MASK DETECTION USING MACHINE LEARNING

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR'S OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Jasraj Singh Chhabda (111915051)

Harsh Prakash (111915041)

Aryan Sharma (111915018)

Amarendra Shendkar (111915013)

Ashish Biswal (111916013)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Indian Institute of Information And Technology Pune

Near Bopdev Ghat, Kondhwa Annexe, Yewalewadi, Pune, Maharashtra 411048

JUNE 2020

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"Image Processing for Mask Detection"** submitted by **Jasraj Singh Chhabda, Harsh Prakash, Aryan Sharma, Amrendra Shendkar, Ashish Biswal** bearing the **MIS Nos: 111915051, 111915041, 111915018, 111915013, 111916013** respectively, in completion of his project work under the guidance of **Dr. Rahul Dixit** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering in the Department of Computer Science and Engineering, Indian Institute of Information Technology,Pune, during the academic year 2020-21.

Dr. Rahul Dixit	**Dr. Tanmoy Hazra**
Project Guide	Head of the Department
Professor	Assistant Professor
Department of CSE	Department of CSE
IIIT Pune	IIIT Pune

Project Viva-voce held on ________________________

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. I would like to thank the people who helped me directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our beloved director, **Dr.Anupam Shukla**, for providing his kind support in various aspects.

I would like to express my gratitude to my project guide **Dr. Rahul Dixit**, Professor, Department of CSE, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this M.Tech project.

I would like to express my gratitude to the head of department **Dr. Tanmoy Hazra**, Professor, Department of CSE, for providing his kind support in various aspects.

I would also like to thank all the faculty members in the Dept. of CSE and my classmates for their steadfast and strong support and engagement with this project.

Abstract

Main aim of Image processing project is to extract important data from images. Using this extracted information description, interpretation and understanding of the scene can be provided by the machine. Main point of image processing is to modify images in to desired manner. This system allows users to take hard copy of the image using printer routines and provides option for users to store file in to disk in different formats. In other words image processing is called as altering and analyzing pictorial information of images. In our daily life we come across different type of image processing best example of image processing in our daily life is our brain sensing lot of images when we see images with eyes and processing is done is very less time.

In existing system there are many techniques which are available for extracting information from images but there are no exact processing is defined. In proposed system we will come across different new techniques in image processing.

Conclusion

Keywords : Image, Processing, extract, data, information, modify,

TABLE OF CONTENTS

List of Figures

List of Tables

Listings

Chapter 1
Introduction

1.1 Image Processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

1.2 Basic Steps

Image processing basically includes the following three steps:
1. Importing the image via image acquisition tools.
2. Analysing and manipulating the image.
3. Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

1.3 Overview

In this lecture we will talk about a few fundamental definitions such as image, digital image, and digital image processing. Different sources of digital images will be discussed and examples for each source will be provided. The continuum from image processing to computer vision will be covered in this lecture. Finally we will talk about image acquisition and different types of image sensors.

Chapter 2
Motivation

In the new world of coronavirus, multidisciplinary efforts have been organized to slow the spread of the pandemic. The AI community has also been a part of these endeavors. In particular, developments for monitoring social distancing or identifying face masks have made-the-headlines.

But all this hype and anxiety to show off results as fast as possible, added up to the usual AI overpromising factor (see AI winter), may be signaling the wrong idea that solving some of these use cases is almost trivial due to the mighty powers of AI.

In an effort to paint a more complete picture, we decided to show the creative process behind a solution for a seemingly simple use case in computer vision:

1. Detect people that pass through a security-like camera.
2. Identify face mask usage.
3. Collect reliable statistics (people wearing masks).

Chapter 3

Literature Review

3.1 What is Image Classification?

A common use of machine learning is to identify what an image represents. For example, we might want to know what type of animal appears in the following photograph.

Figure 3.1: Sample Image of a Dog

The task of predicting what an image represents is called image classification. An image classification model is trained to recognize various classes of images. For example, a model might be trained to recognize photos representing three different types of animals: rabbits, hamsters, and dogs.

When we subsequently provide a new image as input to the model, it will output the probabilities of the image representing each of the types of animal it was trained on. An example output might be as follows:

Animal Type	Animal Probability
Rabbit	0.07
Hamster	0.02
Dog	0.91

For simple understanding:

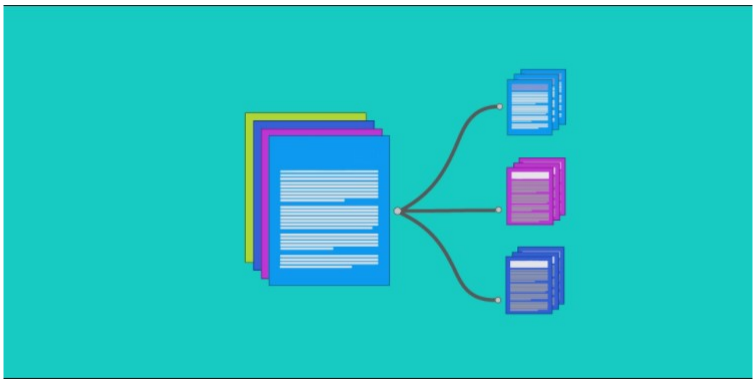

Figure 3.2: Simpler portrayal of Classification of an Image

3.2 What is Object Detection?

Given an image or a video stream, an object detection model can identify which of a known set of objects might be present and provide information about their positions within the image.

For example, this screenshot of our example application shows how two objects have been recognized and their positions annotated:

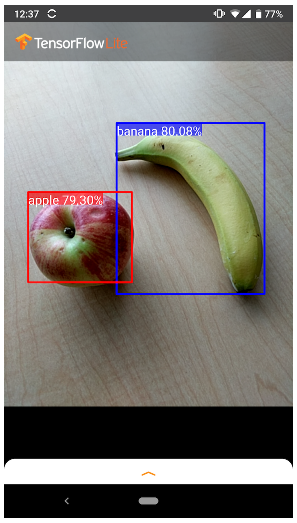

Figure 3.3: Image Detection Screenshot

An object detection model is trained to detect the presence and location of multiple classes of objects. For example, a model might be trained with images that contain various pieces of fruit, along with a label that specifies the class of fruit they represent (e.g. an apple, a banana, or a strawberry), and data specifying where each object appears in the image.

When we subsequently provide an image to the model, it will output a list of the objects it detects, the location of a bounding box that contains each object, and a score that indicates the confidence that detection was correct.

3.3 Model Output

Imagine a model has been trained to detect apples, bananas, and strawberries. When we pass it an image, it will output a set number of detection results - in this example,

Class	Score	Locations
Apple	0.92	[18, 21, 57, 63]
Banana	0.88	[100, 30, 180, 150]
Strawberry	0.0.87	[7, 82, 89, 163]
Banana	0.23	[42, 66, 57, 83]
Apple	0.11	[6, 42, 31, 58]

3.4 Confidence Score

To interpret these results, we can look at the score and the location for each detected object. The score is a number between 0 and 1 that indicates confidence that the object was genuinely detected. The closer the number is to 1, the more confident the model is.

Depending on your application, you can decide a cut-off threshold below which you will discard detection results. For our example, we might decide a sensible cut-off is a score of 0.5 (meaning a 50 percent probability that the detection is valid). In that case, we would ignore the last two objects in the array, because those confidence scores are below 0.5

Chapter 4

What Is Machine Learning?

4.1 Introduction

Machine-learning algorithms use statistics to find patterns in massive* amounts of data. And data, here, encompasses a lot of things—numbers, words, images, clicks, what have you. If it can be digitally stored, it can be fed into a machine-learning algorithm.

Machine learning is the process that powers many of the services we use today — recommendation systems like those on Netflix, YouTube, and Spotify; search engines like Google and Baidu; social-media feeds like Facebook and Twitter; voice assistants like Siri and Alexa. The list goes on.

In all of these instances, each platform is collecting as much data about you as possible—what genres you like watching, what links you are clicking, which statuses you are reacting to—and using machine learning to make a highly educated guess about what you might want next. Or, in the case of a voice assistant, about which words match best with the funny sounds coming out of your mouth.

Frankly, this process is quite basic: find the pattern, apply the pattern. But it pretty much runs the world. That's in big part thanks to an invention in 1986, courtesy of Geoffrey Hinton, today known as the father of deep learning.

4.2 What is Deep Learning?

Deep learning is machine learning on steroids: it uses a technique that gives machines an enhanced ability to find—and amplify—even the smallest patterns. This technique is called a deep neural network—deep because it has many, many layers of simple computational nodes that work together to munch through data and deliver a final result in the form of the prediction.

4.3 What are the different types of Machine Learning?

Machine learning algorithms are often categorized as supervised or unsupervised.

4.3.1 Supervised Machine Learning

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

4.3.2 Unsupervised Machine Learning

In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

4.3.3 Semi-Supervised Machine Learning

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.
Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

4.4 Linear Regression

Linear regression is perhaps one of the most well known and well understood algorithms in statistics and machine learning.
Linear regression is an attractive model because the representation is so simple.

The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

y = B0 + B1*x

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. B0 and B1 in the above example).

It is common to talk about the complexity of a regression model like linear regression. This refers to the number of coefficients used in the model.

When a coefficient becomes zero, it effectively removes the influence of the input variable on the model and therefore from the prediction made from the model (0 * x = 0). This becomes relevant if you look at regularization methods that change the learning algorithm to reduce the complexity of regression models by putting pressure on the absolute size of the coefficients, driving some to zero. Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

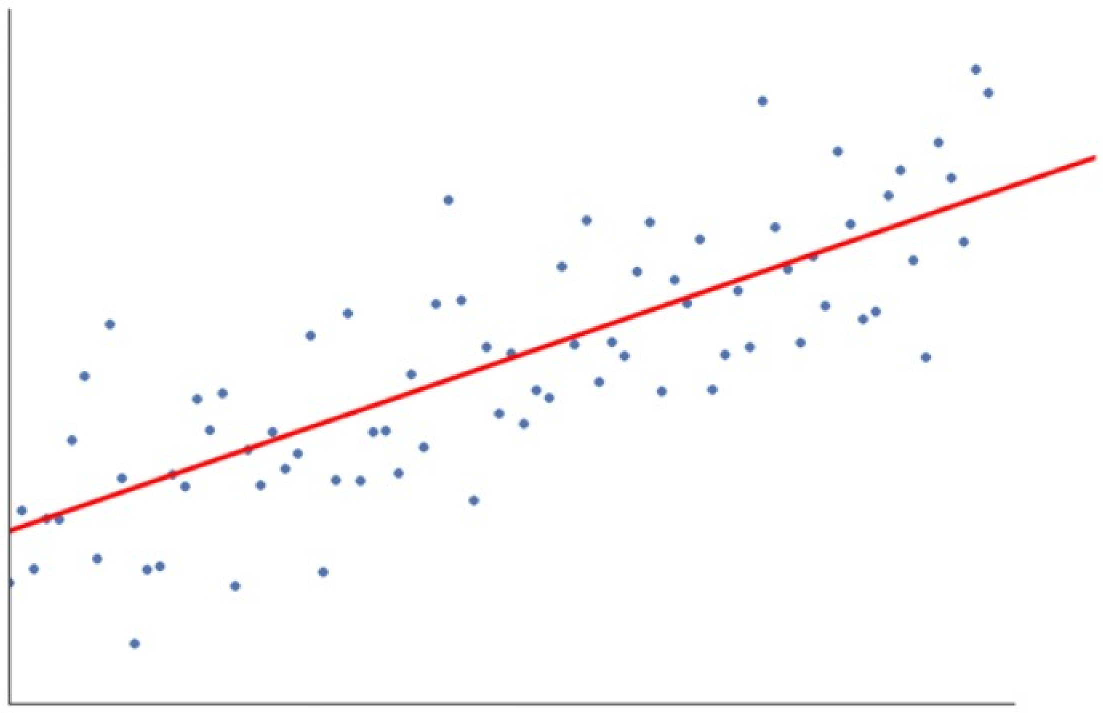

Figure 4.1: Linear Regression

4.5 Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types

4.5.1 Binary or Binomial

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

4.5.2 Multinomial

In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance. For example, these variables may represent "Type A" or "Type B" or "Type C".

4.5.3 Ordinal

In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent "poor" or "good", "very good", "Excellent" and each category can have the scores like 0,1,2,3.

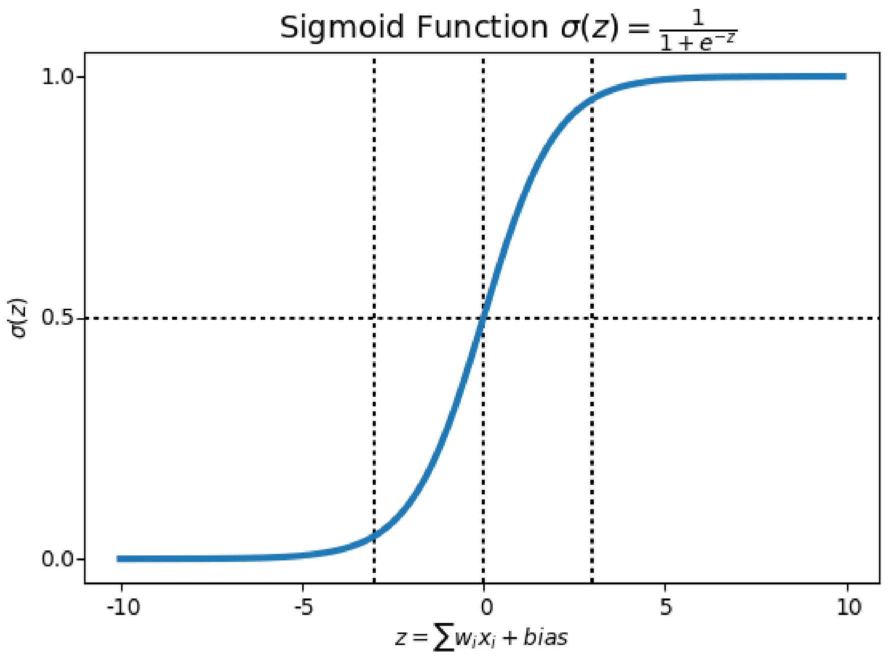

Figure 4.2: Logistic Regression

4.6 Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.

KEY TAKEAWAYS

Neural networks are a series of algorithms that mimic the operations of a human brain to recognize relationships between vast amounts of data.

They are used in a variety of applications in financial services, from forecasting and marketing research to fraud detection and risk assessment. Use of neural networks for stock market price prediction varies.

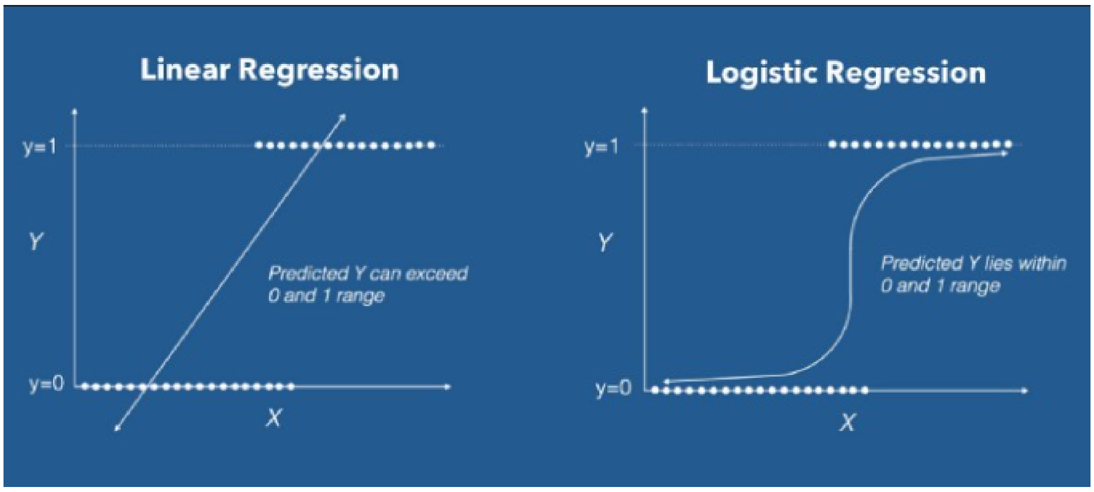

Figure 4.3: Logistic Regression

4.6.1 Basics of Neural Networks

Neural networks, in the world of finance, assist in the development of such process as time-series forecasting, algorithmic trading, securities classification, credit risk modeling and constructing proprietary indicators and price derivatives.

A Neural network works similarly to the human brain's Neural network. A "neuron" in a Neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

A Neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map. For instance, the patterns may comprise a list of quantities for technical indicators about a security; potential outputs could be "buy," "hold" or "sell."

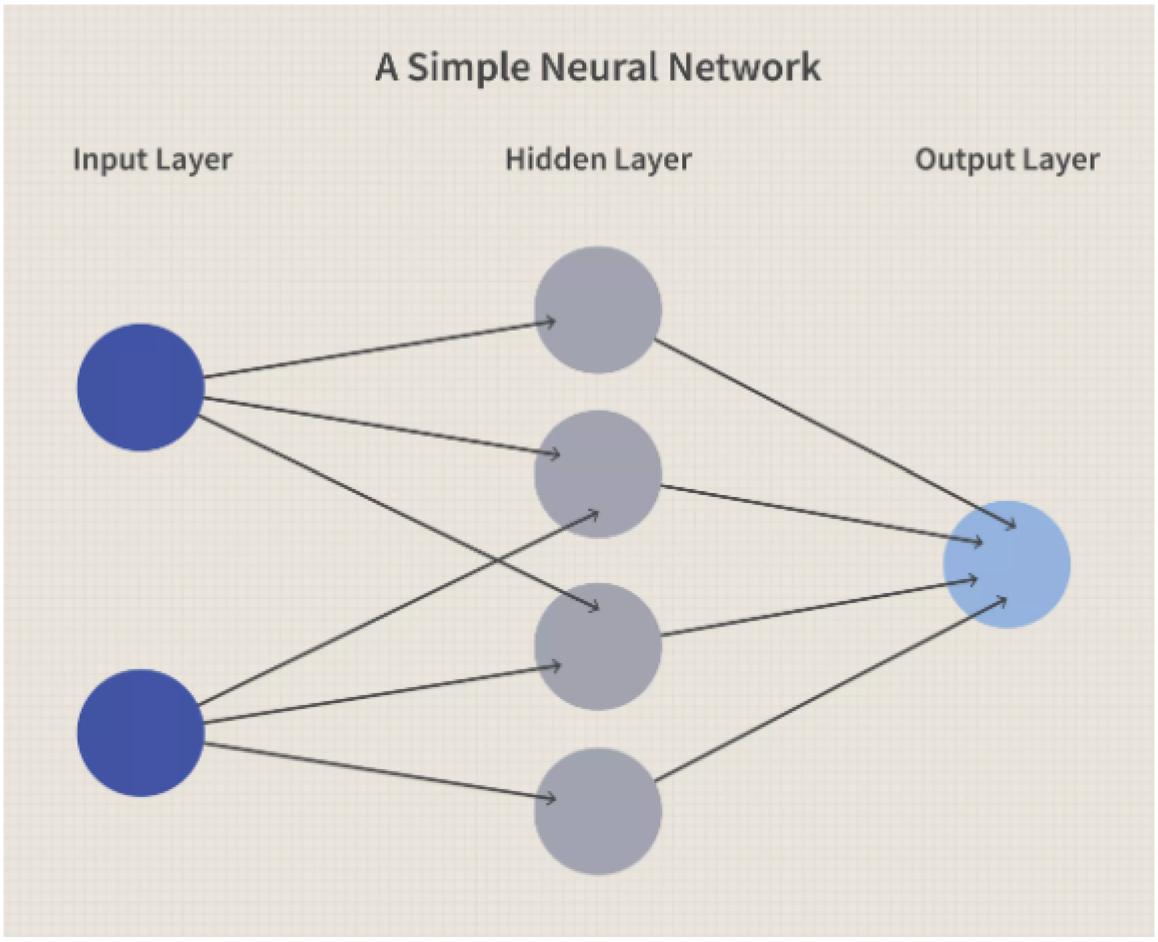

Figure 4.4: neural Networks

Hidden layers fine-tune the input weightings until the Neural network's margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis.

4.6.2 Applications of Neural Networks

Neural networks are broadly used, with applications for financial operations, enterprise planning, trading, business analytics and product maintenance. Neural networks have also gained widespread adoption in business applications such as forecasting and marketing research solutions, fraud detection and risk assessment.

A Neural network evaluates price data and unearths opportunities for making trade decisions based on the data analysis. The networks can distinguish subtle nonlinear inter-dependencies and patterns other methods of technical analysis cannot. According to research, the accuracy of Neural networks in making price predictions for stocks differs. Some models predict the correct stock prices 50 to 60 percent of the time while others are accurate in 70 percent of all instances. Some have posited that a 10 percent improvement in efficiency is all an investor can ask for from a Neural network.

There will always be data sets and task classes that a better analyzed by using previously developed algorithms. It is not so much the algorithm that matters; it is the well-prepared input data on the targeted indicator that ultimately determines the level of success of a Neural network.

4.7 Use Cases

4.7.1 Speech Recognition

Speech occupies a prominent role in human-human interaction. Therefore, it is natural for people to expect speech interfaces with computers. In the present era, for communication with machines, humans still need sophisticated languages which are difficult to learn and use. To ease this communication barrier, a simple solution could be, communication in a spoken language that is possible for the machine to understand.

Great progress has been made in this field, however, still such kinds of systems are facing the problem of limited vocabulary or grammar along with the issue of retraining of the system for different speakers in different conditions. ANN is playing a major role in this area. Following ANNs have been used for speech recognition

Multilayer networks

Multilayer networks with recurrent connections

Kohonen self-organizing feature map

The most useful network for this is Kohonen Self-Organizing feature map, which has its input as short segments of the speech waveform. It will map the same kind of phonemes as the output array, called feature extraction technique. After extracting the features, with the help of some acoustic models as back-end processing, it will recognize the utterance.

4.7.2 Character Recognition

It is an interesting problem which falls under the general area of Pattern Recognition. Many Neural networks have been developed for automatic recognition of handwritten characters, either letters or digits. Following are some ANNs which have been used for character recognition

1. Multilayer Neural networks such as Backpropagation Neural networks.
2. Neocognitron

Though back-propagation Neural networks have several hidden layers, the pattern of connection from one layer to the next is localized. Similarly, neocognitron also has several hidden layers and its training is done layer by layer for such kind of applications.

4.7.3 Signature Verification Application

Signatures are one of the most useful ways to authorize and authenticate a person in legal transactions. Signature verification technique is a non-vision based technique.

For this application, the first approach is to extract the feature or rather the geometrical feature set representing the signature. With these feature sets, we have to train the Neural networks using an efficient Neural network algorithm. This trained Neural network will classify the signature as being genuine or forged under the verification stage.

4.7.4 Human Face Recognition

It is one of the biometric methods to identify the given face. It is a typical task because of the characterization of “non-face” images. However, if a Neural network is well trained, then it can be divided into two classes namely images having faces and images that do not have faces.
First, all the input images must be preprocessed. Then, the dimensionality of that image must be reduced. And, at last it must be classified using Neural network training algorithm. Following Neural networks are used for training purposes with preprocessed image
Fully-connected multilayer feed-forward Neural network trained with the help of back-propagation algorithm.
For dimensionality reduction, Principal Component Analysis PCA is used.

4.8 Deep Neural Networks

The human brain, its functions, and the way it works served as the inspiration for the creation of the neural network. Artificial intelligence and machine learning, which is a subset of AI, play an essential part in its functionality. It starts working when a developer enters data and builds a machine learning algorithm, mostly using the “if ... else ...” principle of building a program. The deep neural network does not only work according to the algorithm but also can predict a solution for a task and make conclusions using its previous experience. In this case, you do not need to use programming or coding to get an answer.

Nodes are little parts of the system, and they are like neurons of the human brain. When a stimulus hits them, a process takes place in these nodes. Some of them are connected and marked, and some are not, but in general, nodes are grouped into layers.

The system must process layers of data between the input and output to solve a task. The more layers it has to process to get the result, the deeper the network is considered.

There is a concept of Credit Assignment Path (CAP) which means the number of such layers needed for the system to complete the task. The neural network is deep if the CAP index is more than two.

A deep neural network is beneficial when you need to replace human labor with autonomous work without compromising its efficiency. The deep neural network usage can find various applications in real life. For example, a Chinese company Sensetime created a system of automatic face recognition system to identify criminals, which uses real-time cameras to find an offender in the crowd. Nowadays, it has become a popular practice in police and other governmental entities.

The American company Pony.ai is another example of how you can use DNN. They developed a system for AI cars that can work without a driver. It requires more than just a simple algorithm of actions, but a much deeper learning system, which should be able to recognize people, road signs and other markings like trees, and other important objects.

The famous company UbiTech creates AI robots. One of their creations is the Alpha 2 robot that can live in a family, speak with its members, search for information, write messages, and execute voice commands.

4.8.1 What is the Difference Between the Neural Network and Deep Neural Network?

You can compare a neural network to a chess game with a computer. It has algorithms, according to which it determines tactics, depending on your moves and actions. The programmer enters data on how each figure moves into the computer's database, determines the boundaries of the chessboard, introduces a huge number of strategies that chess players play by. At the same time, the computer may, for example, be able to learn from you and other people, and it can become a deep neural network. In a while, playing with different players, it can become invincible.

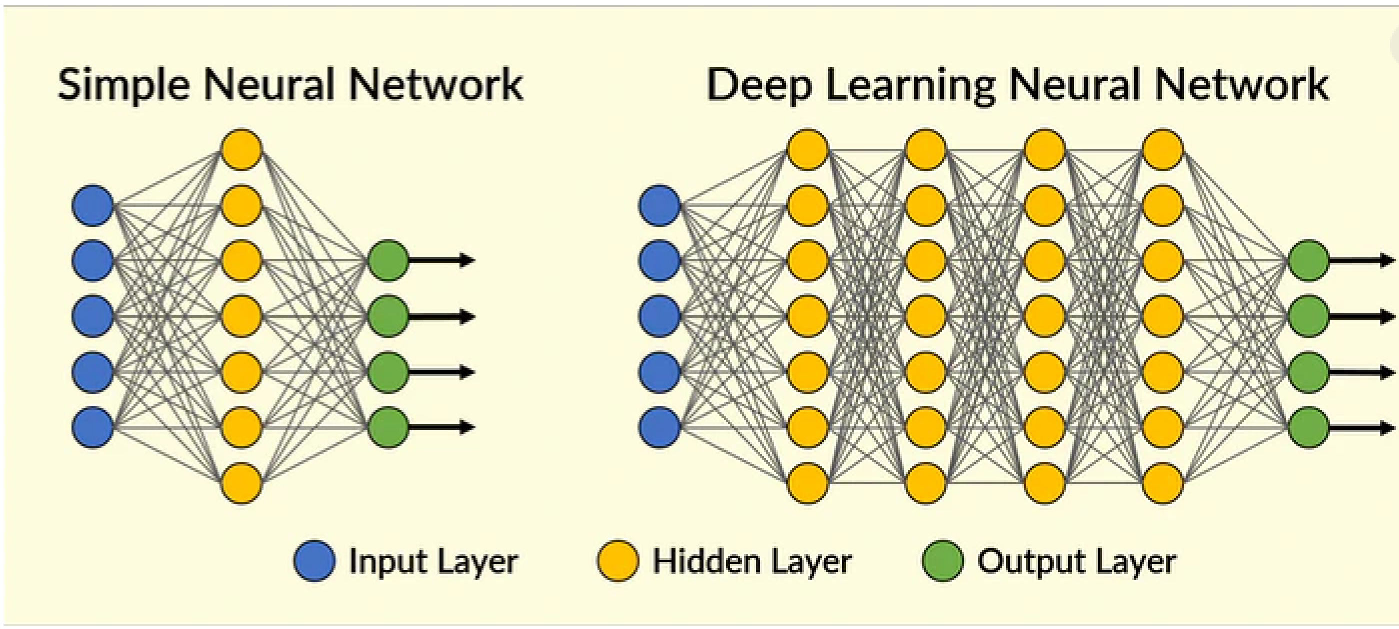

Figure 4.5: neural Networks

The neural network is not a creative system, but a deep neural network is much more complicated than the first one. It can recognize voice commands, recognize sound and graphics, do an expert review, and perform a lot of other actions that require prediction, creative thinking, and analytics. Only the human brain has such possibilities. The neural network can get one result (a word, an action, a number, or a solution), while the deep neural network solves the problem more globally and can draw conclusions or predictions depending on the information supplied and the desired result. The neural network requires a specific input of data and algorithms of solutions, and the deep neural network can solve a problem without a significant amount of marked data.

4.8.2 What Is Deep Learning Neural Network?

The neural network needs to learn all the time to solve tasks in a more qualified manner or even to use various methods to provide a better result. When it gets new information in the system, it learns how to act accordingly to a new situation.

Learning becomes deeper when tasks you solve get harder. Deep neural network represents the type of machine learning when the system uses many layers of nodes to derive high-level functions from input information. It means transforming the data into a more creative and abstract component.

In order to understand the result of deep learning better, let's imagine a picture of an average man. Although you have never seen this picture and his face and body before, you will always identify that it is a human and differentiate it from other creatures. This is an example of how the deep neural network works. Creative and analytical components of information are analyzed and grouped to ensure that the object is identified correctly. These components are not brought to the system directly, thus the ML system has to modify and derive them.

4.9 Convolutional Deep neural Networks

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

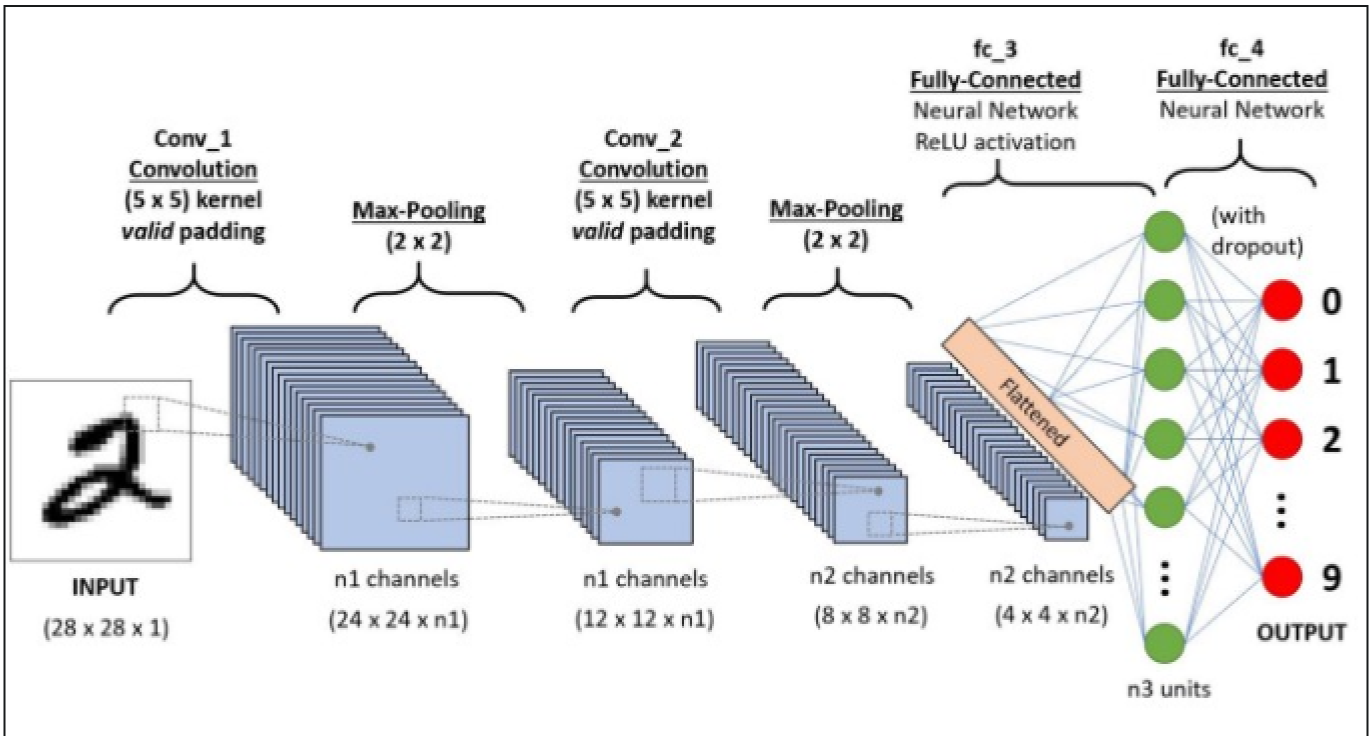

Figure 4.6: convolutional Deep neural Networks

4.9.1 Why ConvNets over Feed-Forward Neural Nets?

An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really.

In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Input Image:

In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which im-

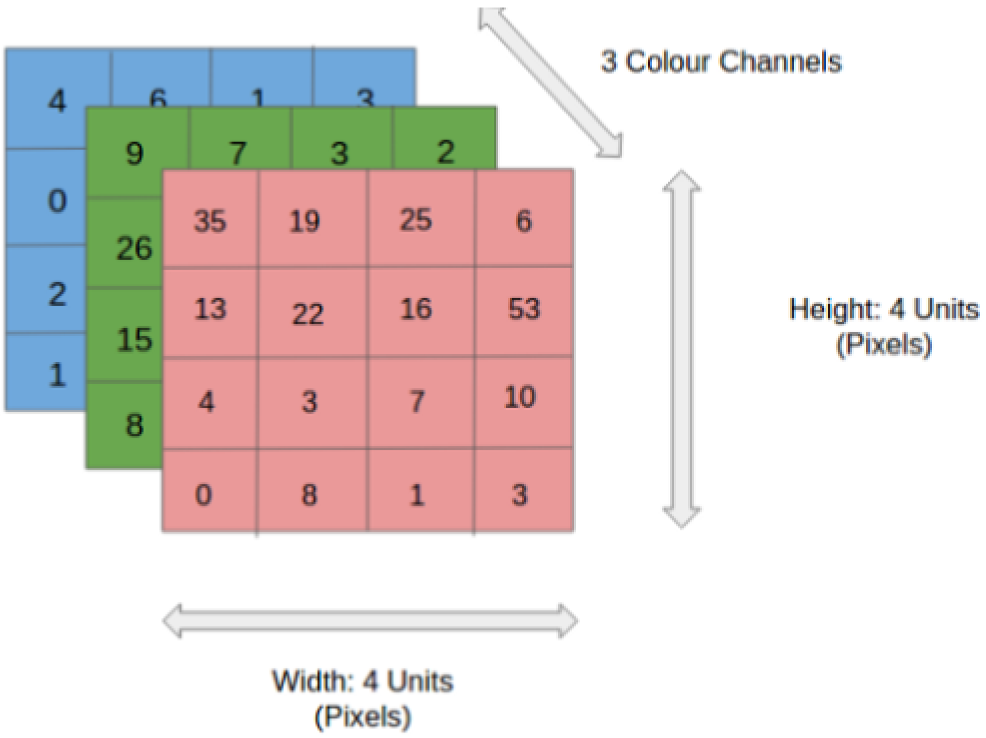

Figure 4.7: convolutional Deep neural Networks

ages exist — Grayscale, RGB, HSV, CMYK, etc.
You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (76804320). The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

convolutional Layer - The Kernel

Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB) In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix.

Pooling Layer:

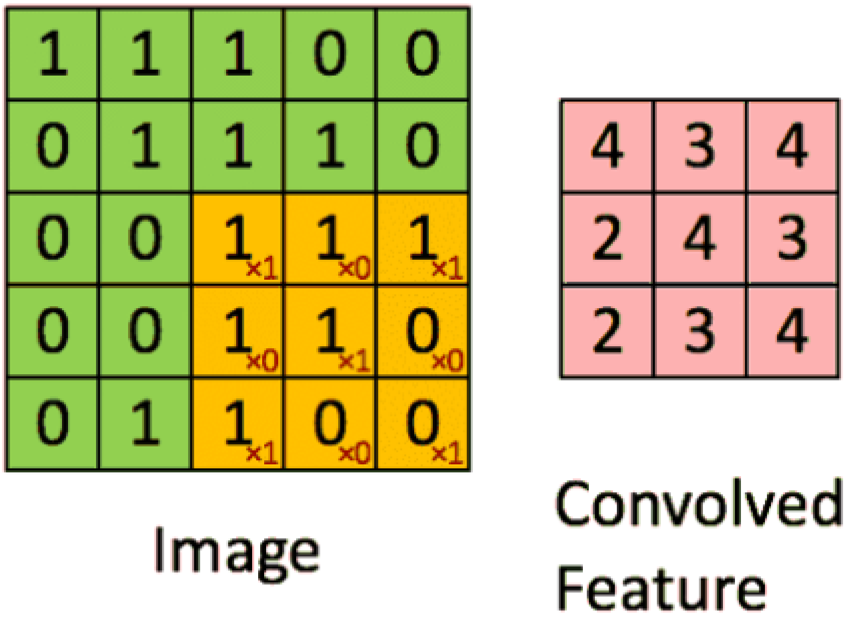

Figure 4.8: convolutional Deep neural Networks

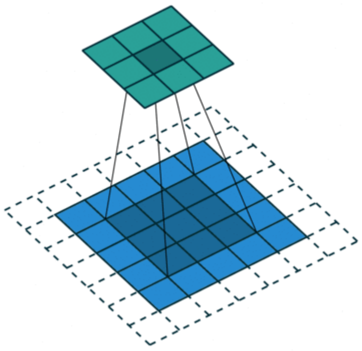

Figure 4.9: Pooling

Pooling Layer

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other

hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.
Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

Final Classification: Classification — Fully Connected Layer (FC Layer)

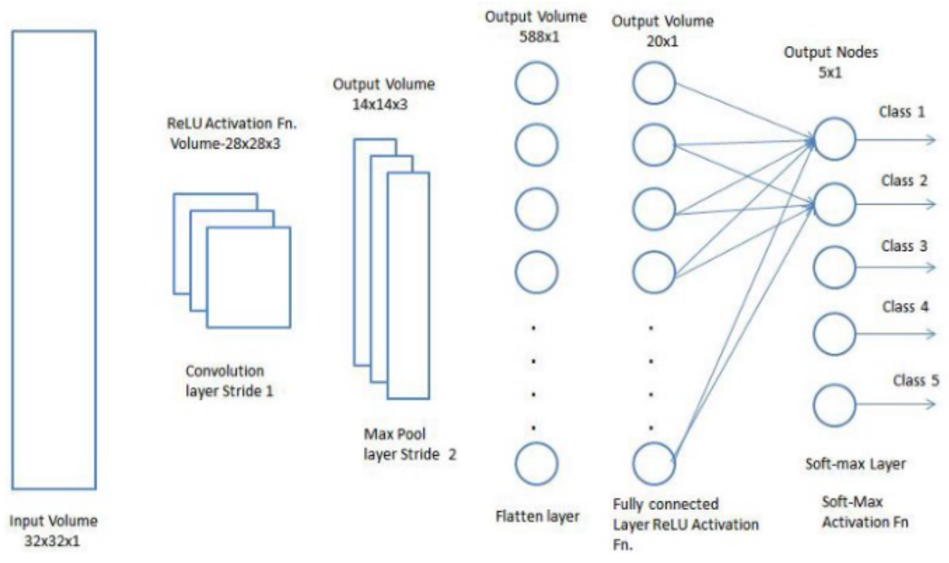

Figure 4.10: Fully Connected Layer

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.
Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

4.10 Mobilenets : (referenced from MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications by Andrew G. Howard, Menglong Zhu et al.)

MobileNets are a class of efficient models for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks. They introduce two simple global hyperparameters that efficiently trade off between latency and accuracy. These hyperparameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. The authors present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification.

4.10.1 Use Cases for MobileNets

MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. They can be built upon for classification, detection, embeddings and segmentation similar to how other popular large scale models, such as Inception, are used.

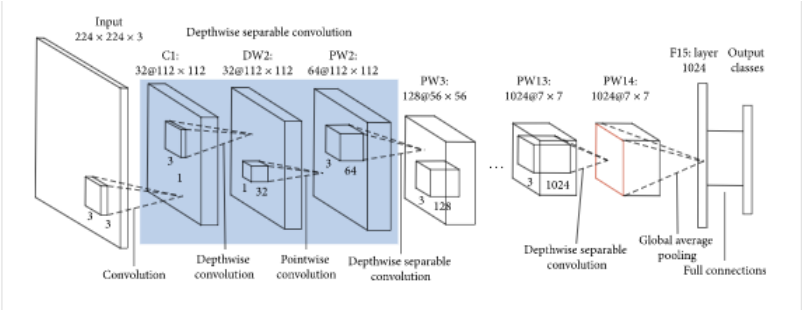

Figure 4.11: Mobile Nets

Chapter 5

Supplementary Libraries

5.1 TensorFlow

Figure 5.1: TensorFlow Logo

TensorFlow is an open source framework developed by Google researchers to run machine learning, deep learning and other statistical and predictive analytics workloads. Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers.

The TensorFlow software handles data sets that are arrayed as computational nodes in graph form. The edges that connect the nodes in a graph can represent multidimensional vectors or matrices, creating what are known as tensors.
Because TensorFlow programs use a data flow architecture that works with generalized intermediate results of the computations, they are especially open to very large-scale parallel processing applications, with neural networks being a common example.

The framework includes sets of both high-level and low-level APIs. Google recommends using the high-level ones when possible to simplify data pipeline development and application programming.

However, knowing how to use the low-level APIs – called TensorFlow Core – can be valuable for experimentation and debugging of applications, the company says; it also gives users a ”mental model” of the machine learning technology's inner workings, in Google's words.

TensorFlow applications can run on either conventional CPUs or higher-performance graphics processing units (GPUs), as well as Google's own tensor processing units (TPUs), which are custom devices expressly designed to speed up TensorFlow jobs.
Google's first TPUs, detailed publicly in 2016, were used internally in conjunction with TensorFlow to power some of the company's applications and online services, including its RankBrain search algorithm and Street View mapping technology.

In early 2018, Google furthered its external TensorFlow efforts by making the second generation of TPUs available to Google Cloud Platform users for training and running their own machine learning models. TensorFlow-based workloads are billed on a per-second basis; the Cloud TPU service initially was launched as a beta program with only ”limited quantities” of the devices available for use, according to Google.

5.2 OpenCV

Figure 5.2: OpenCV Logo

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDAand OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

5.3 SkLearn

Figure 5.3: SkLearn Logo

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

NumPy: Base n-dimensional array package
SciPy: Fundamental library for scientific computing
Matplotlib: Comprehensive 2D/3D plotting
IPython: Enhanced interactive console
Sympy: Symbolic mathematics
Pandas: Data structures and analysis

Extensions or modules for SciPy care conventionally named SciKits. As such, the module provides learning algorithms and is named scikit-learn.

The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as easy of use, code quality, collaboration, documentation and performance.

Chapter 6

Objectives

6.1 Our Aim

To detect, classify and differentiate masked and unmasked people with a reasonably acceptable accuracy using various supplementary libraries:

6.1.1 Python

Figure 6.1: Python Logo

6.1.2 TensorFlow

Figure 6.2: TensorFlow Logo

6.1.3 OpenCV

Figure 6.3: OpenCV Logo

Chapter 7

Research Gaps

7.1 Dataset with enough variety to deal with current variety of Masks

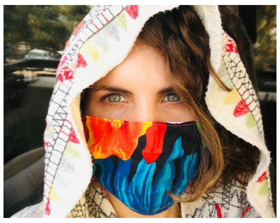

Figure 7.1: Sample 1

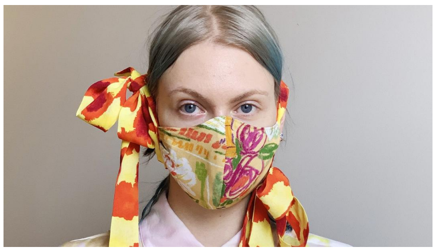

Figure 7.2: sample 2

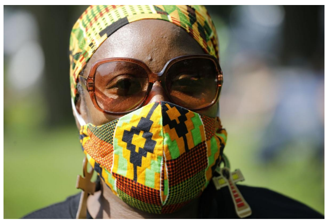

Figure 7.3: Sample 3

7.2 Datasets are resourced from Kaggle, Github which are Independent and therefore we lack total control over it.

7.2.1 Kaggle

Figure 7.4: Kaggle Logo

7.2.2 GitHub

Figure 7.5: GitHub Logo

Chapter 8
Methodology

8.1 Decided on the problem statement after much though around the Current affairs happening around the world vis-à-vis the Pandemic and how we could work around that.

Businesses are constantly overhauling their existing infrastructure and processes to be more efficient, safe, and usable for employees, customers, and the community. With the ongoing pandemic, it's even more important to have advanced analytics apps and services in place to mitigate risk. For public safety and health, authorities are recommending the use of face masks and coverings to control the spread of COVID-19.
In the present scenario due to Covid-19, there is no efficient face mask detection applications which are now in high demand for transportation means, densely populated areas, residential districts, large-scale manufacturers and other enterprises to ensure safety. Also, the absence of large datasets of 'with mask' images has made this task more cumbersome and challenging

8.2 Sourcing the Dataset from pre-existing images on relevant sites.

Datasets are a collection of instances that all share a common attribute. Machine learning models will generally contain a few different datasets, each used to fulfill various roles in the system.

For machine learning models to understand how to perform various actions, training datasets must first be fed into the machine learning algorithm, followed by validation datasets (or testing datasets) to ensure that the model is interpreting this data accurately.

Once you feed these training and validation sets into the system, subsequent datasets can then be used to sculpt your machine learning model going forward. The more data you provide to the ML system, the faster that model can learn and improve.

8.3 Processing the dataset to optimize it for our Problem statement.

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.
On a predictive modeling project, machine learning algorithms learn a mapping from input variables to a target variable.

The most common form of predictive modeling project involves so-called structured data or tabular data. This is data as it looks in a spreadsheet or a matrix, with rows of examples and columns of features for each example.

We cannot fit and evaluate machine learning algorithms on raw data; instead, we must transform the data to meet the requirements of individual machine learning algorithms. More than that, we must choose a representation for the data that best exposes the unknown underlying structure of the prediction problem to the learning algorithms in order to get the best performance given our available resources on a predictive modeling project.

Given that we have standard implementations of highly parameterized machine learning algorithms in open source libraries, fitting models has become routine. As such, the most challenging part of each predictive modeling project is how to prepare the one thing that is unique to the project: the data used for modeling.

8.4 Model Selection for an appropriate fit.

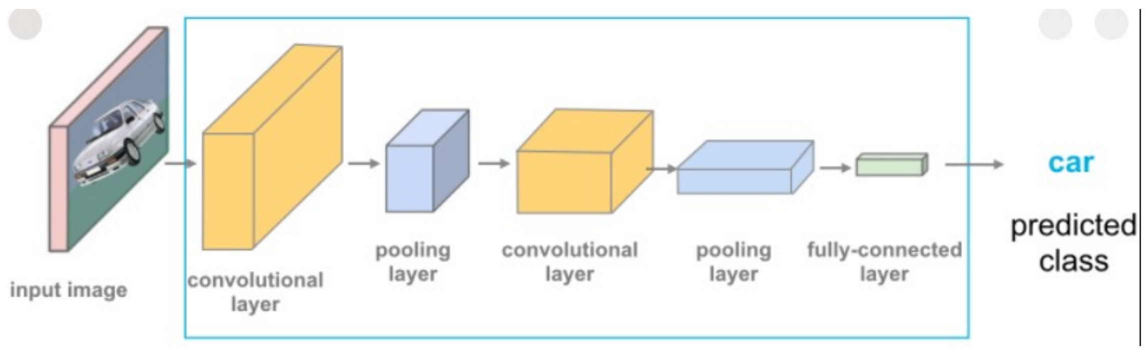

Figure 8.1: Model Selection

Above is the chronology that is being followed.

Model selection is the process of selecting one final machine learning model from among a collection of candidate machine learning models for a training dataset.
Model selection is a process that can be applied both across different types of models (e.g. logistic regression, SVM, KNN, etc.) and across models of the same type configured with different model hyperparameters (e.g. different kernels in an SVM).
Fitting models is relatively straightforward, although selecting among them is the true challenge of applied machine learning.
Firstly, we need to get over the idea of a “best” model.
All models have some predictive error, given the statistical noise in the data, the incompleteness of the data sample, and the limitations of each different model type. Therefore, the notion of a perfect or best model is not useful.
Instead, we must seek a model that is “good enough.”
What do we care about when choosing a final model?
The project stakeholders may have specific requirements, such as maintainability and

limited model complexity. As such, a model that has lower skill but is simpler and easier to understand may be preferred.
Alternately, if model skill is prized above all other concerns, then the ability of the model to perform well on out-of-sample data will be preferred regardless of the computational complexity involved.

Therefore, a "good enough" model may refer to many things and is specific to your project, such as:

A model that meets the requirements and constraints of project stakeholders.
A model that is sufficiently skillful given the time and resources available.
A model that is skillful as compared to naive models.
A model that is skillful relative to other tested models.
A model that is skillful relative to the state-of-the-art.

8.5 Optimising it for the Dataset.

Figure 8.2: Original Picture

Currently our code can work on just black and white images

Figure 8.3: Downsizing

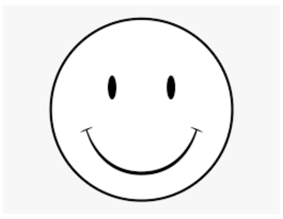

Figure 8.4: Color Optimization

Model optimization is one of the toughest challenges in the implementation of machine learning solutions. Entire branches of machine learning and deep learning theory have been dedicated to the optimization of models.

Hyperparameter optimization in machine learning intends to find the hyperparameters of a given machine learning algorithm that deliver the best performance as measured on a validation set. Hyperparameters, in contrast to model parameters, are set by the machine learning engineer before training. The number of trees in a random forest is a hyperparameter while the weights in a Neural network are model parameters learned during training. I like to think of hyperparameters as the model settings to be tuned so that the model can optimally solve the machine learning problem.

Some examples of model hyperparameters include:

The learning rate for training a Neural network.
The C and hyperparameters for support vector machines.
The k in k-nearest neighbors.
Hyperparameter optimization finds a combination of hyperparameters that returns an optimal model which reduces a predefined loss function and in turn increases the accu-

racy on given independent data.

8.6 Using OpenCV to employ a realtime detection for a video input

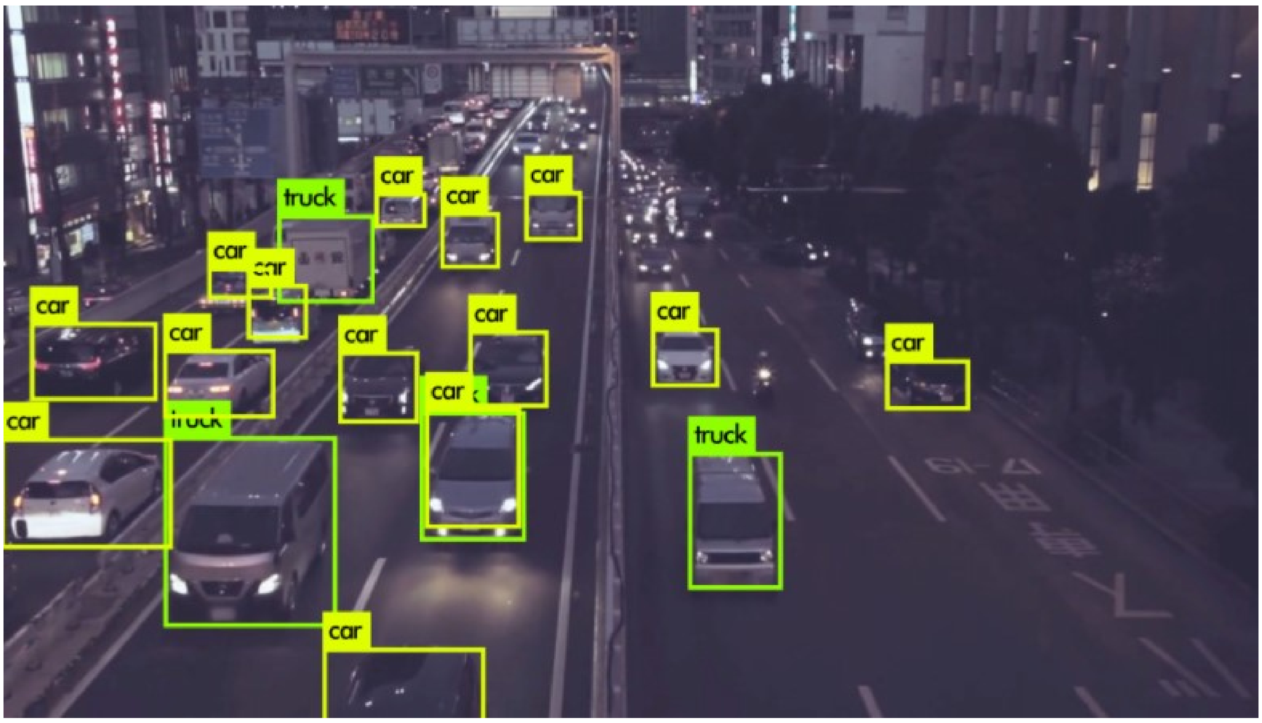

Figure 8.5: Realtime Detection

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it. If we talk about the basic definition of image processing then “Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality”.

Digital-Image :

An image may be defined as a two-dimensional function f(x, y), where x and y are spa-

tial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.
In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.
Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.

8.7 Testing it for a varying choice of parameters and Hyperparameters.

Figure 8.6: Parameters

Hyperparameter tuning is an art as we often call as "black function". Choosing appropriate hyperparameters will make the algorithm shine and produce maximum accuracy. Hyperparameter optimization techniques mostly use any one of optimization algorithms
a) Grid Search
b) Random Search
c) Bayesian Optimization

8.8 Deploying the final model.

Chapter 9

Take Aways From This Project

9.1 We got familiar with a subject yet not taught to us. (Machine Learning)

9.2 In a time where masks have become an absolute necessity, we need to bear in mind the importance of this simple thing. Not only do they prevent the spread of the infection from the infected, they also protect the healthy person wearing one.

9.3 In a place like a hospital, where all kinds of infections can be said to exist in the vicinity, it is a missing thing. So, we can look to incorporate this practice whenever entering a hospital.

9.4 From the government's perspective, this project can also aid in identifying the violation of the simple rule of "Mask Wearing".

Chapter 10

Conclusion

We hope to achieve a tangible impact over a real world problem that is actively plaguing the world using the recent advances in Data collection, Processing, Analysis and the application of Machine learning on the data.

10.1 REFERENCES

10.1.1 Andrew Ng/Deeplearning.ai

Figure 10.1: Parameters

10.1.2 Coursera

Figure 10.2: Parameters

10.1.3 Ashish Jhangra/Kaggle

10.1.4 ThePerceptron/YouTube

10.1.5 https://towardsdatascience.com/

10.1.6 MachineLearningMastery.com

10.1.7 www.pyimagesearch.com

10.1.8 www.tutorialspoint.com

10.1.9 www.googleblog.com

10.1.10 Instagram/MachineLearningIndia

10.1.11 www.hindawi.com

Chapter 11

References

We hope to achieve a tangible impact over a real world problem that is actively plaguing the world using the recent advances in Data collection, Processing, Analysis and the application of Machine learning on the data.

11.1 REFERENCES

11.1.1 Andrew Ng/Deeplearning.ai

deeplearning.ai

Figure 11.1: Parameters

11.1.2 Coursera

coursera

Figure 11.2: Parameters

11.1.3 Ashish Jhangra/Kaggle

11.1.4 ThePerceptron/YouTube

11.1.5 https://towardsdatascience.com/

11.1.6 MachineLearningMastery.com

11.1.7 www.pyimagesearch.com

11.1.8 www.tutorialspoint.com

11.1.9 www.googleblog.com

11.1.10 Instagram/MachineLearningIndia

11.1.11 www.hindawi.com