A Project Report

On

REAL TIME DETECTION OF FACE MASK FOR COVID SAFETY USING MACHINE LEARNING

Submitted in partial fulfillment of the requirement
for the award of the degree of

Bachelor of Technology

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

GALI NARESH	18755A0402
S TASNEEM FATHIMA	17751A04E3
V L YOGASREE	17751A04G2
ABRAHAM RICHIE	17751A04D0

Under the Guidance of

Mr.K.M.Hemambaran M.Tech (Ph.D)

ASSISTANT PROFESSOR, DEPT. OF ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES

(Autonomous)

Affiliated to J.N.T.U.A, Anantapuramu& NBA Accredited

CHITTOOR-517 127

(2020-2021)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES

(Autonomous)

Affiliated to J.N.T.U.A, Anantapuramu& NBA Accredited

CHITTOOR-517 127

CERTIFICATE

This is to certify that the project report entitled **"REAL TIME DETECTION OF FACE MASK FOR COVID SAFETY USING MACHINE LEARNING"** that is being submitted by GALI NARESH (18751A0402), S TASNEEM FATHIMA(17751A04E3),V L YOGASREE (17751A04G2), ABRAHAM RICHIE(17751A04D0) in partial fulfillment for the award of the Degree of **Bachelor of Technology** in **ELECTRONICS AND COMMUNICATION ENGINEERING** to the Jawaharlal Nehru Technological University Anantapuramu, Anantapuramu is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree.

Mr.K.M.Hemambaran, M.Tech(Ph.D)
Assistant Professor, Dept. of ECE
Internal Guide

Dr.M.Saravanan, M.E., Ph.D
Head of the Department
Dept. of ECE

Submitted for University Examination (Viva-Voice) held on_____

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

The successful completion of this project work was made possible with the help and guidance received from various quarters. We would like to avail this opportunity to express our sincere thanks and gratitude to all of them.

We would like to express the heartfelt gratitude to our Management,Chairperson **Smt.D.A.Sathyaprabha Garu** for providing excellent institutional support to complete this project.

We would like to express profound gratitude to our beloved principal **Dr.P.Ramesh Kumar, Ph.D** for giving us the opportunity to embark upon this course and make use of the facilities available in college.

We pay our sincere note of felicitation to **Dr.M.Saravanan, ME,Ph.D, Professor & Head** of the ECE Department for providing excellent departmental support as well as motivation to all of us which enhance our thrust to complete the project.

We thank our project guide **Mr.K.M.Hemambaran, M.Tech, (Ph.D)** for his valuable guidance, support and offering necessary material and willingly giving us advice whenever needed.

Finally we thank all the teaching faculty and non-teaching staff of the Department of Electronics and Communication Engineering for giving their advice and their help during the course of our project work as well as studies.

GALI NARESH	18755A0402
S TASNEEM FATHIMA	17751A04E3
V L YOGASREE	17751A04G2
ABRAHAM RICHIE	17751A04D0

COURSE OUTCOMES FOR PROJECT WORK

On completion of project work we will be able to,

CO1. Demonstrate in-depth knowledge on the project topic.

CO2. Identify, analyze and formulate complex problems chosen for project work to attain substantiated conclusions.

CO3. Design solutions to the chosen project problem.

CO4. Undertake investigation of project problems to provide valid conclusions.

CO5. Use the appropriate techniques, resources and modern engineering tools necessary for project work.

CO6. Apply project results for sustainable development of the society.

CO7. Understand the impact of project results in the context of environmental sustainability.

CO8. Understand professional and ethical responsibilities while executing the project work.

CO9. Function effectively as an individual and a member in the project team.

CO10. Develop communication skills, both oral and written for preparing and presenting project reports.

CO11. Demonstrate knowledge and understanding of cost and time analysis required for carrying out the project.

CO12. Engage in lifelong learning to improve knowledge and competence in the chosen area of the project.

CO – PO MAPPING

COs \ Pos	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	√											
CO2		√										
CO3			√									
CO4				√								
CO5					√							
CO6						√						
CO7							√					
CO8								√				
CO9									√			
CO10										√		
CO11											√	
CO12												√

Evaluation Rubrics for Project work:

Rubric (CO)	Excellent (Wt = 3)	Good (Wt = 2)	Fair (Wt = 1)
Selection of Topic (CO1)	Select the latest topic through complete knowledge of facts and concepts.	Select a topic through partial knowledge of facts and concepts.	Select a topic through improper knowledge of facts and concepts.
Analysis and Synthesis (CO2)	Thorough comprehension through analysis/ synthesis.	Reasonable comprehension through analysis/ synthesis.	Improper comprehension through analysis/ synthesis.
Problem Solving (CO3)	Thorough comprehension about what is proposed in the literature papers.	Reasonable comprehension about what is proposed in the literature papers.	Improper comprehension about what is proposed in the literature.
Literature Survey (CO4)	Extensive literature survey with standard references.	Considerable literature survey with standard references.	Incomplete literature survey with substandard references.
Usage of Techniques & Tools (CO5)	Clearly identified and has complete knowledge of techniques & tools used in the project work.	Identified and has sufficient knowledge of techniques & tools used in the project work.	Identified and has inadequate knowledge of techniques & tools used in project work.
Project work impact on Society (CO6)	Conclusion of project work has a strong impact on society.	Conclusion of project work has considerable impact on society.	Conclusion of project work has a feeble impact on society.
Project work impact on Environment (CO7)	Conclusion of project work has a strong impact on Environment.	Conclusion of project work has considerable impact on the environment.	Conclusion of project work has a feeble impact on the environment.

Ethical attitude (CO8)	Clearly understands ethical and social practices.	Moderate understanding of ethical and social practices.	Insufficient understanding of ethical and social practices.
Independent Learning (CO9)	Did literature survey and selected topic with a little guidance	Did literature survey and selected topic with considerable guidance	Selected a topic as suggested by the supervisor
Oral Presentation (CO10)	Presentation in logical sequence with key points, clear conclusion and excellent language	Presentation with key points, conclusion and good language	Presentation with insufficient key points and improper conclusion
Report Writing (CO10)	Status report with clear and logical sequence of chapters using excellent language	Status report with logical sequence of chapters using understandable language	Status report not properly organized
Time and Cost Analysis (CO11)	Comprehensive time and cost analysis	Moderate time and cost analysis	Reasonable time and cost analysis
Continuous learning (CO12)	Highly enthusiastic towards continuous learning	Interested in continuous learning	Inadequate interest in continuous learning

ABSTRACT

COVID-19 pandemic has rapidly affected our day-to-day life disrupting world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. Most recent and advanced face mask detection approaches are designed using deep learning. In this project we detect whether the person is wearing a mask or not. If the person wears a mask it says “Cheers wearing mask” else “please wear mask” Various libraries of python such as OpenCV, Tensor flow, Keras, speech recognition and gtts. In Deep Learning Convolution Neural Networks is a class Deep Neural Networks which is used to train the models used for this project.

CONTENTS

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Corona viruses are a family of viruses that can cause respiratory illness in humans. They get their name, “corona,” from the many crown-like spikes on the surface of the virus. Severe acute respiratory syndrome (SARS), Middle East respiratory syndrome (MERS) and the common cold are examples of corona viruses that cause illness in humans.

The new strain of coronavirus, COVID-19, was first reported in Wuhan, China in December 2019. The virus has since spread to all continents (except Antarctica).

1.1 Spreading of Coronavirus:

COVID-19 is likely spread:

- When the virus travels in respiratory droplets when an infected person coughs, sneezes, talks, sings or breathes near you (within six feet). This is thought to be the main way COVID-19 is spread.
- When the virus travels in small respiratory droplets that linger in the air for minutes to hours from an infected person who is more than six feet away or has since left the space. This method of spread is more likely to occur in enclosed spaces with poor ventilation.
- From close contact (touching, shaking hands) with an infected person.
- By touching surfaces that the virus has landed on, then touching your eyes, mouth, or nose before washing your hands. (Not thought to spread easily by this method.)

COVID-19 enters your body through your mouth, nose or eyes (directly from the airborne droplets or from transfer of the virus from your hands to your face). The virus travels to the back of your nasal passages and mucous membrane in the back of your throat. It attaches to cells there, begins to multiply and moves into lung tissue. From there, the virus can spread to other body tissues.

Governments, health agencies, researchers and healthcare providers are all working together to develop policies and procedures to limit the spread of this virus both globally and from individual to individual.

1.2. Symptoms:

The CDC says you may have coronavirus if you have these symptoms or combination of symptoms:

- Fever or chills.
- Cough.
- Shortness of breath or difficulty breathing.
- Tiredness.
- Muscle or body aches.
- Headaches.
- New loss of taste or smell.
- Sore throat.
- Congestion or runny nose.
- Nausea or vomiting.
- Diarrhea.

Additional symptoms are possible.

Symptoms may appear between two and 14 days after exposure to the virus. Children have similar, but usually milder, symptoms than adults. Older adults and people who have severe underlying medical conditions like heart or lung disease or diabetes are at higher risk of more serious complications from COVID-19.

1.3. Precautions to protect from COVID-19

Right now, the best defense to prevent getting COVID-19 is to follow some of the same steps you would take to prevent getting other viruses, such as the common cold or the flu.

- Wash your hands for at least 20 seconds— especially before eating and preparing food, after using the bathroom, after wiping your nose, and after coming in contact with someone who has a cold.

- Avoid touching your eyes, nose and mouth to prevent the spread of viruses from your hands.
- Cover your mouth and nose with a tissue when sneezing and coughing or sneeze and cough into your sleeve. Throw the tissue in the trash. Wash your hands afterward. Never cough or sneeze into your hands!
- Avoid close contact (within six feet) with those who have coughs, colds or are sick. Stay home if you are sick.
- If you are prone to sickness or have a weakened immune system, stay away from large crowds of people. Follow the directions of your healthcare authorities especially during outbreaks.
- Clean frequently used surfaces (such as door knobs and countertops) with a virus-killing disinfectant.
- Use hand sanitizers that contain at least 60% alcohol if soap and water are not available.
- Greet people with a friendly gesture instead of shaking hands.
- Get enough sleep, eat a healthy diet, drink plenty of liquids and exercise if you are able. These steps will strengthen your immune system and enable you to fight off infections easier.

1.3. Importance of face mask:

The CDC recommends wearing cloth face coverings in public, especially in places where it's hard to maintain at least six feet of distance between yourself and another person. Face masks protect both you and the people around you. Cloth face masks are being recommended because we now know individuals with COVID-19 could have mild or no symptoms, while still spreading the virus to others.

The cloth face coverings recommended by the CDC are not surgical masks or N-95 respirators, which should be reserved for healthcare workers and first responders

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

In earlier days face detection models were implemented using edge,line and centre near features and patterns are recognized from those features. These approaches are used to find binary patterns locally. These approaches are very effective to deal with Gray-scale images and the computation effort required also very less[1]

AdaBoost is a regression based classifier which is going to fit regression function on original data set even some miss classified objects waits also adjusted during back propagation to optimize the results[2]

Viola Jones Detector proposed a real time object model used to detect different classes of objects. It uses 24x24 base window size to evaluate any image with edge ,line and four rectangular features. Haar-like features are like convolutions to check whether a given feature is available in the image or not[3].This model fails to work when image brightness varies even if it exhibits poor performance when images are in different orientations.

Convolution networks are mainly used for classification problems there are various kinds of CNN architectures such as VGG-16 this architecture consists of 2 convolution layers with input size224 kernel (64,3x3) followed by max pool with size 2x2 then again two convolution layers followed by max pool then three convolution layers with max pool again three convolution layers and max pool and three fully connected layers final FC is soft max this architecture works fine when compared to AlexNet[4,7].

Google Net architecture fundamentally uses the inception method by constructing small convolution layers to reduce the number of parameters it has around 22 layers with convolution and max pooling etc . It can work effectively over Alexnet. It is able to bring down 60 million features in Alex net to 4 million features[8].

In this paper Deepnueral networks which adopts residual learning to train the models deeper around 152 layers are used in this which is 8 times more than VGGnet with minimum complexity. This approach achieved relatively better performance in object detection over COCO dataset[9]

In this paper UNet and SEnet are used to perform segmentation of heart ventricular segmentation . This model is arrange the weights in such a way like more weights are given to useful features and less weights are given to unimportant features.[10]

Support vector machines are used to perform classification on objects which is going to build an equation for the construction line and classifies the objects based on the values mapping to this line. Semantic segmentation method was used to detect facial masks in this paper they have used VGG net for training and FCN is used to semantically segment the faces available in the image[11] performed experiments on multi parsing human data sets and achieved higher accuracy.

In this paper medical image processing was done. They have taken human brain images and are trained by using FCN to identify tumours very effectively. In this paper rather than using 2D segmentation for detecting tumour we have used 3D segmentation[12].

Tumuluru,Lakshmi Ramaniet. al.[13], used CNN model to detect human face which is used efficiently in security related applications. In this paper they have collected various facial features such as mouth, nose, eyes stored as facial templates and used it for detecting differences between faces.

Malathi, J. et. al., [14] mainly focused on identifying forgery images used in different places like in social media, and other publicity required places. In this paper various techniques are proposed to find out features of a forgery image like image spicing,copy move attack which can be handled by using correlation analysis to find duplicate features.Platelet. al., [15]proposed a model to find out the quality of the iron ore by extracting the features from sample material in the mining industry. It is very important to assess the quality of the ore . SVR support vector regressor used for online measure of the quality of the ore. In this process they have extracted 280 features extracted for object identification, SFFS model was developed using SVR.

Object detection has become an important area in the field of image investigation. There are various techniques for image analysis in [15]. In this paper the author introduced a wavelet based neural network for feature extraction and learning which is working efficiently in object detection.

Satapathy, Sandeep Kumar, et al.,[16]proposed a model to detect number plates which is a very important problem helping police to chase many criminal cases. Authors used OCR based approach to detect characters in the number plate and they are stored and processed to client server based model for collecting the details of the owner.Pathak Et.al.,[17] proposed multi dimension biometric authentication system which will work effectively in low lighting conditions here accuracy was improved by using entropy based CNN.

Medical plant detection is becoming a very important problem which will help ordinary people to detect spices [18].In this paper authors proposed a model using CNN. It was trained with medical leaf images and it can able to detect medical plants more accurately.

Human pose detection is one of the important research are nowadays drawing lot of attention [19].In this paper author proposed a model which is going to detect traditional dance based on human pose. To achieve this, they have used CNN and various steps in traditional dance was trained and model learned from it and effectively able to detect traditional dance.

Ravi, Sunitha, et al. [20], sine language detection was implemented by training a CNN model which can able to detect signs in the real world video. Which is very much useful in the driver less cars also. Even it is useful in sign language in machine translation.

Joint angular displacement approach was used through CNN for further enhanceing the capabilities of CNN to capture 3D motion sign language in real time which can be applied to many real time applications in these days[21].Patel, Ashok Kumar et. al.,[22]proposed a model to find out the quality of the ironore by extracting the features from sample material in the mining industry. It is very important to asses the quality of the ore . SVR support vector regressor used for online measure of the quality of the ore. In this process they have extracted 280 features extracted for object identification, SFFS was model was developed using SVR.

CHAPTER 3

TECHNIQUES OF FACE DETECTION

TECHNIQUES OF FACE DETECTION

3.1 MACHINE LEARNING

Machine learning (ML) is a category of an algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available.

3.1.1 Importance of Machine Learning

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry.

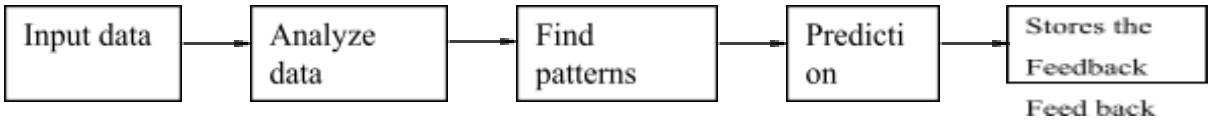
Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms**.**

Machine learning example – Google is able to provide you with appropriate search results based on browsing habits. Similarly, Netflix is capable of recommending the films or shows that you would want to watch based on the machine learning algorithms that perform predictions based on your watch history.

Furthermore, machine learning has facilitated the automation of redundant tasks that have taken away the need for manual labor. All of this is possible due to the massive amount of data that you generate on a daily basis. Machine Learning facilitates several methodologies to make sense of this data and provide you with accurate results.

3.1.2 Machine Learning Working

With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it.



These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

Next in Machine Learning tutorial are its types. Have a look –

3.1.3 Types of Machine Learning

Machine Learning Algorithms can be classified into 3 types as follows –

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised learning

In Supervised Learning, the dataset on which we train our model is labeled. There is a clear and distinct mapping of input and output. Based on the example inputs, the model is able to get trained in the instances. An example of supervised learning is spam filtering. Based on the labeled data, the model is able to determine if the data is spam or ham. This is an easier form of training. Spam filtering is an example of this type of machine learning algorithm.

Unsupervised Learning

In Unsupervised Learning, there is no labeled data. The algorithm identifies the patterns within the dataset and learns them. The algorithm groups the data into various clusters based on their density. Using it, one can perform visualization on high dimensional data. One example of this type of Machine learning algorithm is the Principle Component Analysis. Furthermore, K-Means Clustering is another type of Unsupervised Learning where the data is clustered in groups of a similar order.

The learning process in Unsupervised Learning is solely on the basis of finding patterns in the data. After learning the patterns, the model then makes conclusions.

Reinforcement Learning

Reinforcement Learning is an emerging and most popular type of Machine Learning Algorithm. It is used in various autonomous systems like cars and industrial robotics. The aim of this 54ralgorithm is to reach a goal in a dynamic environment. It can reach this goal based on several rewards that are provided to it by the system.

It is most heavily used in programming robots to perform autonomous actions. It is also used in making intelligent self-driving cars. Let us consider the case of robotic navigation. Furthermore, the efficiency can be improved with further experimentation with the agent in its environment. This is the main principle behind reinforcement learning. There are similar sequences of action in a reinforcement learning model.

3.2 Artificial Intelligence

“Artificial Intelligence is the science of machines do things that would require intelligence if done by man.”

That means, AI is not specifically related to computer science. This is a field of study that encompasses human behaviour, biology, psychology, and even language and linguistics. There’s still not a common consensus among academicians about its definition.

In this blog post, we try to give a broader picture of AI. How it is organized and it's various areas and fields of study. First we will discuss the terminologies associated with AI and then we will discuss the techniques used in implementing AI.

3.2.1 Techniques Used in AI

Myriads of AI techniques have emerged in the past decade for implementing and building AI systems.

1. Natural Language Processing

In a one-liner, natural language processing is the study of how a computer interacts with a human language. Broadly, in application sense, it refers to speech recognition and speech synthesis in human language.

This field of study is already in application phase and companies are using it in their voice assistants. Apple's Siri, Google Assistant, Microsoft's Crotona, and Amazon's Alex relies a lot on natural language processing.

Natural language processing further uses different techniques for implementation like parsing techniques, text recognition, and part-of-speech tagging.

2. (Artificial) Neural Networks

Neural networks are available in living beings. Humans and animals uses a complex network of billions of neurons (which makes neural systems) to take decisions in day-to-day life and learn new things to do. Building artificial neural networks is an attempt to create neural networks modelled on our own brains!

These networks can identify patterns in inputs as it processes a lot of data and learn from it. It uses different learning methods: supervised learning, unsupervised learning, and reinforced

learning. Neural networks have wide applications in pattern recognition, machine learning, and deep learning.

3. Vector machines

Vector machines are really capable in solving classification problems. For instance, an email system like Gmail for classifying an email as ‘Social’ or ‘Promotion’ or ‘Personal’ in nature and categorizing them in their respective categories.

The fundamental of vector (or sometimes called support vector) machines is to create parameters that draws the line between do distinct objects dividing them into two classes. This technique of AI has wide applications in image recognition, face recognition, and text recognition systems.

3.3 Neural Networks

You’ve probably already been using neural networks on a daily basis. When you ask your mobile assistant to perform a search for you—say, Google or Siri or Amazon Web—or use a self-driving car, these are all neural network-driven. Computer games also use neural networks on the back end, as part of the game system and how it adjusts to the players, and so do map applications, in processing map images and helping you find the quickest way to get to your destination.

A neural network is a system or hardware that is designed to operate like a human brain. Neural networks can perform the following tasks:

- Translate text
- Identify faces
- Recognize speech
- Read handwritten text
- Control robots

3.3.1 Working of Neural Network

A neural network is usually described as having different layers. The first layer is the input layer, it picks up the input signals and passes them to the next layer. The next layer does all kinds of calculations and feature extractions—it's called the hidden layer. Often, there will be more than one hidden layer. And finally, there's an output layer, which delivers the final result.

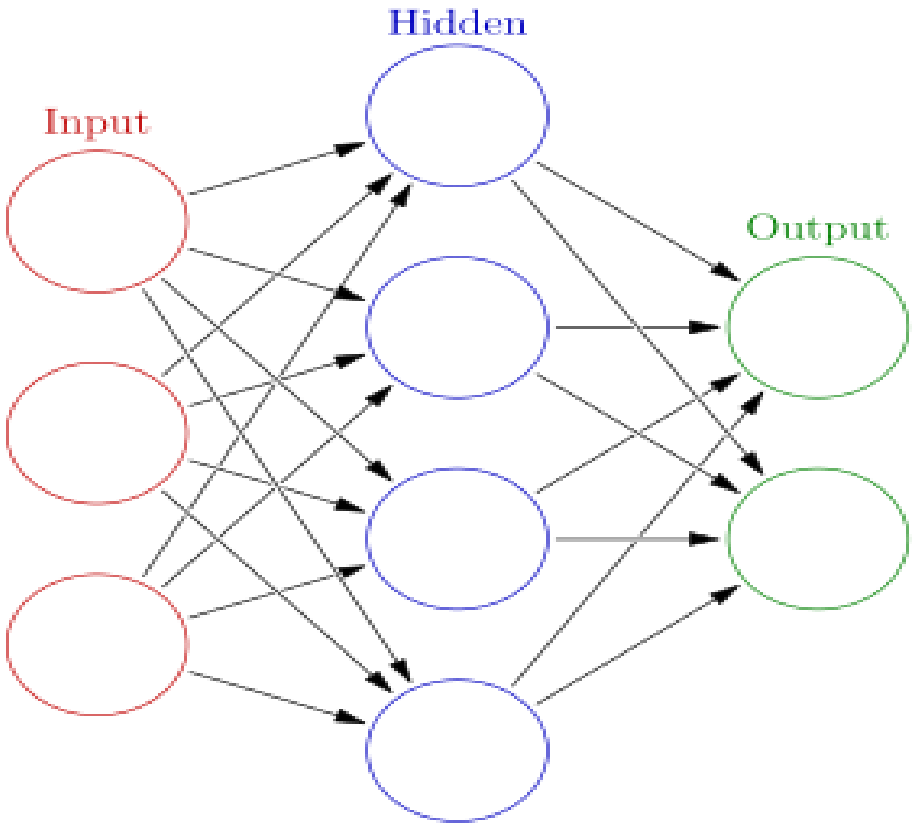


Fig 3.1 Simple Neural Networks

With the help of neural network learn things

Information flows through a neural network in two ways. When it's learning (being trained) or operating normally (after being trained), patterns of information are fed into the network via the input units, which trigger the layers of hidden units, and these in turn arrive at the output units.

This common design is called a feed forward network. Not all units "fire" all the time. Each unit receives inputs from the units to its left, and the inputs are multiplied by the weights of the connections they travel along. Every unit adds up all the inputs it receives in this way and (in the simplest type of network) if the sum is more than a certain threshold value, the unit "fires" and triggers the units it's connected to (those on its right).

For a neural network to learn, there has to be an element of feedback involved—just as children learn by being told what they're doing right or wrong. In fact, we all use feedback, all the time. Think back to when you first learned to play a game like ten-pin bowling. As you picked up the heavy ball and rolled it down the alley, your brain watched how quickly the ball moved and the line it followed, and noted how close you came to knocking down the skittles. Next time it was your turn, you remembered what you'd done wrong before, modified your movements accordingly, and hopefully threw the ball a bit better. So you used feedback to compare the outcome you wanted with what actually happened, figured out the difference between the two, and used that to change what you did next time ("I need to throw it harder,""I need to roll slightly more to the left,""I need to let go later," and so on). The bigger the difference between the intended and actual outcome, the more radically you would have altered your moves.

3.3.2 Types of Neural Networks

Different types of neural networks use different principles in determining their own rules. There are many types of artificial neural networks, each with their unique strengths. You can take a look at this video to see the different types of neural networks and their applications in detail.

Here are some of the most important types of neural networks and their applications.

1. **Feed Forward Neural Network**

This is one of the simplest types of artificial neural networks. In a feed forward neural network, the data passes through the different input nodes till it reaches the output node.

In other words, data moves in only one direction from the first tier onwards until it reaches the output node. This is also known as a front propagated wave which is usually achieved by using a classifying activation function.

Unlike in more complex types of neural networks, there is no back propagation and data moves in one direction only. A feed forward neural network may have a single layer or it may have hidden layers

2. **Radial Basic Function Neural Network**.

Bottom of Form

A radial basis function considers the distance of any point relative to the centre. Such neural networks have two layers. In the inner layer, the features are combined with the radial basis function.

Then the output of these features is taken into account when calculating the same output in the next time-step.

The radial basis function neural network is applied extensively in power restoration systems. In recent decades, power systems have become bigger and more complex.

This increases the risk of a blackout. This neural network is used in the power restoration systems in order to restore power in the shortest possible time.

3. Multilayer perceptron

A multilayer perceptron has three or more layers. It is used to classify data that cannot be separated linearly. It is a type of artificial neural network that is fully connected. This is because every single node in a layer is connected to each node in the following layer.

A multilayer perceptron uses a nonlinear activation function (mainly hyperbolic tangent or logistic function).This type of neural network is applied extensively in speech recognition and machine translation technologies.

4. Convolution Neural Network

A convolution neural network (CNN) uses a variation of the multilayer perceptron. A CNN contains one or more than one convolution layers. These layers can either be completely interconnected or pooled.

Before passing the result to the next layer, the convolution layer uses a convolution operation on the input. Due to this convolution operation, the network can be much deeper but with much fewer parameters.

Due to this ability, convolution neural networks show very effective results in image and video recognition, natural language processing, and recommender systems.

Convolution neural networks also show great results in semantic parsing and paraphrase detection. They are also applied in signal processing and image classification.

CNNs are also being used in image analysis and recognition in agriculture where weather features are extracted from satellites like LSAT to predict the growth and yield of a piece of land.

5. Recurrent Neural Network

A Recurrent Neural Network is a type of artificial neural network in which the output of a particular layer is saved and fed back to the input. This helps predict the outcome of the layer. The first layer is formed in the same way as it is in the feed forward network. That is, with the product of the sum of the weights and features. However, in subsequent layers, the recurrent neural network process begins. From each time-step to the next, each node will remember some information that it had in the previous The neural network begins with the front propagation as usual but remembers the information it may need to use later. This type of neural network is very effective in text-to-speech conversion technology.

6. Modular Neural Network

A modular neural network has a number of different networks that function independently and perform sub-tasks. The different networks do not really interact with or signal each other during the computation process. They work independently towards achieving the output.

As a result, a large and complex computational process can be done significantly faster by breaking it down into independent components. The computation speed increases because the networks are not interacting with or even connected to each other.

CHAPTER 4

EXISTING METHOD AND PROPOSED METHOD

EXISTING METHOD AND PROPOSED METHOD

4.1EXISTING METHOD:

4.1.1 ARTIFICIAL NEURAL NETWORK:

Single perceptron (or neuron) can be imagined as a Logistic Regression. Artificial Neural Network, or ANN, is a group of multiple perceptron/ neurons at each layer. ANN is also known as a Feed-Forward Neural network because inputs are processed only in the forward direction:

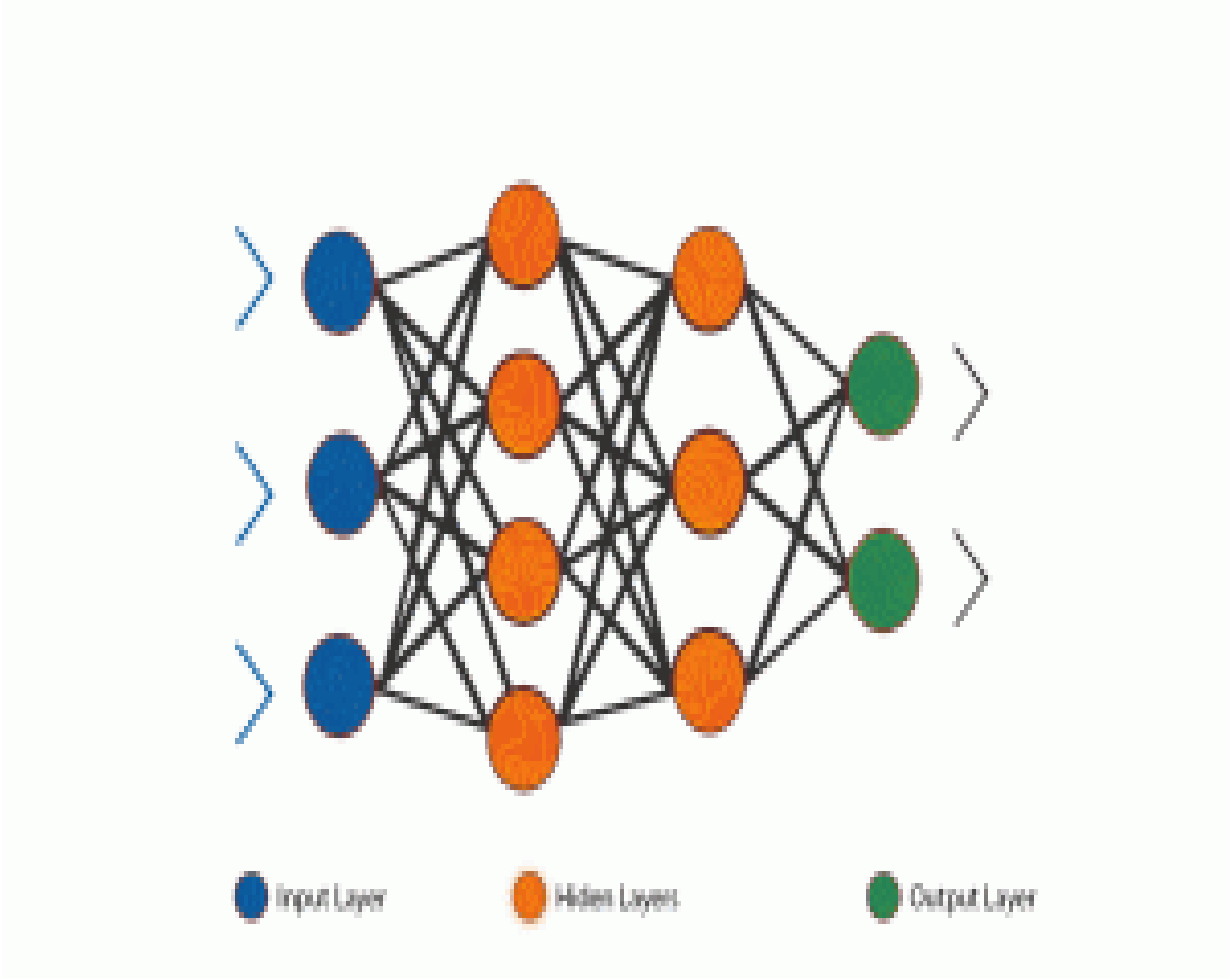


Fig 4.1 Layers ofANN

As you can see here, ANN consists of 3 layers – Input, Hidden and Output. The input layer accepts the inputs, the hidden layer processes the inputs, and the output layer produces the result. Essentially, each layer tries to learn certain weights.

Drawbacks of Artificial Neural Network (ANN)

- While solving an image classification problem using ANN, the first step is to convert a 2-dimensional image into a 1-dimensional vector prior to training the model. This has two drawbacks:
- The number of trainable parameters increases drastically with an increase in the size of the image
- Let us consider the size of the image is 224*224, then the number of trainable parameters at the first hidden layer with just 4 neurons is 602,112. That's huge!

- ANN loses the spatial features of an image. Spatial features refer to the arrangement of the pixels in an image. I will touch upon this in detail in the following sections

- One common problem in all these neural networks is the Vanishing and Exploding Gradient. This problem is associated with the back propagation algorithm. The weights of a neural network are updated through this back propagation algorithm by finding the gradients:

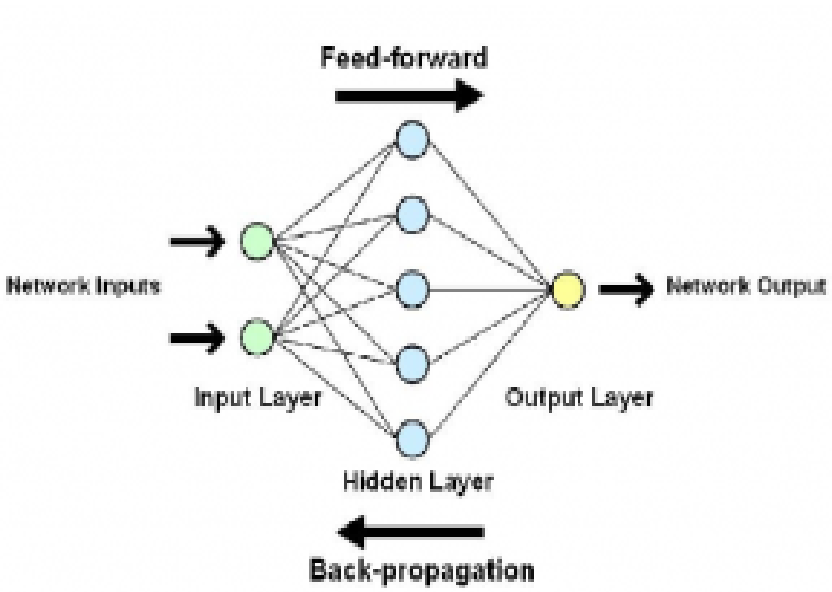


Fig 4.2 Backward Propagation

So, in the case of a very deep neural network (network with a large number of hidden layers), the gradient vanishes or explodes as it propagates backward which leads to vanishing and exploding gradient.

- ANN cannot capture sequential information in the input data which is required for dealing with sequence data.

In order overcome the disadvantage of ANN we go for CNN the advantages of this are

- Parameter sharing
- Sparsity of connection

4.2 PROPOSED METHOD:

CONVOLUTION NEURAL NETWORK:

The convolution neural network, or CNN or ConvNet for short, is a specialized type of neural network model designed for working with two-dimensional image data, although they can be used with one-dimensional and three-dimensional data.

It's a deep, feed-forward artificial neural network. Feed-forward neural networks are also called multi-layer perceptron (MLPs). The models are called "feed-forward" because information flows right through the model. There are no feedback connections in which outputs of the model are fed back into itself. CNN use many filters to extract features the first layers learn basic feature detection filters: edges, corners, etc after that the middle layers learn filters that detect parts of objects finally the last layers have higher representations: they learn to recognize full objects, in different shapes and positions. There are two types of pooling techniques Max pooling and Average pooling.

These neural networks have proven to be successful in many different real-life case studies and applications, like:

- Image classification, object detection, segmentation, face recognition;
- Self-driving cars that leverage CNN based vision systems;

4.2.1 CNN ARCHITECTURE:

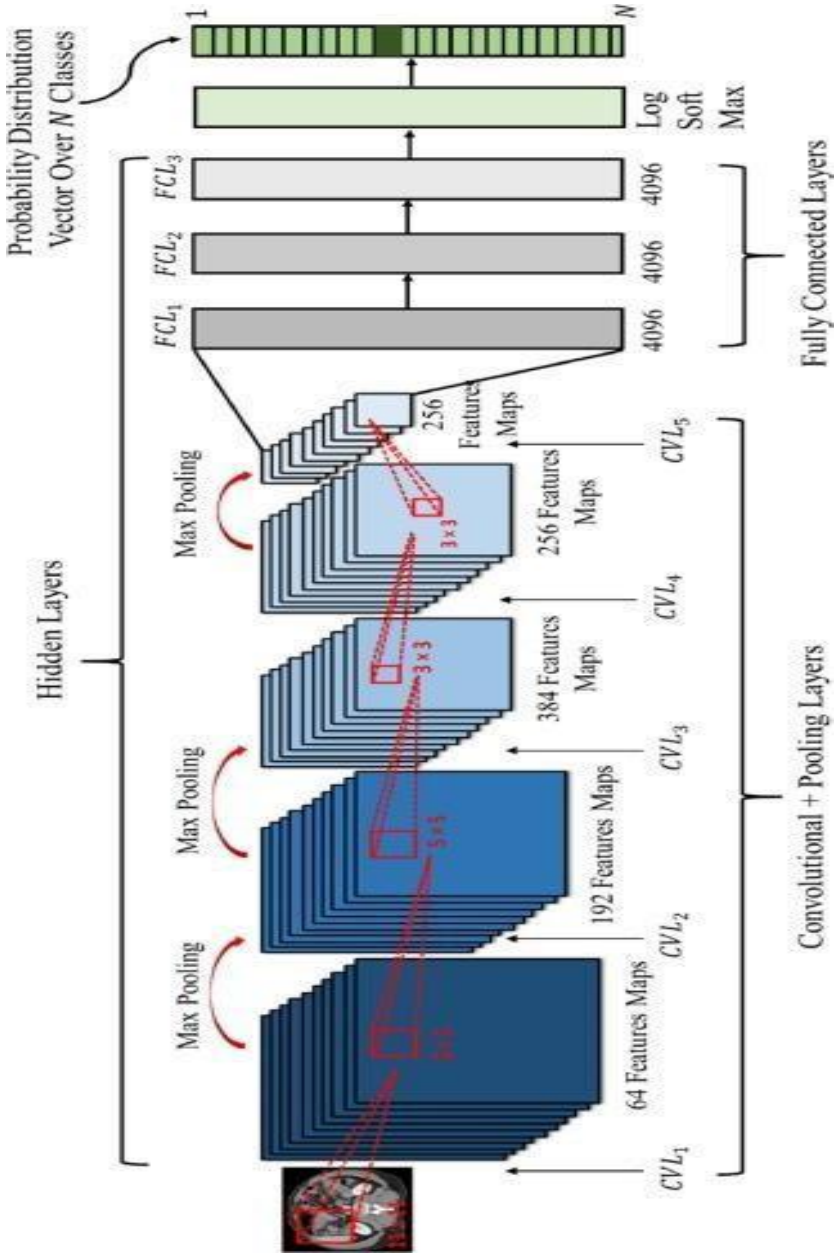


Fig 4.3 CNN Architecture

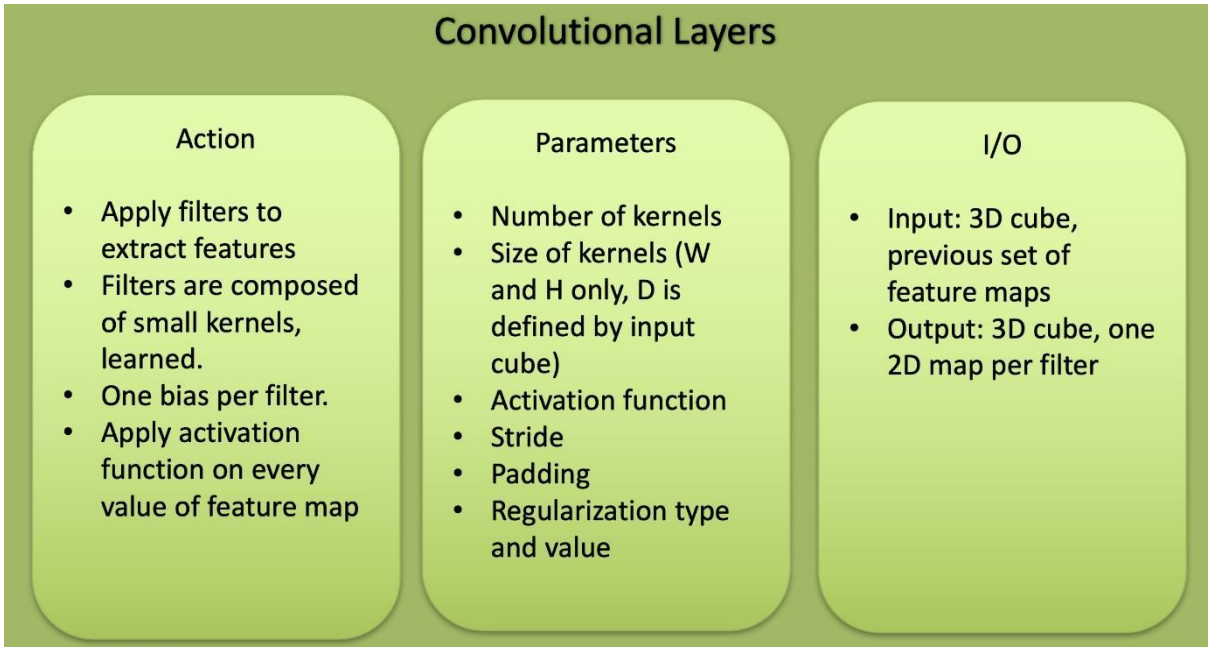
There are five types of layers in a convolutional neural network:

1. Convolution layer.
2. Rectified linear unit layer (ReLU).

3. Sub sampling layer/Pooling layer.
4. Fully connected layer.
5. SoftMax layer.

Each of these layers has different parameters that can be optimized and performs a different task on the input data.

4.2.2 CONVOLUTIONAL LAYER:



Central to the convolutional neural network is the convolutional layer that gives the network its name. This layer performs an operation called a "*convolution* ". In the context of a convolutional neural network, a convolution is a linear operation that involves the multiplication of a set of weights with the input, much like a traditional neural network. Given that the technique was designed for two-dimensional input, the multiplication is performed between an array of input data and a two-dimensional array of weights, called a filter or a kernel.

The filter is smaller than the input data and the type of multiplication applied between a filter-sized patch of the input and the filter is a dot product.

Adotproducts the element-wise multiplication between the filter-sized patch of the input and filter, which is then summed, always resulting in a single value. Because it results in a single value, the operation is often referred to as the "*scalar product* ".

Using a filter smaller than the input is intentional as it allows the same filter (set of weights) to be multiplied by the input array multiple times at different points on the input. Specifically, the filter is applied systematically to each overlapping part or filter-sized patch of the input data, left to right, top to bottom. NxN input layer is convoluted with MxM filter. Then, the convolutional layer output will be of size (N-m+1) x(N-m+1).

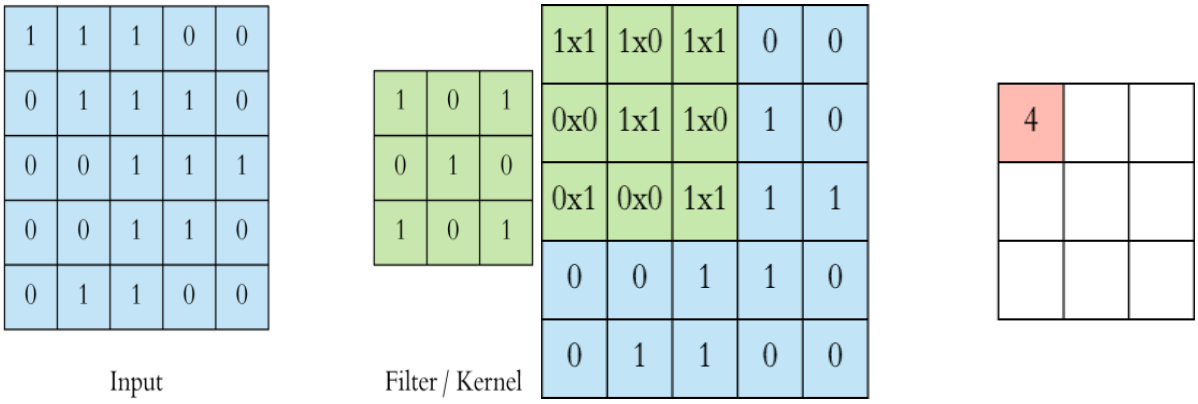


Fig 4.3 Working of Convolutional layer

4.2.3 RECTIFIED LINEAR UNIT:

Generally, Convolution is linear operation, in real world everything is nonlinear. For this non-linearity we use Rectified Linear unit (Re LU).

The weighted sum of the convolution operation is passed through ReLU activation function. So, the values in the final feature maps are not actually the sums, but the ReLU function applied to sum to achieve original sum of feature map.

$$f(x)=\max(0, x)$$

Graphically it looks like,

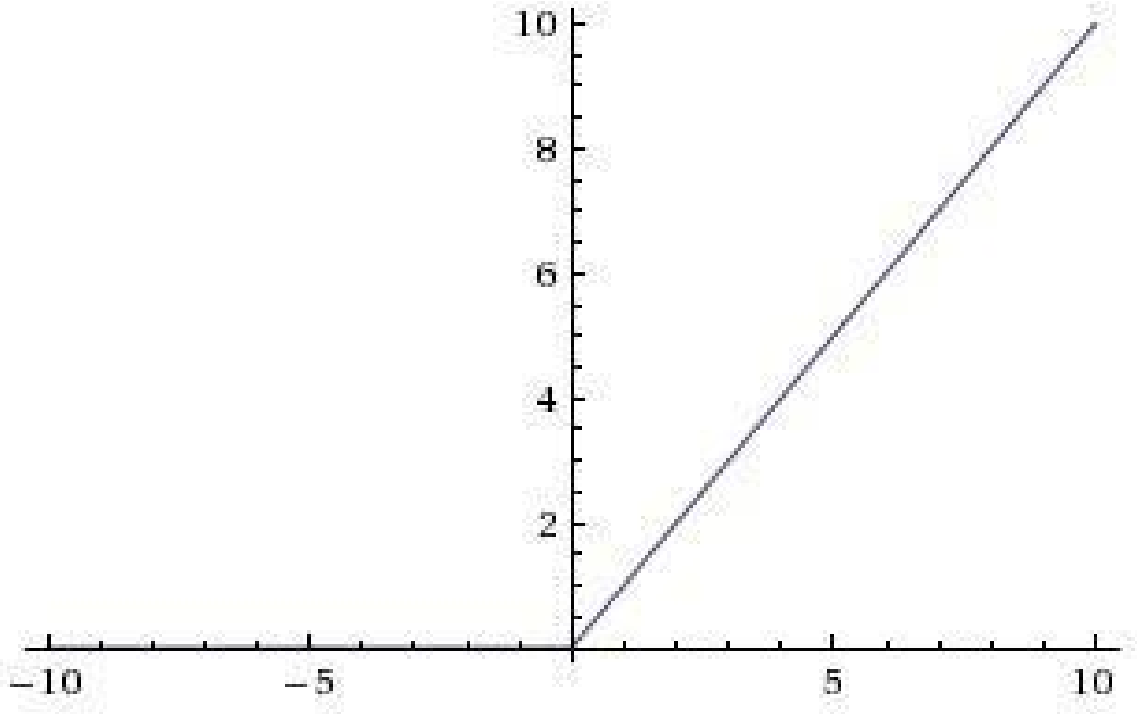


Fig 4,4 Graph of ReLU

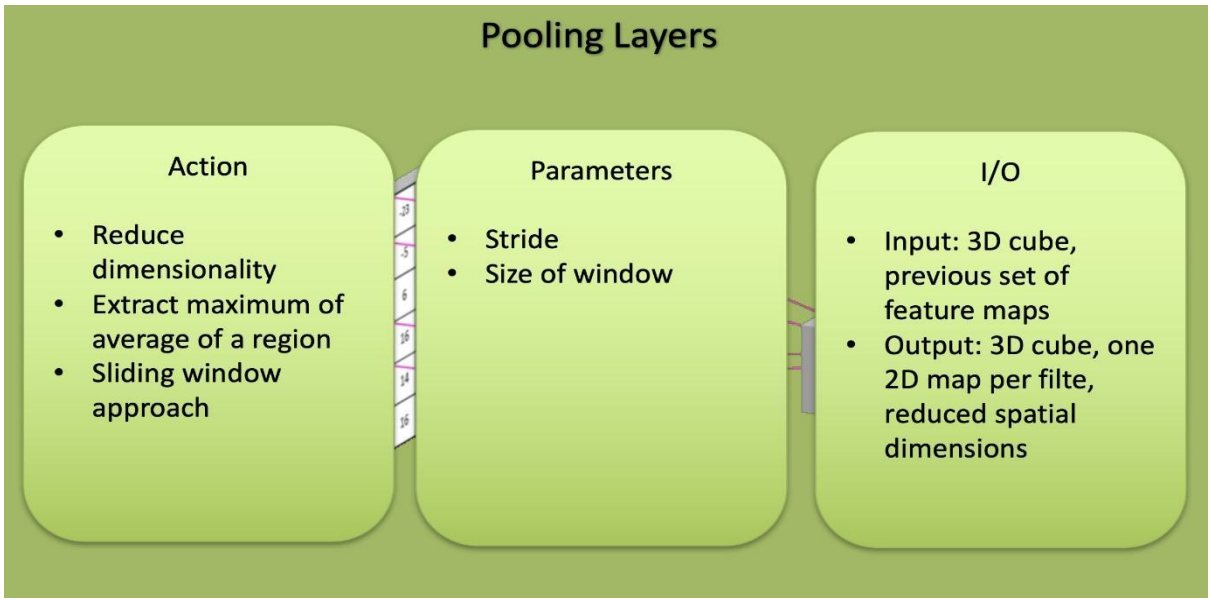
4.2.4 SUB SAMPLING LAYER:

This Sub sampling layer uses pooling technique. There are two types in pooling layer. They are:

1. Max pooling

2. Average pool

In this Max pooling is used. Max pooling selects the maximum pixel intensity from the image.



The objective of subsampling is to get an input representation by reducing its dimensions, which helps in reducing overfitting. One of the techniques of subsampling is max pooling. With this technique, highest pixel value is selected from a region depending on its size. This technique takes the largest value from the window of the image currently covered by the filter.

For example, if NxN input layer, that will give output layer of $\frac{N}{K} X \frac{N}{K}$ layer. The only difference is the function that is applied to the filter and the image window isn't linear.

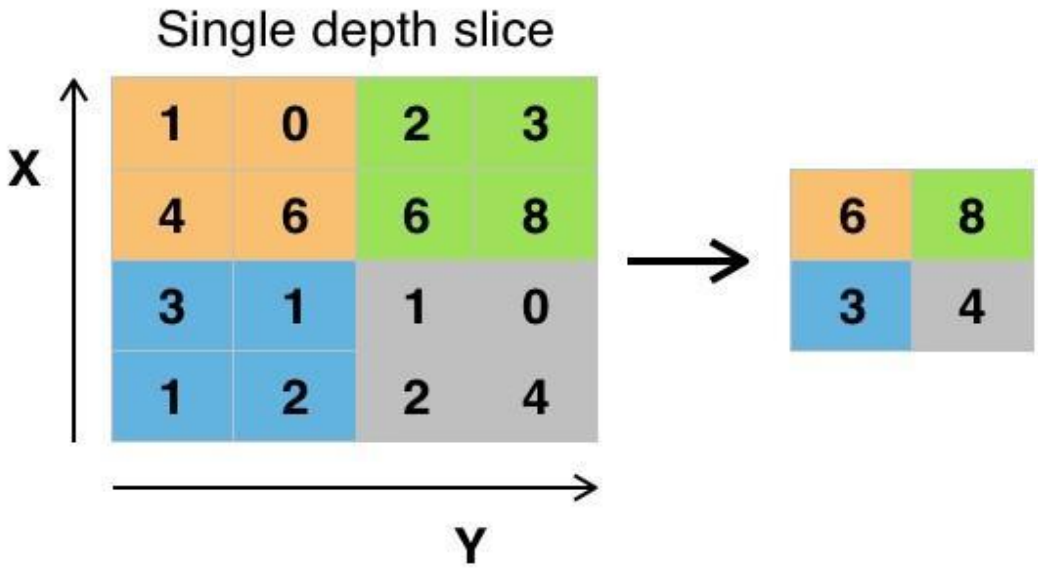
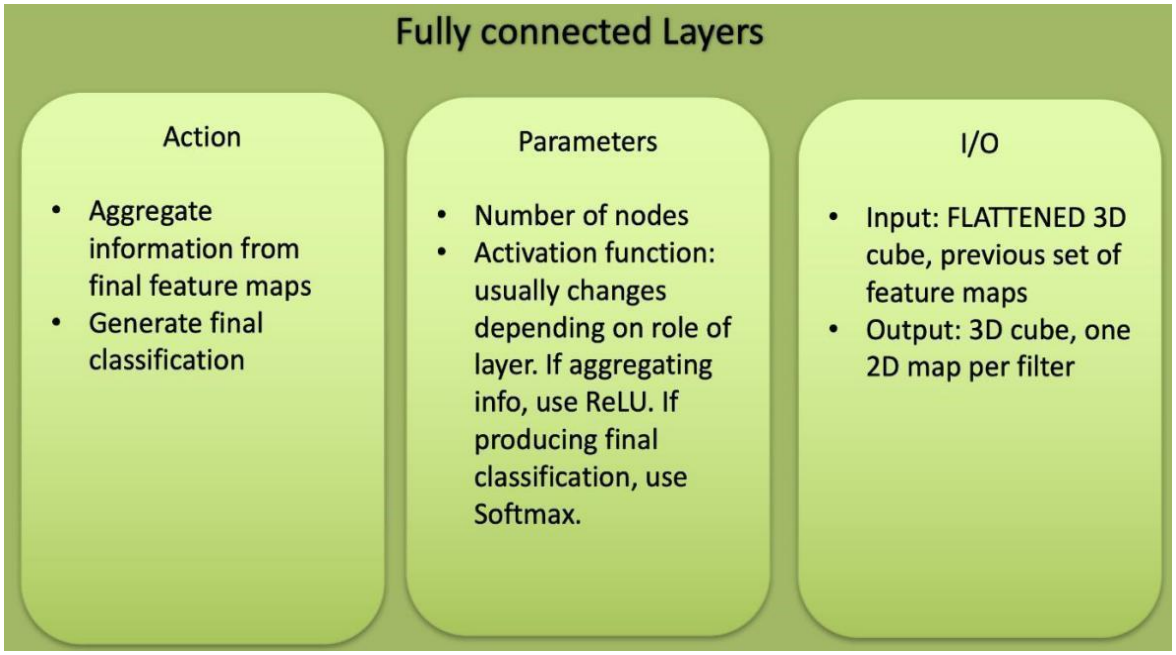


Fig 4.5 Working of Sampling Layer

4.2.5 FULLY CONNECTED LAYER

The objective of the fully connected layer is to flatten the high-level features that are learned by convolutional layers and combining all the features. It passes the flattened output to the output layer. A fully connected layer takes the entire previous layer from max-pooling layer and connects it to every node in fully connected layer it has. Fully connected layers are not spatially connected anymore. It converts 2D or 3D images to 1D and it visualize the image as one-dimensional layer.



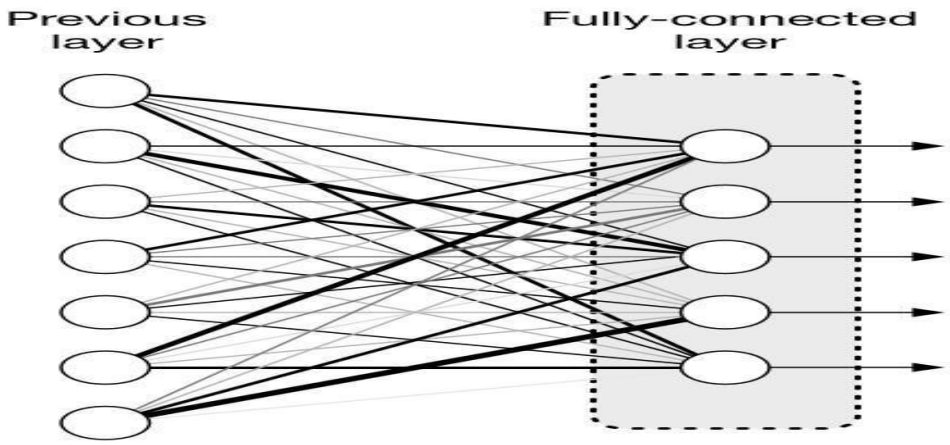


Fig 4.6 Fully Connected Layer

4.2.6 SOFTMAX LAYER

The objective of the softmax classification layer is simply to transform all the activations of final output layer to a series of values that can be interpreted as probabilities. This function will calculate the probabilities of each target class over all possible target classes. The main advantage of using Softmax is the output probabilities range.

CHAPTER 5

SOFTWARE REQUIREMENTS

Software Requirements

5.1 ANACONDA

It is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

Anaconda distribution comes with more than 1,500 packages as well as the Conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the Command Line Interface (CLI).

The big difference between Conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason Conda exists. Pip installs all Python package dependencies required, whether or not those conflict with other packages you installed previously. So your working installation of, for example, Google TensorFlow, can suddenly stop working when you pip install a different package that needs a different version of the NumPy library. More insidiously, everything might still appear to work but now you get different results from your data science, or you are unable to reproduce the same results elsewhere because you didn't pip install in the same order.

Conda analyzes your current environment, everything you have installed, any version limitations you specify (e.g. you only want TensorFlow>= 2.0) and figures out how to install compatible dependencies. Or it will tell you that what you want can't be done. Pip, by contrast, will just install the thing you wanted and any dependency, even if that breaks other things. Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or your own private repository or mirror, using the conda install command. Anaconda Inc compiles and builds all the packages in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. You can also install anything on PyPI into a Conda environment using pip, and Conda knows what it has installed and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with

conda.

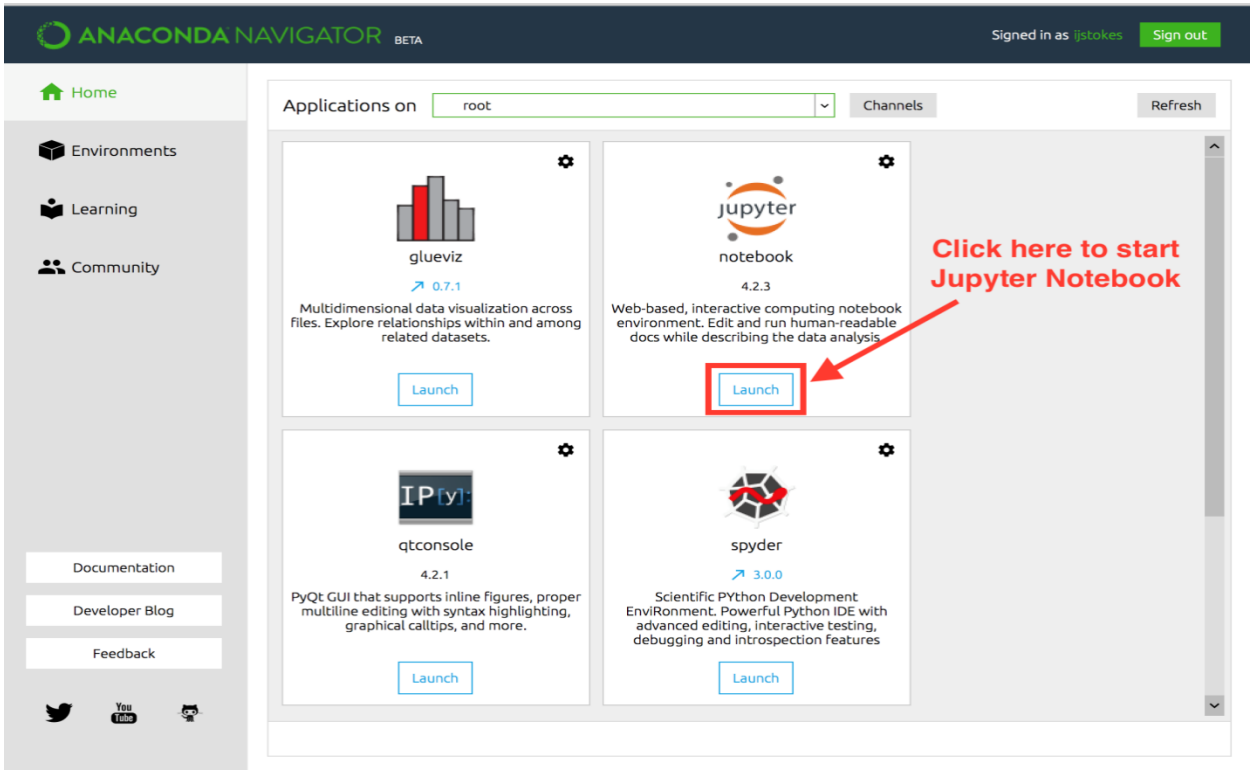


Fig 5.1 Anaconda Navigator

Anaconda Navigator is a desktop Graphical User Interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glueviz
- Orange
- Rstudio
- Visual Studio Code

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

".NET" is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

Microsoft VISUAL STUDIO is an Integrated Development Environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

Python is a powerful multi-purpose programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

Python features are:

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted
- Large Standard Library
- Dynamically Typed Language

5.2 PYTHON

- Python is a powerful multi-purpose programming language created by Guido van Rossum.
- It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

5.2.1 FEATURES OF PYTHON

1. Easy to code

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

2. Free and Open Source

Python language is freely available at official website and you can download it from the given download link below click on the Download Python keyword.

Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3. Object-Oriented Language

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

4. GUI Programming Support

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python.

PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature

Python is a **Extensible** language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language

Python language is also a portable language. For example, if we have python code for windows

and if we want to run this code on other platform such as Linux, UNIX and Mac then we do not need to change it, we can run this code on any platform.

8. Python is integrated language

Python is also an integrated language because we can easily integrated python with other language like c, c++ etc.

9. Interpreted Language

Python is an Interpreted Language. Because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing.There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. Because of this feature we don't need to specify the type of variable.

5.2.2 APPLICATIONS OF PYTHON

WEB APPLICATIONS

- You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS.
- Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

SCIENTIFIC AND NUMERIC COMPUTING

- There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on.

- Also, the language is heavily used in machine learning, data mining and deep learning.

CREATING SOFTWARE PROTOTYPES

- Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.
- However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

GOOD LANGUAGE TO TEACH PROGRAMMING

- Python is used by many companies to teach programming to kids and newbie's.
- It is a good language with a lot of features and capabilities. Yet, it's one of the easiest languages to learn because of its simple easy-to-use syntax.

In order to train the images and proceed for convolution neural network(CNN) in python we have to install some of the libraries like Tensor flow, Keras, Pytorch,etc.,

5.3 TensorFlow

About TensorFlow

Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

- Researchers
- Data scientists
- Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so Tensor Flow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

History of TensorFlow

A couple of years ago, deep learning started to outperform all other machine learning algorithms when giving a massive amount of data. Google saw it could use these deep neural networks to improve its services:

- Gmail
- Photo
- Google search engine

They build a framework called Tensorflow to let researchers and developers work together on an AI model. Once developed and scaled, it allows lots of people to use it.

It was first made public in late 2015, while the first stable version appeared in 2017. It is open source under Apache Open Source license. You can use it, modify it and redistribute the modified version for a fee without paying anything to Google.

TensorFlow Architecture

Tensorflow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

It is called Tensorflow because it takes input as a multi-dimensional array, also known as tensors. You can construct a sort of flowchart of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

TensorFlow is the most famous deep learning library these recent years. A practitioner using TensorFlow can build any deep learning structure, like CNN, RNN or simple artificial neural network.

TensorFlow is mostly used by academics, startups, and large companies. Google uses TensorFlow in almost all Google daily products including Gmail, Photo and Google Search Engine.

5.4Keras

Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks.

Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

Features

Keras leverages various optimization techniques to make high level neural network API easier and more per formant. It supports the following features −

- Consistent, simple and extensible API.
- Minimal structure - easy to achieve the result without any frills.
- It supports multiple platforms and back ends.
- It is user friendly framework which runs on both CPU and GPU.
- Highly scalability of computation.

Benefits

Keras is highly powerful and dynamic framework and comes up with the following advantages

- Larger community support.
- Easy to test.
- Keras neural networks are written in Python which makes things simpler.
- Keras supports both convolution and recurrent networks.
- Deep learning models are discrete components, so that, you can combine into many ways.

5.4.1 Keras Installation Steps

Keras installation is quite easy. Follow below steps to properly install Keras on your system.

Step 1: Create virtual environment

Virtualenv is used to manage Python packages for different projects. This will be helpful to avoid breaking the packages installed in the other environments. So, it is always recommended to use a virtual environment while developing Python applications.

Windows

Windows user can use the below command,

py -m venv Keras

Step 2: Activate the environment

This step will configure python and pip executables in your shell path.

Windows

Windows users move inside the “kerasenv” folder and type the below command,

.\env\Scripts\activate

Step 3: Python libraries

Keras depends on the following python libraries.

- Numpy
- Pandas
- Scikit-learn
- Matplotlib
- Scipy
- Seaborn

Hopefully, you have installed all the above libraries on your system. If these libraries are not installed, then use the below command to install one by one.

numpy

pip install numpy

pandas

pip install pandas

matplotlib

pip install matplotlib

scipy

pip install scipy

scikit-learn

It is an open source machine learning library. It is used for classification, regression and clustering algorithms. Before moving to the installation, it requires the following −

- Python version 3.5 or higher
- NumPy version 1.11.0 or higher
- SciPy version 0.17.0 or higher
- joblib 0.11 or higher.

Keras Installation Using Python

As of now, we have completed basic requirements for the installation of Keras. Now, install the Keras using same procedure as specified below −

pip install Keras

Quit virtual environment

After finishing all your changes in your project, then simply run the below command to quit the environment −

Deactivate

Anaconda Cloud

We believe that you have installed anaconda cloud on your machine. If anaconda is not installed, then visit the official link, www.anaconda.com/distribution and choose download based on your OS.

Create a new conda environment

Launch anaconda prompt, this will open base Anaconda environment. Let us create a new conda environment. This process is similar to virtualenv. Type the below command in your conda terminal −

conda create --name PythonCPU

If you want, you can create and install modules using GPU also. In this tutorial, we follow CPU instructions.

Activate conda environment

To activate the environment, use the below command −

activate PythonCPU

Install spyder

Spyder is an IDE for executing python applications. Let us install this IDE in our conda environment using the below command −

conda install spyder

Install python libraries

We already know the python libraries numpy, pandas, etc., needed for keras. You can install all the modules by using the below syntax −

Syntax

conda install -c anaconda <module-name>

For example, you want to install pandas −

conda install -c anaconda pandas

Like the same method, try it yourself to install the remaining modules.

Install Keras

Now, everything looks good so you can start Keras installation using the below command −

conda install -c anaconda Keras

5.5.Imutils:

5.5.1.Definition:

A series of convenience functions to make basic image processing operations such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and Python.

5.5.2. Installation:

pip install imutils

Note: Installing this package depends on NumPy, Opencv, and matplotlib

5.6.Open CV:

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.

5.6.1. Installation:

pip install opencv-python

5.7. Playsound:

Pure Python, cross platform, single function module with no dependencies for playing sounds.

5.7.1. Installation:

Install via pip:

$ pip install playsound

5.8. Speech recognition:

Speech recognition is the process of converting spoken words to text. Python supports many speech recognition engines and APIs, including Google Speech Engine, Google Cloud Speech API, Microsoft Bing Voice Recognition and IBM Speech to Text.

5.8.1: Installation:

Pip install speech recognition

CHAPTER 6

RESULTS

Results:
CASE (1):

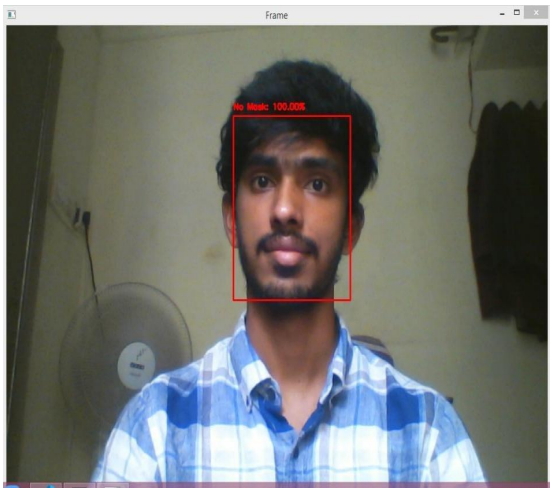


Figure 6.1(a): Result while not Wearing mask

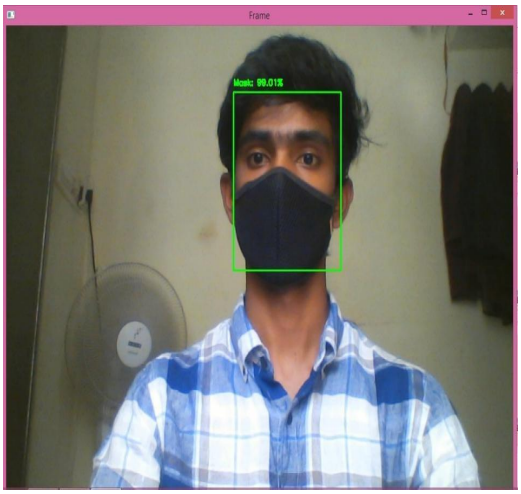
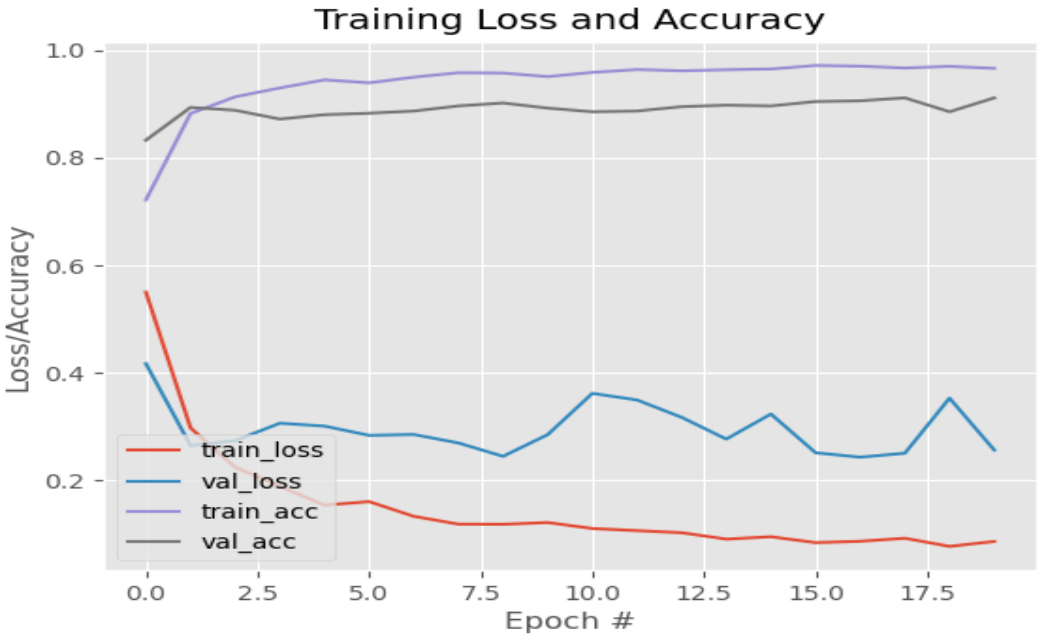


Figure 6.1 (b): Result while wearing mask

In this case if a person wears mask it shows mask with accuracy on the top of the frame which showing in green color and it says “cheers, wearing mask”. And if a person didn’t wear mask it shows no mask with accuracy on the top of the frame in red color and it says “please wear mask”.



CASE (2):

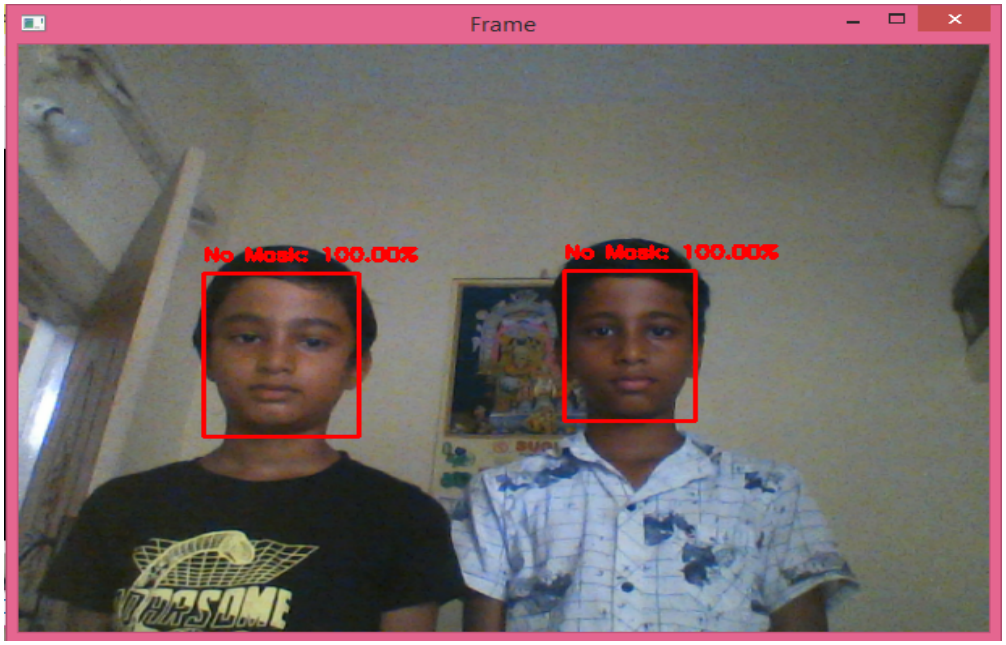


Figure 6.2: Results while both are not wearing mask

In case 2 both the persons are not wearing mask. So it shows no mask with accuracy. Audio doesn't work for more than one person.

CASE (3):

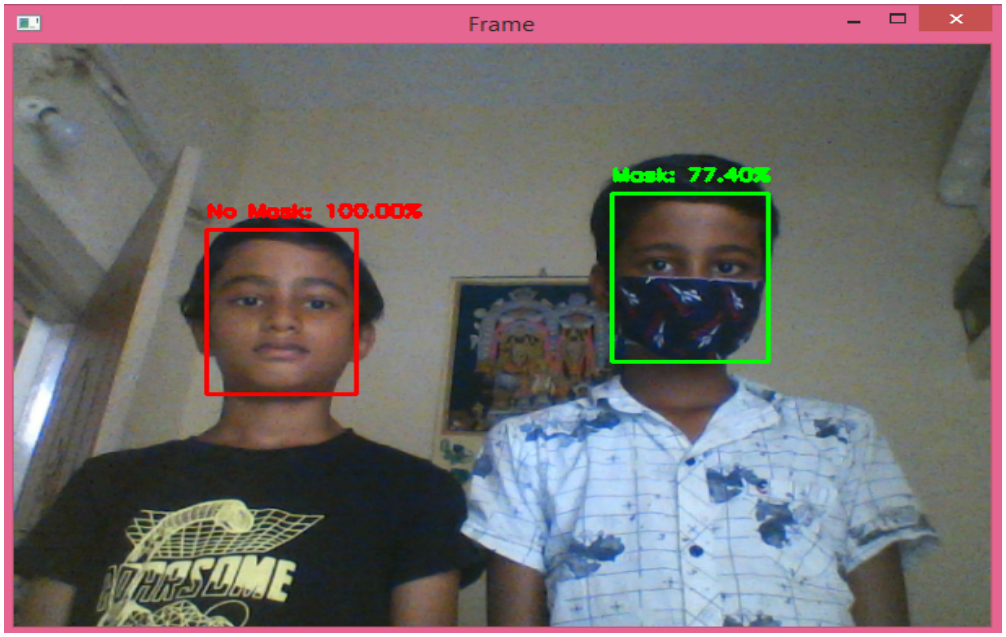


Figure 6.3: Results while one is wearing mask and other not wearing mask

In case 3 one person wearing mask, it shows mask with accuracy. And other person don't wearing, it shows no mask with accuracy. Audio doesn't work for

more than one person.

CASE (4):

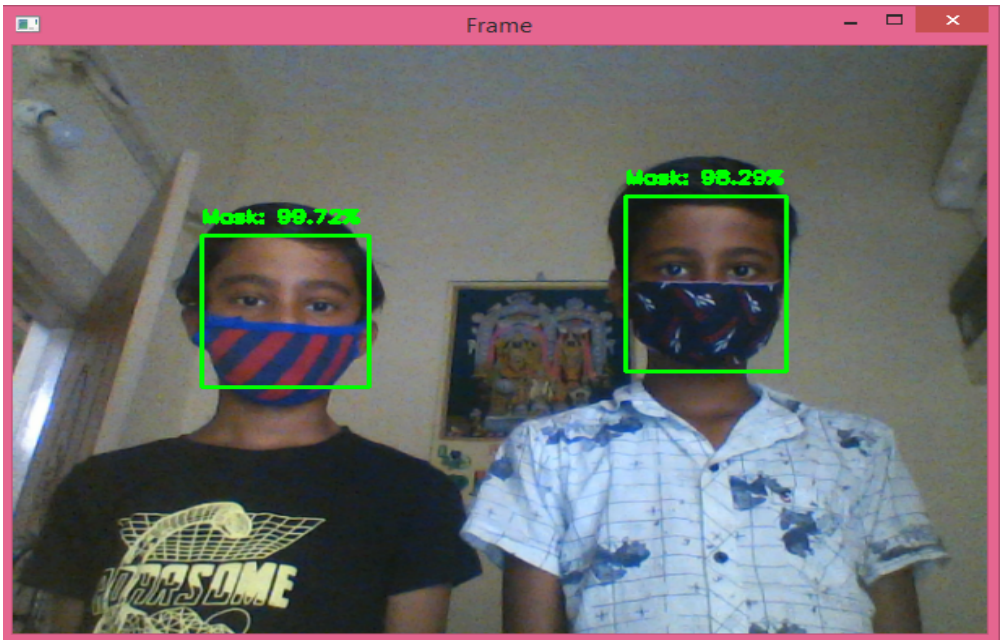


Figure 6.4: Result while both are wearing mask

In case 4 both the persons are wearing mask. So it shows mask with accuracy. Audio doesn't work for more than one person.

CASE (5):

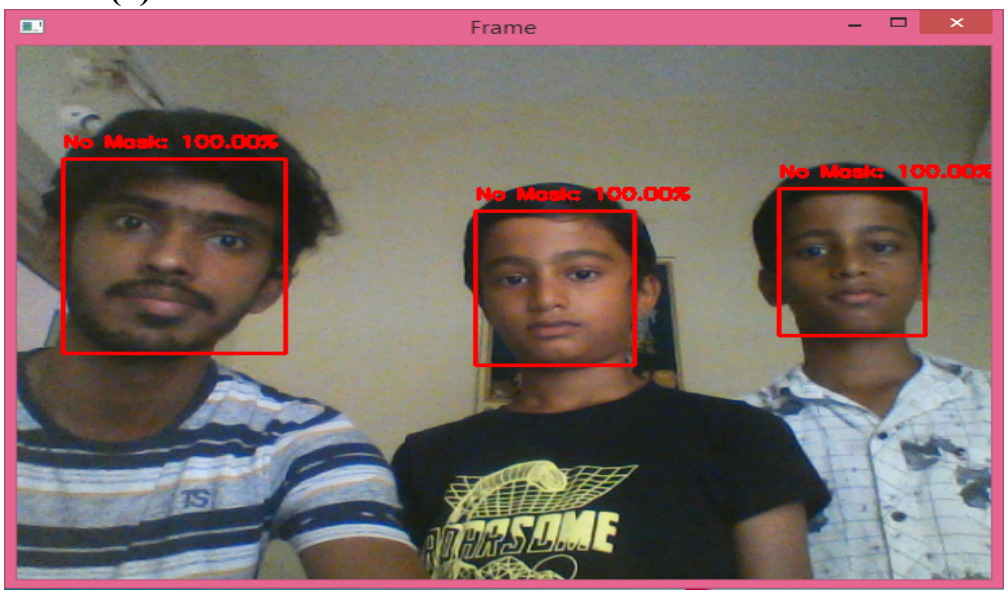


Figure 6.5: Result showing when no one is wearing mask

In case 5 no one is wearing mask. So it shows no mask and accuracy for everyone. Audio doesn't work for more than one person.

CASE (6):

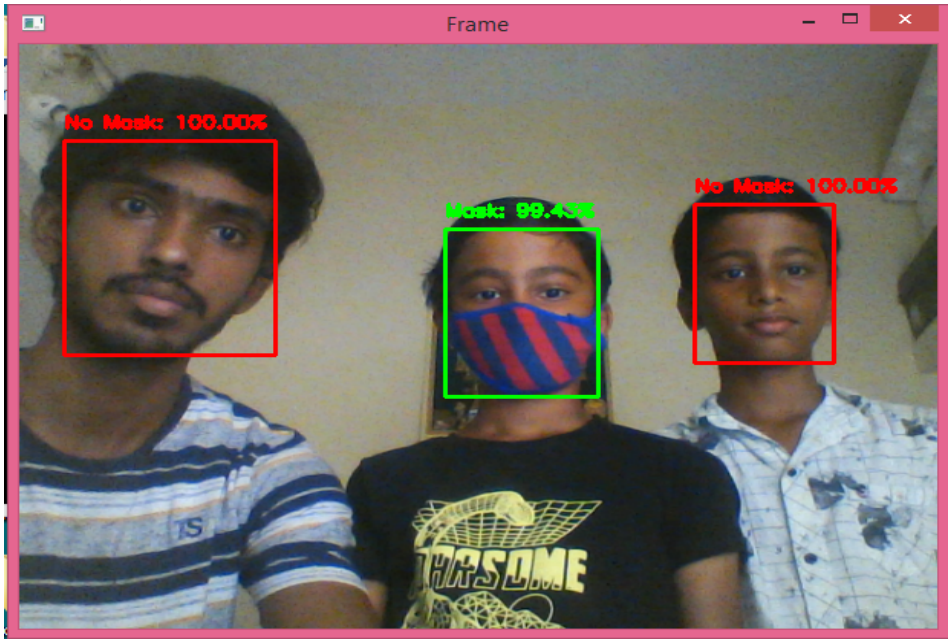


Figure 6.6: Result showing when only one person wearing mask

In case 6 only one person wears a mask for others no masks. Audio doesn't work for more than one person.

CASE (7):

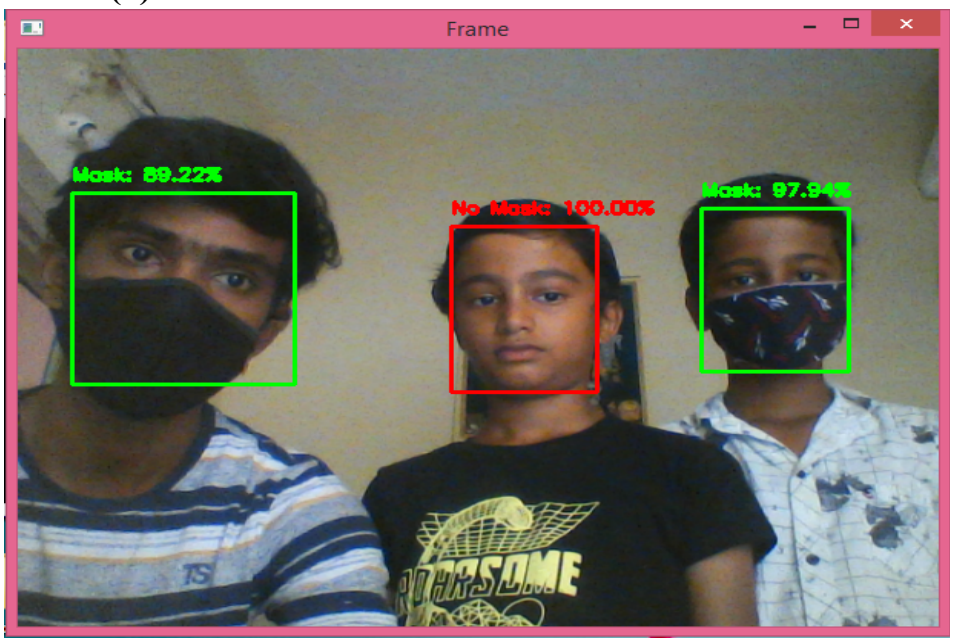


Figure 6.7: Result showing when only one person not wearing mask

In case 7 only one person is not wearing a mask and others are wearing masks. Audio doesn't work for more than one person.

CASE (8):

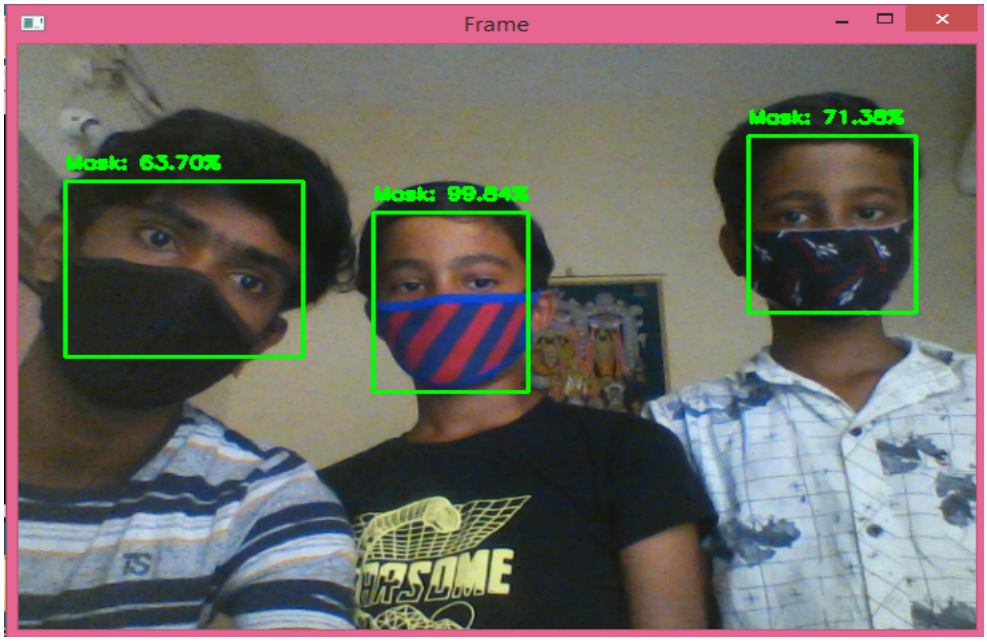


Figure 6.8: Results showing when everyone is wearing mask

In this last case everyone is wearing a mask. So it shows the mask with accuracy. Audio doesn't work for more than one person.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Conclusion and Future Scope

7.1 Conclusion

As the technology are blooming with emerging trends the availability so we have novel face mask detector which can possibly contribute to public health care department. The architecture consists of MobileNetV2 classifier and ADAM optimizer as the backbone it can be used for high and low computation scenarios. The our face mask detection is trained on CNN model and we are used Oven CV, Tensor Flow, Keras and python to detect whether
person is wearing a mask or not along with a audio if wearing mask it says cheers wearing mask or else please wear mask. The model was tested with image and real-time video stream. The accuracy of model is achieved and, the optimization of the model is continuous process. This specific model could be used as use case of edge analytics.

7.2 Future Scope:

The current ongoing system is gracing with MobileNetV2 classifier one of the best system which would be implemented along with the interface of alarm and alerting system in future generation. This system will be integrated with the system implementing social distancing that would make it a complete system which can bring a dramatic impact on the spread of. The new world will be well being of high demand of mask as faceless future and that will be a big security concern. Expertise say, CNN that using face mask proves to be the best solution to mitigate the spread of air borne virus like corona, but as a big security concern headed to challenge the nation as it would create a massive opportunity for people who cover their faces for nefarious reason. And also experts say the mass no of mask wearing in could
complicate in crime investigation in the coming days, as facial recognition is an important part in tracking of the criminals. When the pandemic covid-19 getting over, then this system comes into play for chemical factories, bank, glass factories etc. If a person enters the bank while wearing a mask he would be not allowed to enter and also if the person does not wear masks in glass factories chemical factories and etc. then the person would not be allowed to
enter to the industry. A mind concept of human being have been proved out to be very good at recognizing familiar faces and facial recognition algorithms are getting better in identifying pattern. So thus this challenge would create a scope to new face mask detection algorithms which can identify aces which are covered with greater accuracies and precisions.

.

CHAPTER 8
REFERENCES

REFERENCES:

[1] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, B.J. "Cowling Rational use of face masks in the COVID-19 pandemic" Lancet Respirat. Med., 8 (5) (2020), pp. 434-436

[2] X. Liu, S. Zhang, COVID-19: Face masks and human-to-human transmission, Influenza Other Respirat. Viruses, vol. n/a, no. n/a, doi: 10.1111/irv.12740

[3] "WHO Coronavirus Disease (COVID-19) Dashboard." https://covid19.who.int/ (accessed May 21, 2020).

[4] Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning Symmetry, 12 (4) (2020), p. 651

[5] D.M. Altmann, D.C. Douek, R.J. Boyton What policy makers need to know about COVID-19 protective immunity Lancet, 395 (10236) (2020), pp. 1527-1529.

[6] "Paris Tests Face-Mask Recognition Software on Metro Riders," Bloomberg.com, May 07, 2020.

[7] B. QIN and D. Li, Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19, May 2020, doi: 10.21203/rs.3.rs-28668/v1

[8] C. Li, R. Wang, J. Li, L. Fei, Face detection based on YOLOv3, in:: Recent Trends in Intelligent Computing, Communication and Devices, Singapore, 2020, pp. 277–284, doi: 10.1007/978-981-13-9406-5_34.

[9] M.K.J. Khan, N. Ud Din, S. Bae, J. Yi, Interactive removal of microphone object in facial images, Electronics 8 (10) (2019) , Art. no. 10, doi: 10.3390/electronics8101115

[10] S. A. Hussain, A.S.A.A. Balushi, A real time face emotion classification and recognition using deep learning model, J. Phys.: Conf. Ser. 1432 (2020) 012087, doi: 10.1088/1742-6596/1432/1/012087

[11] Z. Wang, et al., Masked face recognition dataset and application, arXiv preprint arXiv:2003.09093, 2020.

[12] M. Kawulok, M.E. Celebi, B. Smolka (Eds.), Advances in Face Detection and Facial Image Analysis, Springer International Publishing, Cham (2016), pp. 189-248

[13] L. Wen, X. Li, L. GaoA transfer convolutional neural network for fault diagnosis based on ResNet-50 Neural Comput. Appl., 32 (10) (2020), pp. 6111-6124.

[14] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation

invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, July 2002.

[15] T.-H. Kim, D.-C. Park, D.-M. Woo, T. Jeong, and S.-Y. Min, "Multiclass classifier-based adaboost algorithm," in Proceedings of the Second Sinoforeign-interchange Conference on Intelligent Science and Intelligent Data Engineering, ser. IScIDE'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 122–127.

CHAPTER 9
Annexure

9. PROGRAM FOR DETECT_MASK_VIDEO.PY:

import the necessary package

```
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
import os
import playsound
import speech_recognition as sr
from gtts import gTTS

def detect_and_predict_mask(frame, faceNet, maskNet):
	# grab the dimensions of the frame and then construct a blob
	# from it
	(h, w) = frame.shape[:2]
	blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
		(104.0, 177.0, 123.0))

	# pass the blob through the network and obtain the face detections
	faceNet.setInput(blob)
	detections = faceNet.forward()
	print(detections.shape)

	# initialize our list of faces, their corresponding locations,
	# and the list of predictions from our face mask network
	faces = []
	locs = []
	preds = []
```

```
# loop over the detections
for i in range(0, detections.shape[2]):
	# extract the confidence (i.e., probability) associated with
	# the detection
	confidence = detections[0, 0, i, 2]

	# filter out weak detections by ensuring the confidence is
	# greater than the minimum confidence
	if confidence > 0.5:
		# compute the (x, y)-coordinates of the bounding box
		# for the object
	box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
		(startX, startY, endX, endY) = box.astype("int")

	# ensure the bounding boxes fall within the dimensions of
		# the frame
		(startX, startY) = (max(0, startX), max(0, startY))
		(endX, endY) = (min(w - 1, endX), min(h - 1, endY))

	# extract the face ROI, convert it from BGR to RGB channel
		# ordering, resize it to 224x224, and preprocess it
		face = frame[startY:endY, startX:endX]
		face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
		face = cv2.resize(face, (224, 224))
		face = img_to_array(face)
		face = preprocess_input(face)

		# add the face and bounding boxes to their respective
		# lists
		faces.append(face)
		locs.append((startX, startY, endX, endY))

# only make a predictions if at least one face was detected
if len(faces) > 0:
	# for faster inference we'll make batch predictions on *all*
	# faces at the same time rather than one-by-one predictions
```

```
			# in the above `for` loop
			faces = np.array(faces, dtype="float32")
			preds = maskNet.predict(faces, batch_size=32)

		# return a 2-tuple of the face locations and their corresponding
		# locations
		return (locs, preds)

# load our serialized face detector model from disk
prototxtPath = r"face_detector\deploy.prototxt"
weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk
maskNet = load_model("mask_detector.model")

# initialize the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()

# loop over the frames from the video stream
while True:
		# grab the frame from the threaded video stream and resize it
		# to have a maximum width of 400 pixels
		frame = vs.read()
		frame = imutils.resize(frame, width=400)

		# detect faces in the frame and determine if they are wearing a
		# face mask or not
		(locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

		# loop over the detected face locations and their corresponding
		# locations
		for (box, pred) in zip(locs, preds):
			# unpack the bounding box and predictions
			(startX, startY, endX, endY) = box
```

```
                (mask, withoutMask) = pred

                # determine the class label and color we'll use to draw
                # the bounding box and text
                label = "Mask" if mask > withoutMask else "No Mask"
                color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

                # include the probability in the label
                label = "{}: {:.2f}%".format(label, max(mask, withoutMask) *
100)

                # display the label and bounding box rectangle on the output
                # frame
                cv2.putText(frame, label, (startX, startY - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
                cv2.rectangle(frame, (startX, startY), (endX, endY), color,
2)

        # show the output frame
        cv2.imshow("Frame", frame)
        text=result
        if text==”MASK”:
             text=’cheers, wearing mask’
        elif text==”NO MASK”:
             text=’please wear mask’
        tts=gTTS(text=text, lang=”en”)
        filename=”voice”+str(i)+”.mp3”
        tts.save(filename)
        playsound.playsound(filename)
        key = cv2.waitKey(1) & 0xFF

        # if the `q` key was pressed, break from the loop
        if key == ord("q"):
                Break

# do a bit of cleanup
```

```
cv2.destroyAllWindows()
vs.stop()
```