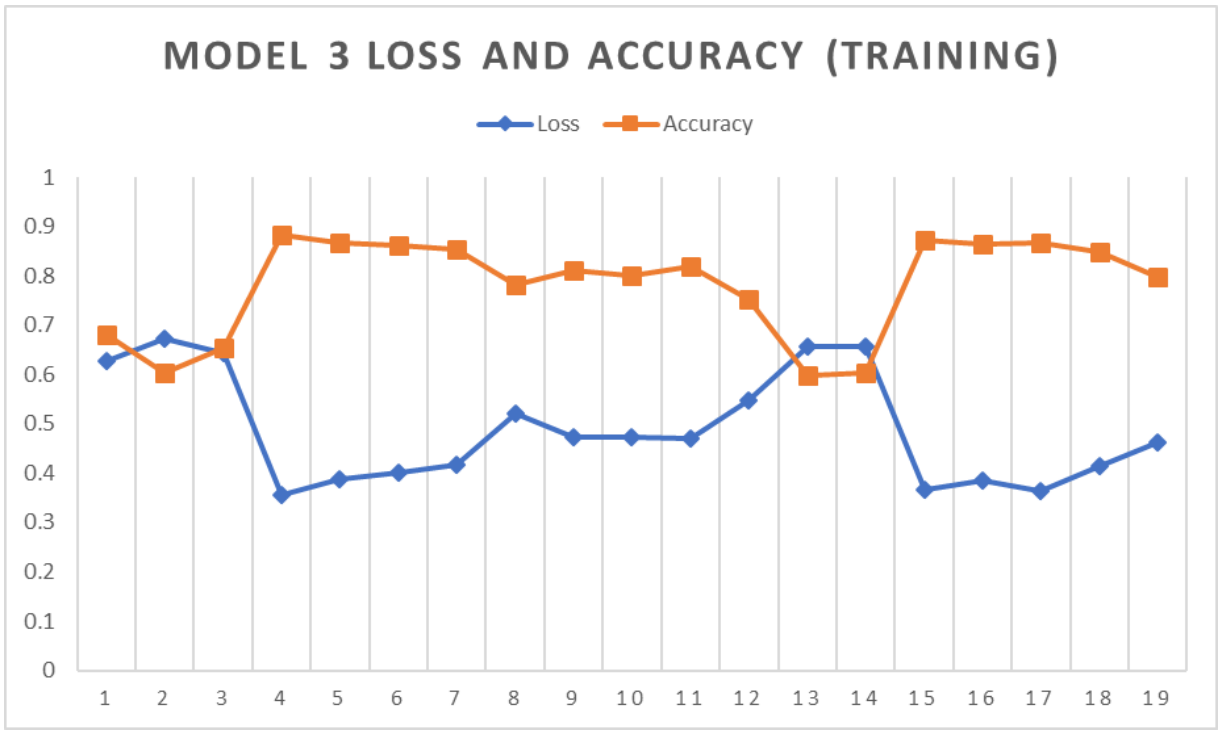


Fig. 6.10 Model 3 Training Loss And Accuracy

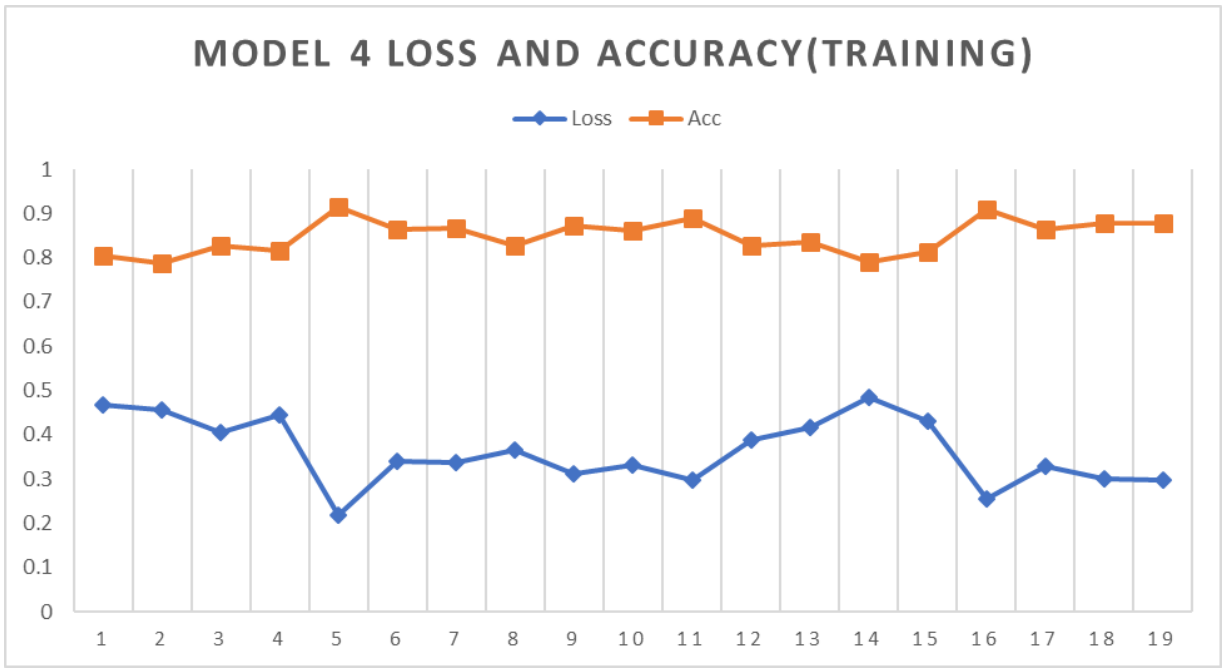


Fig. 6.11 Model 4 Training Loss And Accuracy

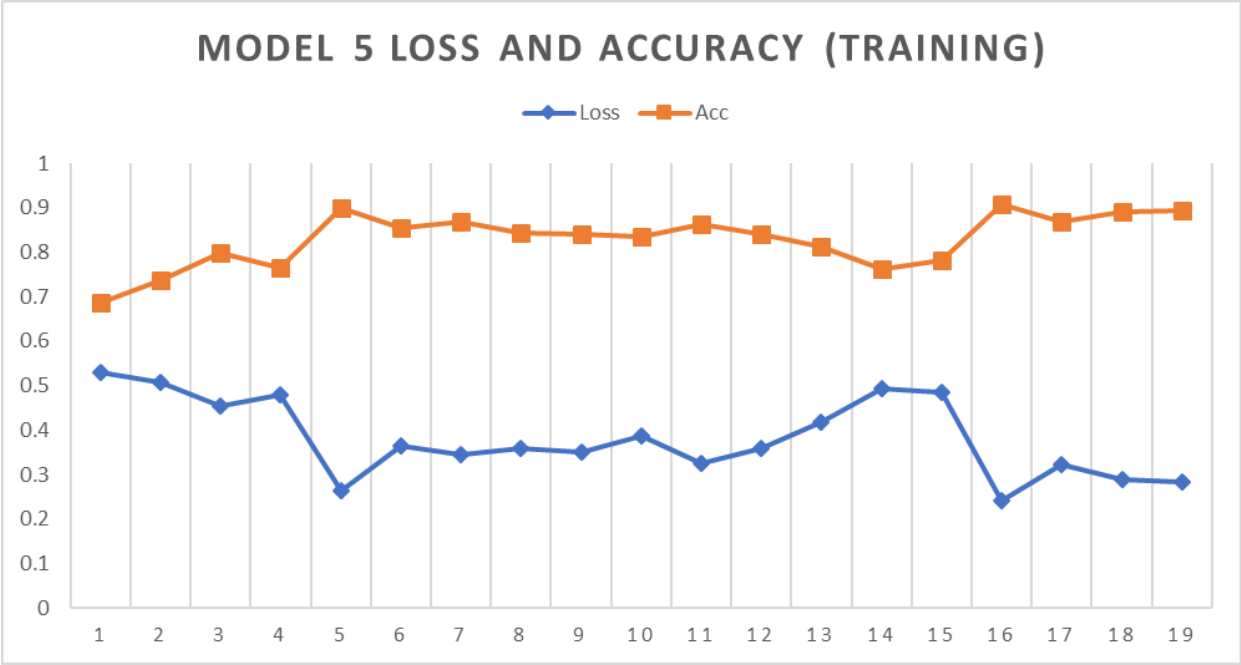


Fig. 6.12 Model 5 Training Loss And Accuracy

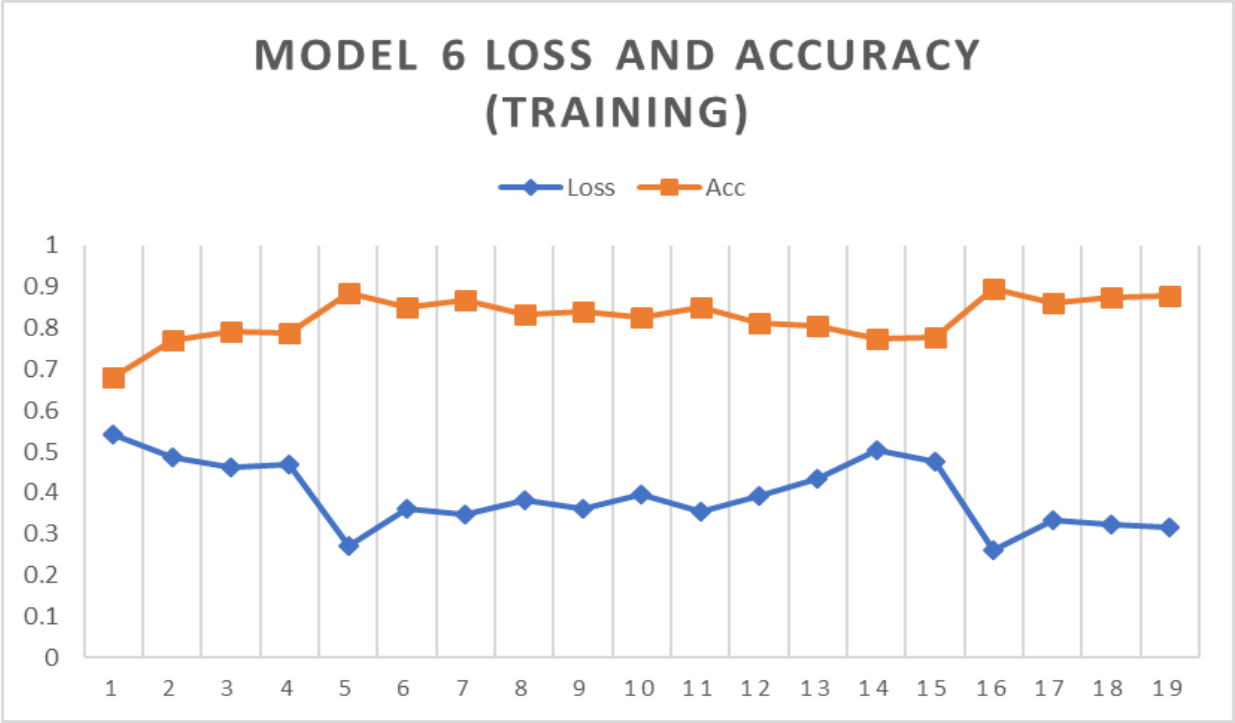


Fig. 6.13 Model 6 Training Loss And Accuracy

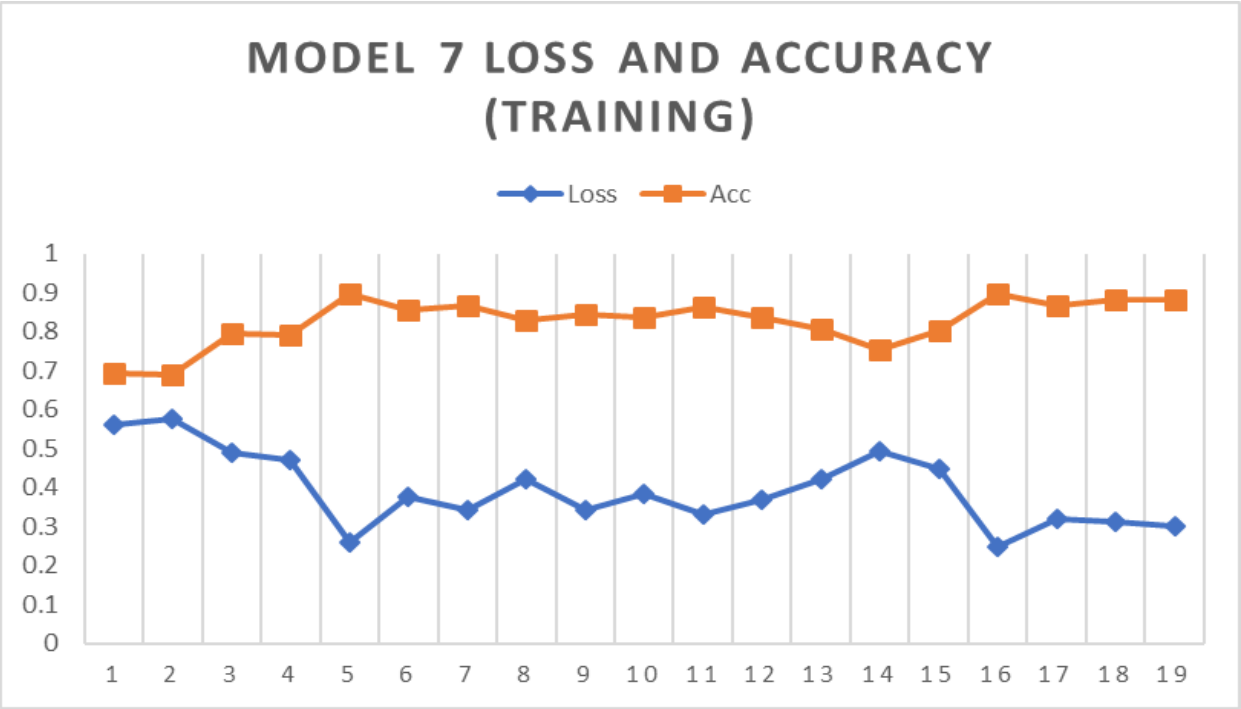


Fig. 6.14 Model 7 Training Loss And Accuracy

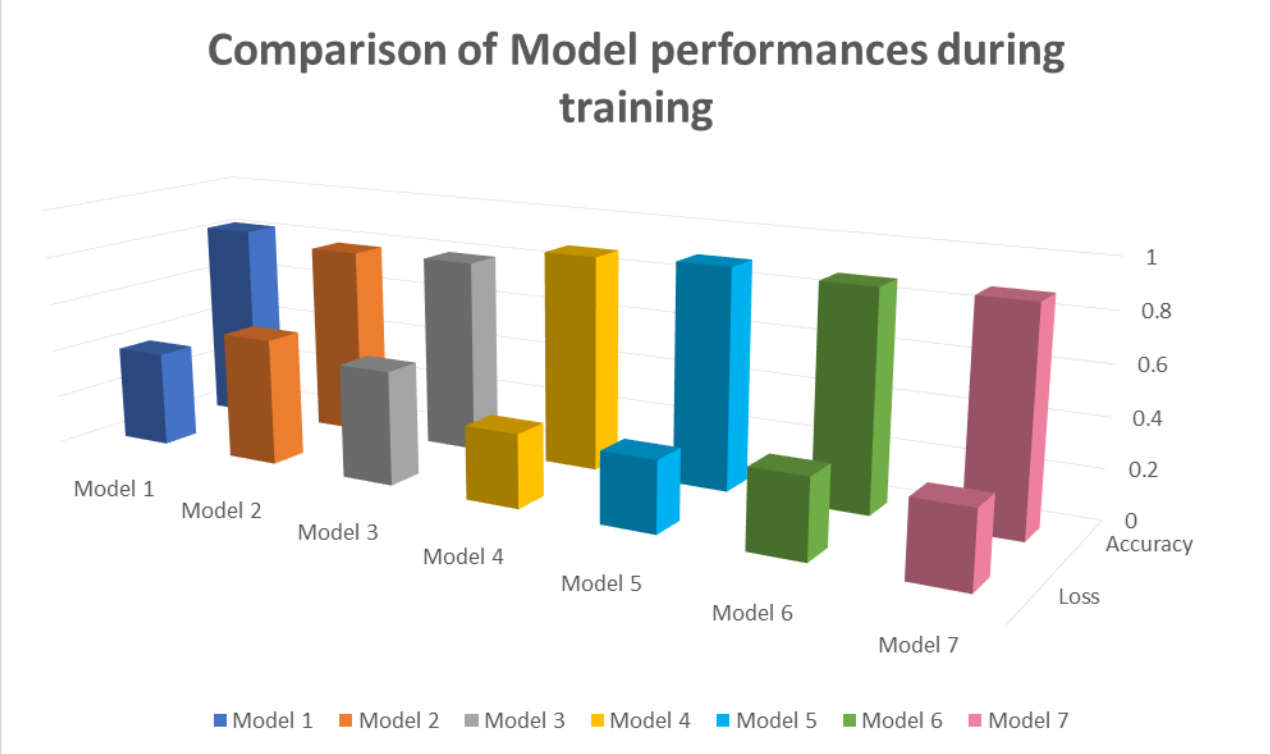


Fig. 6.15 Comparison of Model performances during training

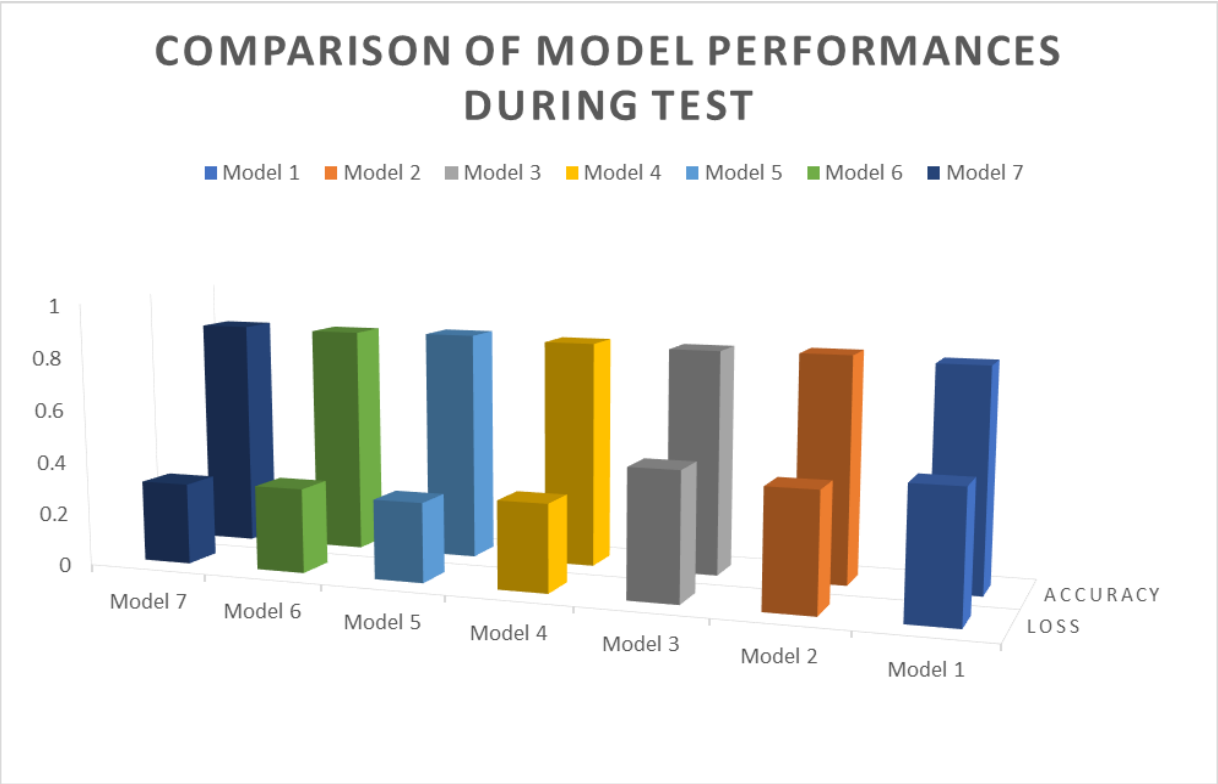


Fig. 6.16 Comparison of Model performances during test

Models	Convolution	ConvLayers	Kernel	Pooling	Pool_size	Dense	No. of Neurons	Dropout	Optimizer	Tr_loss	Tr_acc	Ts_loss	Ts_acc
Model 1	2	40,40	3,3	maxPool	2,2	4	112,112,112,2	30%	adam	0.283597569	0.892053973	0.484120443	0.829835083
Model 2	4	16,16,16,16	5,3,3,3	maxPool	2,2	3	112,112,2	30%	adam	0.519089331	0.785488958	0.442265492	0.847076462
Model 3	4	16,16,16,16	3,3,3,3	maxPool	2,2	3	70,70,2	30%	adam	0.46258016	0.796529969	0.480476893	0.843328336
Model 4	2	60,60	3,3	maxPool	2,2	3	70,70,2	30%	adam	0.296158261	0.877061469	0.329268639	0.851574213
Model 5	4	16,16,16,16	3,3,3,3	maxPool	2,2	4	70,70,70,2	30%	adam	0.28459209	0.893553223	0.300162118	0.865067466
Model 6	4	16,16,16,16	3,3,3,3	maxPool	2,2	4	70,70,70,2	30%	adagrad	0.315410157	0.876311844	0.319970452	0.85982009
Model 7	4	16,16,16,16	3,3,3,3	maxPool	2,2	4	70,70,70,2	30%	adadelta	0.301175343	0.88305847	0.307514041	0.865817092

Fig. 6.17 model comparisons

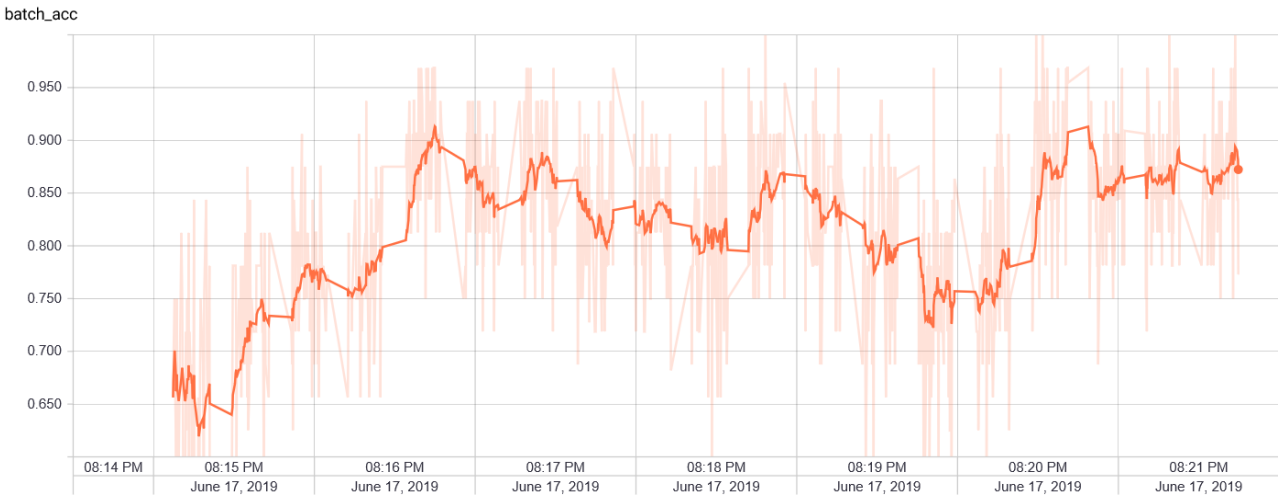


Fig. 6.18 Batch Accuracy for Model 5 Training

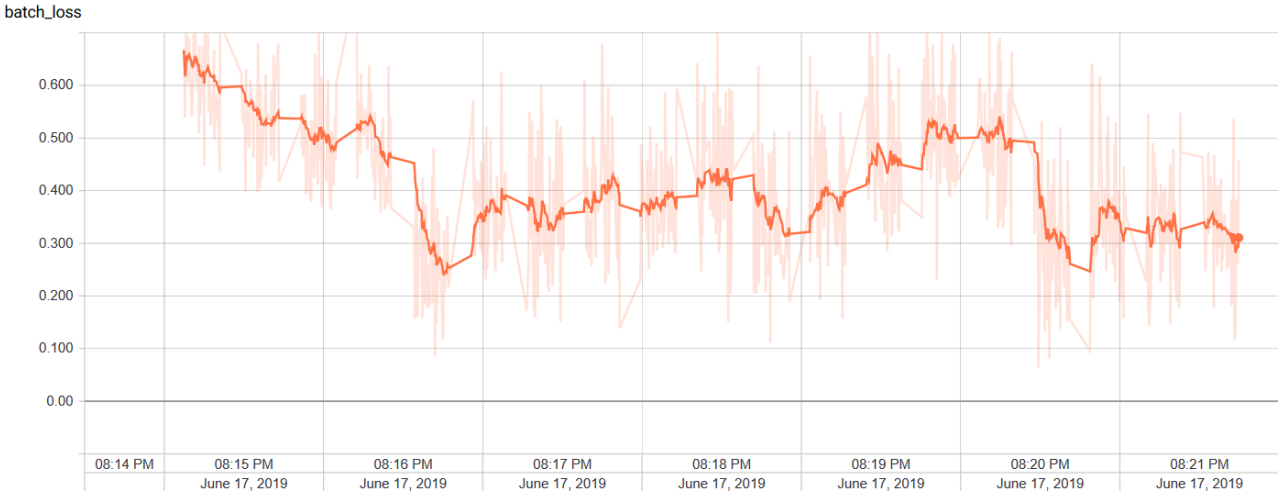


Fig. 6.19 Batch Loss for Model 5 Training

Chapter No. 7

CONCLUSION

Pneumonia accounts for significant portion of mortality, even after advances in medical field. The most common examination technique is chest X-rays; considering all the basic medical tests.

So, the PulmoX software is designed and implemented to aid the radiologist while examining doubtful X-rays. The software assists by providing the probability of existence of pneumonia. The software is based on a neural network model that is 86.51% accurate.

A total of 7 different models were created among which Model 5 with the following configuration performed the best.

- Layers of Convolution with 16 filters and kernel size (3,3) along with Maxpooling filter of size (2,2)
- Three Dense layers with 70 Neurons each activated by RELU activation functions
- One Dense layer with 2 Neurons activated by Softmax activation function
- Optimizer: Adam

This configuration was tested against 1334 test images of DICOM format.

The software computes the probability of X-Ray being normal or affected with Pneumonia.

Future Scope

The further steps would be to improve the accuracy of the model and decrease the loss.

This would be done by experimenting with more model structures and combinations of optimizers and activation functions.

Another feature we would like to add is providing bounding boxes to mark the areas of infection to show thea radiologist which are the critical areas.

Detection of more possible pulmonary diseases in the chest such as enlarged cardio mediastinum, edema, pleural effusion etc.

REFERENCES

Papers:

[1] Saad ALBAWI, Tareq Abed MOHAMMED, Saad AL-ZAWI "**Understanding of a Convolutional Neural Network**", ICET2017, Antalya, Turkey

[2] Mihalj Bakator and Dragica Radosav, "**Deep Learning and Medical Diagnosis: A Review of Literature**", University of Novi Sad, Serbia

[3] Rahib H. Abiyev and Mohammad Khaleel Sallam Ma'aitah, "**Deep Convolutional Neural Networks for Chest Diseases Detection**", Hindawi Journal of Healthcare Engineering, 2018

[4] Jeremy Irvin, Pranav Rajpurkar, Michael Ko,Yifan Yu, Silviana Ciurea-Ilcus,Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren,y Andrew Y. Ng, "**CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison**", Department of Computer Science, Department of Medicine, Department of Radiology, Stanford University

[5] Michal Horný, "**Bayesian Networks**", Boston University School of Public Health

[6] Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V, … Ng, A. Y. (2012). "**Large Scale Distributed Deep Networks**" NIPS 2012: Neural Information Processing Systems

[7] Zhang, S., Choromanska, A., & LeCun, Y. (2015). "**Deep learning with Elastic Averaging SGD**", Neural Information Processing Systems Conference (NIPS 2015)

[8] Dinggang Shen, Guorong Wu, and Heung-Il Suk, "**Deep Learning in Medical Image Analysis**", Annual Review of Biomedical Engineering

[9] Geert Litjens, Thijs Kooi , Babak Ehteshami Bejnordi , Arnaud Arindra Adiyoso Setio , Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, Clara I. Sánchez, "**A survey on deep learning in medical image analysis**", G. Litjens et al. / Medical Image Analysis

[10] Keiron O'Shea and Ryan Nash, "**An Introduction to Convolutional Neural Networks**", Department of Computer Science, Aberystwyth University, Ceredigion,2 School of Computing and Communications, Lancaster University, Lancashire

[11] Cocklin M, Gourlay A, Jackson P, Kaye G, Miessler M, Kerr I, Lams P, "**An image processing system for digital chest X-ray images**", Comput Programs Biomed. 1984.

[12] Murphy K. , " **A Brief Introduction to Graphical Models and Bayesian Networks.**" (1998)

[13] A. Helwan, D. U. Ozsahin, R. Abiyev, and J. Bush, "**One-year survival prediction of myocardial infarction**," International Journal of Advanced Computer Science and Applications, vol. 8, no. 6, pp. 173–178, 2017.

[14] Ben Heil, "**Convolutional neural networks for detecting lung cancer"**

[15] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy, Ding1Aarti, Bagul1Robyn, L. Ball, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, Andrew Y. Ng, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rayswith Deep Learning"

Websites:

https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f

https://medium.com/datadriveninvestor/convolutional-neural-networks-e0d25a3799f6

https://machinelearningmastery.com/crash-course-convolutional-neural-networks/

https://medium.com/machine-learning-world/convolutional-neural-networks-for-all-part-ii-b4cb41d424fd

https://medium.com/x8-the-ai-community/cnn-9c5e63703c3f

https://medium.com/@phidaouss/convolutional-neural-networks-cnn-or-convnets-d7c688b0a207

https://medium.com/datadriveninvestor/deep-learning-deep-belief-network-dbn-ab715b5b8afc

https://www.youtube.com/watch?v=5v1JnYv_yWs&list=PLtBw6njQRU-rwp5__7C0oIVt26ZgjG9NI

https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204

https://www.verywellhealth.com/pneumonia-overview-3156882

https://www.udacity.com/

https://www.coursera.org/

Libraries:

https://keras.io/

https://www.tensorflow.org/

https://www.numpy.org/devdocs/

http://pandas.pydata.org/pandas-docs/stable/

http://docs.h5py.org/en/stable/

https://pydicom.github.io/pydicom/stable/getting_started.html

https://docs.anaconda.com/

https://tkdocs.com/tutorial/

https://pillow.readthedocs.io/

https://pydotplus.readthedocs.io/reference.html

https://graphviz.gitlab.io/documentation/

Plagiarism Checker X Originality Report

Similarity Found: 23%

Date: Thursday, June 20, 2019
Statistics: 1639 words Plagiarized / 7277 Total words
Remarks: Medium Plagiarism Detected - Your Document needs Selective Improvement.

ABSTRACT Challenge in detecting pneumonia using chest X-rays is that, pneumonia in most cases is not confined to one particular area. It is difficult to locate the consolidations on X-rays. Radiologists have to examine a lot of medical images and may miss some features that are essential for the diagnosis.

This may happen mostly from human fatigue or strain from examining many patients. Our software can potentially overcome these drawbacks by using deep learning to determine if the patient is infected with pneumonia. In this way, the software will use the chest X-ray images and conclude with the probability of pneumonia infection, which then will be examined by the radiologist.

The major outcome of this project should be a successful setup to make the best use of the computerized system and its usage in medicine.