

COVID DETECTION USING DEEP LEARNING TECHNIQUE

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in Biomedical Engineering.

By

DEEPAK V S (37240019)

JEEVA J (37240034)



**DEPARTMENT OF BIOMEDICAL ENGINEERING
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC | 12B Status by UGC | Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Deepak V S (37240019)**, **Jeeva J (37240034)** who have done the project work as a team who carried out the project entitled “**Covid Detection Using Deep Learning Technique**” under my supervision from **October 2020 to March 2021**

Dr. J. PREM KUMAR, M.Sc., Ph.D.,
Internal Guide

Dr.T.SUDHAKAR, M.Sc., Ph.D.,
Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **Deepak V S (37240019)** and **Jeeva J (37240034)** hereby declare that the Project Report entitled “**COVID Detection using Deep Learning Technique**” done by us under the guidance of **Dr. J. PremKumar**, Department of Biomedical Engineering, Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Biomedical Engineering.

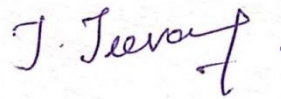
DATE:

PLACE:

1.



2.



SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T. Sudhakar**, M.Sc., Ph.D., **Dr. J. Prem Kumar**, M.Sc., Ph.D., **Heads of the Department, Dept. of Biomedical Engineering** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. J. Prem Kumar**, M.Sc., Ph.D., **Department of Biomedical Engineering** for his valuable guidance, suggestions and constant encouragement paving the way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways for the completion of the project

ABSTRACT

Deep neural networks herald remarkable accuracy in different applications, but the performance also relies on heavy network architectures. During In this way, a transportable student network with much fewer parameters will achieve a level of accuracy that is comparable to that of an instructor network. However, for practical applications, the learned student network's robustness to perturbation is also significant. Existing mechanisms for teacher-student learning are primarily focused on ratios of precision and compression, but ignore the robustness. during this paper, we make the model to predict the utmost efficiency using the mix of three different architecture networks where two to 3 blocks are trained together to urge the very best accuracy. the employment of novel inception model will increase the performance of the model prediction and its robustness. Here the various combination of blocks are trained in parallel and at last concatenation of the blocks are done to improvise the efficiency of the model which successively are going to be accustomed predict the presence of COVID-19 and tuberculosis within the patients effectively in low cost.

Keywords: COVID, Tuberculosis, Neural network, Deep learning, Teacher Student framework

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
1.1	Deep Learning Layers	3
1.2	Deep Learning Process	4
1.3	Types of Deep Learning Networks	5
1.4	CNN	7
1.5	Type of CNN	8
1.6	ReactJS architecture	9
4.1	System Architecture	21
4.2	Data augmentation flow diagram	26
4.3	VGG-16 architecture diagram	27
4.4	A Residual Block of Deep Residual Network	28
4.5	Intuitive representation of Residual Function	29
4.6	Architecture diagram of ResNet50	31
4.7	Inception layer	33
4.8	Architecture diagram of mongoDB	34

LIST OF SYMBOLS AND ABBREVIATIONS

CNN	Convolutional Neural Network
RNNS	Recurrent Neural Network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
ReLU	Rectified Linear Unit
SARSA	State-Action-Reward-State-Action
DDPG	Deep Deterministic Policy Gradient

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO.
	ABSTRACT	I
	LIST OF FIGURES	II
	LIST OF SYMBOLS AND ABBREVIATION	III
1	INTRODUCTION	1
	1.1 OUTLINE OF THE PROJECT	1
	1.2 IMAGE PROCESSING	2
2	LITERATURE REVIEW	10
3	AIM AND SCOPE	18
	3.1 AIM	18
	3.2 SCOPE	18
	3.3 EXISTING SYSTEM	18
	3.4 DISADVANTAGE OF EXISTING SYSTEM	19
	3.5 PROPOSED SYSTEM	19
	3.6 ADVANTAGE OF PROPOSED SYSTEM	20
4	MATERIALS AND METHOD	21
	4.1 SYSTEM ARCHITECTURE	21
	4.2 WORKING	22
	4.3 MODULE DESCRIPTION	22
	4.3.1 Data collection	23

	4.3.2 Data splitting and alignment	24
	4.3.3 Data augmentation	24
	4.3.4 Training architecture	26
	4.3.5 Training with the inception module	32
	4.3.6 Database Integration	33
	4.3.7 Web Development	35
	4.3.8 Software List	36
	4.3.9 Microsoft Visual Studio	36
	4.3.10 Python	37
5	RESULTS AND DISCUSSION	39
	5.1 Final results obtained	39
6	SUMMARY AND CONCLUSION	46
	6.1 CONCLUSION	46
	6.2 FUTURE WORK	46
	REFERENCES	47
	APPENDIX	49

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT

WHO named the Novel Coronavirus Disease (COVID-19) outbreak a deadly disease on March 11, 2020, and reaffirmed the decision for countries to take urgent action and rescale their response to treat, diagnose, and minimise transmission in order to save many lives. Coronavirus disease (COVID-19) is a newly discovered coronavirus that causes a communicable disease. Most people infected with the COVID-19 virus will have a mild to moderate respiratory infection that will go away without treatment. Ravneet Punia et al (2020) developed a system for detecting the novel coronavirus that relies on radiology, or X-rays. We also release a dataset for the scientific community and further growth, which was collected from various medical research hospital facilities that treat COVID-19 patients, along with the paper. Severe disease is more likely to strike the elderly and those with underlying medical issues such as disorder, diabetes, chronic respiratory illness, and cancer. Rachna Jain et al. (2020) conducted research. Deep learning is the most effective machine learning method for analysing a large number of chest x-ray images, which can have a significant effect on Covid-19 screening. When an infected person coughs or sneezes, the COVID-19 virus spreads mainly through droplets of saliva or discharge from the nose, so it's important that you also follow respiratory etiquette (for example, by coughing into a flexed elbow). In recent years there has been significant progress within the field of deep learning. Antonios Makris et al. (2020) looked into the possibility of using technology to help in the diagnosis of virus-infected patients. As a result, the ability of several state-of-the-art pre-trained convolutional neural networks to identify contaminated patients from chest X-Ray images was tested. Since the breakthrough in 2012 ImageNet competition achieved by Alex Net using five convolutional layers and three fully connected layers, a series of more advanced deep neural networks are developed to stay rewriting the record, e.g., VGGNet, Google Net, and ResNet. However, their excellent performances require the support of a large amount of computation. for example, AlexNet contains

about 232 million parameters and desires. 7.24×10^8 multiplications to process a picture with resolution of 227×227 . As a result, deep neural networks' maximum potential can only be realised on high-performance GPU servers or clusters. In contrast, the bulk of the mobile devices employed in our lifestyle usually have rigorous constraints on the storage and computational resource, which prevents them from fully taking advantages of the deep neural network. As a result, networks with lower hardware requirements but comparable accuracy are of great interest to the machine learning and computer vision communities. Deep learning-based detection and study of COVID-19 on chest X-ray images, Jain, R., Gupta, M., Taneja, S., et al.,(2021). Compressing convolutional neural networks may be achieved by vector quantization, decomposing weight matrices, and encoding with hashing tricks. Unimportant weights may be pruned to realize the identical goal by removing the subtle weights reducing the redundancy between weights within the frequency domain, and using the binary networks. Another straightforward approach is to style a compact network directly, e.g., ResNet, Exception network, and Mobile Nets. These networks are often deep and thin with fewer parameters in each layer, and also the nonlinearity of those networks are strengthened by increasing the quantity of layers, which guarantees the performance of the network. Teacher–student learning framework, introduced in knowledge distillation (KD), is one in all the foremost popular approaches to appreciate model compression and acceleration. A student network of sunshine architecture can learn better under the guidance of a significant neural network, such as GoogleNet or ResNet, that has already been well trained with massive data and computing sources as the teacher network. To inherit the benefits of the teacher network, different methods are proposed to encourage the consistency between teacher and student networks.

1.2 IMAGE PROCESSING:

Deep learning

Deep learning is a form of computer software that mimics the network of neurons in the brain. It is a sub-discipline of machine learning and It's called deep learning because it uses deep neural networks. Connected layers are used to construct deep learning

algorithms. The Input Layer is the first layer, the Output Layer is the last layer. All layers in between are known as Hidden Layers. The word deep means the network join neurons in more two layers

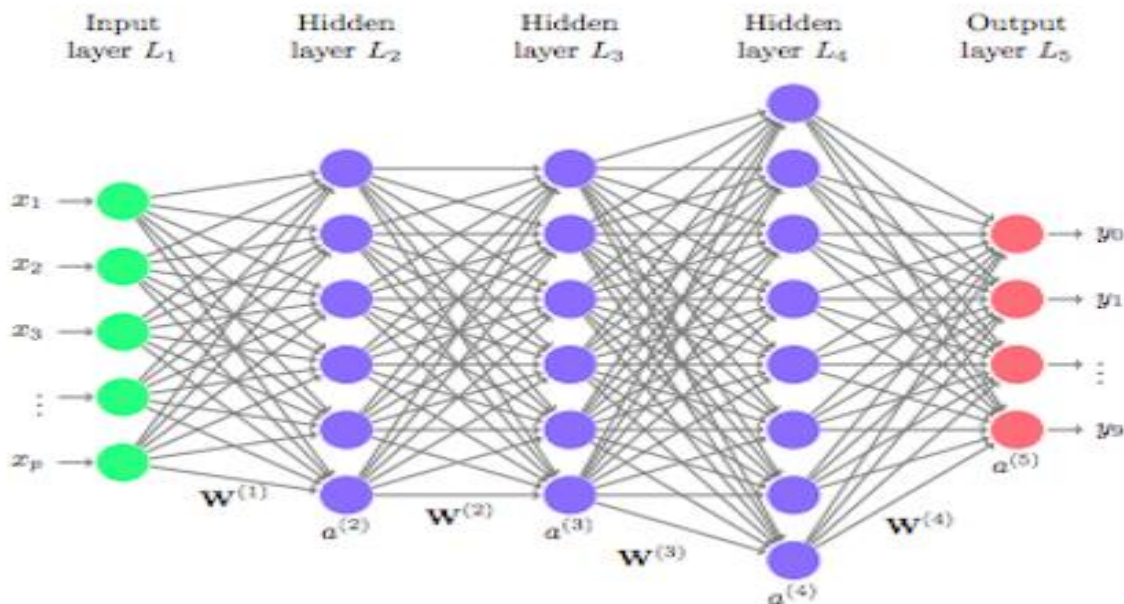


Figure 1.1 Deep Learning Layers

Each Hidden layer is composed of neurons. The neurons are connected to each other. The neuron will process and then propagate the input signal it receive layer on top of it The intensity of the signal sent to the next layer neuron is determined by the bias, weight, and activation mechanism. The network collects vast quantities of input data and processes it across several layers; at each layer, the network can learn highly complex data features. Deep learning is a powerful tool to make an prediction actionable result. Deep learning excels in pattern discovery and knowledge-based prediction. Big data is the fuel for deep learning. When these two factors are combined, a company can achieve unparalleled levels of growth, management, revenue, and creativity. Deep learning can outperform traditional method. For instance, deep learning algorithms are 40% more accurate than machine learning algorithm in image classification, 27 % more accurate in facial recognition and 25% in voice recognition.

Deep Learning Process

From object detection to speech recognition, a deep neural network offers state-of-the-art precision in a variety of tasks. They will learn on their own, without the programmers having to specifically code predefined information.

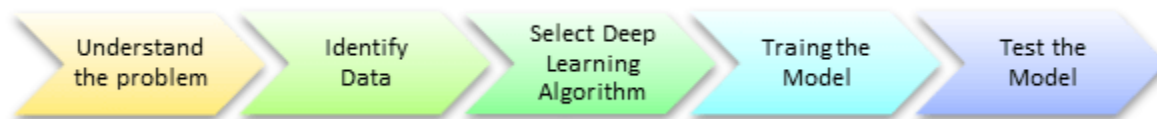


Figure 1.2 Deep Learning Process

To grasp the thought of deep learning, imagine a family, with an infant and fogeys. The toddler points objects together with his finger and always says the word 'cat.' As its parents are concerned about his education, they keep telling him 'Yes, that's a cat' or 'No, that's not a cat. The infant continues to point to things, but with cats, he becomes more precise.. the small kid, deep down, doesn't know why he can say it's a cat or not. He has just learned a way to hierarchy complex features bobbing up with a cat by observing the pet overall and still specialize in details like the tails or the nose before to form up his mind. A neural network works quite the identical. Each layer represents a higher level of data, or the information hierarchy. A four-layer neural network can learn more complex features than a two-layer neural network. There are two stages of the learning process. • The first phase entails applying a nonlinear transformation to the input and producing a mathematical model as an output; the second phase entails refining the model using a machine learning algorithm mathematical method referred to a derivative of Until reaching a level of precision, the neural network repeats these two steps hundreds to thousands of times that is appropriate. The repeat of this two-phase is named an iteration.

Classification of Neural Networks

Shallow neural network: The Shallow neural network has only one hidden layer between the input and output.

Deep neural network: Deep neural networks have more than one layer. For instance, Google LeNet model for image recognition counts 22 layers. Nowadays, deep learning is used in many ways like a driverless car, mobile phone, Google Search Engine, Fraud detection, TV, and so on.

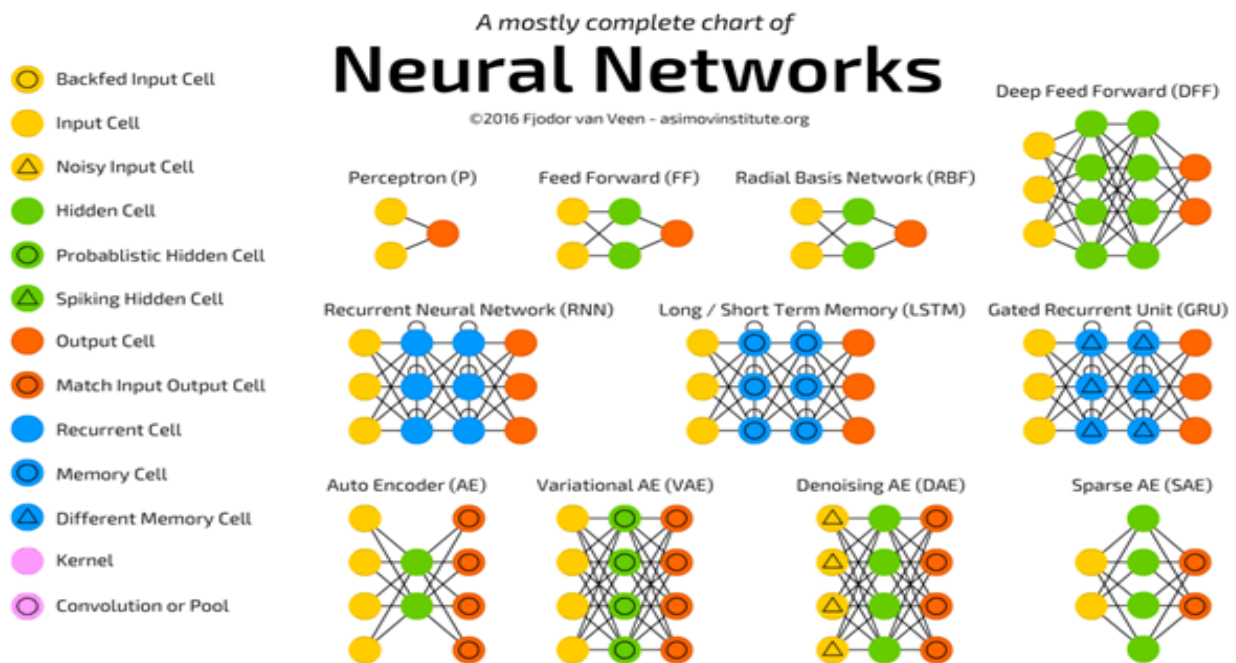


Figure 1.3 Types of Deep Learning Networks

Feed-Forward Neural Networks

The simplest form of artificial neural network. With this kind of architecture, information flows in mere one direction, forward. It means that data flows from the input layer to the "hidden" layers and finally to the output layer. The network doesn't have a loop. Information stops at the output layers.

Recurrent neural networks (RNNs)

RNN could be a multi-layered neural network which will learn data sequences and produce variety or another sequence by storing information in background nodes. RNNs are compatible for processing sequences of inputs.

The process of RNNs are explained below:

The RNN neurons will receive a proof that time to the beginning of the sentence. The network receives the word "Do" as an input and produces a vector of the quantity. This vector is fed back to the neuron to supply a memory to the network. This stage aids the network in remembering that it obtained "Do" within the first place, which it'll move to the subsequent words within the same manner. The end occurs after receiving the word "a." a." The neural network would measure the likelihood of each English word that could be used to finish the sentence. A well-trained RNN would almost certainly assign a high likelihood value.

Common uses of RNN

- Assist securities traders in producing analytic reports
- Identify irregularities within the contract of monetary statement
- Detect fraudulent credit-card transaction
- Provide a caption for images
- Power chatbots
- When practitioners are dealing with time-series data or sequences, RNN is commonly used.

Convolutional Neural Networks (CNN)

CNN is a multi-layered neural network with a specific architecture that defines the output by extracting increasingly complex features of the data at each layer. CNN's are well suited for perceptual tasks

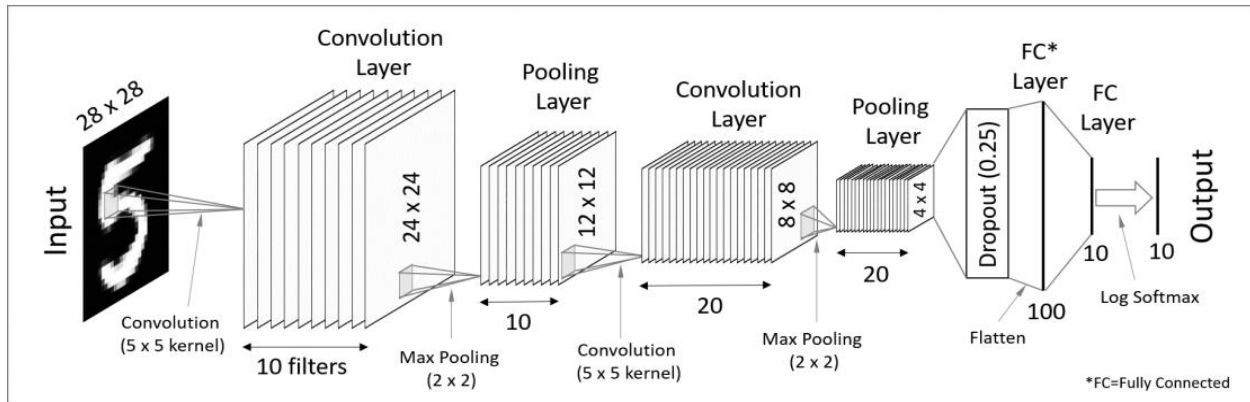


Figure 1.4 CNN

When there is an unstructured data set (e.g., images) and practitioners need to derive information from it

For instance, if the task is to predict a picture caption:

CNN receives a photograph of a cat to illustrate a point. A pixel array is a machine word for a collection of pixels. In general, one layer for a greyscale image and three layers for a colour image. The network can recognise specific features during feature learning (i.e., hidden layers), as an example, the tail of the cat, the ear, etc. When the network has mastered the art of image recognition, it may assign a probability to each image it has encountered. The network's prediction will be determined by the mark with the highest likelihood. A Convolutional Neural Network (CNN, or ConvNet) is a form of multi-layer neural network that is designed to recognise visual patterns directly from pixel images with little pre-processing. The ImageNet project may be a massive graphic database intended for use in visual applications study on visual perception software. The ImageNet project hosts an international software competition known as the ImageNet Large Scale. The ILSVRC is a competition in which software programmes compete to correctly identify and detect objects and scenes.

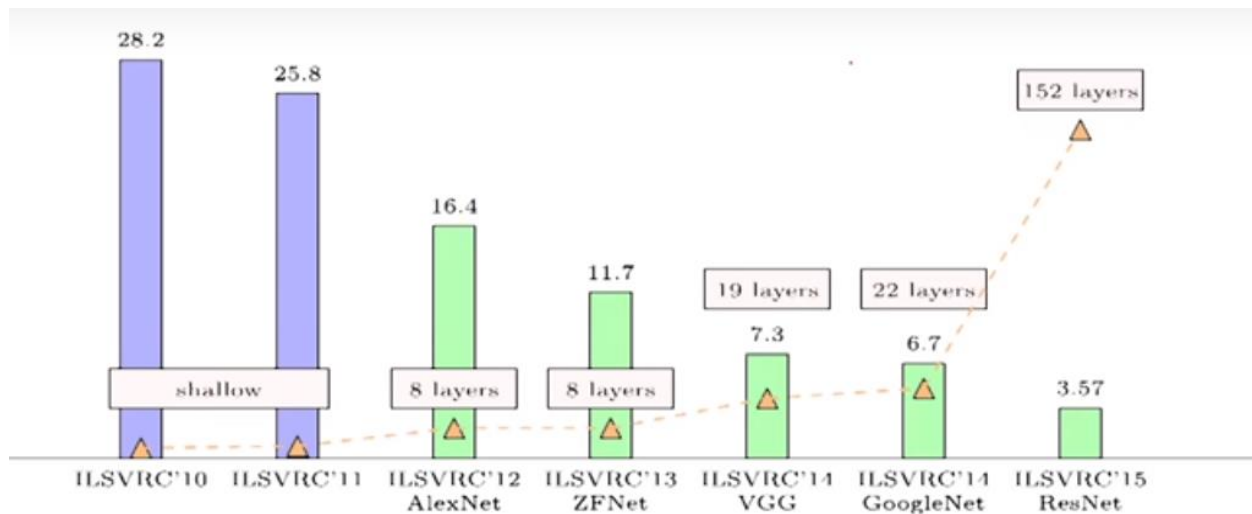


Figure 1.5 Types of CNN

ReactJS

ReactJS Basically, it's an open-source JavaScript library that's used to build single-page applications' user interfaces. It is in charge of the display layer for both network and mobile React can also be used to build interchangeable user interface elements. React was developed by Jordan Walke, a Facebook software developer, was the first to build React. React's The main goal of React is to be quick, scalable, and flexible. It only operates on application user interfaces. This corresponds to In the MVC template, there is a view. Other JavaScript modules or frameworks, such as Angular JS, can be used for it in MVC applications.

Features

- JavaScript XML is abbreviated as JSX. It's an XML/HTML-like syntax that React uses. It goes on and on. ECMAScript to allow for the coexistence of XML/HTML-like text and JavaScript react code. Pre-processors like Babel use this syntax to convert HTML-like text contained in JavaScript files into standard JavaScript objects. We can take it a step further with JSX by embedding the HTML code within the JavaScript once more. This makes HTML codes easier to read and increases JavaScript's performance making robust

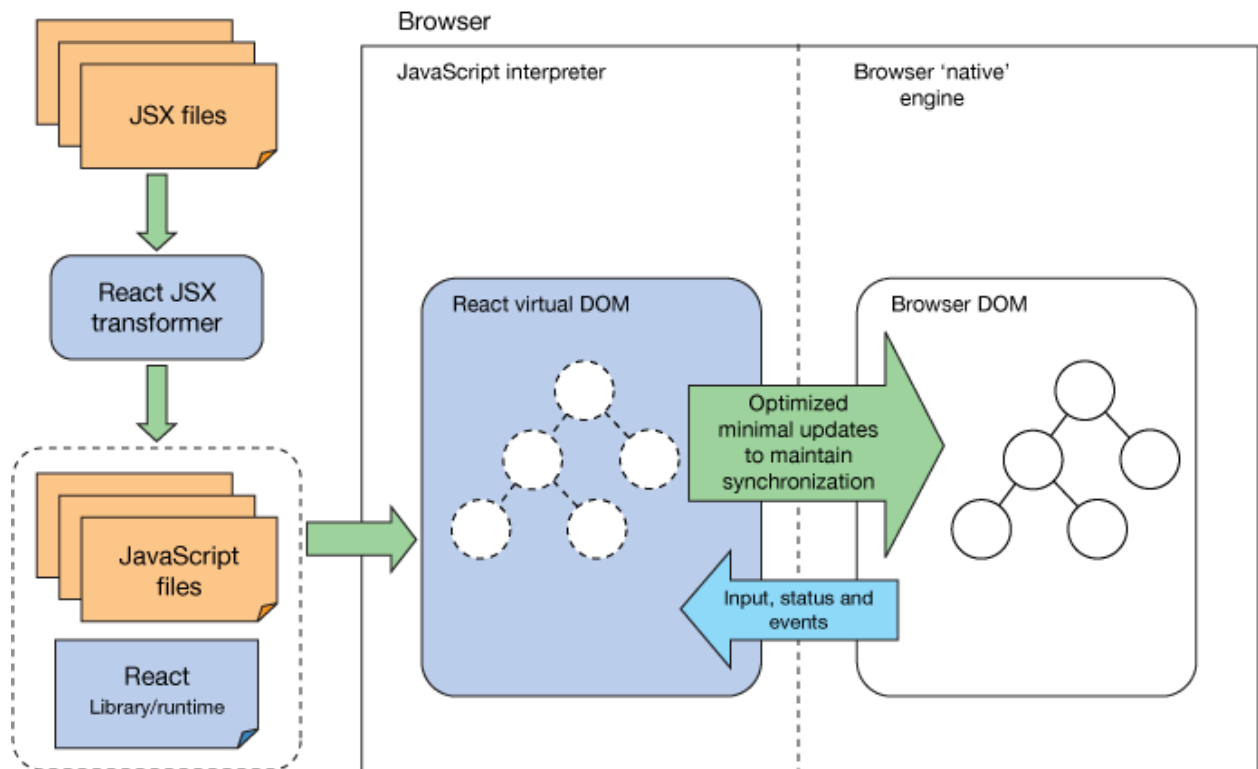


Figure 1.6 ReactJS architecture

CHAPTER - 2

LITERATURE REVIEW

Ji-Hoon Bae *et al*, (2020) We suggest a new knowledge transfer approach based on the teacher-student framework (TSF), in which knowledge is transferred in the form of dense flow through layers..was extracted from a pre-trained "teacher" deep neural network (DNN) and passed to a "worker" DNN. Multiple overlapped flow-based pieces of information from the pre-trained instructor DNN are densely extracted across layers in the case of distilled knowledge. The TSF then achieves transference of the densely collected teacher knowledge by repeated sequential training from the bottom up. between the DNN models of the instructor and the student To put it another way, we conduct bottom-up step-by-step transfer of densely distilled knowledge to the student DNN in order to efficiently transmit extracted valuable teacher information to the student DNN.The proposed method is compared to existing TSF-based information transfer methods for application to reliable image datasets, such as CIFAR-10, CIFAR-100, MNIST, and SVHN, in terms of image classification accuracy and quick optimization.For all benchmark datasets considered in this analysis, when the dense flow-based sequential knowledge transfer scheme is used in the TSF, the qualified student ResNet more accurately represents The pre-trained teacher ResNet contains a wealth of information and outperforms current TSF-based knowledge transfer approaches in terms of accuracy.

Tianyu Guo *et al*,(2018) Benchmark data sets show that the proposed method for learning robust student networks with satisfying accuracy and small sizes is efficient. With significantly fewer parameters. The accuracy of a portable student network can be equivalent to that of a teacher network. However, for practical applications, the learned student network's robustness to perturbation is also significant. Existing teacher-student learning systems place a premium on precision and compression ratios, but they neglect robustnessIn this paper, we use the teacher network to help the student network make more confident predictions, and we investigate the lower bound of the perturbation that will break the student network's 11 faith. To optimise this lower cost,

two important objectives regarding prediction scores and example gradients have been established. bound, to improve the student network's robustness without sacrificing efficiency.

Yan-Hui Tu *et al*, (2019) A novel teacher-student learning framework for the preprocessing of a speech recognizer, leveraging the online noise tracking capabilities of improved minima controlled recursive averaging (IMCRA) and deep learning of nonlinear interactions between speech and noise. First, a teacher model with deep architectures is built to learn the target of ideal ratio masks (IRMs) using simulated training pairs of clean and noisy speech data. Next, a student model is trained to learn an improved speech presence probability by incorporating the estimated IRMs from the teacher model into the IMCRA approach. The student model can be compactly designed in a causal processing mode having no latency with the guidance of a complex and noncausal teacher model. Moreover, the clean speech requirement, which is difficult to meet in real-world adverse environments, can be relaxed for training the student model, implying that noisy speech data can be directly used to adapt the regression-based enhancement model to further improve speech recognition accuracies for noisy speech collected in such conditions

Zhong-Qiu Wang *et al*, (2019) Time-frequency (T-F) masking based on deep learning has significantly improved monaural (single-channel) speech isolation and enhancement. This research looks at its ability for estimating direction of arrival (DOA) in noisy and reverberant environments. We investigate how T-F masking and traditional localization algorithms, such as generalised cross correlation with phase transform, can be used together. Additionally, new algorithms based on steered-response SNR and steering vectors have been proposed. The central concept is to use deep neural networks (DNNs) to find speech dominant T-F units with relatively clean phase for DOA estimation. Our DNN is trained using only monaural spectral details, which means that the trained model can be applied to arrays of different numbers of microphones. Although only monaural information is used for training, experimental results show strong robustness of the proposed approach in new environments within tense noise and room reverberation, outperforming traditional DOA estimation methods by large margins. Our findings also indicate that the ideal ratio mask and its variants are still good training targets for accurate speaker localization.

YongliangYang *et al*,(2020) The robust stabilisation problem of discrete-time linear dynamical systems with bounded and mismatched instability has a model-free solution. An optimal controller design method is derived to solve the robust control problem, which results in solving an algebraic Riccati equation (ARE). The optimal controller obtained by solving the ARE is shown to be capable of robustly stabilising the uncertain system. Off-policy reinforcement learning (RL) is used to solve the problem at hand without the need for system dynamics in order to create a model-free solution to the translated ARE. In addition, comparisons of on-policy and off-policy RL approaches in terms of robustness are provided to investigate noise and its relationship to device dynamics. Finally, a simulation example is used to demonstrate the efficacy of the off-policy RL method proposed. is used to solve the problem at hand without the need for system dynamics in order to create a model-free solution to the translated ARE. In addition, comparisons of on-policy and off-policy RL approaches in terms of robustness are provided.

MahbubulAlam *et al*,(2018) A novel biologically relevant sparse-deep simultaneous recurrent network (S-DSRN) for robust facial expression recognition. The feature sparsity is obtained by adopting dropout learning in the proposed DSRN as opposed to usual handcrafting of additional penalty terms for the sparse representation of data. Theoretical analysis of S-DSRN shows that the dropout learning offers desirable properties such as sparsity, and prevents the model from overfitting. Experimental results also suggest that the proposed method yields better performance accuracy, requires reduced number of parameters, and offers reduced computational complexity than that of the previously reported state-of-the-art feed-forward DNNs using two of the most widely used publicly available facial expression data sets. Furthermore, we show that by combining the proposed neural architecture with a state-of-the-art metric learning technique significantly improves the overall recognition performance. Finally, a graphical processing unit (GPU)-based implementation of S-DSRN is obtained for real-time applications.

Amin Azmoodeh *et al*,(2018) In military environments, the Internet of Things (IoT) usually consists of a wide variety of Internet-connected devices and nodes.(e.g.

medical devices and wearable combat uniforms).Cyber criminals, especially state-sponsored or nation-state actors, see these IoT devices and nodes as a lucrative goal. The use of malware is a popular attack vector.A common attack vector is the use of malware.We present a deep learning-based method for detecting Internet Of Battlefield Things (IoBT) malware via the device's Operational Code (OpCode) sequence in this paper. To distinguish malicious and benign applications, we convert OpCodes into a vector space and use a deep Eigenspace learning approach. We also show the robustness and long-term viability of our proposed malware detection strategy.against the insertion of junk code Finally, we open-source our malware sample on Github. in the hopes that it will aid future research efforts. (e.g. to facilitate evaluation of future malware detection approaches).

Qi Wang *et al*,(2012) Vehicular Management relies heavily on congestion detection. The majority of current algorithms, on the other hand, are insufficient for real-world applications. Traditional features are not discriminative which results in rather poor performance under complex scenarios. While deep features can better reflect high-level details, deep network regression training is difficult. A robust hierarchical deep learning approach is proposed for the task of congestion detection.In this method, a deep network is designed for hierarchical semantic feature extraction. Different from traditional deep regression networks which usually directly utilize mean squared error as loss function, a robust metric learning is employed to effectively train the network.Multiple networks are combined based on this to boost generalisation capacity even further. Extensive tests are carried out, and the effectiveness of the proposed model is verified.

Dong Kyun Shin¹ *et al*, (2017) Unseen object detection using incremental active semi-supervised learning (IASSL). It uses a combination of batch-based active learning (AL) and bin-based semi-supervised learning (SSL) to take advantage of AL's discovery and SSL's exploitation capabilities. To assess the ambiguity and diversity of AL, as well as the confidence in SSL, a collaborative sampling approach is used.We can pick more detailed, confident, and representative samples at a low cost with batch-based AL Bin-based SSL divides streaming image samples into multiple bins, with each bin

transferring the discriminative information of a convolutional neural network multiple times (CNN) deep understanding to the next bin until the performance criterion is reached. In unknown, cluttered data distributions, IASSL can resolve noisy and biased labels. In comparison to state-of-the-art technologies such as Faster RCNN, SSD300, and YOLOv2, we achieve superior efficiency.

Jordi Sanchez-Riera *et al*, (2017) An optimization function that minimises the error between the model and the data is used to fit a 3D hand model to the observed RGB-D data. However, since these algorithms are so reliant on the initialization stage, they are impractical to use in a natural setting. To solve these kind of problems, it is common to use an offline dataset with pre-learned gestures that will serve as a first rough estimate. In particular, we present an algorithm that employs an articulated ICP minimization feature, with parameters derived from a dataset of hand gestures trained using a deep learning system. This set up has two strong points. Deep learning, for starters, offers a very fast and accurate estimation of performed hand gestures. Second, the articulated ICP algorithm captures the potential variability of a gesture performed by different people or by slightly different gestures. Several methods are used to test and validate our proposed algorithm. Independent analyses are carried out for the deep learning system and articulated ICP. Furthermore, various real-world sequences are registered to validate our method, and finally, state-of-the-art algorithms are used to perform quantitative and qualitative comparisons.

Vanika Singhal *et al*, (2018) A new deep learning architecture that is specifically designed for hyperspectral image classification. In a systematic manner, we study various levels of dictionaries. The discriminative layer, which learns a linear classifier, is the final layer. The training proceeds greedily; at a time, one level of dictionary is learned and therefore the coefficients accustomed train the following level. For classification, the coefficients from the final stage are used. Robustness is incorporated by minimizing absolutely the deviations rather than the more popular Euclidean norm. The built-in robustness aids in the reduction of mixed noise (Gaussian and sparse) in hyperspectral images. Results show that our proposed techniques outperform all other deep learning methods—deep belief network, stacked auto encoder, and

convolutional neural network. The experiments are dispensed on both benchmark deep learning data sets (MNIST, CIFAR-10, and Street View House Numbers) similarly as on real hyperspectral imaging data sets.

Jennifer E. Rosen* *et al*, (2016) Thyroid nodules are normal, and fine needle aspiration biopsy (FNAB) is often used to assess the existence of malignancy and to guide treatment. Unfortunately, cytology alone does not distinguish between benign and malignant thyroid nodules in around 15–30% of FNAB-evaluated nodules. These patients need surgery solely for the purpose of diagnosis; the majority of these nodules turn out to be benign in the end. elastic ligature .Elastic light scattering spectroscopy (ESS) that measures the spectral differences between benign and malignant thyroid nodules has shown promise in improving preoperative determination of benign status of thyroid nodules. We present the findings of a massive, prospective, blinded study that validated the ESS algorithm in thyroid nodule patients .An ESS system was used to acquire spectra from human thyroid gland tissue The histopathology of the biopsy samples was related to the spectroscopic findings.

Micha Feigin *et al*, (2016) Ultrasound elastography is gaining popularity as a cost-effective and practical diagnostic method for cancer detection and differentiation, as well as thyroid disease diagnosis. Unfortunately, state of the art shear wave imaging techniques, essential to market this goal, are limited to high-end ultrasound hardware because of high power requirements; are extremely sensitive to patient and sonographer motion, and customarily, suffer from low frame rates. We present another method using single-sided pressure-wave sound speed, which is motivated by research and theory showing that longitudinal wave sound speed has diagnostic abilities similar to shear wave imaging. measurements from channel data. Methods: during this paper, we present a single-sided sound speed inversion solution employing a fully convolutional deep neural network. For training, we use simulations, which enable us to generate an infinite amount of ground truth data. We show that inverting for longitudinal sound speed in soft tissue at high frame rates is possible. On simulated results, we test the strategy. On the basis of minimal real-world evidence, we present highly encouraging results. Final Thoughts: With deep learning, sound speed inversion on

channel data has a lot of promise. technologies. technological advancements
Importance: In many places, specialised shear wave ultrasound systems are still unavailable. Longitudinal sound speed and deep learning technology allow a new way to diagnose tissue elasticity that is sponsored. It is possible to achieve high frame rates.

Amir Mirbeik-Sabzevari *et al*, (2015) This work introduces new, stable, and broadband skin-equivalent semisolid phantoms for mimicking interactions of milli-meter waves with the human skin and skin cancer. Realistic skin phantoms serve as an invaluable tool for exploring the feasibility of new technologies and improving design concepts related to milli-meter-wave skin cancer detection methods. Normal and malignant skin tissues are separately mimicked by using appropriate mixtures of deionized water, oil, gelatin powder, formaldehyde, TX-150 (a gelling agent, widely referred to as 'super stuff'), and detergent. The dielectric properties of the phantoms are characterized over the frequency band of 0.5–50 GHz using a slim-form open-ended coaxial probe in conjunction with a milli-meter-wave vector network analyser. The measured permittivity results show excellent match with ex-vivo, fresh skin (both normal and malignant) permittivities determined in our prior work over the entire frequency range. This work results in the closest match among all phantoms reported in the literature to surrogate human skin tissues. The stability of dielectric properties over time is also investigated. The phantoms demonstrate long-term stability (up to 7 months was investigated). In addition, the penetration depth of milli-meter waves into normal and malignant skin phantoms is calculated. It is determined that milli-meter waves penetrate the human skin deep enough (0.6 mm on average at 50 GHz) to affect the majority of the epidermis and dermis skin structures.

YanboWang *et al*, (2015) Smoking is the major cause of lung cancer and the leading cause of cancer-related death in the world. The most current view about lung cancer is no longer limited to individual genes being mutated by any carcinogenic insults from smoking. Instead, tumorigenesis a phenotype conferred by many systematic and global alterations, leading to extensive heterogeneity and variation for both the genotypes and phenotypes of individual cancer cells. Thus, strategically it is foremost important to develop a methodology to capture any consistent and global

alterations presumably shared by most of the cancerous cells for a given population. This is particularly true that almost all of the data collected from solid cancers (including lung cancers) are usually distant apart over a large span of temporal or even spatial contexts. Here we report a multiple non-Gaussian graphical model to reconstruct the gene interaction network using two previously published gene expression datasets. Our graphical model aims to selectively detect gross structural changes at the level of gene interaction networks. Our methodology is extensively validated, demonstrating good robustness, as well as the selectivity and specificity expected based on our biological insights. In summary, gene regulatory networks are still relatively stable during presumably the early stage of neoplastic transformation. But drastic structural differences can be found between lung cancer and its normal control, including the gain of functional modules for cellular proliferations such as EGFR and PDGFRA, as well as the loss of the important IL6 module, supporting their roles as potential drug targets. Interestingly, our method can also detect early modular changes, with the ALDH3A1 and its associated interactions being strongly implicated as a potential early marker, whose activations appear to alter LCN2 module as well as its interactions with the important TP53-MDM2 circuitry. Our strategy using the graphical model to reconstruct gene interaction work with biologically-inspired constraints exemplifies the importance and beauty of biology in developing any bio-computational approach.

CHAPTER - 3

AIM AND SCOPE

3.1 AIM

The aim of the project is to present a deep learning methodology with least complexity and easy implementation using advanced algorithm to test for the occurrence of COVID-19 infection in a patient accurately. To do parallel processing of the filters in order to increase efficiency. To make the model more robust to error rate. Analyze the generated model by giving real time input data. To develop a web application using reactJS as a frontend

3.2 SCOPE:

- Can be used in hospitals for effective COVID-19 and tuberculosis testing
- Can be used at laboratories.
- Early prediction of COVID-19 and tuberculosis.

3.3 EXISTING SYSTEM:

We suggest a teacher student system (TSF)-based information transfer approach as a substitute, in which knowledge is distilled in the style of dense flow across layers from a pre-trained “teacher” deep neural network (DNN) and transferred to a different “student” DNN. within the case of distilled knowledge, multiple overlapped flow-based items of data from the pre-trained teacher DNN are densely extracted across layers. The TSF then uses repetitive sequential training from bottom to top between the teacher and student DNN models to transfer the densely extracted teacher knowledge. In other words, to efficiently transmit extracted useful teacher information to the coed DNN, we perform bottom-up step-by-step transfer of densely distilled knowledge. For application to reliable image classification, the proposed method's output in terms of image classification accuracy and quick optimization is compared to that of existing TSF-based information transfer methods. image datasets, including CIFAR-10, CIFAR-100, MNIST, and SVHN. The trained student ResNet more

accurately represents the rich information of the pre-trained teacher ResNet when the dense flow-based sequential knowledge transfer scheme is used within the TSF, and exhibits superior accuracy to the prevailing TSF-based knowledge transfer methods for all benchmark datasets considered during this study.

3.4 DISADVANTAGES OF EXISTING SYSTEM:

One of the most important drawbacks is that transfer learning only works if both models' initial and goal problems are sufficiently similar. If the primary round of coaching data needed for the new task differs significantly from the old data, then the trained models might perform worse than expected. Only benign and malignant variety of thyroid presence is detected within the existing system. Accuracy decreases with decrease within the amount of information.

3.5 PROPOSED SYSTEM:

Deep neural networks promise high accuracy in a number of applications, but their performance is often dependent on complex network architectures. The classical teacher–student learning model attempts to say a lightweight yet accurate student network using well-trained heavy networks as students. This way, a transportable student network with substantially less parameters can do considerable accuracy. However, beyond precision, the learned student network's robustness to perturbation is also important for realistic applications. Current teacher-student learning systems focus primarily on accuracy and compression ratios, but fail to consider robustness. We make the model to predict the utmost efficiency using the mix of three different architecture networks where two to a few blocks are trained together to induce the very best accuracy. The utilization of novel inception model will increase the performance of the model prediction and its robustness. Here the various combination of blocks are trained in parallel and eventually concatenation of the blocks are done to improve the efficiency of the model which successively are wont to predict the presence of COVID-19 and tuberculosis within the patients effectively in low cost.

3.6 ADVANTAGES OF PROPOSED SYSTEM:

- Accurate prediction of covid-19 and tuberculosis .Three different filters will be used for predicting the disease
- Efficiency of the model will increase
- Robustness will increase
- No need of previous memory for further processing
- Parallel processing of the blocks takes place reducing the time needed for computation

CHAPTER: 4

MATERIAL AND METHODS

DETAILED DESIGN OF THE PROJECT:

This chapter describes the overall design. It also describes each module that is to be implemented.

4.1 SYSTEM ARCHITECTURE

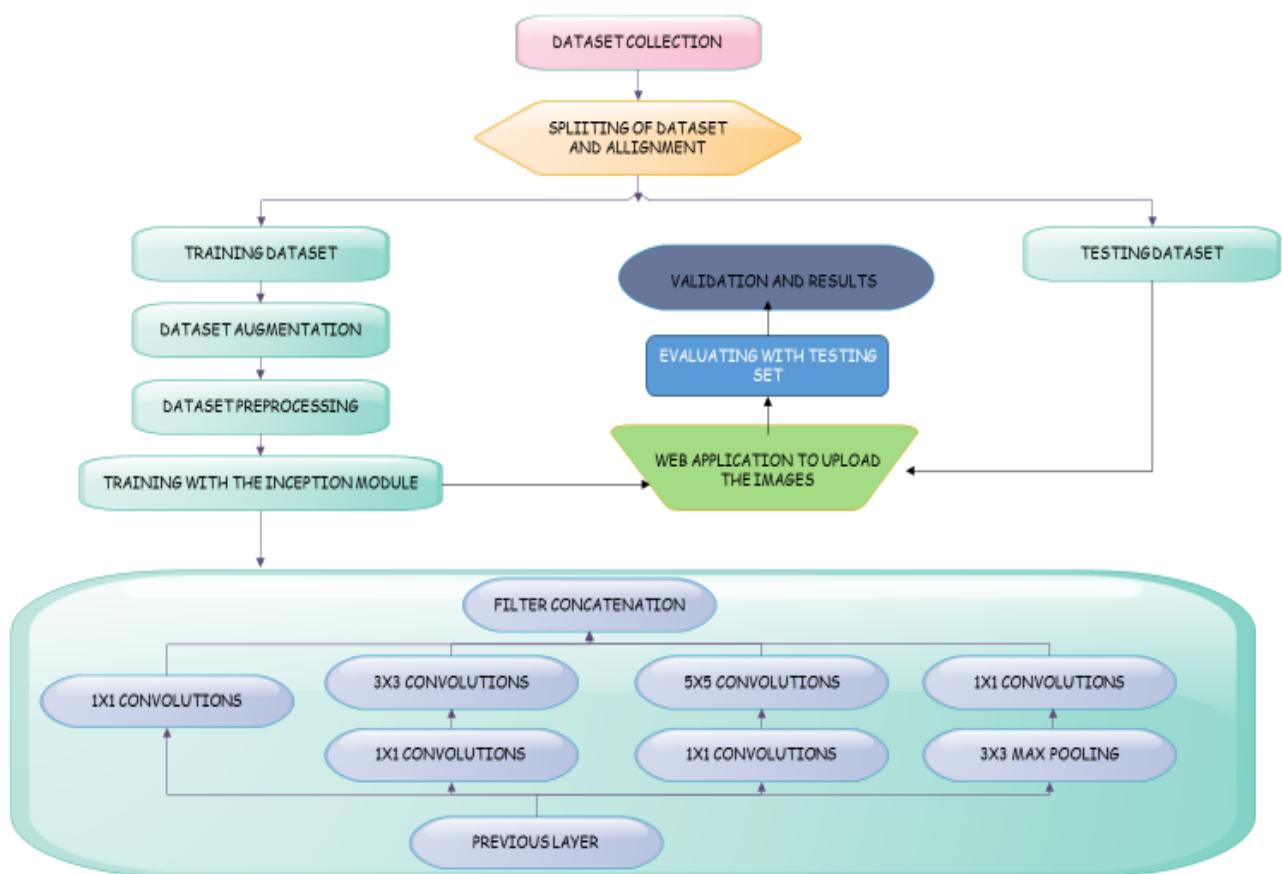


Figure: 4.1: System Architecture

4.2 WORKING

In this project, we are going to develop a novel architecture which will be used for increasing the accuracy behind all the existing deep learning techniques by which we can give highly efficient COVID-19 & tuberculosis disease prediction. So, the first step in the project will be collecting the dataset from various resources and then we will be separating these datasets into training as well as testing dataset where the testing dataset will be kept separate and the training dataset will be used to train the model. Then dataset augmentation will be done which will increase the dataset into multiple numbers. Then these datasets are pre-processed using different techniques to align the datasets into single dimensions. Tu, Yan-Hui, *et al.*,(2019) In this paper, we propose a novel teacher-student learning system for speech recognizer preprocessing that takes advantage of enhanced minima controlled recursive averaging (IMCRAonline)'s noise monitoring capabilities and deep learning of nonlinear interactions between speech and noise. After pre-processing our dataset, we will be ready for training with the architecture. Now, we will be using our novel architecture such that three different filters will be used to train the model. Then we will be performing optimization and the method used for optimization is Adam, SGD which will optimize the whole model in order to remove the noises. We will be performing loss minimization such as cross entropy and finally validate the mode by giving real time inputs. At the end of the training, we will be obtaining the highest efficient model which is the output of a combination of three different filters. Antonios Makris and colleagues (2020) A dataset was developed using a mixture of publicly accessible X-ray images from patients with COVID-19 disease, common bacterial pneumonia, and healthy people. We also provide a comparison between the use of one, two and three filters. A web application using a JavaScript framework reactJS is developed to enable the user to view and determine the results of the COVID-19 & tuberculosis disease prediction interactively. Thus, we obtain the accurate model which can be applied in various fields we want.

4.3 MODULE DESCRIPTION

- Dataset Collection

- Dataset Splitting and Alignment
- Dataset Augmentation
- Training with the inception module
- Database integration
- Web application development

4.3.1 Dataset collection:

A data set may be a list of information Deep Learning has risen to popularity as the preferred approach for addressing a wide variety of challenging real-world issues. It's definitely far and away the most effective performing method for computer vision tasks. The image above showcases the ability of deep learning for computer vision. With enough training, a deep network can segment and identify the "key points" of each person within the image. These deep learning machines that are working so well need fuel plenty of fuel; that fuel is data. The more labelled data available, the higher our model performs. the concept of more data resulting in better performance has even been explored at a large-scale by Google with a dataset of 300 Million images! When deploying a Deep Learning model in an exceedingly real-world application, data must be constantly fed to continue improving its performance. And, within the deep learning era, data is incredibly well arguably the foremost valuable resource. There are three steps of collecting data

4.3.1.1 Scraping From the Web

Because of the amount of human effort involved, manually locating and uploading pictures takes a long time. The task probably has some kind of common objects are to be detected. And so that becomes the keyword for web-scraping. It also becomes the object's class name. This seems to be very simple for a task like image classification, where the image annotations are fairly coarse. But to do something like instance segmentation? Every single pixel in the image is required. It's best to use some of the many excellent image annotation resources that are already available. The paper demonstrates how to construct a model that can produce pixel labels for segmentation from a rough collection of polygon points around an object. Deep extreme cut is identical to deep extreme cut, except that only the four extreme points around the object are used. This will result in some cool segmentation and bounding box labels. Another

alternative is to use an image annotation GUI that already exists. Draw both bounding boxes and polygon points for segmentation maps in this common labelling programme. Mechanical Turk on Amazon is also a low-cost choice.

4.3.1.2 Third-party:

In the deep learning age, data has become such a valuable asset, many start-ups have begun to offer their own image annotation services, in which they will collect and mark the information. Given an outline of what quite data and annotations needed. Mighty, a company that specialises in self-driving car image annotation and has grown to be a major player in the field, was also present at CVPR 2018. Payment AI is less specialised than Mighty AI, and it can annotate images from any domain. Jennifer E. Rose and others (2016) The current COVID-19 epidemic has resulted in social stigma and prejudice against people of certain ethnic groups, as well as everyone who is suspected of having come into contact with the virus.

4.3.2 Dataset Splitting And Alignment

Large datasets can be slow to process. Splitting them into smaller datasets will help speed up the reconstruction process. We will call each of the small datasets a sub-model. Smaller datasets run faster since each bundle change iteration involves fewer files. Furthermore, the reconstruction of the various sub-models can be carried out in parallel. The sub-models' reconstructions would not always be consistent with each other since they are performed separately. The orientation will be determined solely by the GPS locations of the images and the ground control points. When the neighbouring reconstructions share cameras or points, it is possible to enforce the alignment of common cameras and points between the different reconstructions.

4.3.3 Data Augmentation

The amount of data available also increases the efficiency of deep learning neural networks.

Data augmentation is a method of Creating new training data from current data artificially. This is achieved by converting examples from the training data into new and special training examples using domain-specific techniques. 25 The best-known is of

data augmentation is image data augmentation, which entails transforming images in the training dataset into transformed that are from the same group as the original picture

Shifts, turns, zooms, and other operations from the field of image processing are included in transforms. The aim is to add new, plausible examples to the training dataset. This refers to variants of the training set images that the model is likely to see. A horizontal flip of a cat photo, for example, would make sense because the photo could have been taken from either the left or right.

As a result, it is clear that the particular data augmentation strategies used for a training dataset must be carefully selected, taking into account the training dataset as well as knowledge of the problem domain. In addition, a small prototype dataset, model, and training run can be used to test data augmentation approaches separately and together to see if they result in a meaningful change in model results.

Modern Deep learning algorithms like the convolutional neural network (CNN) will learn features that aren't visible to the naked eye independent of where they appear in the picture. Nonetheless, augmentation can help with this transform invariant approach to learning by assisting the model in learning features that are also transform invariant, such as left-to-right to top-to-bottom ordering, and light levels in images. Usually, image data augmentation is only used on the training dataset, not the validation or evaluation datasets. Data planning, such as image resizing and pixel scaling, differs in that it must be done uniformly for all datasets that communicate with

the

model.

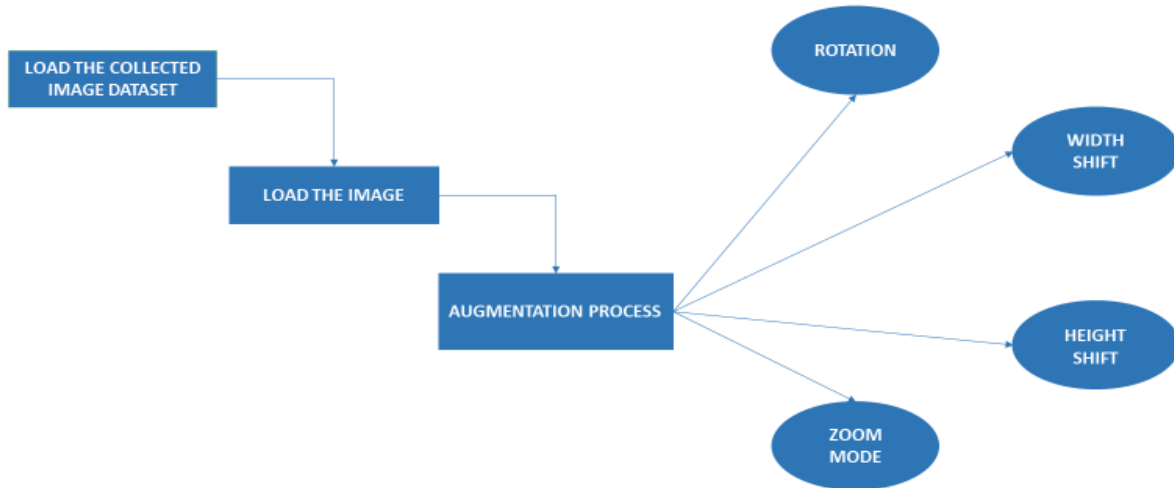


Figure 4.2 Data augmentation flow diagram

4.3.4 Training architecture

4.3.4.1 One Filter Module

In order to allow comparison between differing types of filters and its accuracy, one filter module is been trained using the VGG16 architecture. VGG16 could be a convolutional neural network model introduced The paper "Extremely Deep Convolutional Networks for Large-Scale Image Recognition" by K. Simonyan and A. Zisserman from the University of Oxford. In ImageNet, a dataset of over 14 million images belonging to 1000 groups, the model achieves 92.7 percent top-5 test accuracy. it absolutely was one among the famous model submitted to ILSVRC-2014 It outperforms AlexNet by replacing large kernel-sized filters (11 and 5 in the first 27 and

second convolutional layers, respectively) with small kernel-sized filters multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and using an NVIDIA Titan Black GPU's

The Architecture

The architecture depicted below is VGG16.

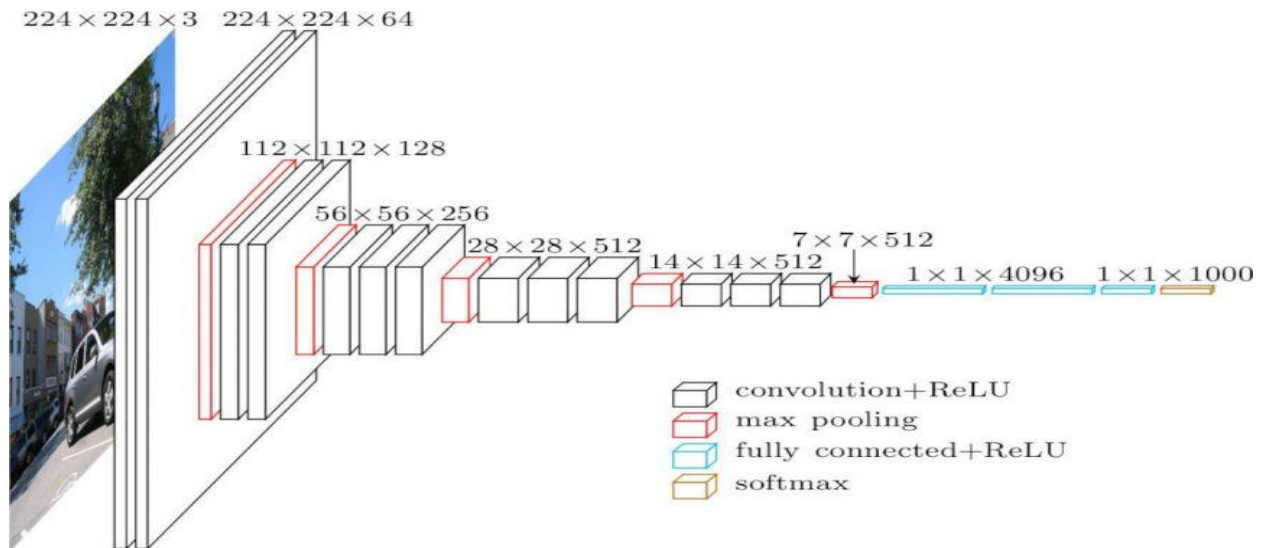


Figure 4.3 VGG16 Architecture

The input to conv1 layer is of fixed size 224×224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a 3×3 (the smallest size to catch the notions of left/right, up/down, and centre) is a very small receptive field. In one of the configurations, it also utilizes 1×1 convolution filters which is a linear transformation of the input channels (followed by non-linearity). The convolution stride is set to 1 pixel, and the spatial padding of convolution layer input is set to 1 pixel for 33 convolution layers so that the spatial resolution is maintained after convolution. Five max-pooling layers, which obey some of the conv. layers, perform spatial pooling. (Max-pooling is not applied to any of the conv. layers.) Max-pooling is achieved with stride 2 over a 22 pixel window. Following a stack of convolutional layers (of varying depth in different architectures), three Fully-Connected (FC) layers are added: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus has 1000 channels. The soft-max layer

is the final layer. In all networks, the completely connected layers are configured in the same way. The rectification (ReLU) non-linearity is present in all hidden layers. It should also be noted that, with the exception of one, none of the networks contain Local Response Normalization (LRN), which does not boost performance on the ILSVRC dataset but does increase memory utilization.

Computer Vision and Radiology for COVID-19 Identification, R. Punia *et al.*, International Conference for Digital Technology, 2020.

4.3.4.2 Two Filter Module

The identity relation between the layers is the only construction that transforms the simple network into a residual network. The residual block used in the network is seen in the screenshot below. The identity relation is represented by a curved arrow that starts at the input and ends at the residual block edge

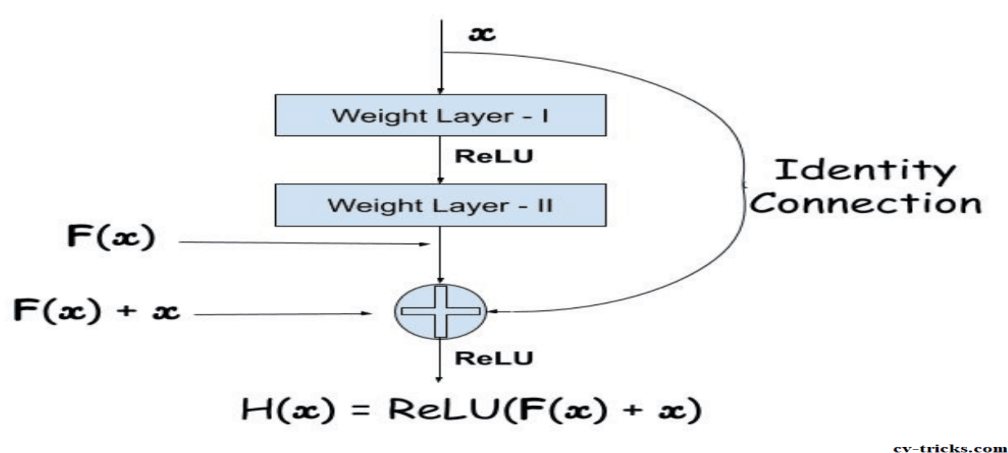


Figure 4.4 A Residual Block of Deep Residual Network

Residual Function

Residual function (also referred to as residual mapping) is that the difference between the input and output of the residual block under question. In other words, residual mapping is that the value that may be added to the input to approximate the ultimate function ($A_1, A_2, A_3, \dots, A_n$) of the block. you'll be able to also assume that the residual mapping is that the amount of error which may be added to input so on reach

the ultimate destination i.e. to approximate the ultimate function. you'll be able to visualize the Residual Mapping as shown within the next figure. you'll see that the Residual Mapping is acting as a bridge between the input and also the output of the block. Note that the burden layers and activation function aren't shown within the diagram but they're actually present

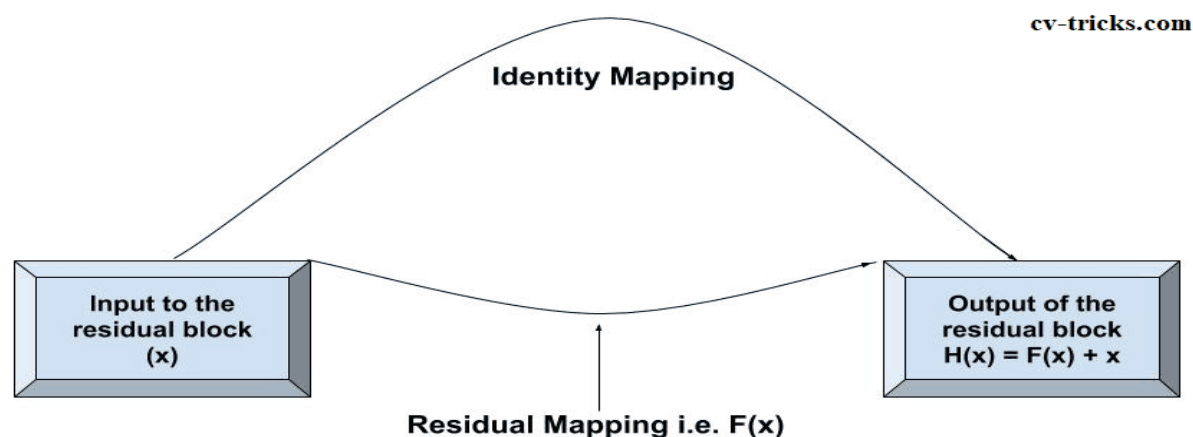


Figure 4.5 Intuitive representation of Residual Function

Let us change our naming conventions to make this post compatible with ResNet paper. Fig-1, shows the residual block. The function which should be learned as a final result of the block is represented as $H(x)$. The input to the block is x and the residual mapping.

Architecture of ResNet-50

Now we'll speak about ResNet50's architecture is shown below. ResNet50's architecture is divided into four levels, as seen in the diagram below. The network can handle images with heights and widths that are multiples of 32 and a channel width of three. We'll say the input size is $224 \times 224 \times 3$ for the sake of clarification. Every ResNet architecture uses 77% and 33% kernel sizes for initial convolution and max-pooling are two examples of initial convolution and max-pooling, respectively. After that, Stage 1 of the network begins, which consists of three Residual blocks, each of three layers. the kernels' size wont to perform The total number of layers in the block of stage 1's convolution operation is 64, 64, and 128. The identity relation is challenged by the

curved 30 arrows. The dashed connected arrow shows that the convolution process in the Residual Block is done with stride 2, which decreases the input dimensions by half. are going to be increased by twofold When we progress from one point to the next, the channel width doubles, reducing the size of the input to half its original size. For deeper networks like ResNet50, ResNet152, etc, bottleneck design is employed. for every residual function F , 3 layers are stacked one over the opposite Convolutions are used in the first, third, and eleventh layers. The 11 convolution layers are in charge of reducing and restoring the size of the image. the size. With smaller input/output measurements, the 33 layer is left as a bottleneck. Finally, the network has a 1000-neuron completely linked layer, which is accompanied by a mean Pooling layer (ImageNet class output).

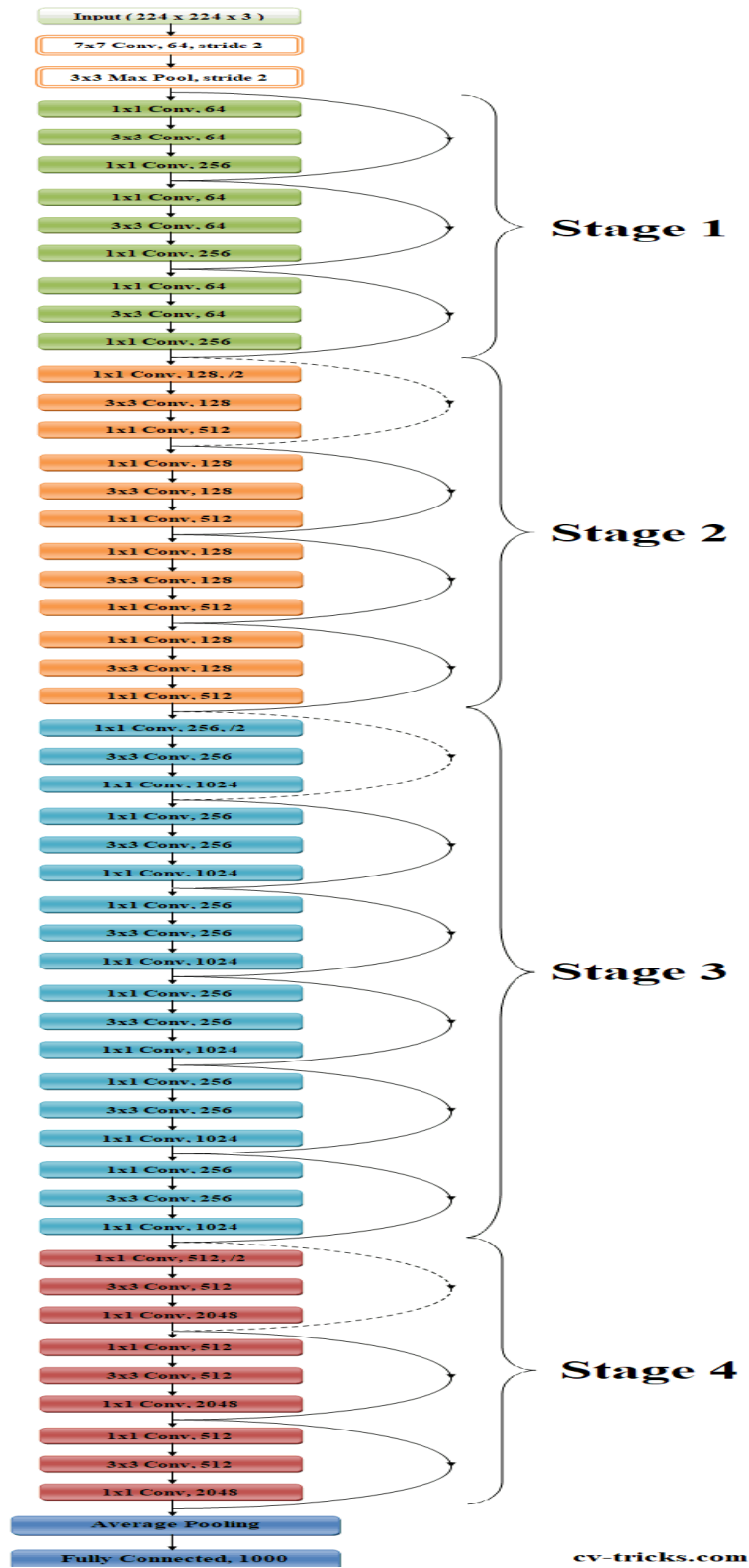


Figure 4.6 Architecture diagram of ResNet50

4.3.5 Training With The Inception Module

The Inception network was a very important milestone within the development of CNN classifiers. before its inception (pun intended), hottest CNNs just stacked convolution layers deeper and deeper, hoping to induce better performance.

The Inception network on the opposite hand, was complex (heavily engineered). It used many tricks to push performance; both in terms of speed and accuracy. Its constant evolution results in the creation of several versions of the network Each new version is a step forward from the previous one. Understanding the enhancements will assist us in creating custom classifiers that are both quick and accurate.. Also, looking on your data, a lower version may fine work better.

Inception Layer

The idea of the inception layer is to hide a much bigger area, but also keep a fine resolution for tiny information on the photographs. So, the thought is to convolve in parallel different sizes from the foremost accurate detailing (1x1) to a much bigger one (5x5).

The concept is that a set of gabor filters of various sizes can manage multiple object scales better. Each filter on the inception layer can be learned, which is an advantage. The simplest way to boost deep learning efficiency is to use more layers and data; GoogleNet uses nine inception modules. the matter is that more parameters also implies that your model is more vulnerable to overfit.As a result, all bottleneck strategies are used to prevent a parameter explosion on the inception layers.

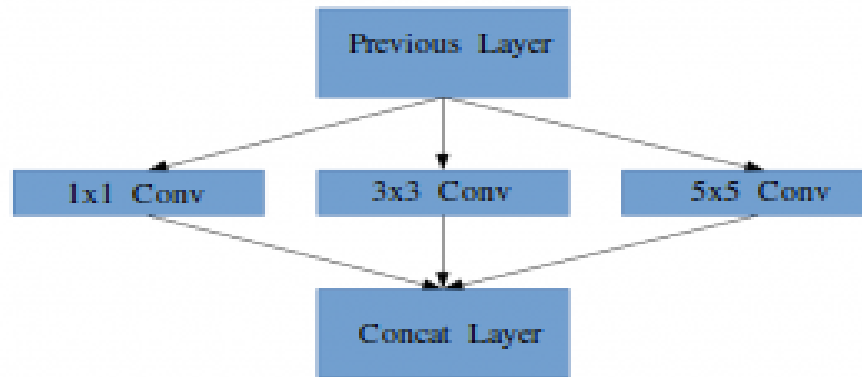


Figure 4.7: Inception layer

As seen above, it's a drastic change from the sequential architectures which we saw previously. In an exceedingly single layer, multiple forms of “feature extractors” are present. This indirectly helps the network perform better, because the network at training itself has many options to settle on from when solving the task. It can either favor to convolve the input, or to pool it directly.

The final architecture contains multiple of those inception modules stacked one over the opposite. Even the training is slightly different during this newly evolved inception module, as most of the topmost layers have their own output layer. This nuance helps the model converge faster, as there's a joint training further as parallel training for the layers itself.

4.3.6 Database integration

MongoDB could be a cross-platform, document-oriented database that has, high performance, high availability, and simple scalability. MongoDB works on concept of collection and document.

Database

A database is a physical storage location for data. On the file system, each database has its own collection of files. Multiple databases are common on a single MongoDB server.

Collection

The term "collection" refers to a group of MongoDB documents. It's the same thing as an RDBMS table. In a single database, there is a set. A schema is not enforced by collections. Different fields can be found in different documents within a series. In most cases, all of the records in a compilation have the same or similar purposes.

Document

A document is made up of a set of key-value pairs. The schema of documents is complex. Documents in a collection don't have to have the same set of common field in a collection's documents may contain a variety of data. The table below depicts the relationship between RDBMS terminology and MongoDB.

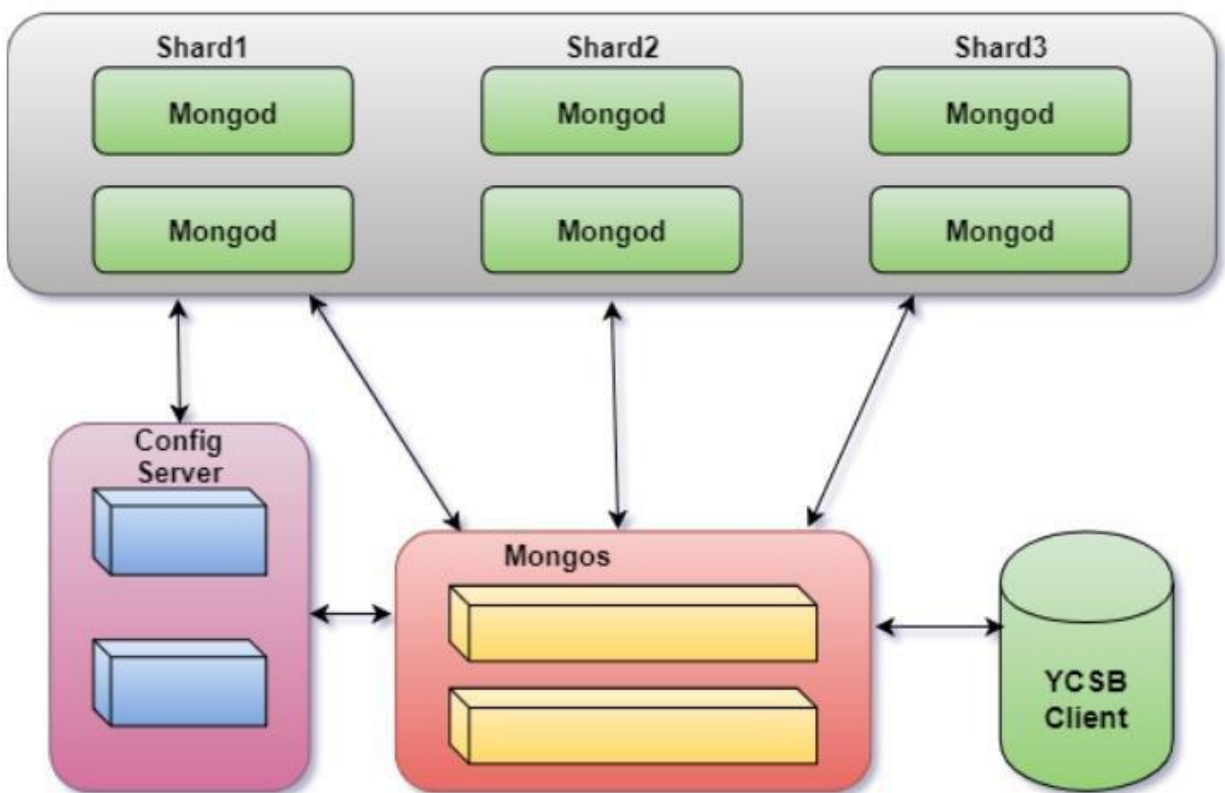


Figure 4.8 Architecture diagram of mongo DB

4.3.7 WEB DEVELOPMENT

ReactJS is JavaScript library used for build a reusable UI components. in line with React official documentation, following is that the definition –

React could be a library for building composable user interfaces. It promotes the creation of reusable user interface components that display data that is likely to change in the future Many people use Respond because of the V in MVC. The DOM is abstracted from you by React, resulting in a simpler programming model with better performance. React can also be used with Node to render on the server and with React Native to power native apps. One-way reactive data flow is implemented in React, which eliminates boilerplate and makes it easier to reason about than conventional data binding. Ji-HoonBae *et al.*,(2020) The proposed method is compared to existing TSF-based information transfer methods for application to reliable image datasets, such as CIFAR-10, CIFAR-100, MNIST, and SVHN, in terms of image classification accuracy and quick optimization.

JSX

JSX stands for JavaScript XML. JSX allows us to write HTML in React. In React, JSX makes it easier to add HTML and write. It is faster than JavaScript since it optimises when converting to Java. It makes easier for us to create templates. Instead of separating the mark-up and logic in separated files, React uses components for this purpose. In subsequent papers, we will go through components in greater depth.

Components

Components are reusable code chunks that are self-contained and self-contained. They do the same things as JavaScript functions, but they run on their own and return HTML via a render function. The two types of components are Class and Function components; we'll concentrate on Class components in this tutorial.

Unidirectional data flow and Flux

React implements one-way data flow, making it easy to think about your app. Flux is a data flow pattern that helps with unidirectional data flow.

ADVANTAGES OF REACT JS

- Makes use of a JavaScript object called virtual DOM. Since the JavaScript virtual DOM is faster than the regular DOM, the app's speed would improve.
- Compatible with other systems and can be used on both the client and server side.
- Component and data patterns increase readability, making larger apps easier to manage.

4.3.8 SOFTWARE LIST

- Microsoft Visual studio
- Python

4.3.9 Microsoft Visual Studio

In this project, the code editor is Visual Studio. The company's integrated development environment is Microsoft Visual Studio (IDE). Computer programmes, as well as websites, web applications, web services, and mobile apps, are frequently created. Visual Studio makes use of Microsoft software development tools such as Windows API, Windows Shapes, Windows Presentation Foundation, Windows Store, and Mic.

In addition to code refactoring, Visual Studio provides a code editor that supports IntelliSense (component for code completion). Both a source-level and a machine-level debugger can be used with the integrated debugger. Visual Studio supports 36 different programming languages, and the code editor and debugger can work with almost any language (to varying degrees) if a language-specific service is available. Built-in languages C, C++/CLI, Visual Basic.NET, C#, F#, JavaScript, TypeScript, XML, HTML, and CSS are only a few examples. Plug-ins are available for a variety of languages, including Python, Ruby, Node.js, and M.

Other languages, including Python, Ruby, Node.js, and M, are supported by plug-ins. Java (and J#) were previously supported. The most basic version of Visual Studio is the Community edition, which is available for free. "Free, fully-featured IDE for students,

open-source, and individual developers," says the Visual Studio Community edition's slogan..

The currently supported Visual Studio version is 2019

Features

- Code editor. Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for variables, functions, methods, loops, and LINQ queries.
- Debugger.
- Designer.
- Other tools.
- Extensibility.
- Previous products.
- Community.
- Professional.

4.3.10 Python

In this project, python is employed because the artificial language. Python is a high-level object-oriented programming language with built-in dynamic semantics that is mainly used for web and app creation. It has a lot of appeal in the field of Rapid Application Development because it allows for binding and dynamic typing Python is a pretty simple language, so it's easy to say because it uses a single syntax that prioritises readability As a result, software maintenance and implementation costs are reduced because teams can collaborate without major language and experience barriers. Python also encourages the use of modules and packages, which means that programmes are often constructed in this way. .in a highly modular design, with code that can be reused through a number of projects. If you've built a module or kit that you like, you can scale it to use in other projects, and you can easily import and export it. One of the most appealing features of Python is that both the quality library and, as a result, the interpreter are available for free, in both binary and text formats and source

code Python and all of the required resources are available on all major platforms, so there's no exclusivity. As a result, it's a desirable choice for developers who don't want to think about high construction costs. Don't worry if this Python definition goes over your head. You'll get it eventually. What you want to remember from this section is that Python is a programming language is a programming language used to build applications for the web and smartphones, like phones. Since it's reasonably simple to be told, the requisite resources are readily available to everyone or some.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 FINAL RESULTS OBTAINED

To begin, we can divide our project into modules of implementation that have already been completed. The method of gathering covid disease datasets is known as dataset processing. To upload the scanned images, a web application using reactJS is developed. Various datasets were gathered, and one example of one of the datasets gather

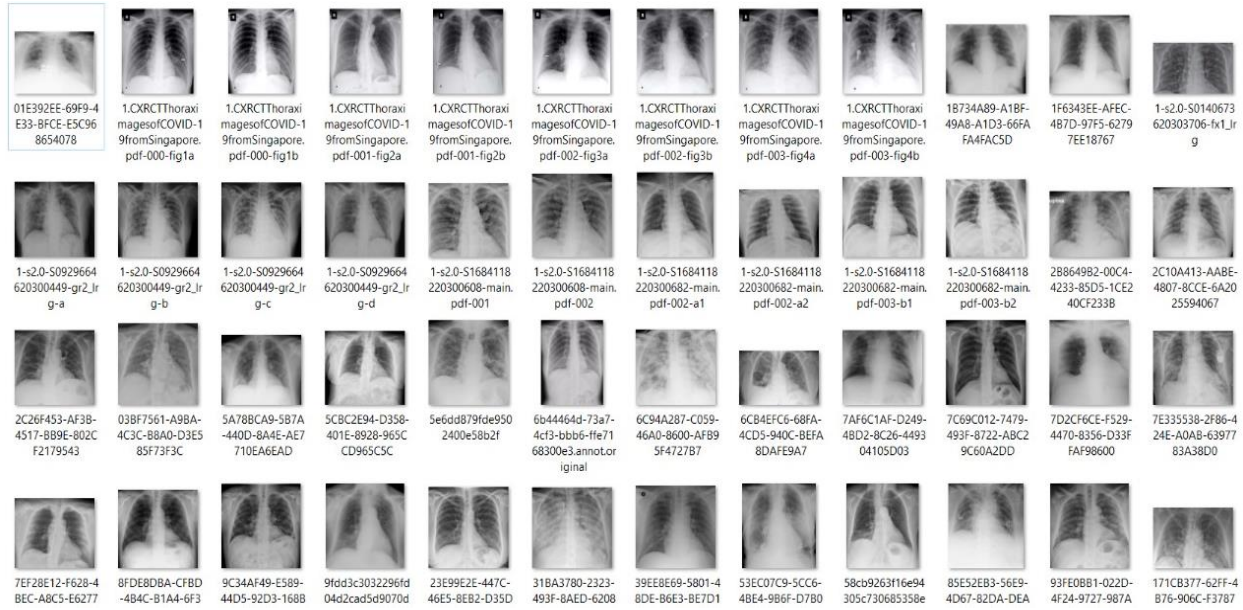


Figure 6.1 Dataset Collection of covid

These datasets are then preprocessed to establish an equivalent aspect ratio in order to prepare them for model training. To begin preprocessing, the datasets are divided into different categories. Following that, the implementation phase takes place, during which the training process is carried out and the results are collected. To run the code, we must first go into the project folder and pick the location to run from there, with the location of the code being the most important factor to consider. The datasets were trained using the VGG architecture and achieved an accuracy of around 90%, as shown in the following figure.

```

C:\Users\Elegant Sivasundaram\Desktop\project\sthybe> python train.py --dataset dataset/images --model single_filter.model --output single_filter.png
2021-02-04 12:32:56.210722: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_100.dll'; dlopen: cudart64_100.dll not found
2021-02-04 12:32:56.210860: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
[INFO] loading images...
[INFO] processed 580/707
WARNING:tensorflow:From C:\Users\Elegant Sivasundaram\anaconda3\lib\site-packages\tensorflow_core\python\ops\runtime_variables_ops.py:1630: calling BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops) with constraint is deprecated and will be removed in a future version.
Instructions for updating:
if using Keras pass *constraint arguments to layers.
2021-02-04 12:33:40.857950: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library nvcuda.dll
2021-02-04 12:33:41.359817: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1618] Found device 0 with properties:
name: GeForce MX130 major: 5 minor: 0 memoryClockRate(GHz): 1.189
pciBusID: 0000:01:00:0
2021-02-04 12:33:41.403933: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_100.dll'; dlopen: cudart64_100.dll not found
2021-02-04 12:33:41.404945: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cublas64_100.dll'; dlopen: cublas64_100.dll not found
2021-02-04 12:33:41.405977: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cuffrt64_100.dll'; dlopen: cuffrt64_100.dll not found
2021-02-04 12:33:41.406121: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'curand64_100.dll'; dlopen: curand64_100.dll not found
2021-02-04 12:33:41.406556: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cusolver64_100.dll'; dlopen: cusolver64_100.dll not found
2021-02-04 12:33:41.407140: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cuspars64_100.dll'; dlopen: cuspars64_100.dll not found
2021-02-04 12:33:41.407646: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudnn64_7.dll'; dlopen: cudnn64_7.dll not found
2021-02-04 12:33:41.407708: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1641] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2021-02-04 12:33:41.408026: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2021-02-04 12:33:41.409507: I tensorflow/core/common_runtime/nnapi/gpu_device.cc:1120] Device Interconnect StreamExecutor with strength 1 edge matrix.
2021-02-04 12:33:41.409637: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1165]
[INFO] compiling model...
[INFO] training head...
2021-02-04 12:33:42.949499: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 411041792 exceeds 10% of system memory.
2021-02-04 12:33:43.079549: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 411041792 exceeds 10% of system memory.
2021-02-04 12:33:43.080754: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 398196736 exceeds 10% of system memory.
2021-02-04 12:33:49.287598: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 398196736 exceeds 10% of system memory.
2021-02-04 12:33:53.585852: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 411041792 exceeds 10% of system memory.
loss
acc
val_loss
val_acc
[INFO] evaluating after initialization...
precision recall f1-score support
Covid 0.66 1.00 0.80 35
ignore 1.00 1.00 1.00 39
normal 1.00 0.85 0.92 118
accuracy 0.89 0.95 0.91 192
macro avg 0.89 0.95 0.90 192
weighted avg 0.94 0.91 0.91 192
[INFO] serializing model...

```

Figure 6.2 Efficiency of First Algorithm

The graph plotted for the number of epochs run can be seen in the following figure

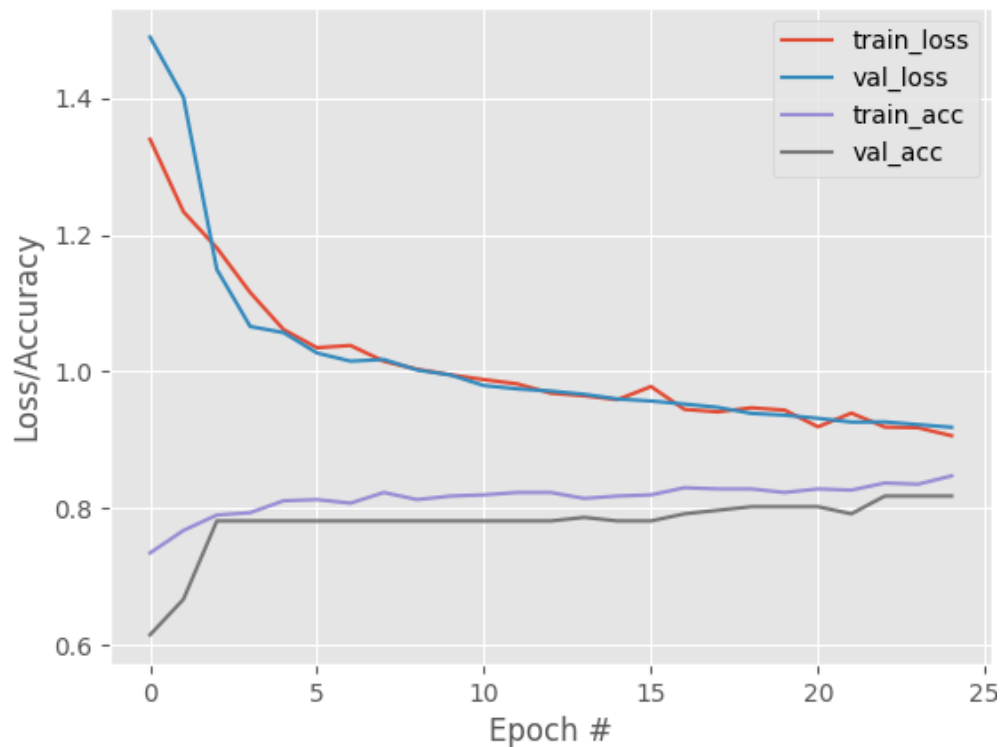


Figure 6.3 Graph Plot After the Training Process

Now the second algorithm, resnet, has been applied to the datasets, resulting in an

accuracy after the training process, as shown in the diagram below.

```
[INFO] evaluating after fine-tuning...
```

	precision	recall	f1-score	support
covid	0.57	1.00	0.73	45
ignore	1.00	1.00	1.00	19
normal	1.00	0.19	0.32	42
accuracy			0.68	106
macro avg	0.86	0.73	0.68	106
weighted avg	0.82	0.68	0.61	106

Figure 6.4 Efficiency of Second Algorithm

The following figure depicts the graph plotted for the number of epochs run.



Figure 6.5 Graph Plot After the Training Process

Now that the novel inception algorithm has been developed and the datasets have been trained with it, we have achieved a higher accuracy than before, as shown in the graph below.

```
[INFO] evaluating after fine-tuning...
```

	precision	recall	f1-score	support
covid	1.00	1.00	1.00	45
ignore	1.00	1.00	1.00	19
normal	1.00	1.00	1.00	42
accuracy			1.00	106
macro avg	1.00	1.00	1.00	106
weighted avg	1.00	1.00	1.00	106

Figure 6.6 Efficiency of Novel Algorithm

The following figure depicts the graph plotted for the number of epochs run.

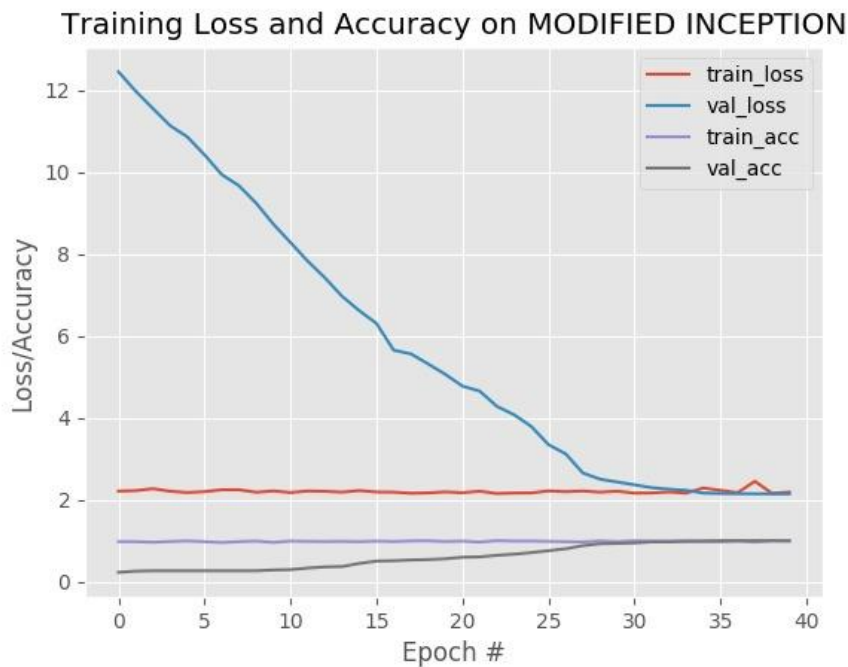


Figure 6.7 Graph Plot After the Training Process

The recognise code is then written, which is used to predict the existence of covid disease by importing the model created during the training phase and integrating it with the frontend created. The login page for the web dashboard is depicted in the diagram below.

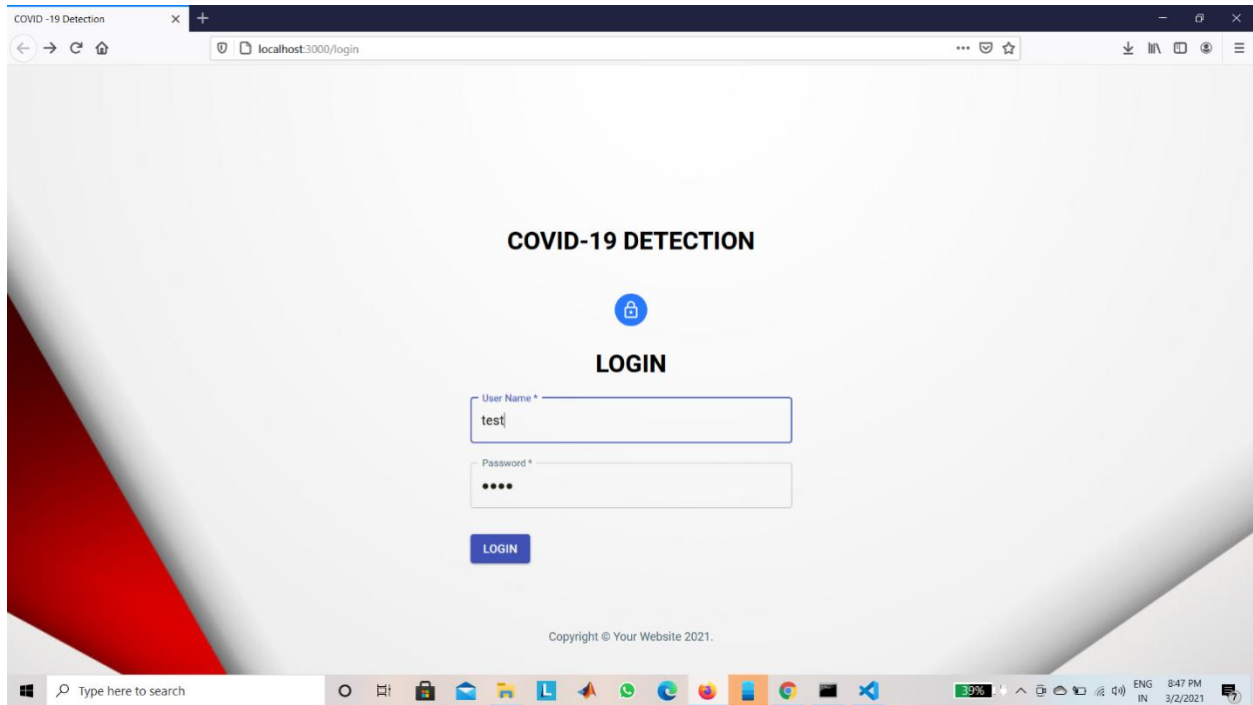


Figure 6.8 Login Page

After logging in, the picture to predict disease can be uploaded as shown in the diagram

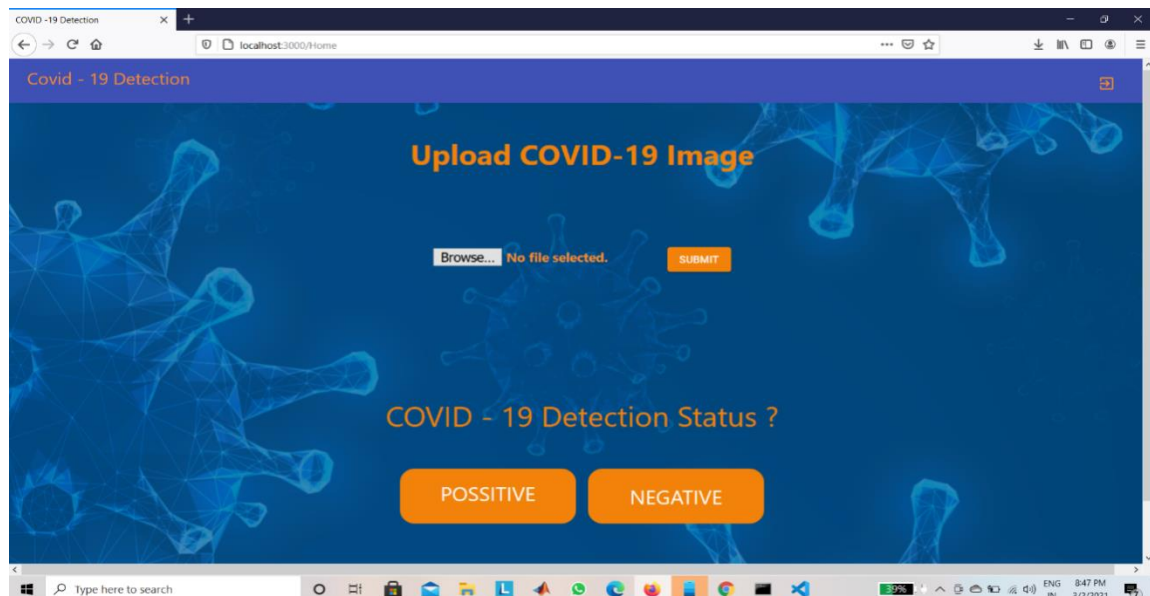


Figure 6.9 Image Upload Option

We can select the type of image we want to predict by clicking on the image upload option, as shown in the below figure.

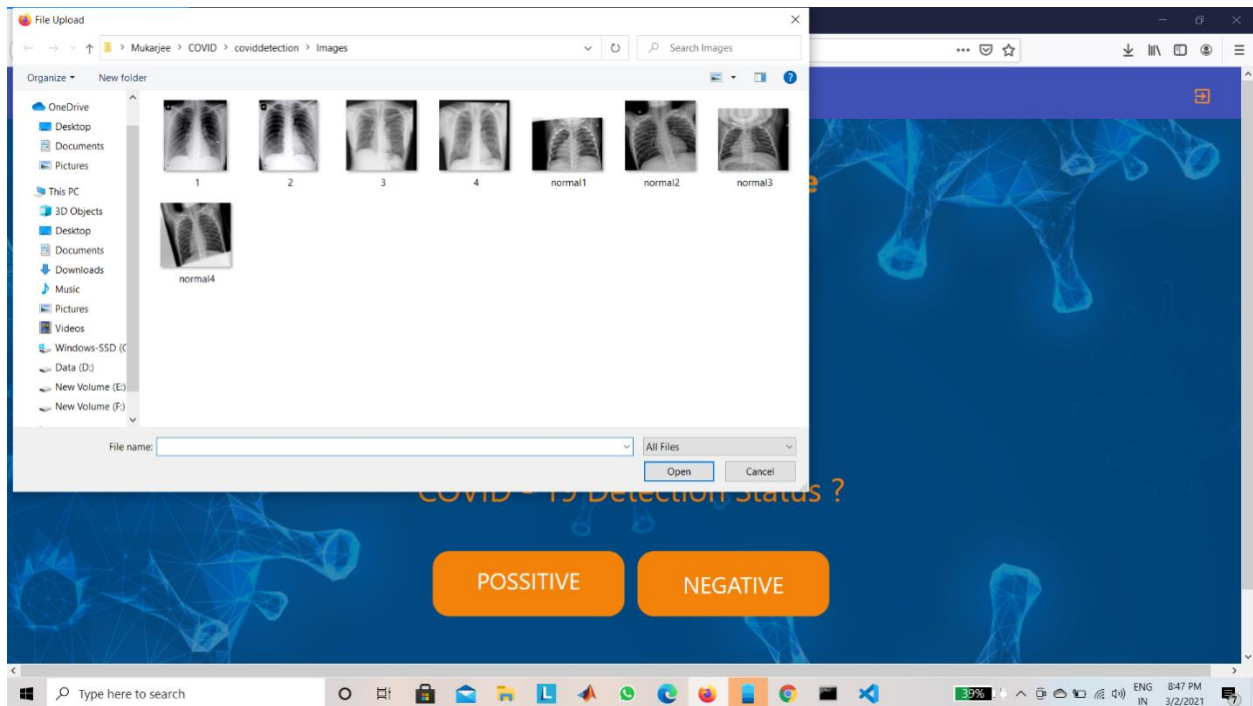


Figure 6.10 Image Uploading for Prediction

The disease is predicted and the findings are shown in the below figure after the image is selected and processed in the backend.

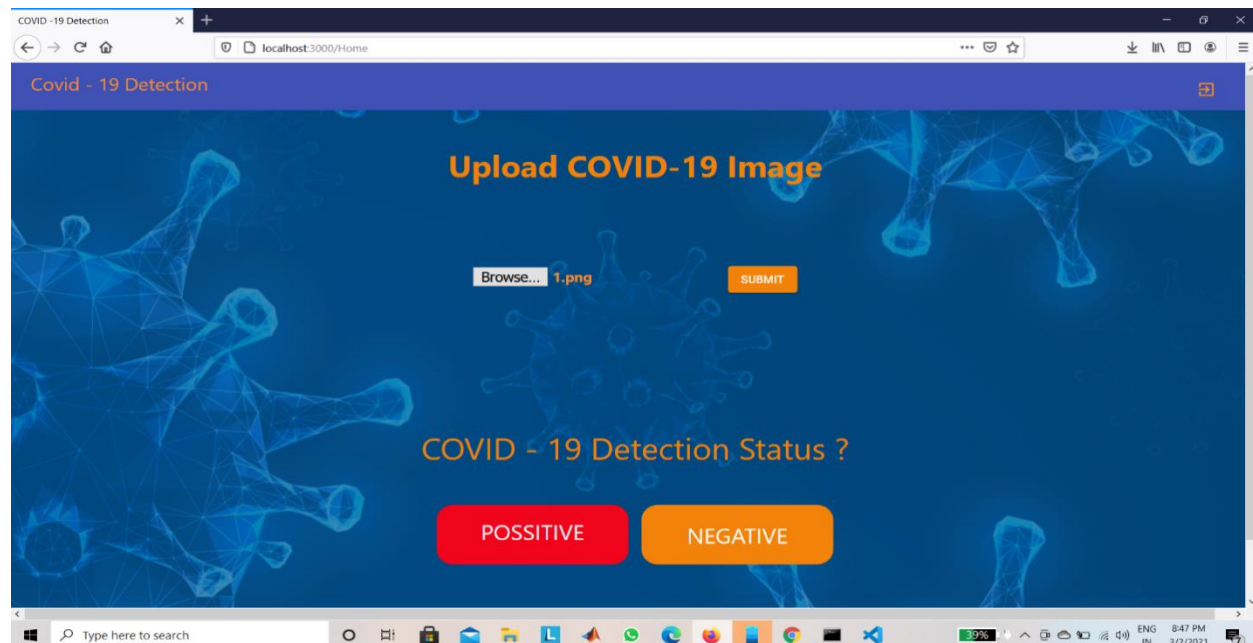


Figure 6.11 Covid Prediction

As a result of the above findings and discussion, it is clear that we have successfully developed a project for predicting covid diseases using a web application that uses reactJS to upload the picture and interactively display the diagnosis. We've also presented a comparison of the accuracy differences between three algorithms, demonstrating that the newly developed algorithm outperforms the others. algorithm and proved that the novel developed algorithm performs the best among the three. Thus, we have successfully implemented the scope of the project.

CHAPTER – 6

SUMMARY AND CONCLUSION

6.1 CONCLUSION

This project is used to identify the existence of COVID-19 and tuberculosis and to provide preventative measures. This also helps in offering successful care at a low cost, thus reducing the time spent identifying COVID-19& tuberculosis in its current state. It's done by hand, which takes longer and has a higher chance of human error. As a consequence, the time required is reduced This project decreases the rate of human error in manual classification.

6.2 SUMMARY

We will study the application of the COVID-19& tuberculosis determine technology in the healthcare field in the near future, and how it can help identify different forms of cancer with greater precision. In the medical sector, there is a greater potential to build or convert this project in a number of ways. As a result, this project has a bright future ahead of it, where manual forecasting can be automated production in a cheap way.

REFERENCE

- [1] Antonios Makris, Ioannis Kontopoulos, Konstantinos Tserpes, "COVID-19 detection from chest X-Ray images using Deep Learning and Convolutional Neural Networks11th Hellenic Conference on Artificial Intelligence, 2020.
- [2] Jain, R., Gupta, M., Taneja, S. *et al.* Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl Intell* **51**, 2021.
- [3] Jennifer E. Rosen*, Hyunsuk Suh, Nicholas J. Giordano, Ousama M. A'amar, Eladio Rodriguez-Diaz, Irving I. Bigio, and Stephanie L. Lee,"Preoperative Discrimination of Benign from Malignant Disease in Covid Nodules With Indeterminate Cytology Using Elastic Light-Scattering Spectroscopy",IEEE Transactions On Biomedical Engineering, Vol. 61, No. 8, August 2016.
- [4] Ji-HoonBae ,Member, IEEE, Doyeob Yeo, JunhoYim, Nae-Soo Kim, Cheol-Sig Pyo, and JunmoKim ,Member, IEEE," Densely Distilled Flow-Based Knowledge Transfer in Teacher-Student Framework for Image Classification", IEEE transactions on image processing, 2020
- [5] Jordi Sanchez-Riera, Kathiravan Srinivasan, Kai-Lung Hua, Wen-Huang Cheng, Senior Member, IEEE, M. Anwar Hossain, Mohammed F. Alhamid "Robust RGB-D Hand Tracking Using Deep Learning Priors" I EEEtransactions on circuits and systems for video technology april 8, 2017.
- [6] MahbubulAlam, Lasitha S. Vidyaratne, and Khan M. Iftekharruddin , Senior Member, IEEE "Sparse Simultaneous Recurrent Deep Learning for Robust Facial Expression Recognition" IEEE transactions on neural networks and learning systems. Digital Object Identifier 10.1109/TNNLS.2017.2776248,2018.
- [7] MichaFeigin, Daniel Freedman, and Brian W. Anthony," A Deep Learning Framework for Single-Sided Sound Speed Inversion in Medical Ultrasound", 0018-9294 2016,2016

- [8] Qi Wang, Senior Member, IEEE, Jia Wan, Xuelong Li, Fellow, IEEE “Robust Hierarchical Deep Learning for Vehicular Management” Citation information: DOI 10.1109/TVT.2883046, IEEE Transactions on Vehicular Technology,2018.
- [9] R. Punia, L. Kumar, M. Mujahid and R. Rohilla, "Computer Vision and Radiology for COVID-19 Detection," International Conference for Emerging Technology (INCET), Belgaum, India,pp. 1-5, 2020
- [10] TianyuGuo, Chang Xu ,Shiyi He, Boxin Shi, Member, IEEE, Chao Xu, and Dacheng Tao , Fellow, IEEE “Robust Student Network Learning” IEEE transactions on neural networks and learning systems,2019.
- [11] VanikaSinghal, Hemant K. Aggarwal, SnigdhaTariyal, and AngshulMajumdar, Senior Member, IEEE “Discriminative Robust Deep Dictionary Learning for Hyperspectral Image Classification” IEEE transactions on geoscience and remote sensing ,2018.
- [12] YanboWang ,Weikang Qian , Bo Yuan,” A graphical model of smoking-induced global instability in lung cancer”,2015.
- [13] Yan-Hui Tu, Jun Du, and Chin-Hui Lee, Fellow, IEEE “Speech Enhancement Based on Teacher-Student Deep Learning Using Improved Speech Presence Probability for Noise-Robust Speech Recognition” IEEE/ACM Transactions On Audio, Speech, And Language Processing, 2019.
- [14] YongliangYang , Member, IEEE, ZhishanGuo , Member, IEEE, HaoyiXiong, Member, IEEE, Da-Wei Ding, Member, IEEE, Yixin Yin , Member, IEEE, and Donald C. Wunsch , II, Fellow, IEEE “Data-Driven Robust Control of Discrete-Time Uncertain Linear Systems via Off-Policy Reinforcement Learning” IEEE transactions on neural networks and learning systems,2019.
- [15] Zhong-QiuWang ,Xueliang Zhang , and DeLiang Wang , Fellow, IEEE “Robust Speaker Localization Guided by Deep Learning-Based Time-Frequency Masking”

IEEE/ACM transactions on audio, speech, and language processing, vol. 27, no. 1, january 2019.

APPENDIX

```
# construct the image generator for data augmentation

aug = ImageDataGenerator(rotation_range=30, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
    horizontal_flip=True, fill_mode="nearest")

# grab the list of images that we'll be describing, then extract
# the class label names from the image paths
print("[INFO] loading images...")

imagePaths = list(paths.list_images(args["dataset"]))

classNames = [pt.split(os.path.sep)[-2] for pt in imagePaths]
classNames = [str(x) for x in np.unique(classNames)]

# initialize the image preprocessors

aap = AspectAwarePreprocessor(64, 64)

iap = ImageToArrayPreprocessor()

# load the dataset from disk then scale the raw pixel intensities to
# the range [0, 1]

sdl = SimpleDatasetLoader(preprocessors=[aap, iap])

(data, labels) = sdl.load(imagePaths, verbose=500)
```

```

data = data.astype("float") / 255.0

# partition the data into training and testing splits using 75% of
# the data for training and the remaining 25% for testing
(trainX, testX, trainY, testY) = train_test_split(data, labels,
                                                  test_size=0.25, random_state=42)

# convert the labels from integers to vectors
trainY = LabelBinarizer().fit_transform(trainY)
testY = LabelBinarizer().fit_transform(testY)

model = ResNet.build(64, 64, 3, 7, (3, 4, 6),
                    (64, 128, 256, 512), "dataset_tyrd6/images")

# loop over all layers in the base model and freeze them so they

print("[INFO] compiling model...")
opt = Adam(lr=0.001)
model.compile(loss="categorical_crossentropy", optimizer=opt,
              metrics=["accuracy"])

# start to become initialized with actual "learned" values

```

```

# versus pure random

print("[INFO] training head...")

model.fit_generator(aug.flow(trainX, trainY, batch_size=32),
                    validation_data=(testX, testY), epochs=20,
                    steps_per_epoch=len(trainX) // 32, verbose=200)

# evaluate the network after initialization

print("[INFO] evaluating after initialization...")

predictions = model.predict(testX, batch_size=32)

print(classification_report(testY.argmax(axis=1),
                             predictions.argmax(axis=1), target_names=classNames))

print("[INFO] re-compiling model...")

opt = SGD(lr=0.0001)

model.compile(loss="categorical_crossentropy", optimizer=opt,
              metrics=["accuracy"])

# train the model again, this time fine-tuning both the final set
# of CONV layers along with our set of FC layers

print("[INFO] fine-tuning model...")

H = model.fit_generator(aug.flow(trainX, trainY, batch_size=32),
                        validation_data=(testX, testY), epochs=15,
                        steps_per_epoch=len(trainX) // 32, verbose=1)

```

```

for key in H.history.keys():
    print(key)

# evaluate the network on the fine-tuned model
print("[INFO] evaluating after fine-tuning...")
predictions = model.predict(testX, batch_size=32)
print(classification_report(testY.argmax(axis=1),
    predictions.argmax(axis=1), target_names=classNames))

# save the model to disk
print("[INFO] serializing model...")
model.save(args["model"])

plt.style.use("ggplot")
plt.figure()

plt.plot(np.arange(0, 15), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, 15), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, 15), H.history["acc"], label="train_acc")
plt.plot(np.arange(0, 15), H.history["val_acc"], label="val_acc")
plt.title("Training Loss and Accuracy on PADDY")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend()
plt.savefig(args["output"])

```

