Project Report

on

AGE, GENDER, COLOR AND PATTERN DETERMINATION USING CNN AND PYTHON

Submitted in the partial fulfilment of the requirements for

the award of the degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

CSN KOUSHIK	**[ROLL NO.16311A04K9]**
D. NARESH	**[ROLL NO.16311A04P2]**
N. BHANU PRAKASH REDDY	**[ROLL NO.17315A0448]**

UNDER THE GUIDANCE OF

Mrs. D. PRADEEPTHI,	**DR. SHRUTI BHARGAVA CHOUBEY**
SR. AI Developer	**Associate Professor, Department of ECE**
SMART BRIDGE	**SNIST.**

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Autonomous)

Yamnampet (V), Ghatkesar (M), Hyderabad – 501 301.

April 2020

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Yamnampet (V), Ghatkesar (M), Hyderabad – 501 301

CERTIFICATE

This is to certify that the project report entitled ***"AGE, GENDER, COLOR AND PATTERN DETERMINATION USING CNN AND PYTHON"*** is being submitted by

CSN KOUSHIK [ROLL NO.16311A04K9]

D. NARESH, [ROLL NO.16311A04P2]

N. BHANU PRAKASH REDDY, [ROLL NO.17315A0448]

in partial fulfilment of the requirements for the award of **Bachelor of Technology** degree in **Electronics and Communication Engineering** to **Sreenidhi Institute of Science and Technology** affiliated to **Jawaharlal Nehru Technological University, Hyderabad** (Telangana)**.** This record is a bona fide work carried out by them under our guidance and supervision. The results embodied in the report have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide

Dr. Shruti Bhargava Choubey,

Associate Professor, ECE Department

Project Coordinator

Dr. Shruti Bhargava Choubey,

Associate Professor, ECE Department

Head of the Department

Dr. S.P.V. SUBBA RAO

Professor, Department of ECE

Signature of the External Examiner

DECLARATION

We hereby declare that the work described in this report, entitled "***AGE, GENDER, COLOR AND PATTERN DETERMINATION USING CNN AND PYTHON***" which is being submitted by me in partial fulfillment for the award of **Bachelor of Technology** in **Electronics and Communication Engineering [ECE], Sreenidhi Institute Of Science & Technology** affiliated to Jawaharlal Nehru Technological University Hyderabad, Kukatpally, Hyderabad (Telangana) -500 085 is the result of investigations carried out by us under the Guidance of **Dr. Shruti Bhargava Choubey,** Associate Professor, ECE Department, Sreenidhi Institute Of Science And Technology, Hyderabad. The work is original and has not been submitted for any Degree/Diploma of this or any other university.

Place: Hyderabad
Date:28/3/2020 Signature

CSN KOUSHIK, [ROLL NO.16311A04K9].
D. NARESH, [ROLL NO.16311A04P2].
N. BHANU PRAKASH REDDY, [ROLL NO.17315A0448].

ACKNOWLEDGEMENTS

We thank **Dr. Shruti Bhargava Choubey (Internal Guide)** and **Dr. Shruti Bhargava Choubey (Project Coordinator)**, Associate Professor, Dept of ECE, Sreenidhi Institute of Science & Technology, Hyderabad for their valuable advice, support and help during our research work. We thank to all my teachers and professors for their valuable comments after reviewing our research papers.

We would like to express our sincere gratitude to **Dr. S.P.V. Subba Rao**, Professor, Head of the department, Electronics & Communication Engineering, Sreenidhi Institute of Science & Technology, Hyderabad for his continued support and valuable guidance and encouragement extended to us during our research work. We thank him for his painstaking efforts to guide us throughout our research work.

We wish to extend our special thanks to the **Executive Director Dr. P. Narasimha Reddy, the Director Shri Abhijit Rao and the Principal Dr. T. Ch. Siva Reddy**, Sreenidhi Institute of Science and Technology, allowing us to be the part of this wonderful opportunity provided to us by our academic curriculum.

We wish to extend my special thanks to all my friends who helped directly or indirectly to complete our research work. We extend our thanks to our parents and all our family members for their unceasing encouragement and support who wished us a lot to complete this work.

CSN KOUSHIK [ROLL NO. 16311A04K9]

D. NARESH [ROLL NO. 16311A04P2]

N. BHANU PRAKASH REDDY [ROLL NO. 17315A0448]

SMARTBRIDGE

Date: 20/05/2020

PROJECT COMPLETION CERTIFICATE

This is to acknowledge that the under mentioned students of **B.Tech (ECE)** 4th year of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** have completed their Major project with great success at our concern, with the title **"Determination of Age, gender, color and pattern using python"** in the Domain **Machine Learning** from **10th Jan 2020** to **20th May 2020 (students were engaged in offline mode from 10th Jan – 20th Mar and online mode from 20th Mar -20th May).**

CSN Koushik	**R.NO 16311A04K9**
D. Naresh	**R.NO 16311A04P2**
N. Bhanu prakash reddy	**R.NO 17315A0448**

Their project is found to be useful in the relevant business and they have Submitted a copy of the project report to us. During their project period we found them Sincere, hardworking and possessing a good behavior.

We wish them grand success in their future endeavors.

Thanking you,

SmartBridge Educational Services Pvt. Ltd. Hyderabad

Jayaprakash.Ch,
Program Manager

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The data which has been collected of tremendous amount in nature is to be handled meticulously such that any errors which maybe generated or received needs are to be handled with utmost care. The data can be unordered or ordered based on the one's requirement for the task or ease of handling the data. The neural networks have been evolved in order to mimic the actions of the human brain, such that based on the amount of the training that it has received; the outcomes are generated for the data which is given as input to it. The images taken are to be processed/ analysed and for that the image needs to be segmented. Each node in the artificial neural network can have their weights updated in order to reduce the cost function to a large extent based on the data it receives at the time of the forward and the backward propagation. There are various activation functions used for the neural network based on the type of the distribution of the data. The ANN used for the image processing is called as the (CNN) convolutional neural networks. The data given to this neural network is processed or analyzed by the help of the convolutions that are done on the data. The convolutions are done on the values of the image which is sampled or segmented based on the intensity values. The coding is done in the python language and the reason it is chosen was that it supports for the high number of the packages like the Keras, Tensorflow, OpenCV etc which play a major role in the coding for the development of the neural network and the processing of the image or the frame that has been captured. The packages are imported by the help of the creation of the instantiation of the package. The determination of the age, gender and the color and the pattern is done by the help of the CNN wherein the data is taken in the form of the video stream and the predictions are done by the use of the pre-built models namely the caffe and the minivggnet model by the use of the opencv package.

CHAPTER-1

INTRODUCTION

1.0 INTRODUCTION

Huge amounts of bulk quality data is produced on a daily basis constituting up to Zeta Bytes of data and they are to be handled as per one's requirement and the data has to be stored or it has to be used for the other applications. Neural networks are to be developed which have the capability to analyse/process the image or input data that is being is given input to it such that it can mimic the functionalities of the human brain. The input images can be given to an artificial neural network (ANN) which altogether is named as convolutional neural network, it performs the functionalities like the max pooling and the feature extraction etc. such that all the properties of the images can be extracted and trained by it. The cost function can be evaluated based on the updation of the weights by the forward and the backward propagations. The activation function namely the RELU can be used in general. The input image needs to be made error free and digitized such that the image can be segmented or resolved into various pixels such that they can be represented in binary values. The images will be represented with numerical values that are taken at every pixel, such that the values are taken and convolved with itself in a matrix manner. The matrix size can be defined by the user based on the number of pixel count or image's resolution. The image resolution will also govern the delay or the processing speed or the performance of the entire network. The entire network will be convolving the input matrices and then later the characteristics can be determined after the computation. Each node in the neural network can be considered as the processing element like the adder and a long short-term memory element such all the weights can be updated accordingly to generate the outcomes. A predictor code also needs to be written for the potential trained network such for the sake of the prediction of outcomes from the input test data input.

Python is a high programming language that is very easier to code when compared to that of the C/ Java programming languages. It supports for

various built in libraries that can be used easily by the programmer. There are various packages available in the python programming language such that they provide the designer or the programmer a large variety of options and the ease of the programming such that they can be used very frequently like the Tensorflow, Keras and Pandas etc. [1-11]

1.1 BRIEF DESCRIPTION

1.1.1 INTRODUCTION TO DATA AND DATA SCIENCE

Data is the characteristic of any particular entity or any physical quantity. The data can be stored and used in various formats depending on the end user requirement. The can be stored in the form of the structured or in an unstructured manner. In the structured manner, it generally represents the data in the form of the tables, in a well organized manner. The data can be accessed by the help of the SQL queries. The other form of the data representations are the semi structured manner and the unstructured manner. The unstructured representation of the data is represented in an unorganized manner wherein the data in this form of an audio or an image or a video files. The data is of various formats like .txt files, .csv files, and .mp4 files. jpg files, .png files etc which will be generated in huge amounts regularly in the order of peta or zeta bytes of data. [1-5]

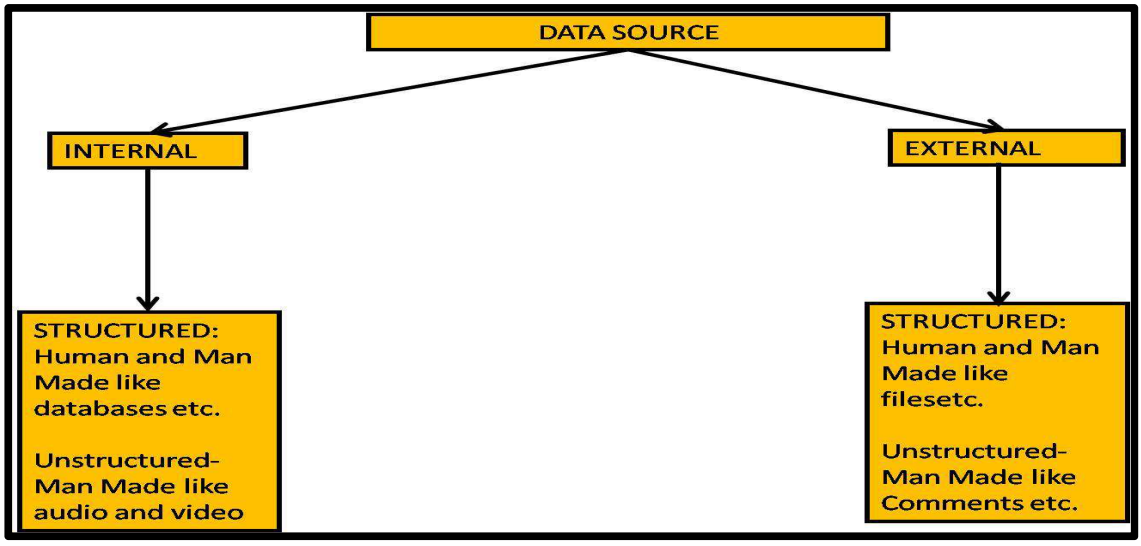


Figure 1.1: Data Formats used in Data Science-1

Data science is science of handling the data wherein the data is handled by a data scientist who implements the principles of mathematics, science, statistics, programming languages and that of the business economics. The data scientist can implement their own algorithms by modifying the already existing algorithms or creating the new algorithms that increase the model's entire accuracy. The data mining algorithms are limited in potential and may tend to give limited outcomes; hence the data scientist can use their enhanced versions to generate better outcomes. [6-11]

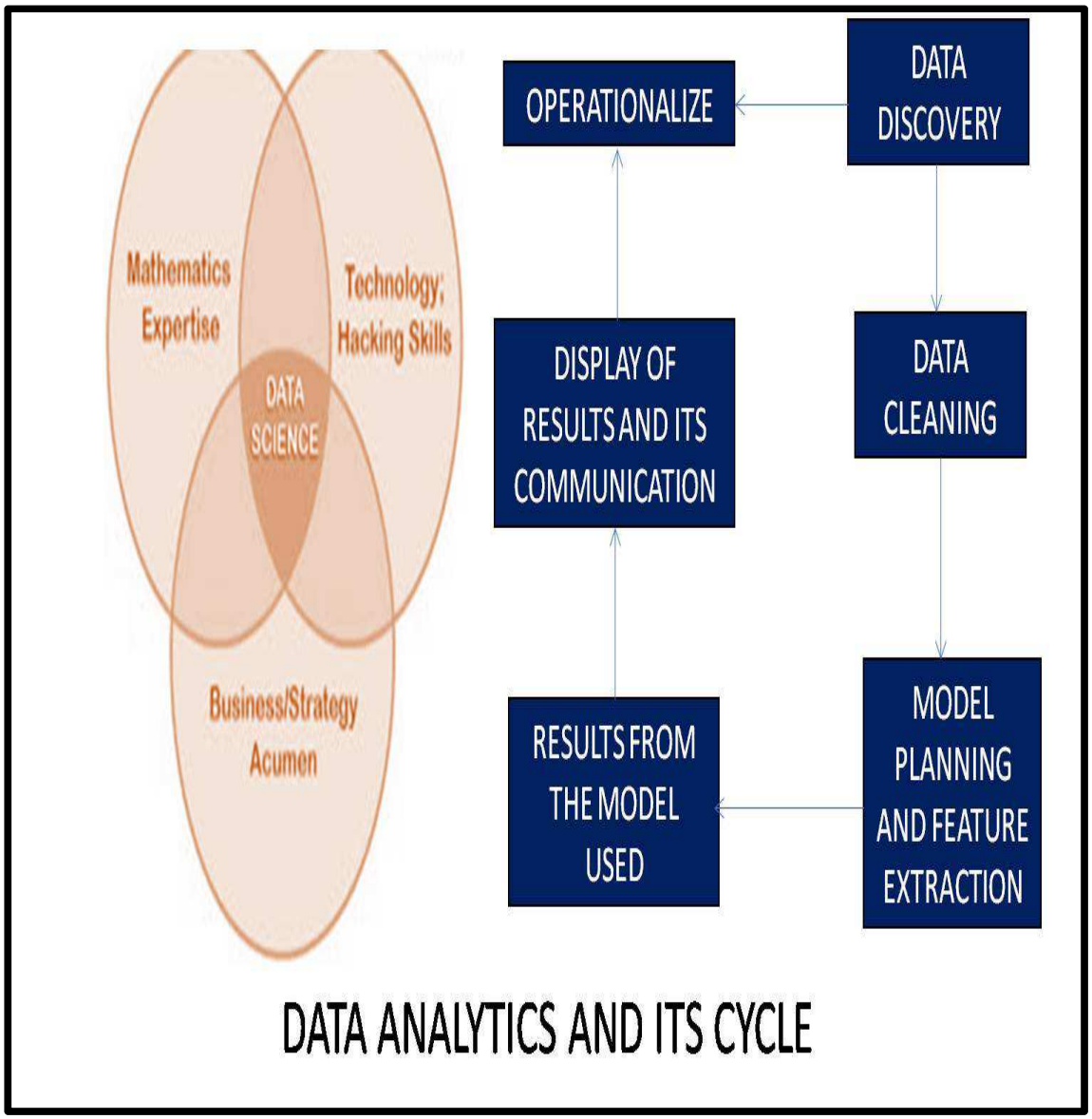
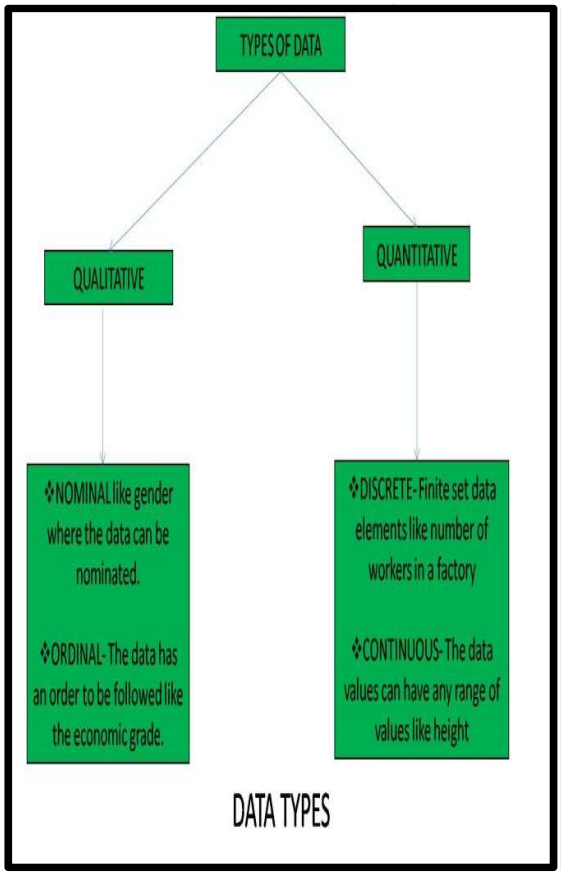


Figure 1.2: Data Science and its Lifecycle



	ANALYSTS	DATA SCIENTISTS
Types of data	Structured and semistructured, mostly numeric data	All types, including unstructured, numeric and nonnumeric data (such as images, sound, text)
Preferred tools	Statistical and modeling tools, usually contained in a data repository	Mathematical languages (such as R and Python), machine learning, natural language processing and open-source tools that access and manipulate data on multiple servers (such as Hadoop)
Nature of work	Report, predict, prescribe and optimize	Explore, discover, investigate and visualize
Typical educational background	Operations research, statistics, applied mathematics, predictive analytics	Computer science, data science, symbolic systems, cognitive science
Mind-set	Percentage who say they: •are entrepreneurial: 69% •explore new ideas: 58% •gain insights outside of formal projects: 54%	Percentage who say they: •are entrepreneurial: 96% •explore new ideas: 85% •gain insights outside of formal projects: 89%

Figure 1.3: Types of Data and its utilization in Data Science

Data Scientist	Data Analyst
Data scientist should be familiar with all the database system like MySQL, Hive etc	Data Analyst should be familiar with Data Warehousing, business intelligence and in-depth analysis to SQL.
All the data scientist should have a basic knowledge on Python, Java, Job Developments for MayReduce.	Should have a strong base on Hadoop Analytics (Hive, HBase, MapReduce job, Impada, Cascading)
Should have strong understanding in various analytical jobs-Ranking, collecting data and using them on data sets.	Should know about data storing, retrieving skill if data is mistakenly lost and having knowledge of new tools
Expert is Mathematics, Statistics, Data Mining, Correlation, predictive analysis and prediction for future of business decision.	Experts in determining tools and components of data architecture.
If a Data Scientist knows R then it acts as an added advantage.	Should know different tools: that will help transforming the source data to analytical data sets.
Have a Deep statistical insights and machine learning knowledge like- Mahout, Bayesian, Clustering.	Able to take Business decision and take critical business features for real time.

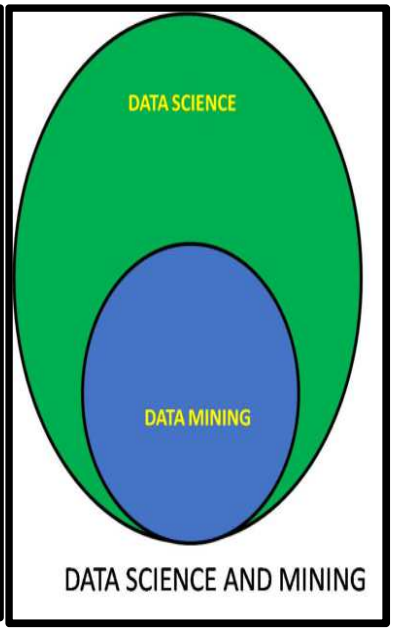


Figure 1.4: Data Science and Data Mining

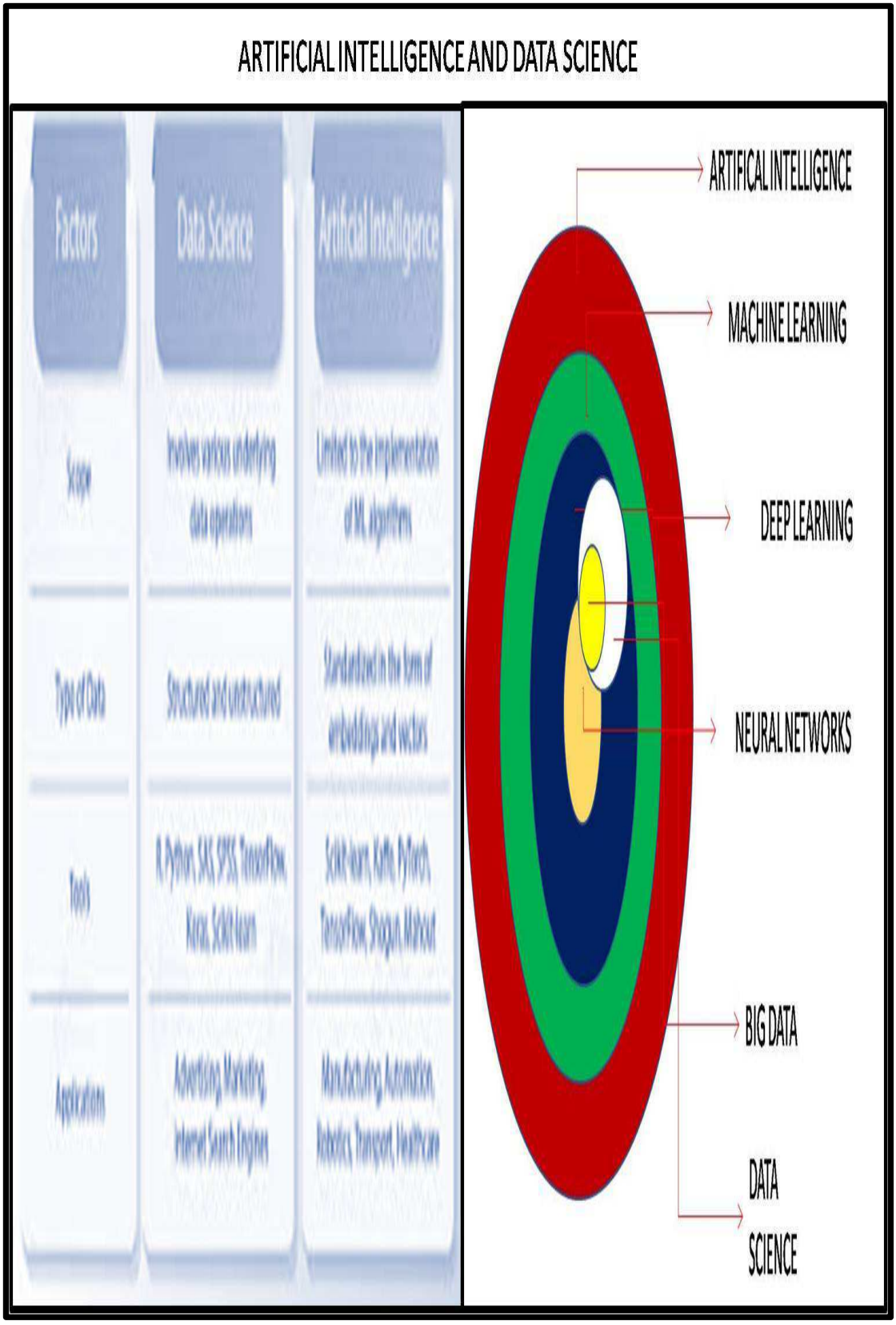


Figure 1.5: Data Science and Artificial Intelligence

1.1.2 PYTHON PROGRAMMING LANGUAGE

Python is a high level programming language which is a high level programming language and it was invented by Guido Van Rossum around the year of 1990's but it got its boom in the late 2000's as people were not ready to accept it and many technologies were predefined for other programming languages. It is very easier to code while compared to that of the C/ Java programming languages. It has a support for enormous pre-constructed libraries that may be used by the programmer. There are various packages available within the python programming language such that they offer the programmer a large style of alternatives and the ease of the programming such that they may be used very frequently. Packages like the Tensorflow, Keras, Pandas, Scikit etc, and many more are the useful available packages. It supports for the object oriented programming language. The main difference between the usage of java and python is the problem of the code redundancy and ease of use and the number of packages available to be used at that instant.

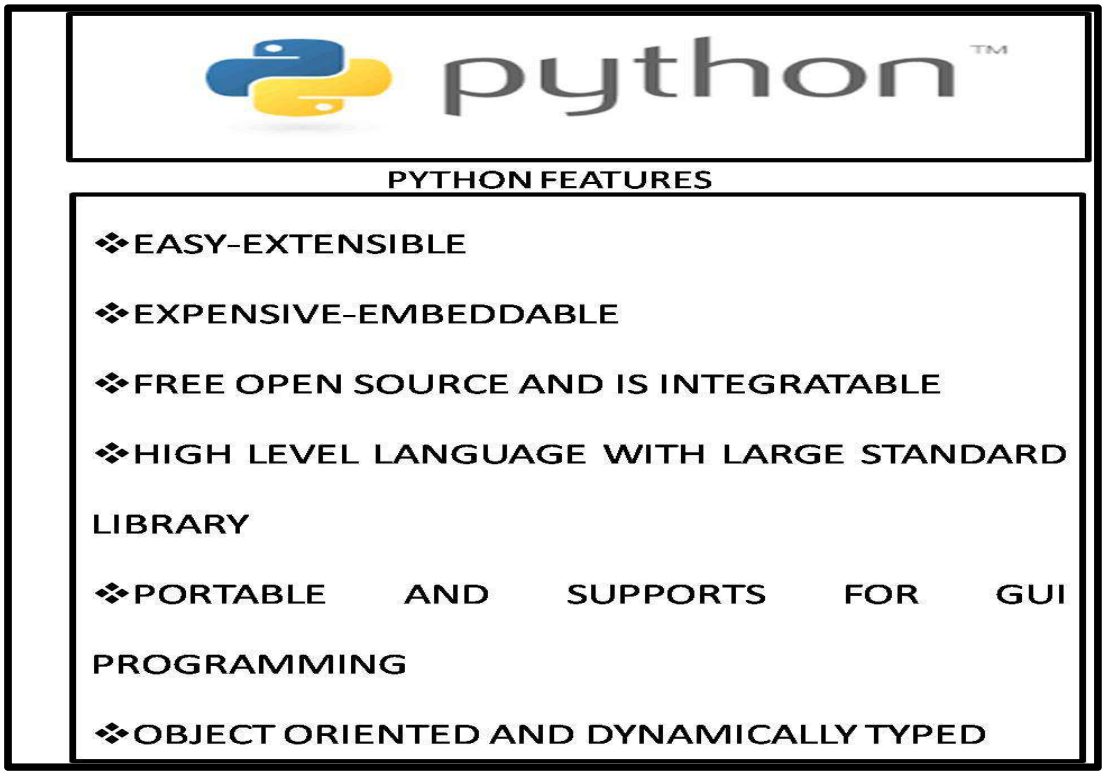


Figure 1.6: Python and its features.

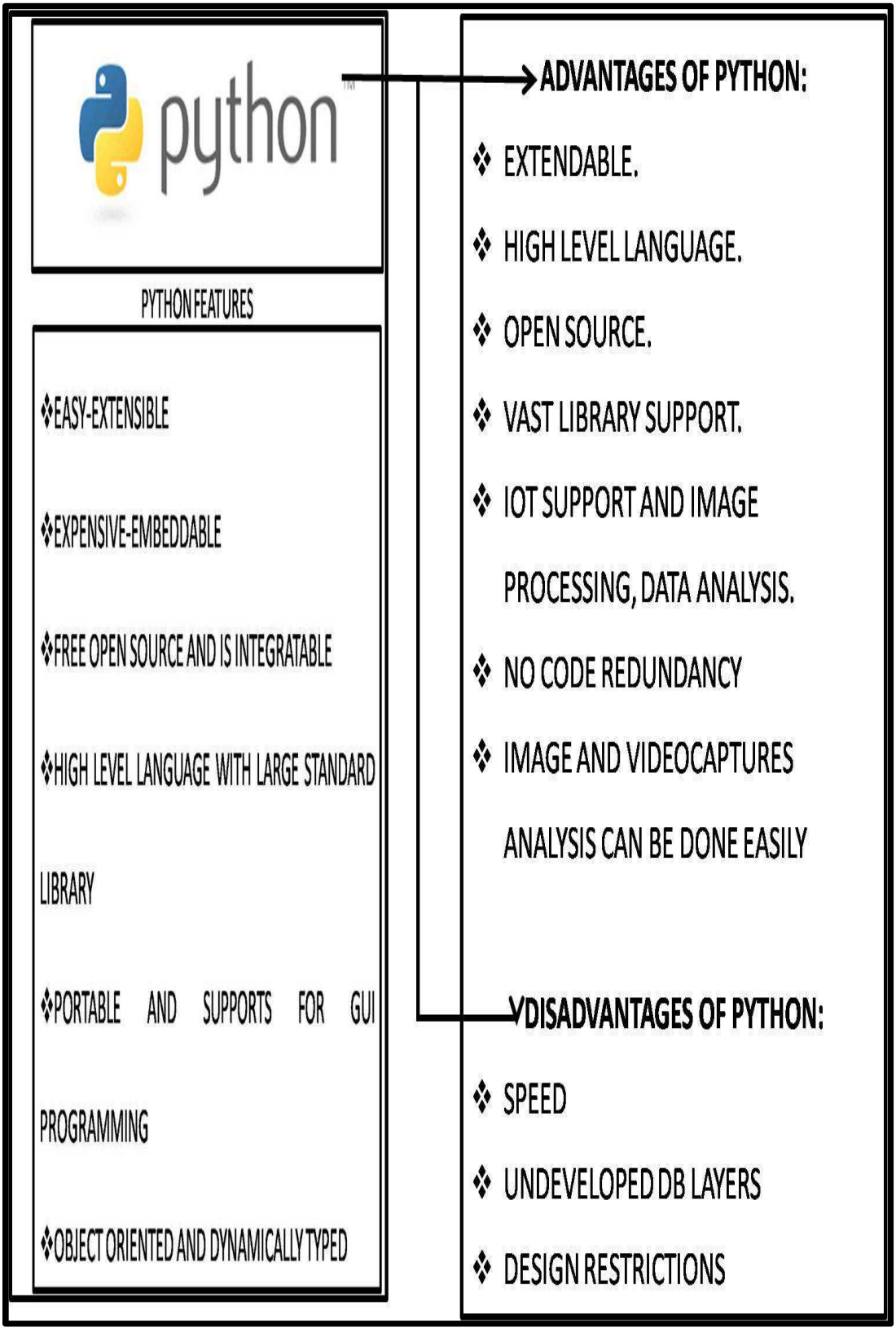


Figure 1.7: Advantages and Disadvantages of using python

1.1.3 ARTIFICIAL INTELLIGENCE

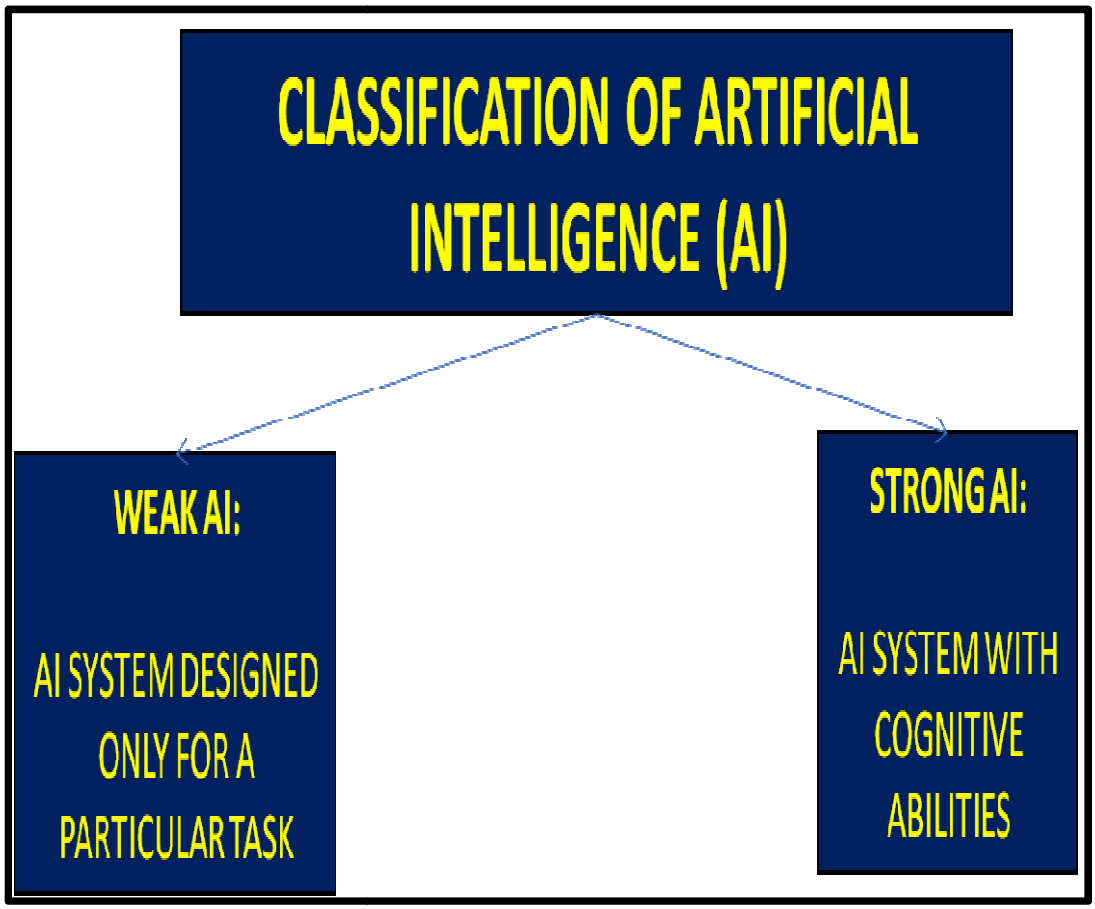


Figure 1.8: Classification of AI

Artificial Intelligence is the software program that is used to generate the outcomes or predict them by mimicking the human intelligence. The base of the AI model is a Neural Network which processes the inputs and generates the outputs which are very sophisticated with the help of the activation functions. It supports for the automations and this upcoming technology makes the life of people much easier. By the affiliation of the picture processing and the neural network, the consequences can be generated as top as that of the human brain belief. As the AI/DL is a requirement for the destiny programs that allows you to guide the automation and technology of the effects by itself without the intervention of the human beings. So, the destiny is generally tend to rely solely in this upcoming technologies such that life of the people may be made a whole lot

extra less difficult. It reveals application inside the sectors just like the clinical sectors, defence or protection purposes most probable to present the info of the man or woman's with recognize to the age, gender, colour and pattern etc. It will find a foremost function within the detection of the above said elements which facts them for the sake of the further analysis. Entire future depends upon it as it supports for the automation of tasks and reduces the manual intervention and reduces the time wastage. It can subdivide into categories of machine and deep learnings to enhance the model's accuracy which is being developed. In order to achieve the mimicking the human intelligence, there is a requirement to develop an artificial brain which is possible only when many perceptron nodes developed must be connected at the synapse artificially in contrast to that of the biological perceptron nodes. [6-11]

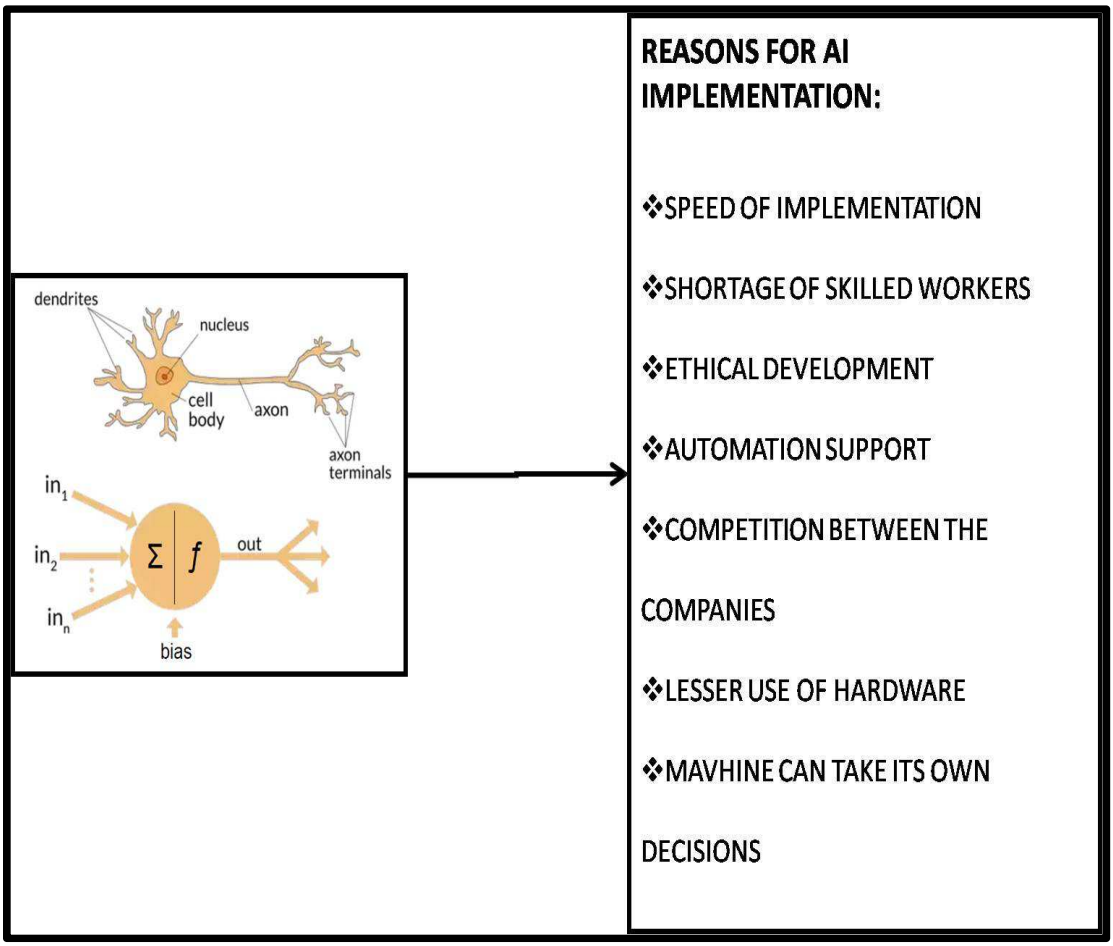


Fig 1.9: Reasons for AI.

1.1.4 MACHINE LEARNING

In order for the Artificial Intelligence to make progress, it must be able to understand the outside world and its surroundings and understand the inputs that it can be given for it to successfully interpret the messages. Multimodal machine learning builds models such it that can process and relate information from multiple models available. From early research to the recent researches on the models, multimodal machine learning is a key multidisciplinary field with increasing importance and potential. The disadvantage with machine learning models is that even though the model may give good accuracy but it does not operate over the previous inputs and predicts only based on the present inputs and that will be the only drawback. [6-11]

If it is used with the help of the deep learning models like that of the neural network, gives better results. Irrespective of the multimodal machine learning models, there are 3 basic types of the machine learning models based on type of data. The data can be either categorical or continuous. They are namely the supervised, unsupervised and reinforcement learning.

In supervised learning, there are algorithms like the classification and the regression which can generate the desired outcome under the control of a parameter or a variable whereas in unsupervised learning, the output generation will be done by the model by operating over the unknown data and there is no parameter or any variable to govern the entire task. For the sake of the type of the data on which it needs to operate, different algorithms have been developed, namely the classification for the sake of the categorical data while the regression is used for the sake of the continuous data element. [6-11]

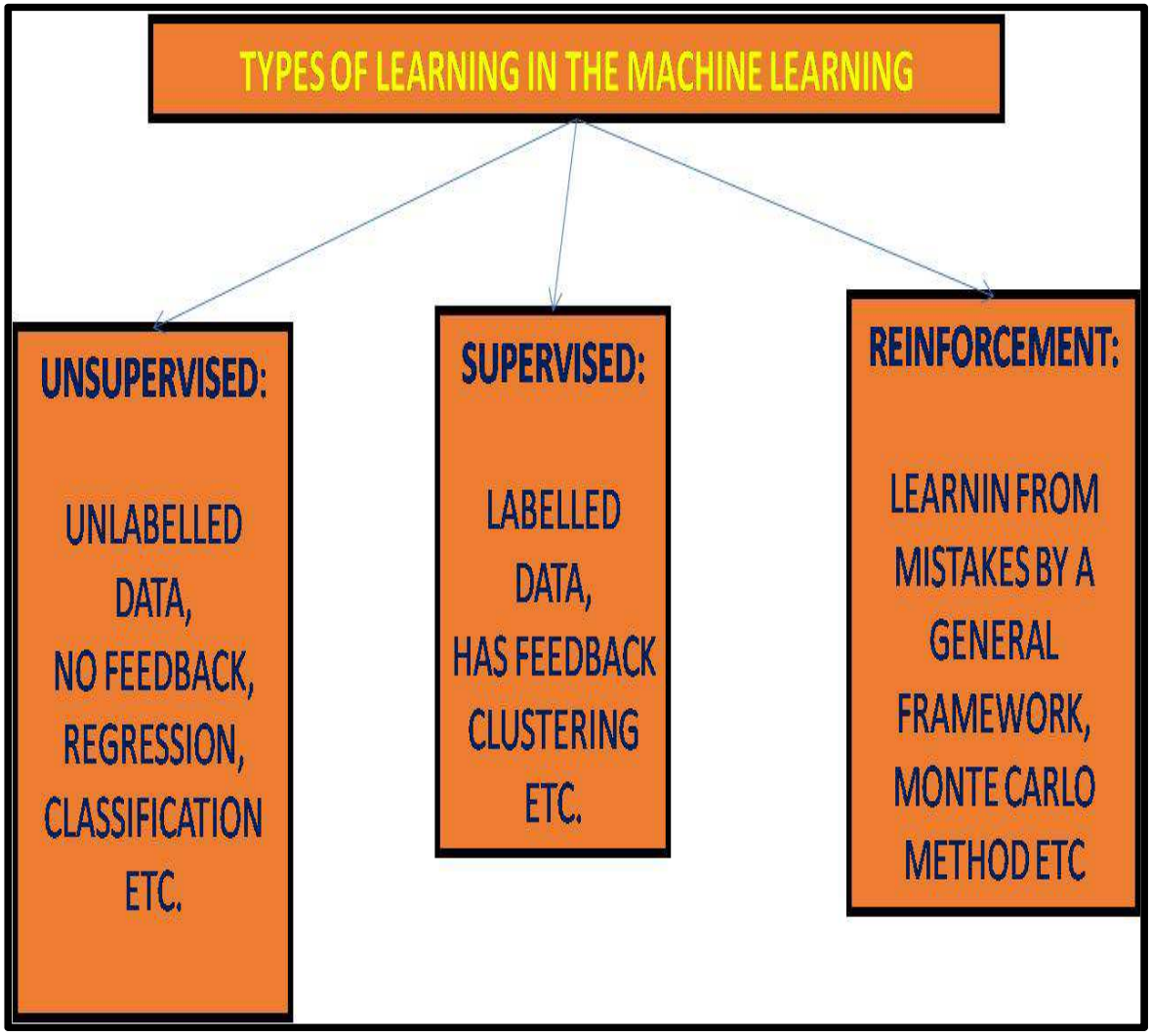


Figure 1.10: Types of Machine Learning-1.

- **Supervised Learning:** The model is given input with the categorized dataset that is a labeled dataset. It already has input and output parameters predefined such that the output is determined based on the input parameters. So, when a new dataset is given, the supervised learning analyses the statistics and produces the best output which is consistent with the categorized data given. Examples of the algorithms are: Regression and Classification etc.
- **Unsupervised Learning:** In unsupervised learning, the machine will not be given any labeled dataset in advance rather it must be able to generate the labeled dataset from the unlabelled dataset. The algorithm is

designed to learn by itself, without any data supervision. Example Clustering etc.

- **Reinforcement Learning:** The system or the model attempts to discover or learn the surest answer and it itself learns from the mistakes. As it learns from the wrong answer, by the help of the feedback path, the accuracy of the model being developed can be enhanced to a large extent.

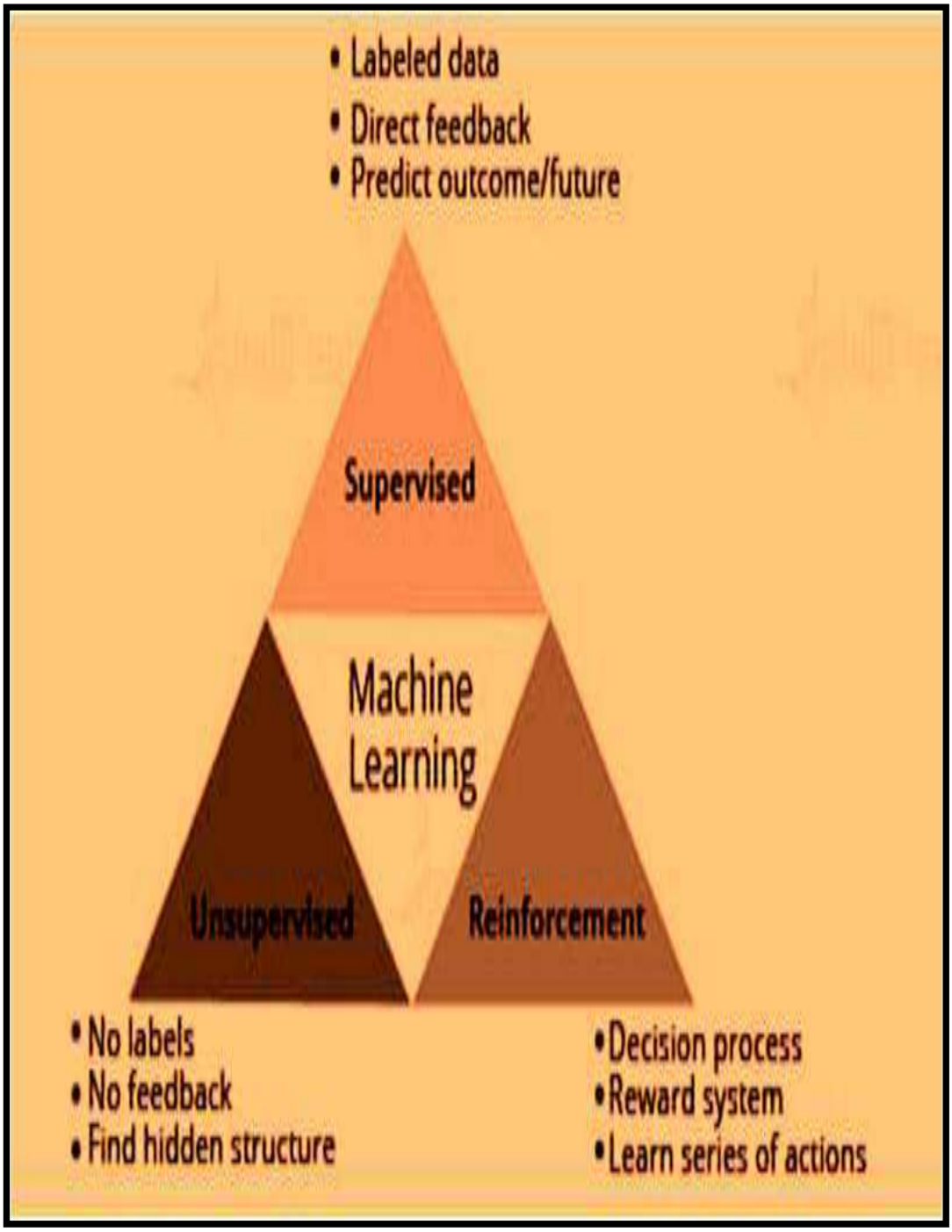


Figure 1.11: Types of Machine Learning-2.

Multimodal Machine Learning Research has some challenges for the computational researchers based on the given the data heterogeneity. There are 5 main challenges in using the machine learning model, namely the:

- Representation of the data for the model,
- Translation of modal data to another modality,
- Alignment of the modal data which is to be mapped among the modalities.
- Fusion of the aligned data of the modalities.
- Co-learning for the sake of the exchange of the knowledge between the modalities.

These are the challenges that one must handle and produce the outputs. These challenges can also be considered to be the procedure of the machine learning model to be implemented. If the data is not suitable to be operated like in case of the strings, the data needs to be encoded by the help of the one hot encoding algorithm. In machine learning, there is no training done on the data and the outputs are predicted on the present inputs only. [6-11]

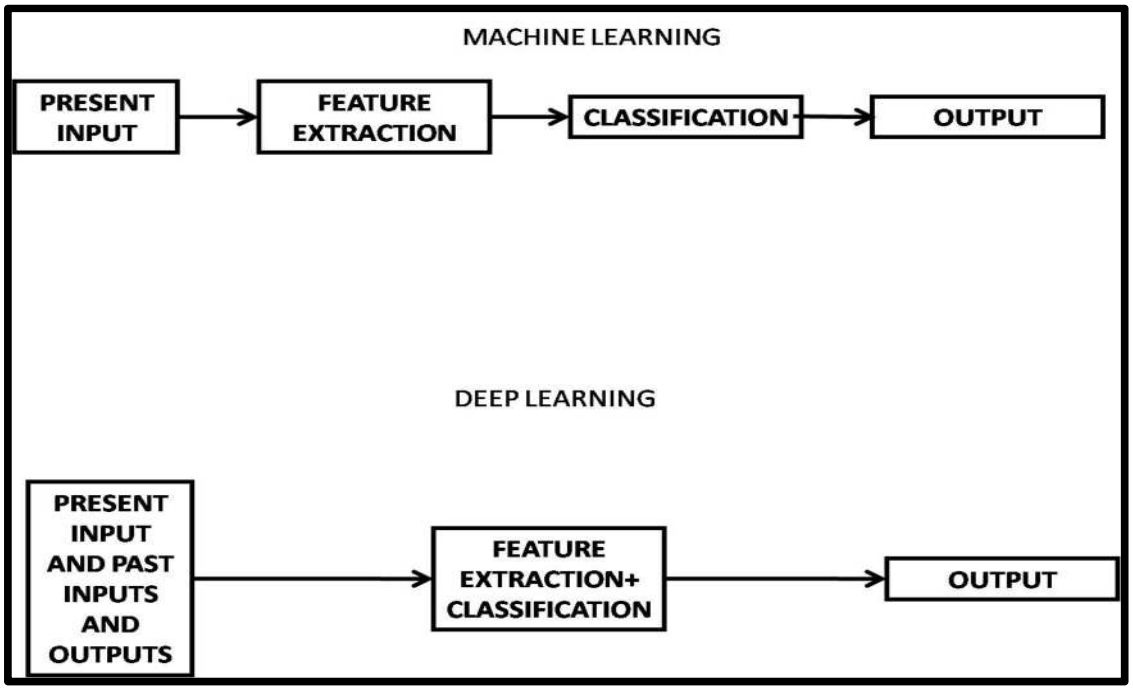


Figure 1.13: Machine learning v/s Deep Learning.

1.1.5 DEEP LEARNING

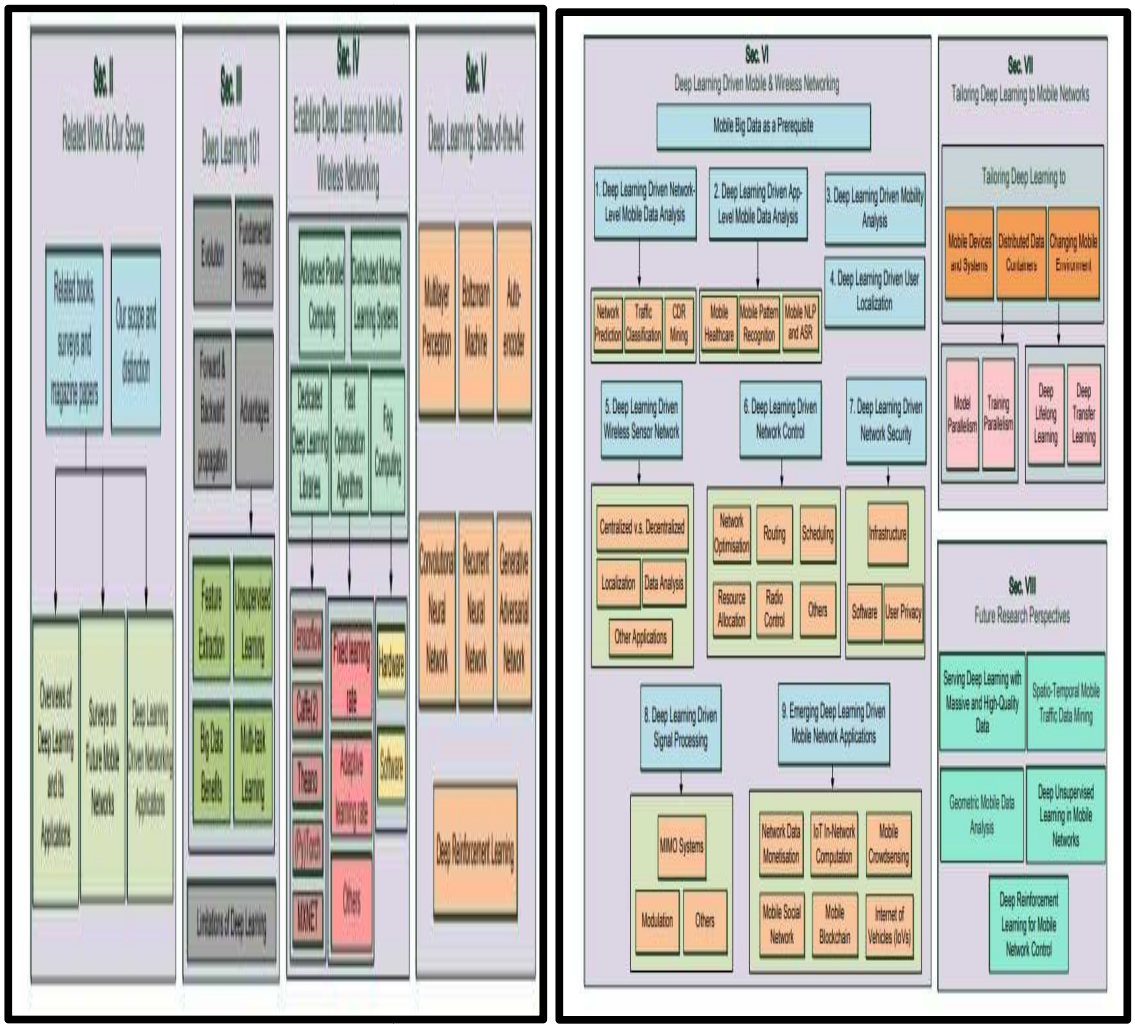
Deep learning is the part of the artificial intelligence wherein the complex algorithms are implemented by the help of the neural networks. They learn about the entire data at a large extent such that the network gets accustomed to it and the data is then later validated. The deep learning have to tend to come into effective usage since the previous 2 decades where in complex problems have been easily solved by the machine to take the decisions and give the desired outcomes. The multi-layer neural networks are always used for the sake of the optimal solutions but they consume much computational resources due to which the overall throughput will reduce but the accuracy of the model is enhanced by a large extent. [6-11]

The increasing demand of computing power and data are the key factors that govern the deep learning usage. Since the big data evolution, large amounts of input data samples are produced and collected which are used especially mostly to train the deep learning models for which high-performance computing systems are needed, they are the GPU-based frameworks, they reduce the training time for models is reduced. [6-11]

Deep learning with its great performance is found to be very useful and used in applications like that of the computer vision, speech recognition and processing. The deep learning models can extract the high-level features and learn the hierarchical representations by combining the low-level input more effectively. Deep learning (DL) has been a increasingly growing field of research since 2006, redefining state-of-the-art performances in a wide variety of areas such as object recognition, image segmentation, speech recognition, and machine translation. Data-driven machine health monitoring is gaining popularity in modern manufacturing systems, due to the widespread deployment of low-cost sensors and their Internet access. In the meantime, deep learning provides valuable tools to process and evaluate these big machinery data [6-11].

As a machine learning branch, deep learning attempts to model hierarchical representations behind data and identify (predict) patterns by stacking in hierarchical architectures multiple layers of information processing modules. Recently, in various fields such as computer vision, automated speech recognition, natural language processing, audio recognition and bioinformatics, deep learning has been widely implemented. Deep learning is simply not a new concept, even dating back to the 1940s.[107]

The main goal of deep neural networks is to approximate complex functions by a composition of basic and predefined unit (or neuron) operations. Such an objective function may be almost of any kind, such as mapping images with their class labels (classification), calculating future stock prices based on historical values (regression), or even deciding on the next optimal step of chess provided the current board status (control) Depending on the model's structure, the operations performed are typically determined by a weighted combination of a particular group of hidden units with a nonlinear activation function. These operations are called "layers" along with the output units. The neural network architecture is analogous to the mechanism of perception in a brain, where a particular collection of units is triggered given the current environment and influences the performance of the neural network model.[106-108] In addition to CUDA coding for GPU parallelization, designing a deep learning model from scratch can be difficult for engineers, as this includes definitions of forwarding behaviors and gradient propagation operations at each layer. With the increasing popularity of deep learning this phase is simplified by many dedicated libraries. Most of these toolboxes operate with multiple programming languages and are built with GPU acceleration and automatic support for differentiation. This removes the need for hand crafted gradient propagation concept.



1.14 Diagrammatic view of Deep Learning in procedure.

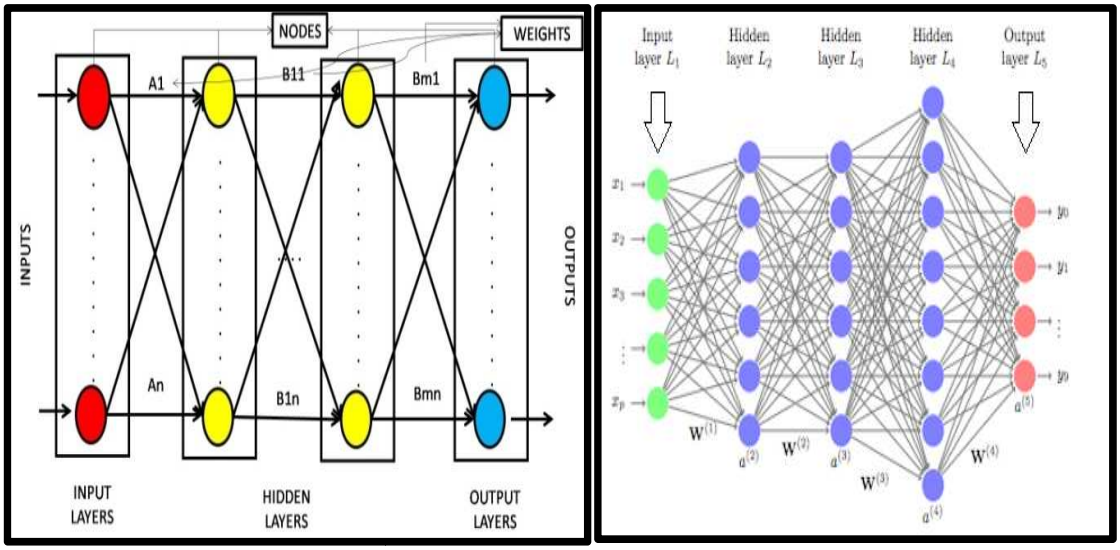


Figure 1.15: Neural Network.

1.1.6 NEED FOR AI/ML/DL AND THEIR DIFFERENCES

Intelligence maybe reduced to a black box that performs a series of insightful decisions. For a decision to be made, a series of rules are needed to be defined. Earlier in the field, rules were predefined via heuristics and/or intelligent engineering design (human intelligence). These were not as useful since the rules were based on human reasoning which comes short when higher dimensions and volumes of data are involved and the patterns in the data which would inform the decision made by the 'Artificial Intelligence' are much more complex to be reasoned by human intelligence (deductive or inductive). Artificial Intelligence, Machine Learning and Deep Learning are the interconnected fields belonging to the category of the AI. They support the Artificial Intelligence models by providing a set of algorithms to the neural networks to solve the complex problems. [6-11]

AI is becoming a need in our daily life. There is a lot of hype about AI since the beginning of IT sector. In the 1950's, people thought that IT is for creating AI. Under this assumption, languages like LISP had come up in computer science since 1958 as the second-oldest high-level programming language. The assumption has made computer scientist work on Artificial Intelligence than Normal Programming. It is going to be used by human in all walks of life. So, by the use of the Artificial Intelligence the lives of the people have been made easier and comfortable and it will be made more once if it is fully deployed in a large scale as it supports for the automation of the tasks and can generate better results than the normal human being. AI is also used for Autonomous cars as an essential component. In AI society, owning a car become more troublesome. An AI model with IoT will take care of your needs at the time of the driving, because it maintains, parks, drives it in a safe manner by the use of the image processing applications due to its decision taking capacity. Thus reducing the traffic and damage to the property by accidents and make like comfortable to the people. [6-11]

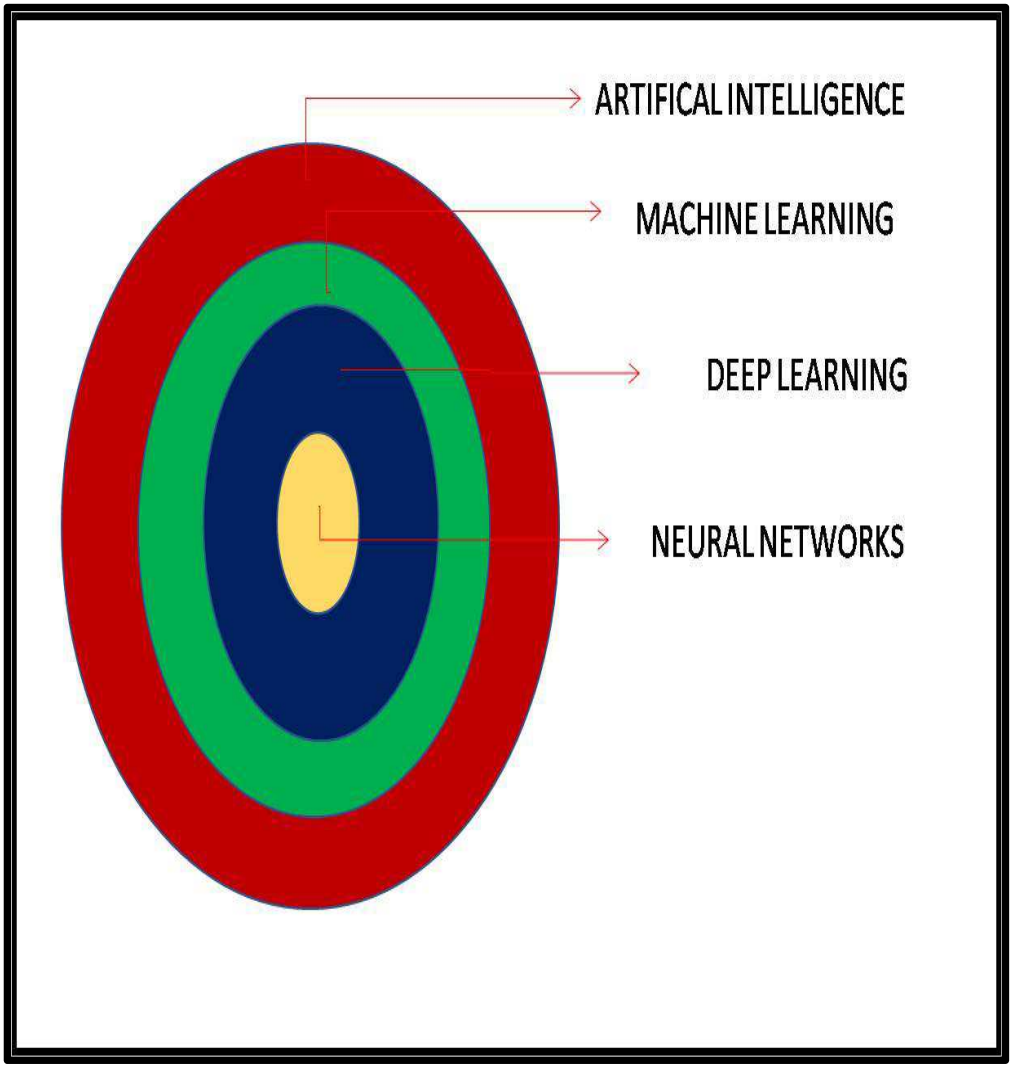


Figure 1.16: Categorization in Artificial Intelligence-1.

Machine learning makes it possible to build systems which can handle tasks which cannot be programmed. Some tasks are simply impossible to code. Imagine trying to build a robot to drive a car, you might be presented with an almost unlimited variety of different situations, each of which would require a split second response. Machine learning not only allows computers to respond to tasks they have been taught, it allows computers to make leaps of intuition - to recognise tasks which are similar enough to previous experience to solve. [6-11]

Deep learning is a constituent of artificial intelligence which solely mimics the working of human mind's intelligence so as to process information and developing the patterns. Deep Learning is a subset of

machine learning which is a subset of artificial intelligence (AI) is in turn. It has networks capable of processing any kinds of data. These estimate the statistical parameters and the probabilities for the large data collected and they are being implemented in many applications which are popular.

- **Artificial Intelligence:** It is the science of enabling machines or systems to take the decisions for it by thinking like a human.
- **Machine learning:** It is the subset of AI, focusing essentially to make the decisions based on the input data only.
- **Deep learning:** It is the subset of ML, essentially used for the sake of the development of the neural networks in order to solve complex problems just by the help of the training of the systems for solving complex problems.

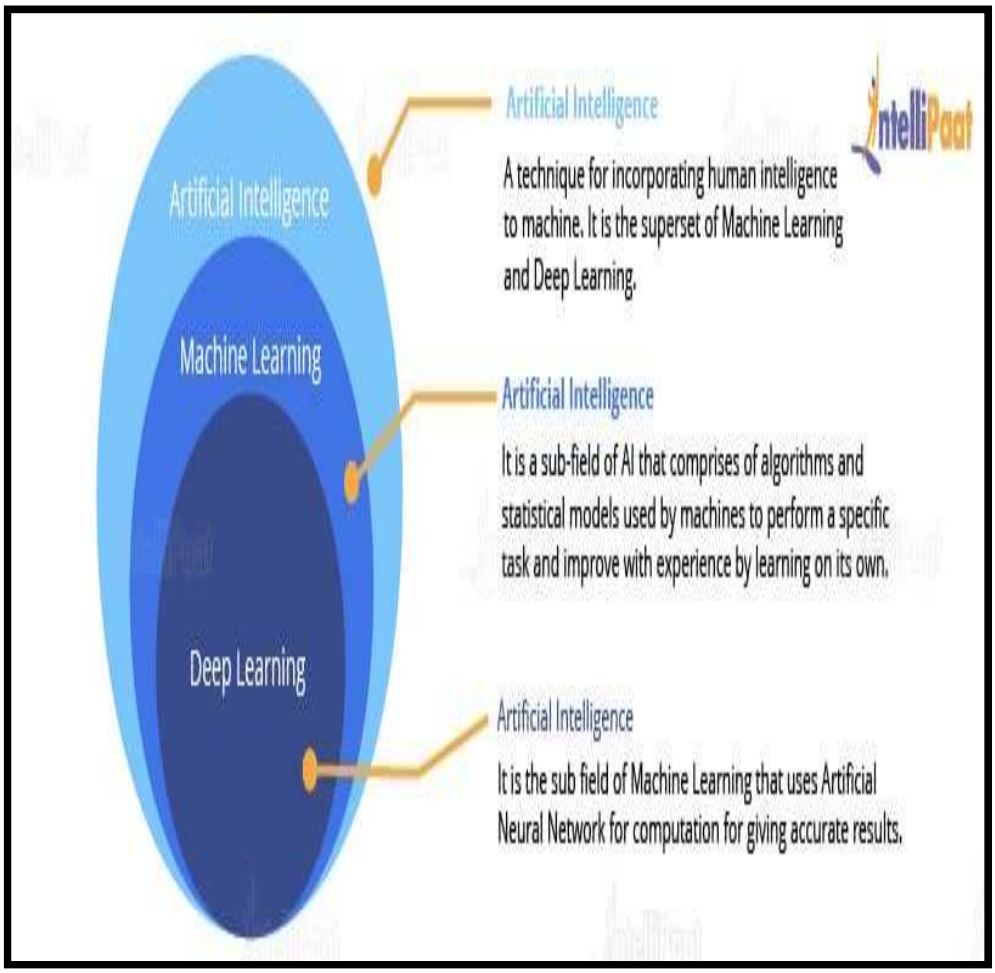


Figure 1.17: Categorization in Artificial Intelligence-2.

1.1.7 NEURAL NETWORKS

A Neural Network is considered as a deep earning algorithm or a machine learning algorithm. Machine Learning is a branch of Artificial Intelligence that dealing with pattern / behaviour understanding and simulation. Machine learning algorithms work upon large sets of data. The machine learning model derives its behaviour/characteristics from the training data that you provide. Once it learns, if you throw it a similar data, it is able to predict the outcome. Neural Networks are developed in order to make decisions in a situation just like how the human mind solves the complex problems instinctively and intelligently by considering all the merits and the demerits of the task to be done. Neural Network mimics the functionality or the decision taking capacity of the human brain. The human brain has many perceptron nodes which are the centre for the analytical and cognitive capability of the individual; similarly, an Artificial Neural Network (ANN) is comprises of multiple "perceptrons" just like the perceptron nodes in the brain. Network must be in a position to predict or generate the outputs based on the inputs given to it after training itself for a large amount of data. [6-11]

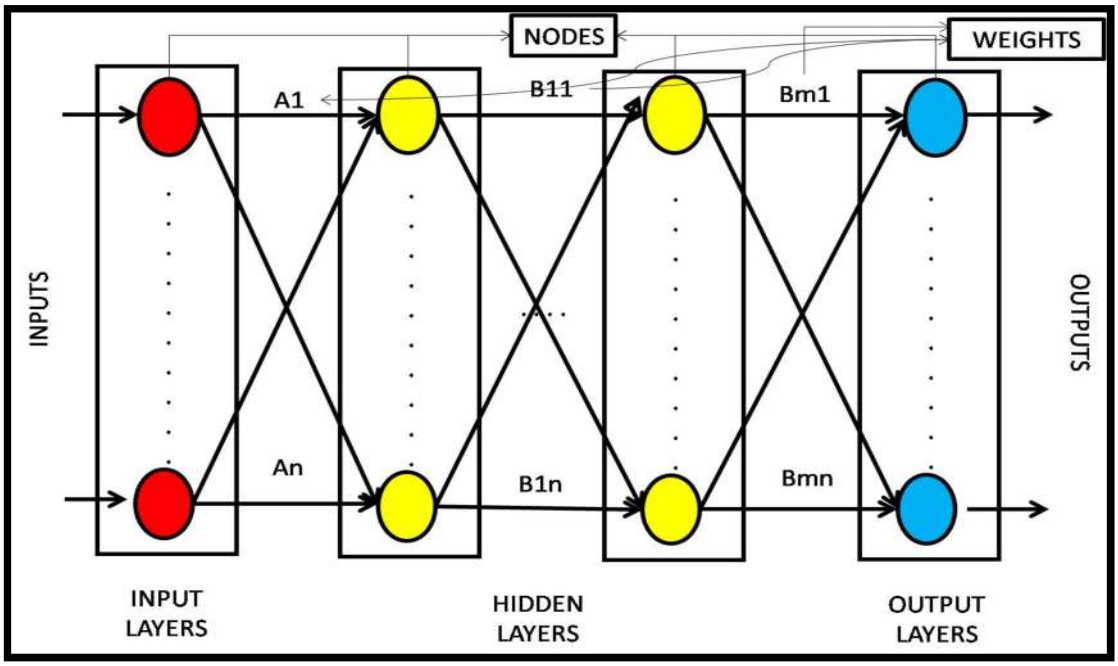


Figure 1.18: Neural Network

A neural network has three main constituent layers mainly and they are the:

- **Input Layer:** Layer to take the input raw test or train data.
- **Hidden Layer:** It is present in between the input and output layer wherein many operations are performed and consist of the perceptrons. The cost function is updated by the forward and backward propagation. There is high demand for the multiple hidden layers due to its better performance.
- **Output Layer:** The layer for the outputs is got and used in the feedback.

Neural networks are understood by the help of "Perceptrons". Perceptrons are a single layer neural network which classifies the linear data. It has 4 main constituents, namely the:

- Input data elements,
- Weights and Bias at the nodes,
- Summation Function at the nodes for computation,
- Activation or transformation Function to activate the node.

Many processors are made to run parallel where huge amount of data are processed at once and the cost function can be updated. These nodes are the ones that tend to do the computations and tend to consist the multiply and Accumulate units or ALU units, interconnected with the other nodes just like the perceptron nodes. A very huge amount of data is analysed and we can achieve deep learning through this process. They understand complex problems and can extract hidden pattern or the data by Complex signal processing problems.

Neural networks are even described on the basis of the depth of layer count that is the hidden layer present between input and output layers. Various combinations of the output with the input at the time of forward and backward propagations generate different types of the neural-network designs allow. Its system is made up of hardware and software, programmed to behave like the perceptron nodes of the human brain. Main application of neural network is to solve complex problems related especially to the complex signal processing or pattern recognition/extraction. Examples like

speech-to-text analytical processing, Data Analysis, Weather predictions and facial characteristic recognition are the applications of neural network. [6-11]

It uses large number of parallely arranged processors operating used in parallel tiers. The first tier gets raw input data where in the each corresponding successive tiers receive output from the preceding tier in a cascaded manner. The last tier gives the system's output. Neural networks are adaptive, where they modify/update themselves by training and subsequent runs. At the time of the training, the inputs are made to be associated with the weighing values called as weights whose values are not fixed and change every time at the time of the forward and the backward propagation. [6-11]

1.8 ARTIFICIAL NEURAL NETWORK (ANN)

Artificial Neural Networks with their self learning mechanisms predict the outcomes without the help of the programmer and it supports for the automation of the tasks. As the ANN are still not yet developed to what is expected to it to be in the future, they have sometimes, failed some tasks. They can be at times complicated and confusing. They tend to operate on the feature extraction of pattern hidden, where they are used to extract the features and use it for the pattern prediction. There is no use of the traditional programming but involves massive parallel networks must be created and they are to be trained for problem specific.

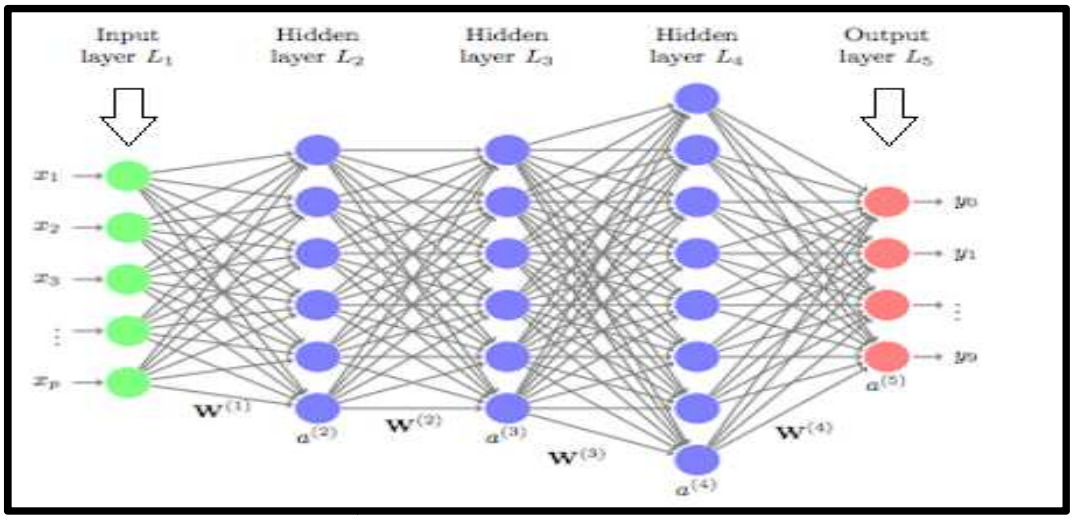


Figure 1.19: Basic ANN-1.

Just like the human body being made up of cells named as perceptron nodes which carry “messages” through an electrochemical process at the synapse, the nodes in ANN are equivalent to perceptron nodes, where nodes are linked to every other node by the help of the Synaptic Weights (w). ANN is hard and fast linked which can enter the input data and generate the output based on value of weights and activation/transformation functions used. At the time of training, the network learns by itself by updating the synaptic weights at the time of the forward and backward propagation at each corresponding node. Neural community gaining knowledge of is also known as connectionist learning, referencing the connections among the nodes. They can be understood by the early design techniques as to how they operate. For standard solutions to the preferred issues, a customized way to particular conditions is to be used, for example, logging in to an e-trade website, we will see customized product recommendations based on previous purchases and searches if any, wish list etc. [6-11]

ANNs had applications in the domains like which include:

- Classification of facts
- Anomaly detection
- Speech popularity
- Audio era
- Time collection evaluation

Convolution Neural Networks (CNN) also known as Le Nets (named after Yann LeCun), are the artificial neural networks having arbitrary connections between layers taking inputs as the images or the video frames. A sure symmetry is maintained such that the perceptron nodes can identical copies with different wide variety of synaptic weights. CNNs operate with pixels where every individual pixel is compared to its surrounding pixels. Conventional computer systems used the algorithmic approach where the computer sequentially follows the commands in solving the problem so, the problem solving capability is restricted or hindered but this disadvantage can sometimes be beneficial as they might do the tasks as per the procedure prescribed in the algorithm and generate better results. Convolutional Neural Network (CNNs or ConvNets), is a special sort of feed-ahead ANN

network which is used usually for the sake of the image processing applications. Convolutional Neural Networks comprise of perceptron nodes having modifiable weights and biases. Instead of take an image's one pixel at a time, ConvNets organization takes numerous pixels together at once so that it will apprehend the temporal pattern and generate the outputs accordingly. ConvNets can observe the organization of pixels forming a line or curve named as the gradient of descent. It can form c=various shapes based on the input and the cost function being evaluated. CNN's require huge datasets of training and testing and consume tremendous computational time to train itself and consume quite a good number of the computational resources as well. So, preferable GPU units are used for their faster computational speed than the normal conventional processors.

Due to transfer learning being a very powerful and practical technique, it utilizes a trained model rather a to be in training network, so thereby minimizing the time for training and completing tasks at greater speeds. Transfer learning uses pre-trained models on a new different or newly created dataset such that it adapts to solve the problem.

In order to build our solution, we will follow transfer learning technique and use the trained model to develop the solution for the problem. We used pre trained models namely the Caffe framework and minivggent framework based on opencv. These deep learning frameworks have great expression of speed, modularity, accuracy and efficiency. Caffe model has been a part of the repository used by researchers and designers to share them in a library called as "Model Zoo".

The tasks must be done with utmost care such that they can reduce the time wastage and can increase the throughput. The traditional computers use cognitive approach to solve problems irrespective of the small vague or unambiguous commands being given to it. These commands are translated to the higher degrees of the language applications making it understandable for the humans and also to the machine code for the machine to understand. These machine codes are easily predictable; so the faults can be due the malfunctions in either software program or hardwares. Neural networks and traditional algorithmic computer complement each

other. Neural network systems perform the tasks which are greatly in accordance to the algorithmic techniques which implement the mathematical operations etc. along with fundamentals of the programming or any technology is required.

1.1.7.1 NEURAL NETWORK ARCHITECTURE

Human's brain is composed of the humongous amount of the perceptron nodes (nerve cells) which replaces short electric pulses for the sake of the message transfer at the synapse. Computer algorithms mimic these biological systems for the sake of the decision taking capability are called artificial neural networks. Neural network research is on the rise so as to reach a point where they are more knowledge than the human brain, and expand computers to address every poorly handled problem. For example, conventional computer systems have trouble understanding speech and recognizing people's faces.

This neural network is comprises of 3 layers, namely the input layer, hidden layer, and output layer. Each layer consists of one or extra nodes, each being associated with the values of the weights. The strains observed at the nodes imply the transferable exchanges of the valuable information from among the nodes. The information is transferred from the input to the hidden layers wherein the data is processed or updated which is then transferred to the output layers.

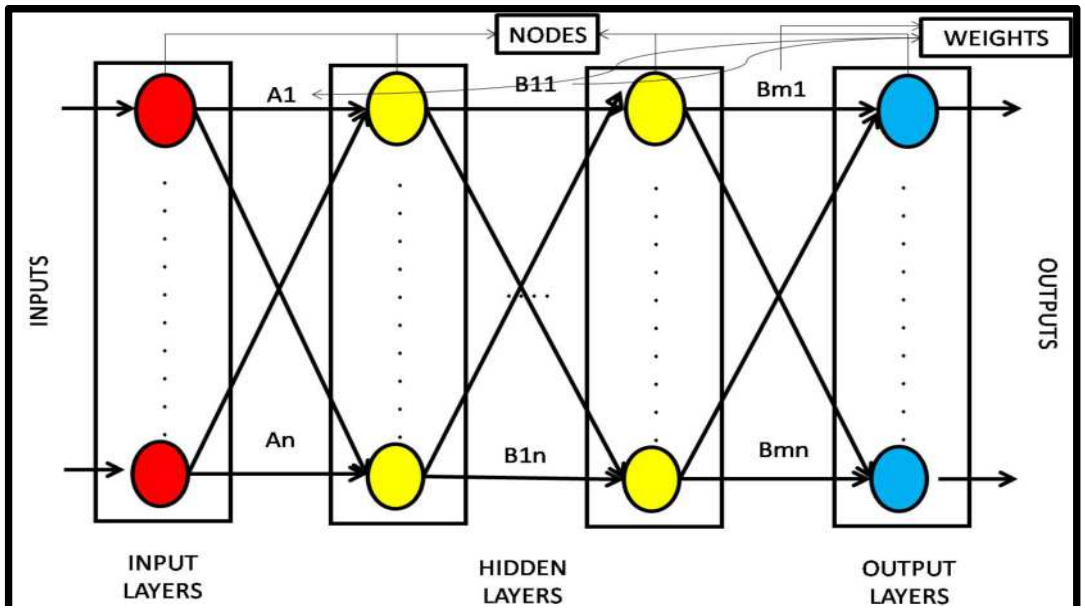


Figure1.20: Neural Network.

The layers like the Input layer and the output layer use buffers for the sake of the optimizing the use of the data elements by storing the elements if required and even enhance the operating efficiency speed. The Hidden layer wherein composed of the perceptron nodes are hidden just how the human brain is hidden in the skull of the human body. So, the system can be considered on the whole as a black box which needs to be programmed or trained on the input dataset for it to generate. On increasing the hidden layers count, the count of the perceptron nodes also increases and therefore more computational resource are consumed like the energy and memory and the throughput for the output to be generated. The layers are having the use of the activation functions for the sake of the input transformation for the output generation and the evaluation of the cost function. These functions are essential for the sake of the cost function to be evaluated and for it to have the least minimum value possible. There are various kinds of the activation functions available which can be chosen as per the type of the input data being used. Cost/Loss function is used to evaluate gradient of descent for the task to be performed. The artificial neural network models are usually specified by three governable attributes:

- Interconnectivity among the perceptron nodes of the hidden layers.
- Activation Functions used for the perceptron nodes at every layer.
- Learning Rules to train the entire network and then evaluating it.

There are five basic perceptron node connection architectures based on the presence of the feedback path connection and the number of the hidden layer, they are:

- ❖ Single-layer feed forward network
- ❖ Multilayer feed forward network
- ❖ Single node with its own feedback
- ❖ Single-layer recurrent network
- ❖ Multilayer recurrent network

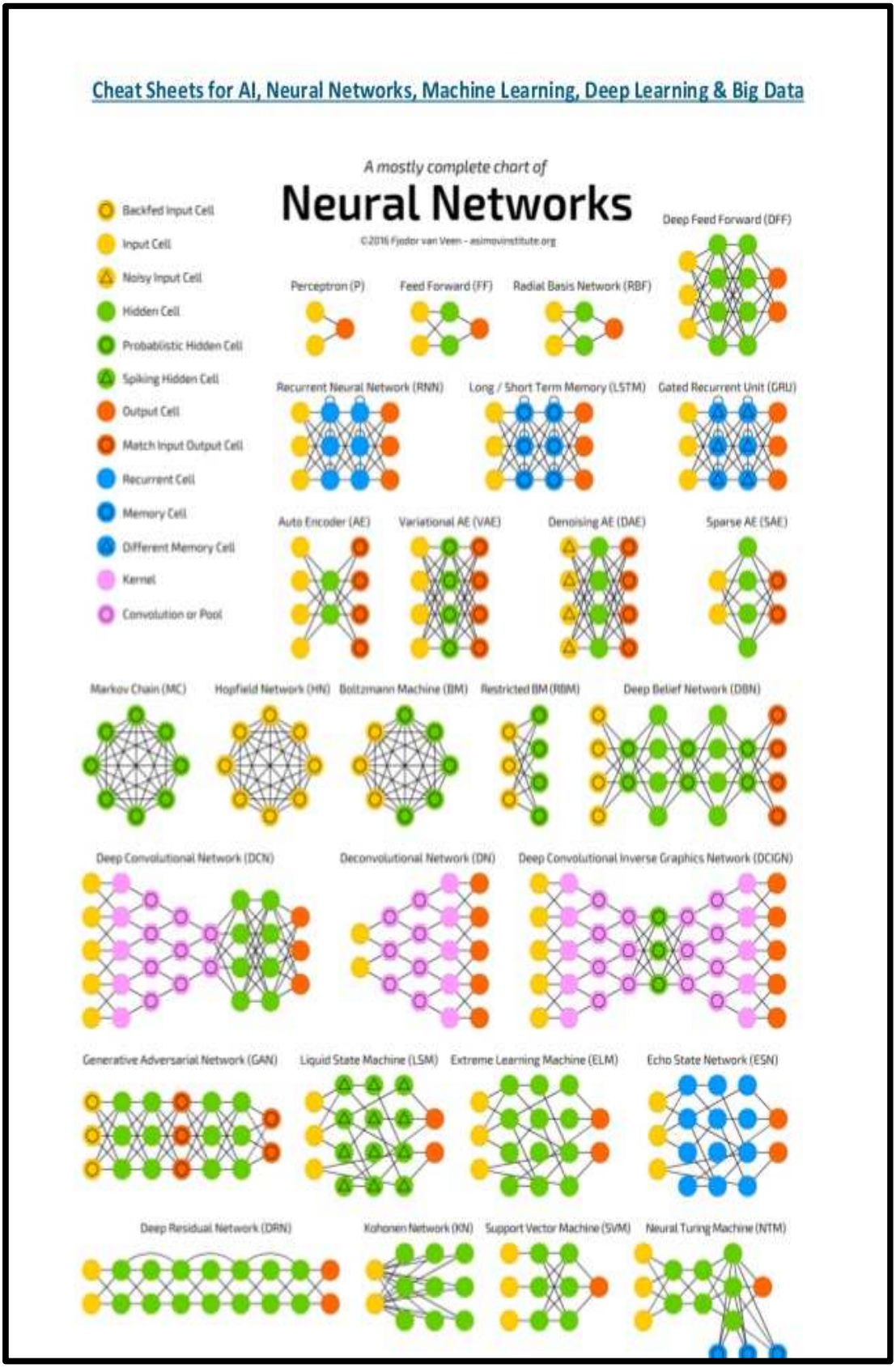


Figure 1.21: Types of the neural networks

❖ **SINGLE-LAYER FEED FORWARD NETWORK**

In this type of the neural network, there are only two layers namely the input and the output layer where the input layer does now not totally rely on the precious inputs and the outputs such that the output layer is shaped as and when the weights are updated on the input nodes. The output layer gets the computed data elements and by the help of the activation functions, the data can be changed for better representation and can be indicated by the help of the output indicators.

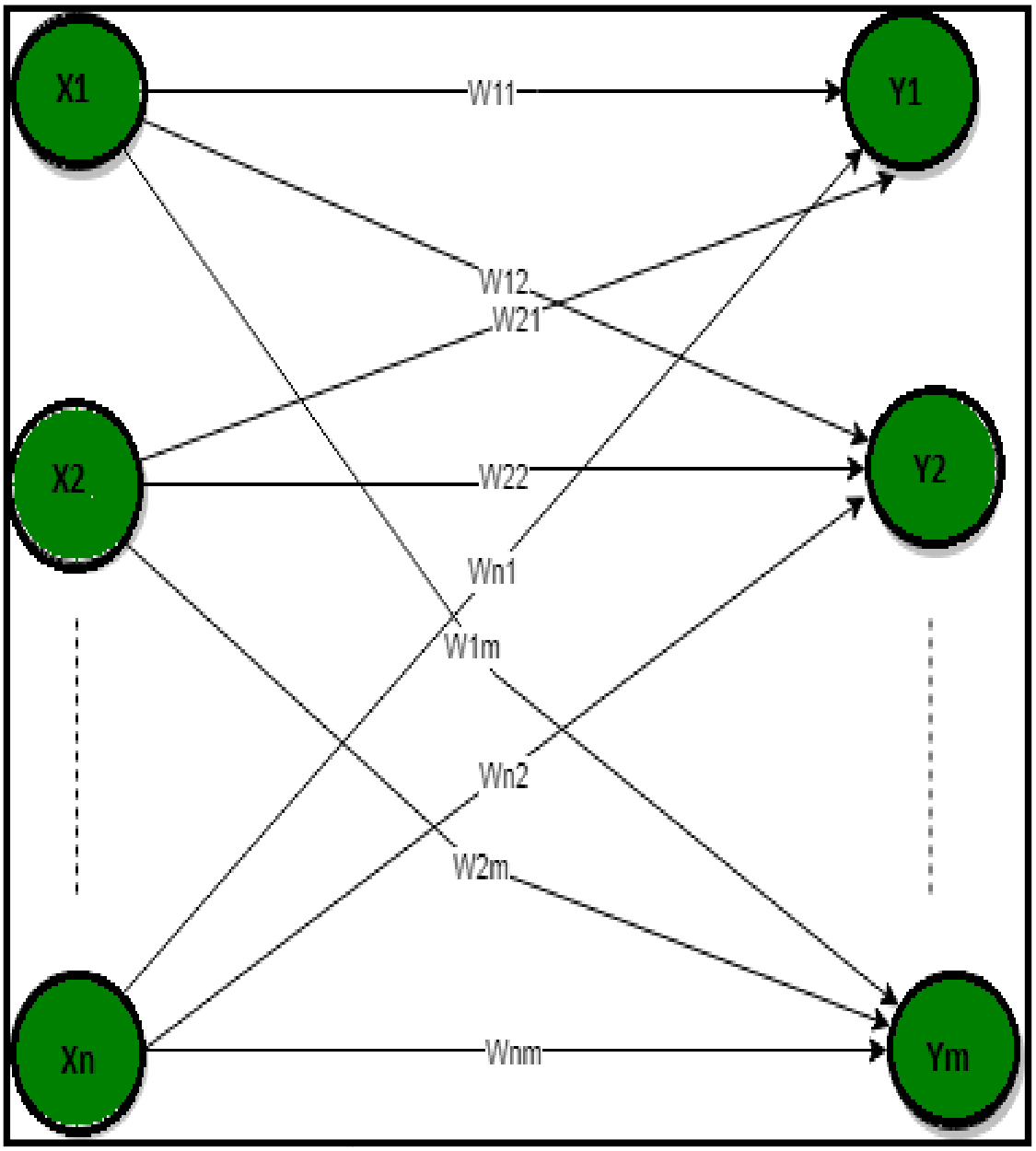


Figure 1.22: Single-Layer Feed Forward Network

❖ **MULTILAYER FEED FORWARD NETWORK**

It is a feed forward neural network having many dense layers which are interior to network, inner of the input layer and with no direct contact to the output layer. The additional hidden layers allow the model being to be computationally stronger in comparison to that of the single feed-forward network. There are few connectivity, in which outputs of the version are fed not back into the input end as it has no feedback paths.

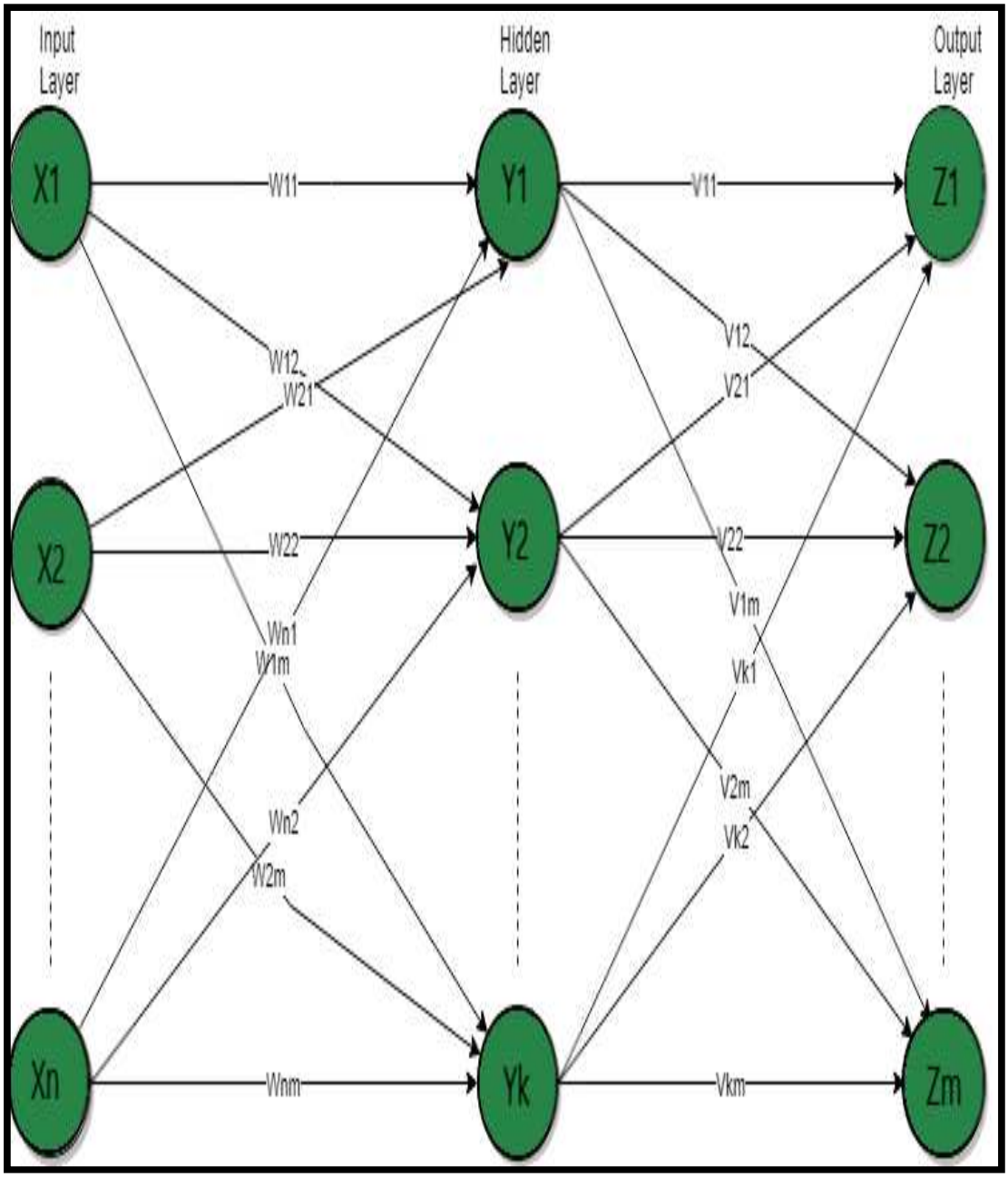


Figure 1.23: Multilayer Feed Forward network.

❖ **SINGLE NODE WITH ITS OWN FEEDBACK**

The outputs are directed again to the input layer as the inputs to the same or preceding layers and it results in feedback networks. Recurrent neural networks (RNN) are feedback networks with a closed loop and are more efficient. The weights values are updated by the help of the feedback path and the cost function can be decreased to a large extent and thereby increasing the accuracy of the model to a large extent.

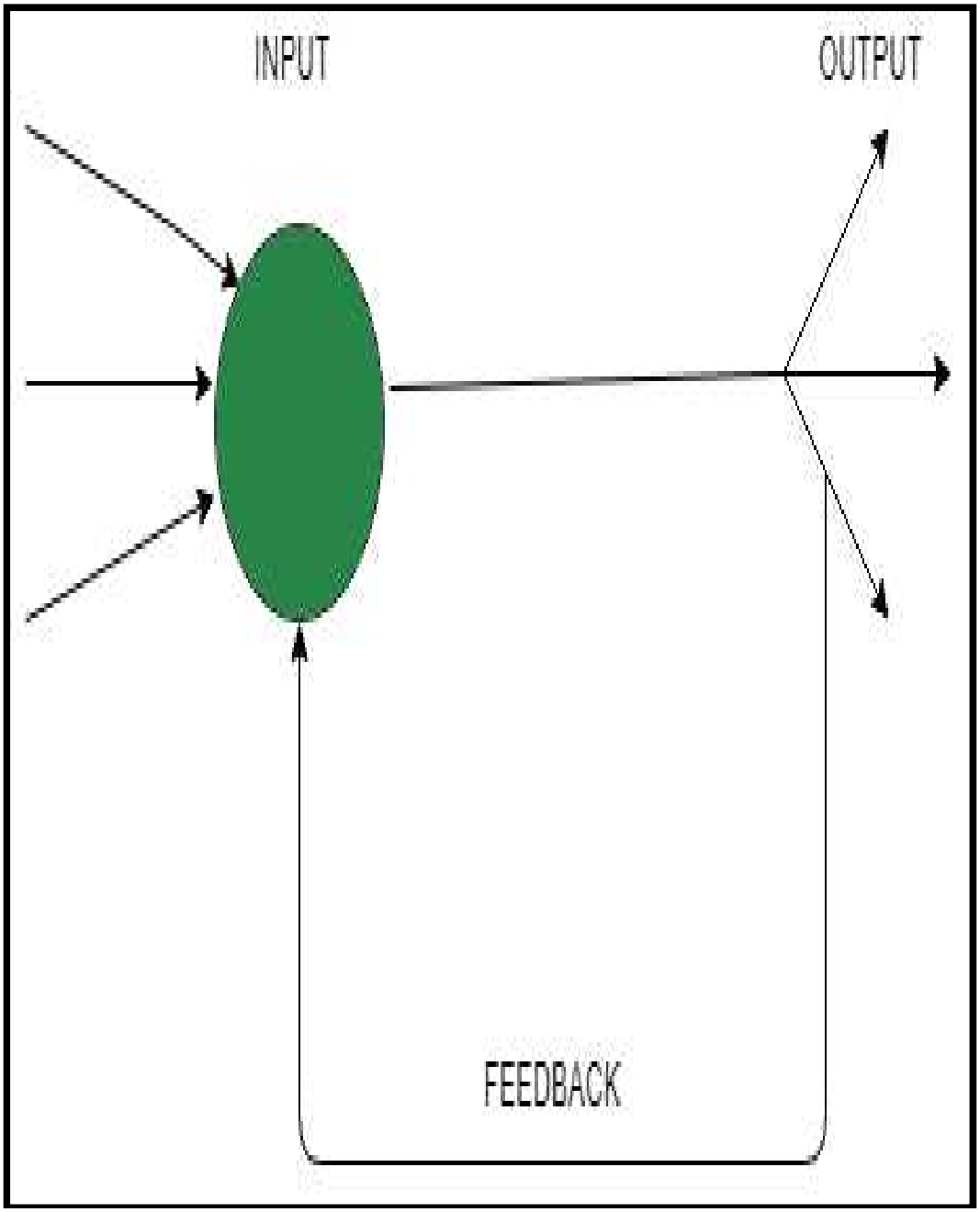


Figure 1.24: Single Node with own Feedback

❖ **SINGLE-LAYER RECURRENT NETWORK**

It single layer network with feedback connection. The processing of the data is done with the updation of the value of the weights, and then they are fed back to the input layer such that this methods reduces the cost function to a large extent due to repetition in the feedback being observed. Recurrent neural network is a class of synthetic neural network wherein the connectivity among nodes form a directed graph in a sequence. It shows dynamic temporal conduct in a time series. RNN's use their internal network in order to sequence the inputs.

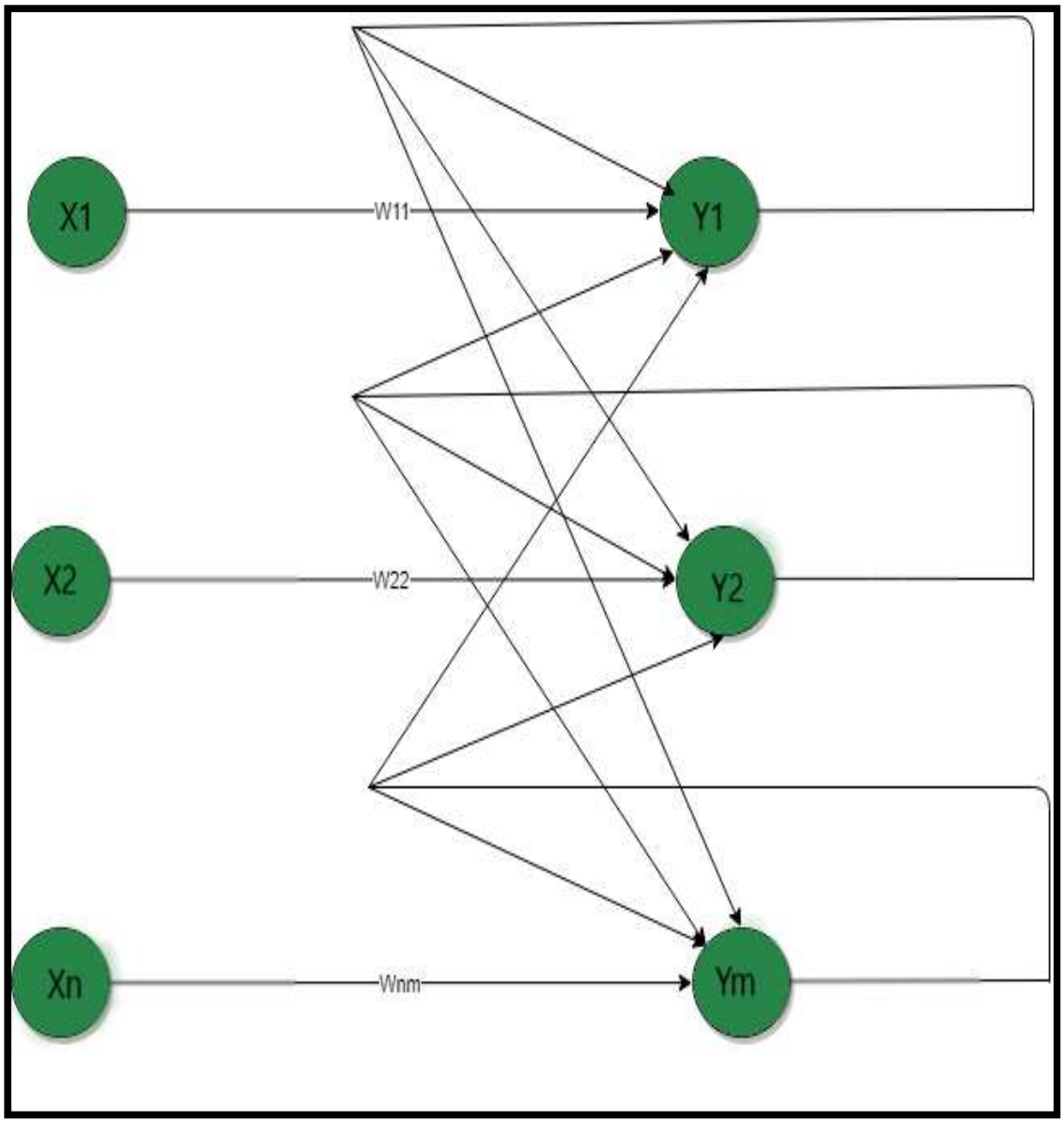


Figure 1.25: Single layer recurrent network.

❖ **MULTI LAYER RECURRENT NETWORK**

In this network, processing element outputs are directed to the input and the preceding layers resulting to a multilayer recurrent network. The feedback is given to every element of the sequence just as in the single layer recurrent network wherein the output is dependent on the previous computation's inputs and the outputs. Inputs aren't essential at every time step. The important aspect of the Recurrent Neural Network (RNN) is its many layer hidden network, which captures elements records approximately at every time series when required.

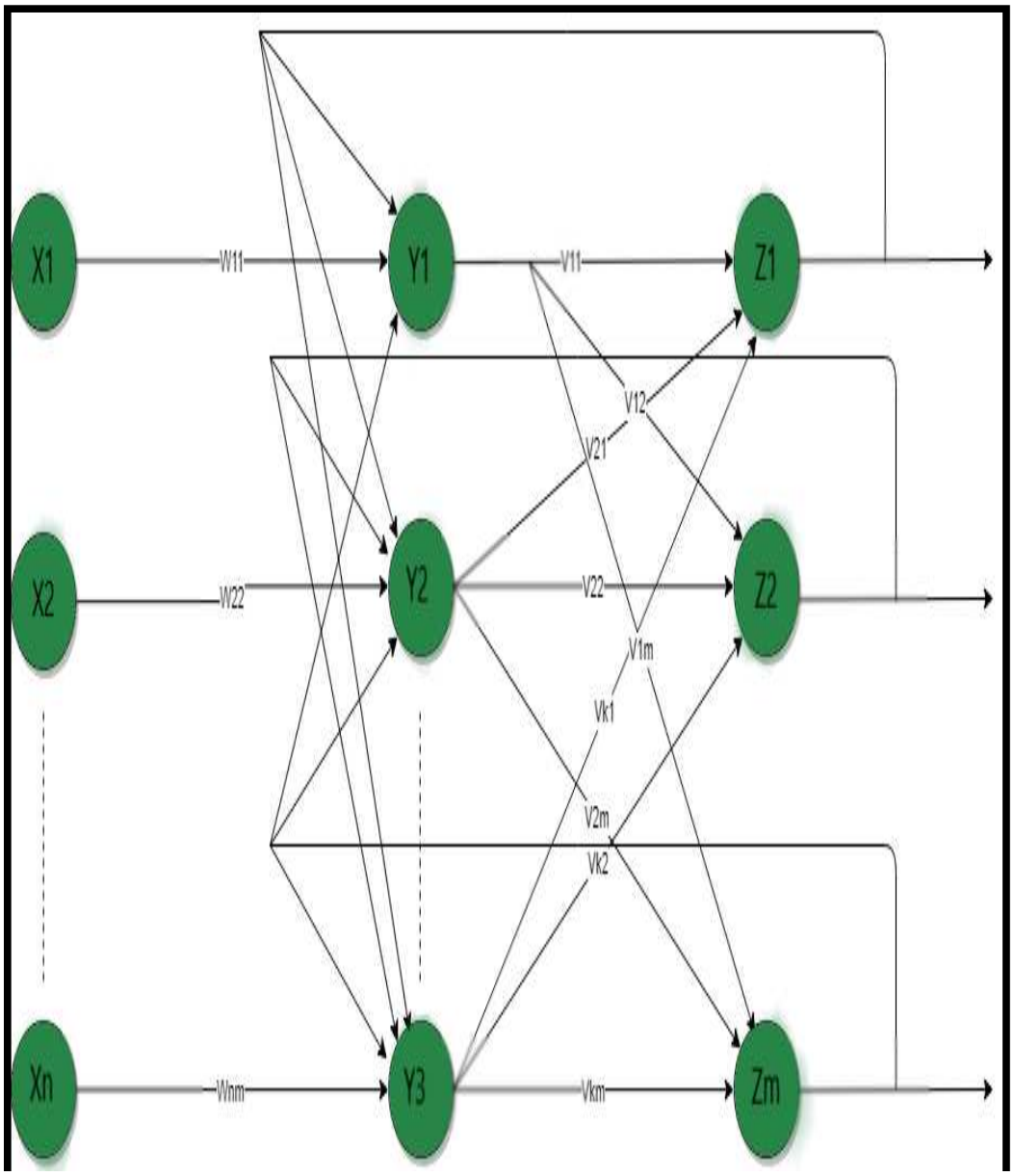


Figure 1.26: Multi layer recurrent network.

1.1.7.3 ACTIVATION FUNCTION

The Activation function is essential for an ANN for learning about the data and does the processing of the data to be done in an efficient manner. Their main role is to convert input to output signal at a node. This output signal is used as input to the following hidden layers or input layer. Activation function makes the decision whether the perceptron node has to be activated or not, such that it can be achieved by the help of synaptic weights at each node. So, a particular perceptron node can be selected to process the input. It introduces non-linearity into the perceptron node output for the given inputs.

The activation function is a polynomial on nth degree where if there is no activation function at the node, it is taken as the linear activation function but they are not very powerful as compared to that of the non linear activation functions due to the linearity of the output achieved for the given inputs. The activation function is quite essential for training neural network such that the network can be used at some good value of the accuracy. It helps to learn the complex records like pixels, videos, audio, speech, etc. Non-Linear activation functions have a response with a curvature.

1.1.7.3.1 TYPES OF ACTIVATION FUNCTIONS

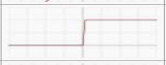
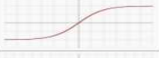
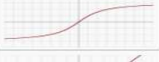
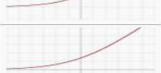
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Figure 1.27: Activation Functions in Neural Networks.

❖ **THRESHOLD ACTIVATION FUNCTION-(BINARY STEP FUNCTION)**

Threshold-based activation function is a binary step function in which the input value is either above or below the value of threshold, the perceptron node activation depends on the value of the inputs, when it is above threshold the signal is buffered to the next layers or stages as the node is considered to be in a activated state. The main difficulty with this function is at the time of the binary classifier creation and at the time of the connection of the multiple perceptron nodes.

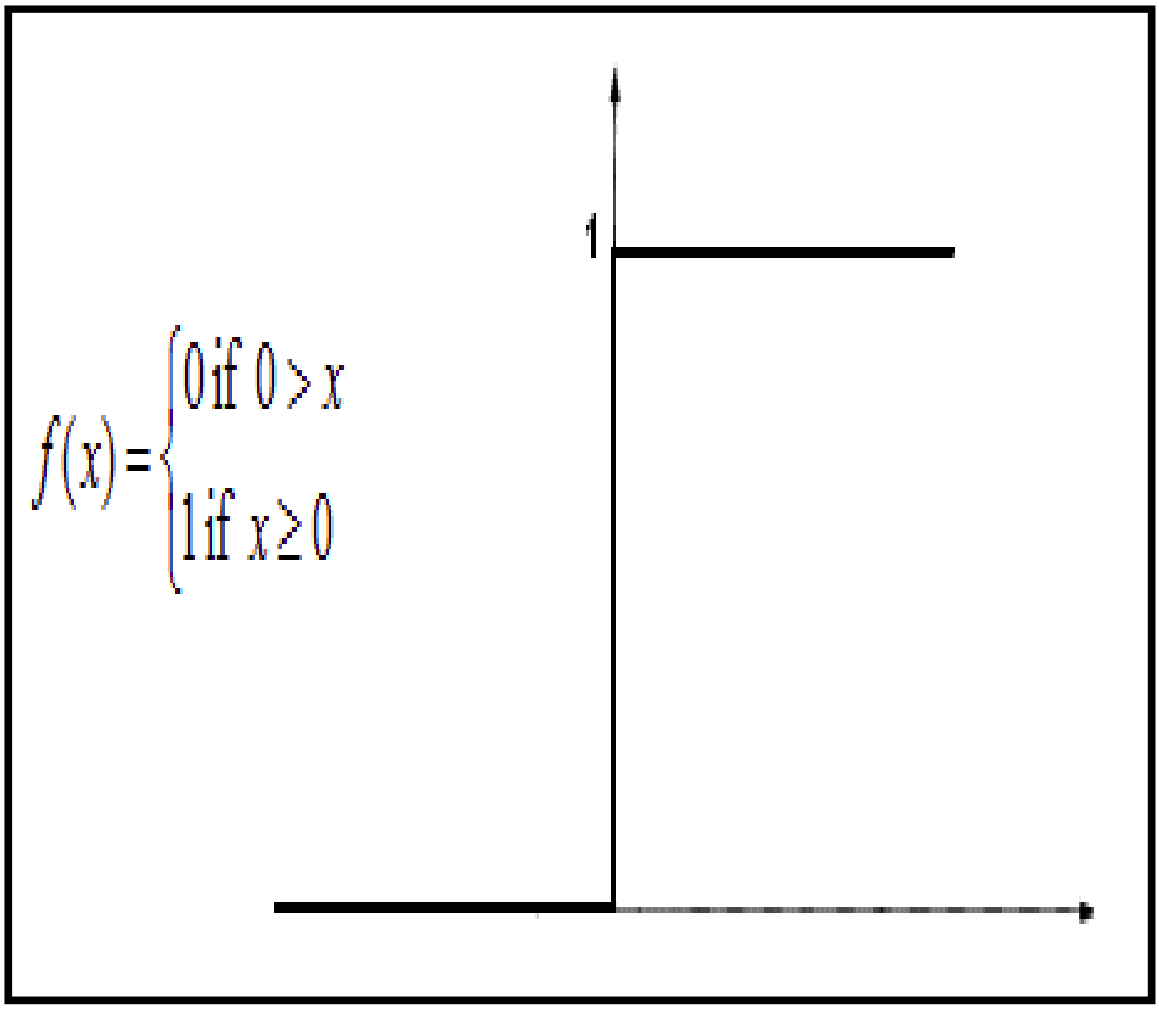


Figure 1.28: A Binary step function

Node is "ACTIVATED" if **-------**

if X> threshold_value

activation_output=1

else if X>threshold_value

activation_output=0.

❖ **SIGMOID ACTIVATION FUNCTION-(LOGISTIC FUNCTION)**

A Sigmoid activation function is a mathematical function having its characteristics shaped in the form of the "S"-shaped curve ranging between 0 and 1. Therefore, it is used for the models in which the probability is to be predicted as an output for the task. This function is differentiable and the slope of the curve can be found using two points. The drawback it is that, the neural network gets stuck at the time of the training, especially when there is a strong negative input given at the input end.

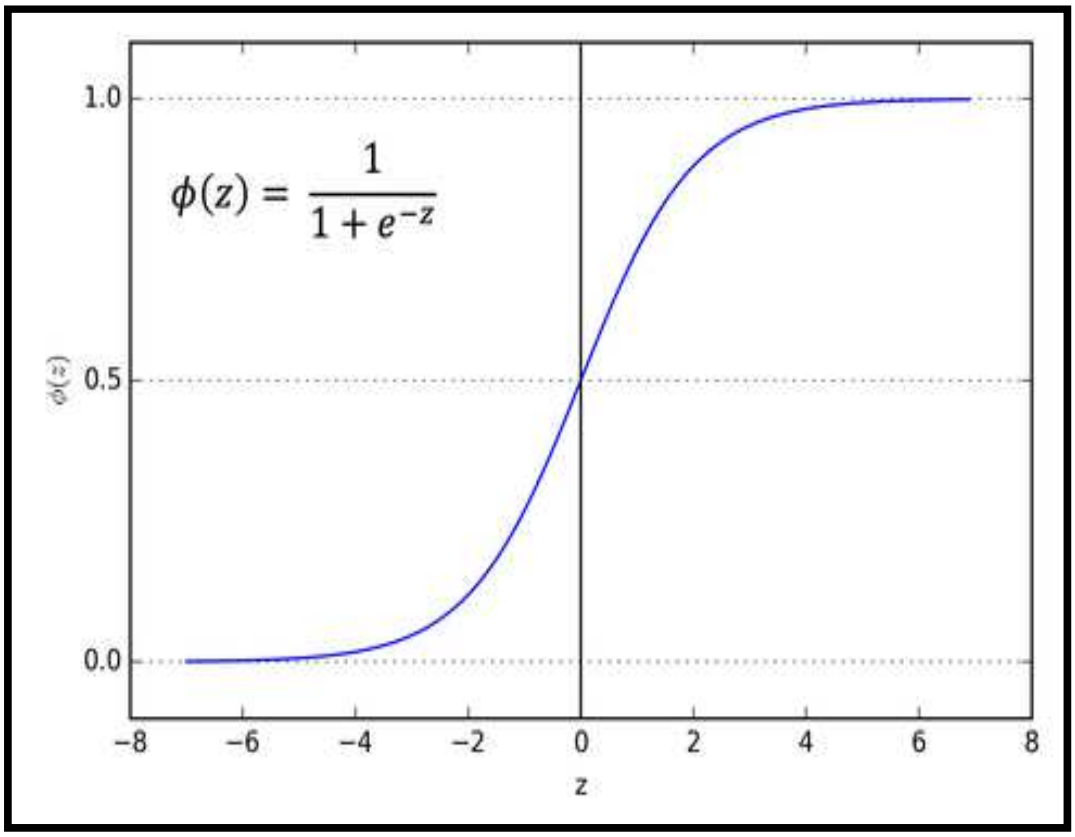


Figure 1.29: Sigmoid Activation Function.

❖ **HYPERBOLIC TANGENT FUNCTION—(TANH)**

It is similar to that of the sigmoid function but it performs better than it. It is non linear o=in nature and can the layers can be stacked. The values of the function ranges in between the intervals of (-1,1). The main advantage is that, at the time of strong negative inputs, the values are mapped to corresponding negative output, for zero-valued inputs to near-zero outputs and positive outputs for non zero positive inputs. So, the chance of getting stuck is less especially at the time of the training.

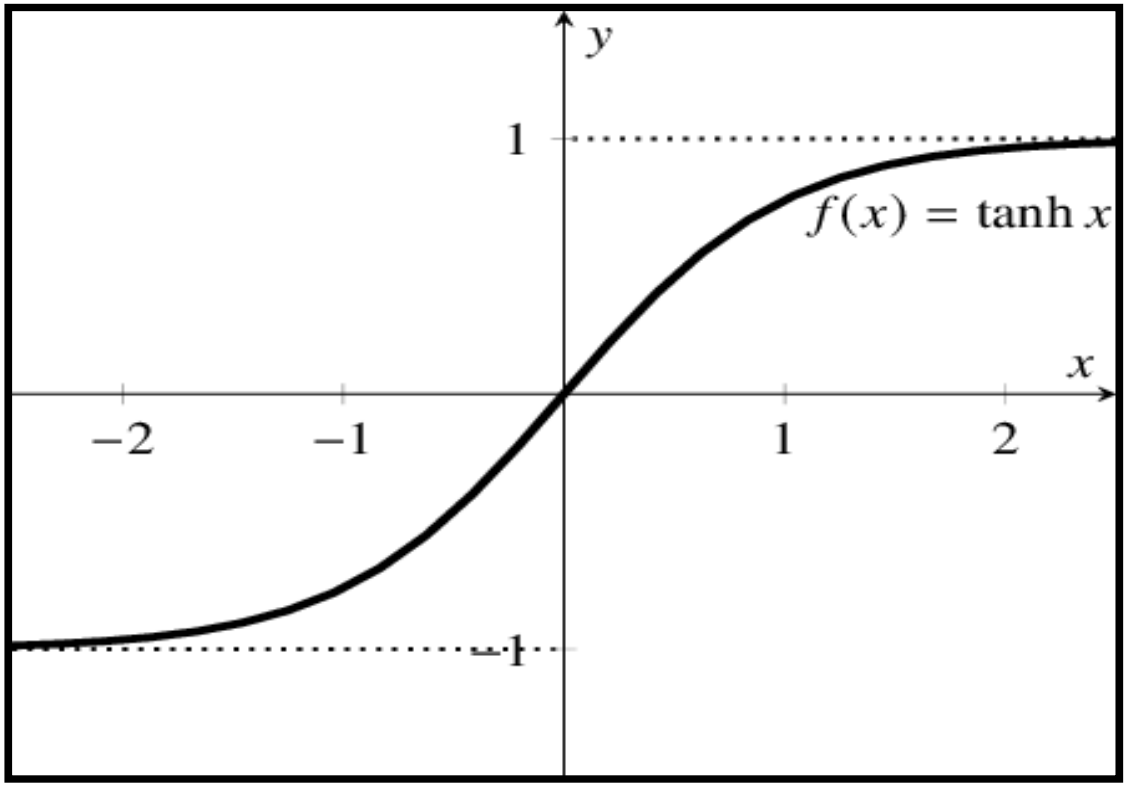


Figure 1.30: Hyperbolic tangent function

❖ **RECTIFIED LINEAR UNITS-(RELU)**

ReLu is widely preferred activation function, especially for CNN and ANN, as the function has a scope of range of values varying from zero to infinity [0, ∞). The output is positive or zero, if the input is positive. Relu is a non-linear function and if used in combination with other functions, then the resultant function is also non-linear hence it is a good approximator, irrespective of combination of functions. ReLu efficiency is 6 times the efficiency of the hyperbolic tangent function. It is applicable to the hidden dense and to the input layers but for output layer, softmax function is used for the classification or regression problems, as it is a linear function. The main limitation of this function is that, the nodes can die for the input being given to it such that the weights of the node will be updated unnecessarily and have a drastic impact over the network accuracy at the time of the training and testing, resulting the network to possess dead perceptron nodes. This problem can be reduced by the help of the "**Leaky ReLu" function** whose small slope keeps the weights updation alive. Leaky ReLu

function ranges from -∞ to +∞. It increases the range of the ordinary ReLu function.

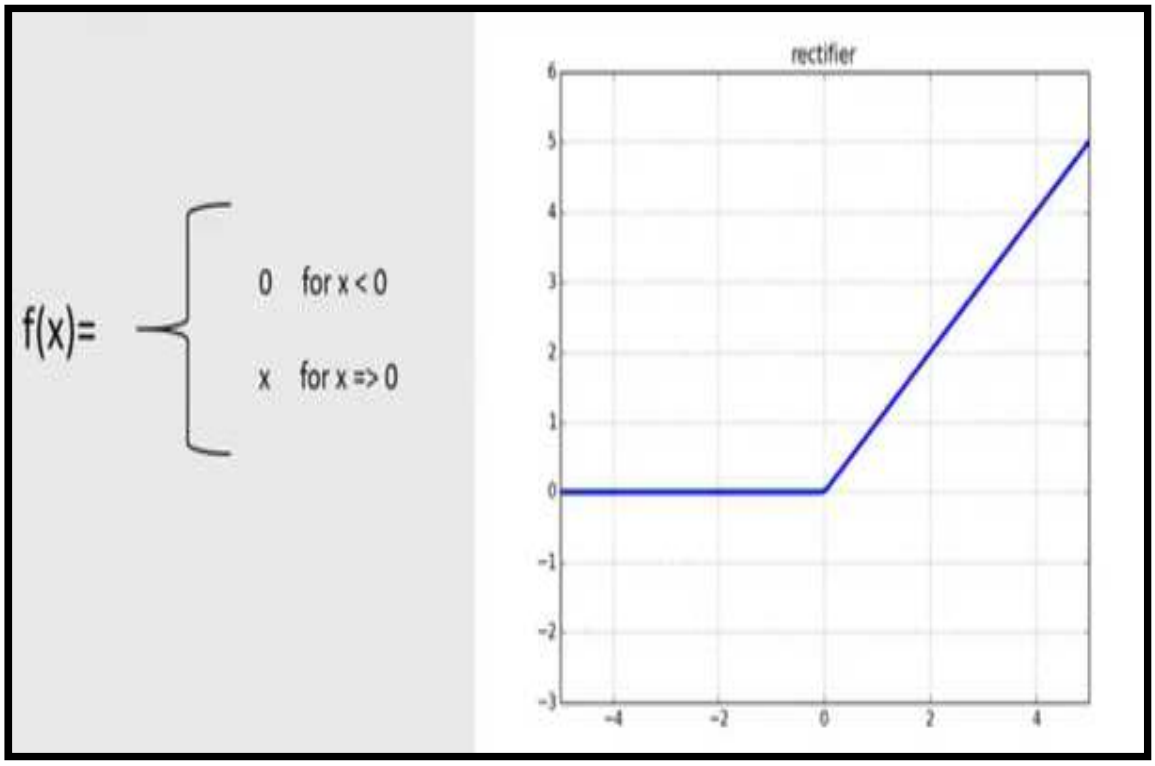


Figure 1.31: ReLu

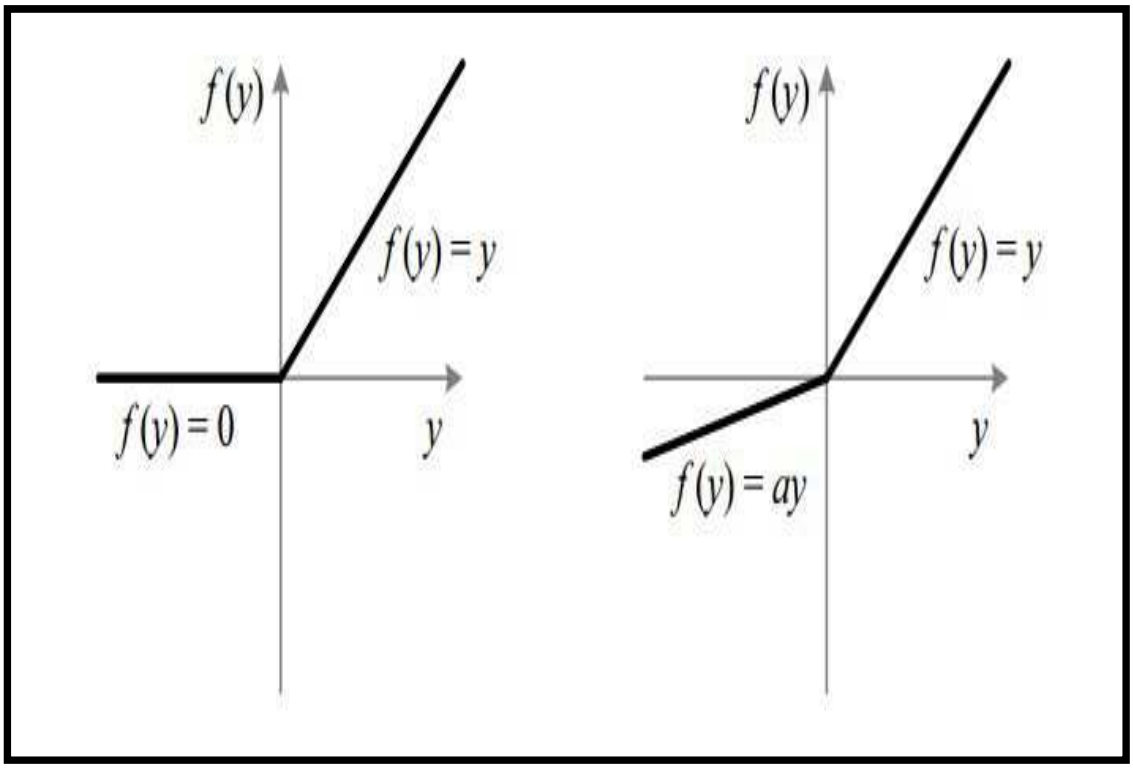


Figure 1.32: ReLu vs. Leaky ReLu

1.2 PROBLEM STATEMENT

Nowadays, by the existing methods, there was no cumulative detection for the age, gender, colour and the pattern so a cumulative method was to be developed especially for the sake of the video analytics. There is huge requirement for the sake of the video analysis and for the surveillance purpose for the latter stages of the procedure. So, initially the code was to be developed for the sake of the cumulative detection.

1.3 MOTIVATION

The reason this topic was chosen for our major project was that, as the Artificial Intelligence and Neural Network processes the inputs and generates the outputs that are very sophisticated, giving support for the automation of tasks and resulting for good demand in the future for it due to its decision taking capability. By the association of the image processing and the neural network, the outcomes can be generated as good as that of the human brain perception. As the AI/DL is a must requirement for the future applications in order to support the automation and generation of the outcomes by itself without the intervention of the humans. So, the future is tend to depend solely on these upcoming technologies such that life of the human beings can be made much more easier.

It finds application in the sectors like the medical sectors, defence or security purposes most likely to give the details of the individual's with respect to the age, gender, color and pattern etc. It will find a major role in the detection of the above said aspects which records them for the sake of the further analysis.

1.4 OBJECTIVES OF THE PROJECT

- To determine age and gender of an individual.
- To determine colour of the upper part of the dress the individual wears.
- To determine the type of clothing type one wears like shirt or dress etc.

1.5 EXISTING METHODS

The existing methods generally depict using the image and that too individually. So in order to determine the age and gender, the recognition was done by the help of the wrinkles or with the help of the facial characteristics respectively whereas the colours are determined using the r, g, b values and the pattern depiction was difficult. The pattern determination was done by the help of the images only. All these methods were not used for trend analysis and for surveillance purposes. The use of the camera is not done and the data which was collected if any was wasted. [6-11, 12-92]

1.6 PROPOSED METHOD

The data is collected from the real time via a live stream from the camera that captures the videos. The data is then given to the models namely the caffe, minivggnet model to determine the age, gender and the pattern of the dress type that one wears, and the model to determine the color. The models are used with high values of the accuracy and used for the sake of the prediction. The model is trained by the help of the fashionmnist dataset for the sake of the pattern and the UTKFace dataset can be used for the sake of the age and gender determination.

CHAPTER-2

REVIEW OF LITERATURE

2.0 INTRODUCTION

LeCun first suggested Convolutionary Neural Networks (CNNs) for image processing, which features two main characteristics: spatially mutual weights and spatial pooling. CNN models have demonstrated their performance in different computer vision applications, where input data is normally 2D data. CNN was also implemented to tackle sequential data including the analysis of the natural language and the understanding of expression. In CNN, the convolutionary layers convert multiple local filters with raw input data and generate local translation invariant features, and the subsequent pooling layers extract features with a fixed duration over sliding windows of the raw input data with multiple rules including average, max so on.[107]

2.1 REVIEW OF LITERATURE ON CNN

Artificial Neural Networks (ANNs) are the computational processing elements which are inspired from the perceptron nodes for the sake of the operation point of view. ANNs are having a wide variety of interconnected computating nodes/perceptron nodes, which work collectively just like in the brain in order to take a decision for the sake of the output generation. Compared to the modern day deep CNN, the restriction of the computational resources and the throughput time along with the algorithmic challenges play a major role for the sake of the training the efficient network for the sake of the output generation. The output can be generated by the help of the convolutions being done on the input signal and the transformation function/activation functions.

$$\mathbf{y = \sum_{i=0}^{N} w_i * x_i * Transform\_Function\_At\_Node_i} \quad \text{---Equation: (2.1)}$$

Through its much capability laid especially in the deeper CNN architectures, these have become a standard for the sake of the output

generation but with the tremendous increase in computational power requirement and the quantity of training input data required irrespective of the complexity. There are available packages like the keras and the tensorflow of the python language, making it possible to code them with an ease due ti the use of the built in libraries. Some applications of the Deep CNN are human face recognition, speech processing and motion detection classification etc. Neural networks are the subsets of the artificial intelligence (AI). Predominant neural networks types are used for the sake of the multidimensional signal processing and image processing etc. are deep convolutional neural networks (CNNs). CNNs are currently wide range of uses, which mostly constitute for the sake of feature extraction and analyses and generalization of the outputs especially at the time of the training. The discovered hidden features are used for the tasks to be done like classifying or clustering etc. A Convolutional Neural Network (CNN) is constituted to have many additional convolutional dense layers including the subsampling step. The CNN architecture is taken to be a 2D shape. It is achieved by the help of the neighbourhood node connectivity and their associated weights accompanied by some shape for pooling leading to the translation invariant capabilities. CNNs are easily trainable and have lesser parameterized constructs than that of the fully connected networks with equal amount of hidden layers. The cost function is to be evaluated such that the accuracy of the model is enhanced to a large extent.

Convolutional neural network (CNN) is the most commonly used form in machine learning and has emerged as the best choice in areas such as image recognition, object detection, semantic segmentation, and speech processing. CNN-based algorithms achieve excellent efficiency with the overhead of Huge computing and memory resources, particularly when the network depth is increased. The computational capacity of training large CNN models under reasonable timing constraints has become viable through general-purpose graphics processing units (GPUs) in the last few years. However, CNNs' forward inference method often entails high computational complexity, and requires good performance in real time. For

example, the VGG-16 model, which includes 138 million parameters, consumes more than 30 billion multiplication and additional operations for a 224 RGB image feed-forward process.[106]

The number of arithmetic operations increases exponentially when approaching applications with larger images. Thus, conventional general-purpose processors can not meet the requirements of real-time applications, and numerous GPU-based CNN accelerators and field-programmable gate array (FPGA)/application-specified integrated circuit (ASIC) have been proposed.[106]

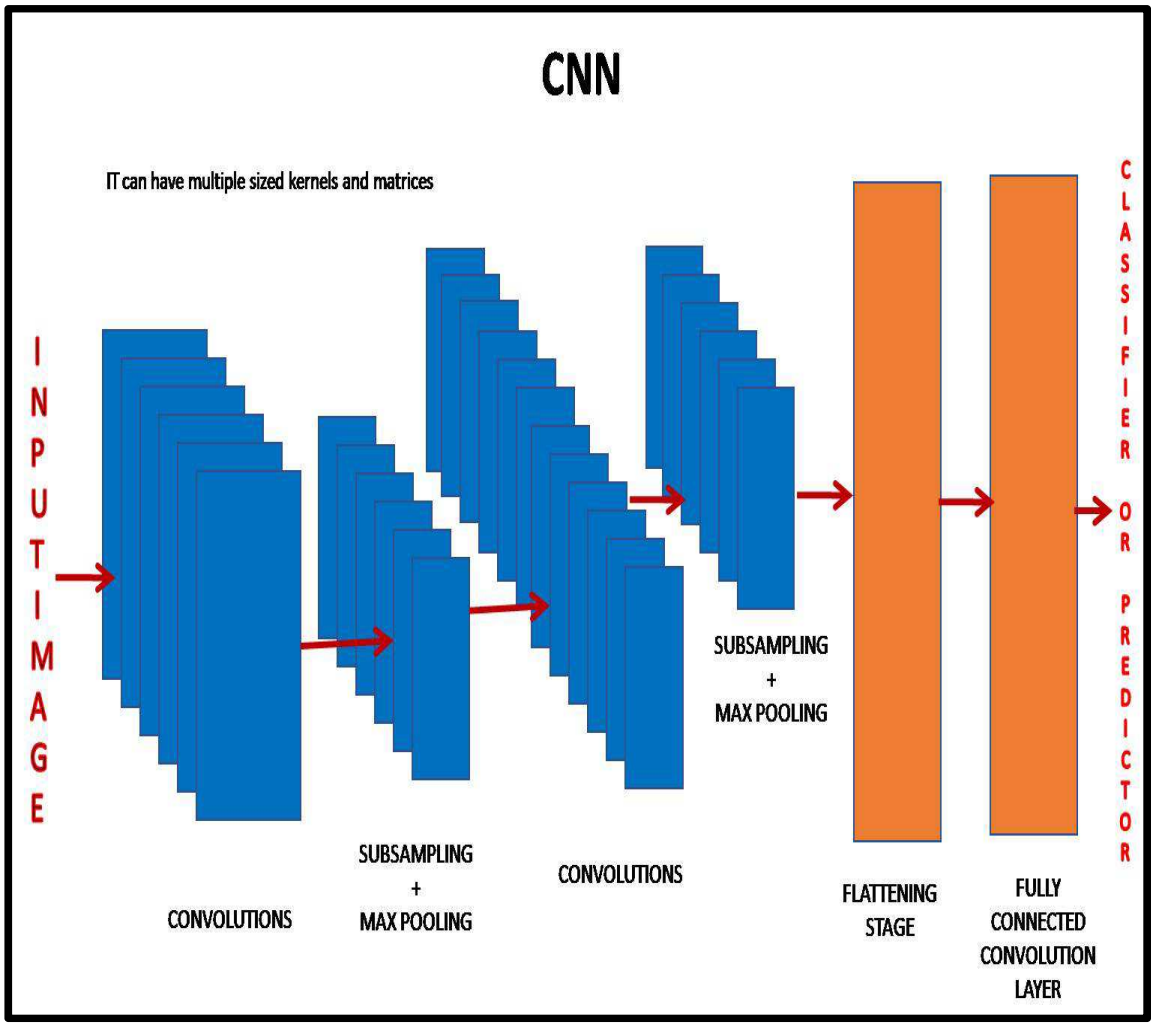


Figure 2.1: CNN-1

Convolution Neural Networks(CNN) has revolutionized the computer vision field with its outstanding vision results. They're non-linear functions designed to form a human eye. Recognizing their significance, they have done work to enhance their results. The system's accuracy improved over time and so did the difficulty behind it's functioning. To boost their performance several complex architectures are being implemented. Present Resnet has hundreds of layers and have 50 times fewer parameters than AlexNet. Such models give less evidence of how they operate internally and bring these remarkable results to bear. One approach to understanding CNN is by leveraging the network's filter activation function maps. Another way is by putting forward the regions the CNN is looking at, in the picture. The common motive behind these approaches is to indicate the regions in the image that correspond to the image recognition objects output of the CNN.[107]

They are analogous to the traditional ANNs as they too have perceptron nodes self-optimising themselves especially at the time of the training. From the input vectors taken from the picture, the final output is either predicted or estimated based on value of synaptic weights. CNN are mainly used for the sake of the pattern or hidden feature extraction from the given input image or the video frame.

All the image related tasks can be performed by the help of the CNN like the encryption of the data. One main drawback of CNN is that it requires huge computational power devices like the GPU units, where the time taken for the training will be reduced to a large extent in comparison with that of the traditional computational complexities which are required to compute the statistics related to the images. Consider a small image within a dimensionality of 28×28, with the use of a single perceptron node, in the initial hidden layer itself, which would tend to give values of about for 784 weights (28×28×1) where 1 is for the MNIST dataset which is normalised to have black and white intensity values whereas in case of the coloured inputs, the number of values tends to increase upto 12, so many hidden

layers are needed and they are to be used to extract every detail in a precise manner.

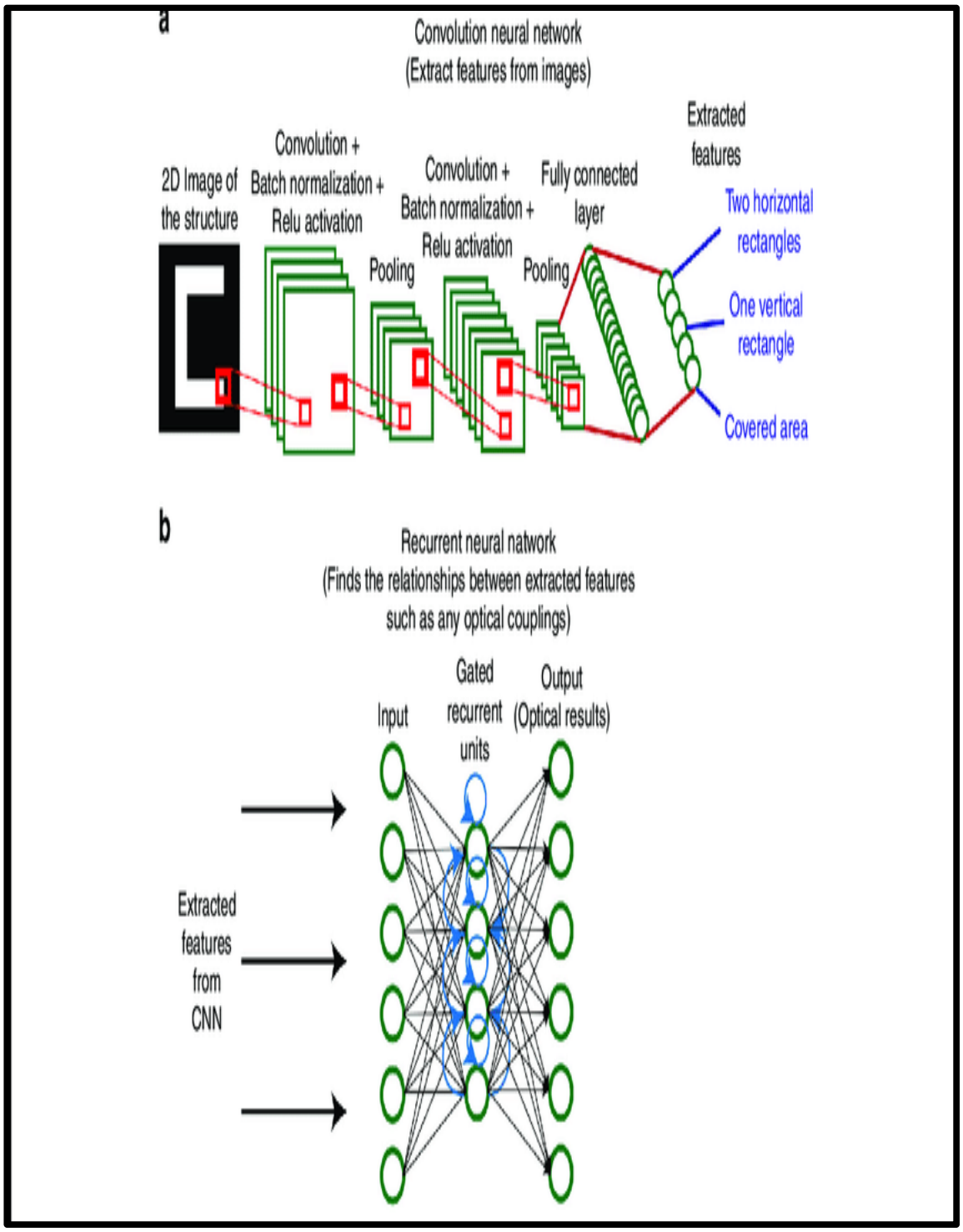


Figure 2.2: CNN-2.

CNNs depend only on the stimuli input values to generate the outputs by extraction of the desired hidden features. The perceptron nodes of the layers are categorized into 3 dimensions namely the:

- the spatial dimensionality of the inputs
- the hidden layer of a depth.
- The output convolutional layer

A particular hidden does not depend on the other layers of the network irrespective of the connection being made. Any CNN network can be divided into 4 regions. CNNs remodel inputs at every stage of the layers by the use of convolutional and down sampling methods to give outputs**.** [6-11]

1. **Input image** is converted into the pixels upon the sampling.

2. **Convolutional layer** determine the computable output at perceptron nodes which is connected with other nodes, inputs and the value of the weights connected to the nodes. Huge quantity of convolutions takes place in this layer in a matrix arithmetic manner such that, based on the selected dimensions of the matrix, those many features can be extracted at greater ease. Each layer will use the activation function named as rectified linear unit (ReLu) and sigmoid as the activation of the output produced by the help of the previous stages.

3. **Pooling layer** to down sample and reduce the spatial dimensionality of the given input, such that it lowers the parameters count within that activation stage, at that time instant. The highest intensity value is chosen from the matrix of set dimension and which is later again used in the computations for better generation of the results.

4. **Fully-connected layers** does perform tasks as observed in a normal ANNs and produce outputs from the activations of the respective nodes and use it for various applications and classification. ReLu activation function can be used for these layers so as to enhance performance tremendously. **[3]**

2.2 Deep convolutional neural network

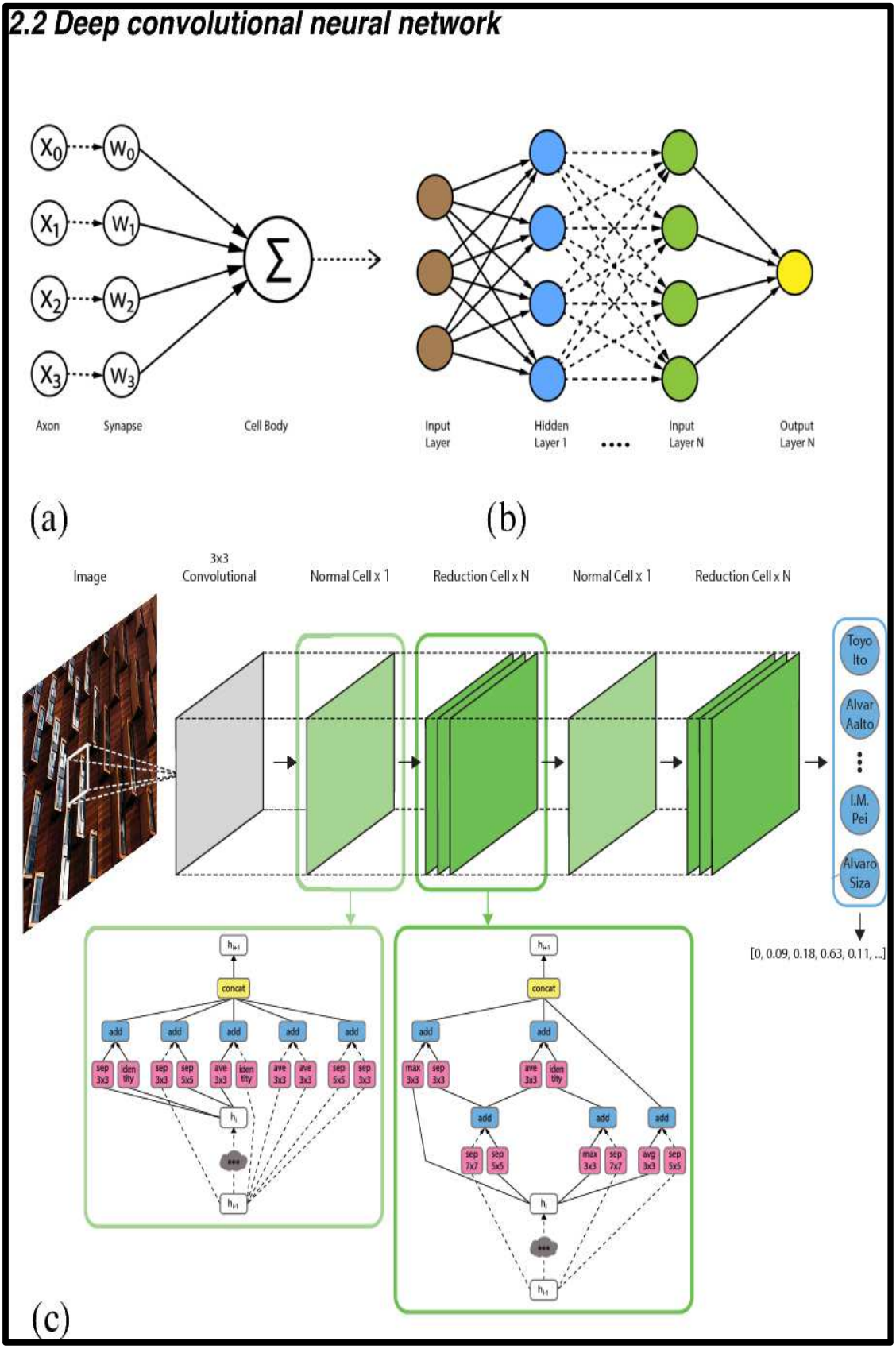


Figure 2.3: CNN-3.

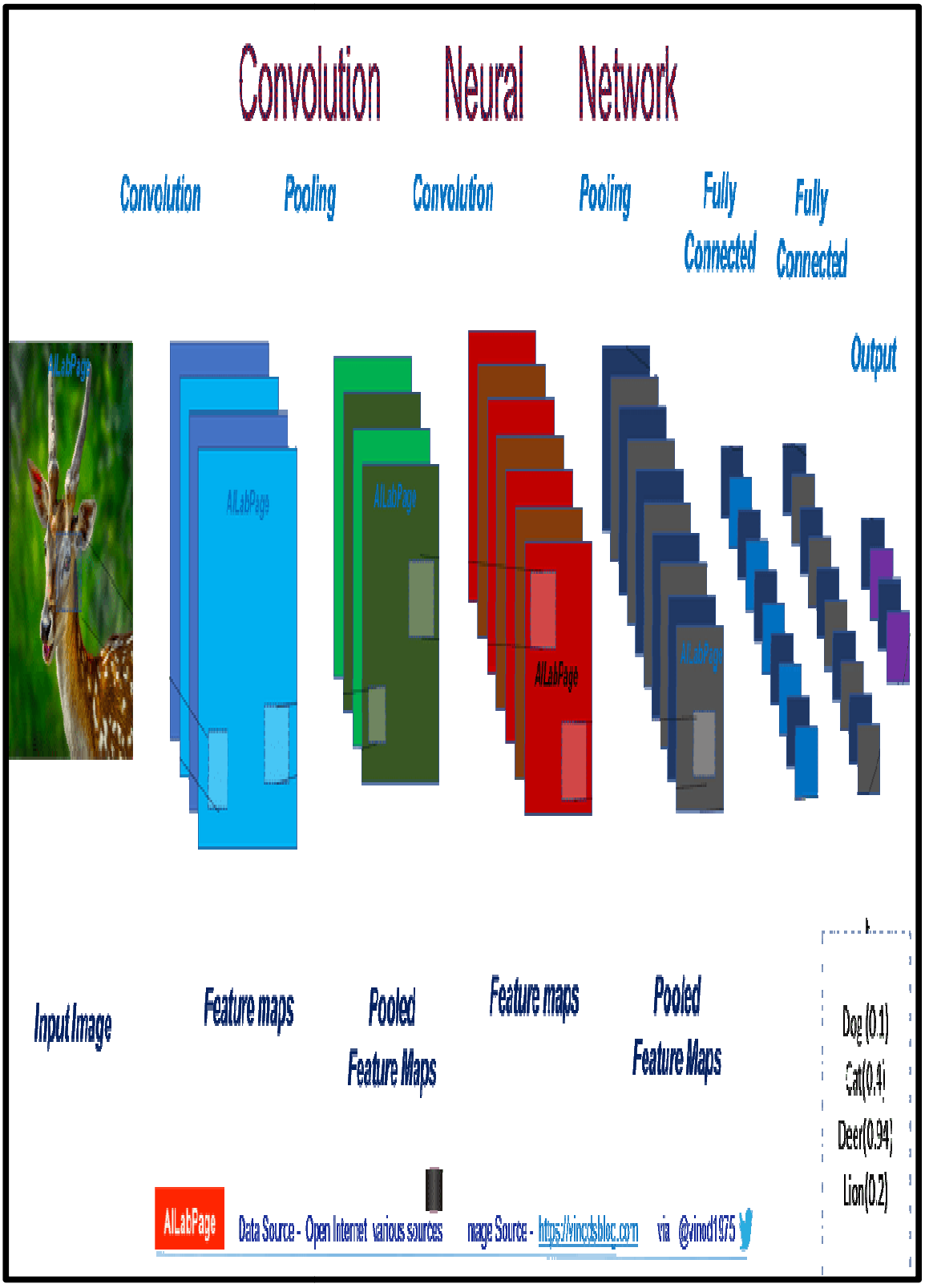


Figure 2.4: CNN Steps.

2.1.1 CONVOLUTIONAL LAYER

It is the most important main layer used in the CNN network as it performs the convolutions on the input data and the transforming functions. The use of the learnable kernels for the sake of the spatial dimensionality reduction is done along the entire range of depth of the input. When inputs are provided to the convolutional layer, the layer then convolves each and every input values and it filters out across the spatial dimensionality and provides a 2D activation map which can be visualised. Scalar product is calculated for every value in kernel with the each input value. Each kernel has its own corresponding activation map and help to shape the output. So to train on these huge input values becomes difficult, we pool the layer outputs in the next step and train the network accordingly later being given to fully connected convolutional layer. [6-11]

The dimensionality of the selected region from the input is called as receptive field for the computing perceptron nodes. The connectivity depends on the input values and its dimensions. For instance, for an input image of size 64, it gives 64*64*3 (coloured RGB image) which has values of weights for the respective field being set 6*6*3 giving 108 values of weights at a perceptron node.

Convolutional layers reduce the complexity and optimise the output and which can be done by 3 means, namely the hyper parameters, the depth of hidden layer, the stride and setting zero-padding to increase the dimensions.

The intensity of output values produced at the end of the convolutional stage by the convolutional layers, depends on the number of the perceptron nodes inside the layer especially to the proximity of the input layer. The connectivity's between the perceptron nodes can be optimized by reducing this hyper parameters substantially thereby reducing the network size and the computational power thereby generating the output in quick manner, but it also reduces the feature recognitions from the inputs.

Outline the stride in which the depth is set across the spatial dimensionality of the input for the receptive area to be regioned. They tend to govern the overlapping aspect of the hidden layers or inputs affecting the activation functions in the receptive region. For example, if the stride is set as 1, it results in a closely overlapped receptive discipline producing extraordinary activation of the perceptron nodes. It the stride is set to a large quantity, it will reduce the frequency of overlapping and produce an output of lower spatial dimensions.

Zero-padding is the procedure to add the zero or any other value of the input. It is a powerful method to manage the dimensionality of the output and the convolutional layers respectively. To calculate the output dimensionality, you can employ the following components as depicted in the equation:

(V1 − Rpt) + 2ZpSt + 1 –---Equation 2.2

Where V1 is the input length (height×width×intensity),

Rpt represents the receptive subject size,

Zp is the amount of zero padding set

St is the stride.

If it is not an integer, then the stride is incorrectly set, indicating that the perceptron nodes are not properly connected across the inputs. [6-11]

The output gradient of descent can be estimated by the help of the dimensions of the input images. If the weights and the dimensions are founds to be equal to that of the activation maps, the data can be reduced in dimensions to a large extent with interconnectivity of the perceptron nodes being the major deterministic parameter.[6-11]

2.1.2 POOLING LAYER

Pooling layers are used to reduce the dimensionality of the convolved input image at the end of the convolutional stage and also the

parameter count and the computational complexity for the network. It operates on every activation map within the inputs scaling the dimensionality with the usage of “MAX” characteristic. In most CNNs, characteristic have kernels being used of dimensionality of 2 × 2 shape, such that it implements it with a stride value of 2, along with the spatial dimensions of the input, scaling the activation map by 25% of its size, maintaining the intensity as to the same value. Usually, stride and filters of pooling layers are set at 2 × 2 dimensions, in order to traverse at the entire spatial dimensionality of the input. Kernel size if set above the value of 3 will typically decrease the overall performance of the network. The next stage can be either the fully connected convolutional layer or to stage having feature extract as per the requirement of the task to be implemented. The pooling perceptron nodes perform operations like the L1/L2-normalisation, and average pooling [6-12].

2.1.3 FULLY-CONNECTED LAYER

As the pooling layers reduce the input image dimensionality at the end of the convolutional stage and also reduce the parameters and computational complexity of the model. It operates over every activation map of the inputs scaling the dimensions using the MAX characteristic feature. The feature extraction stage if not there in the entire model being developed, then the next stage is the fully connected convolutional layers, wherein convolutions are performed on the max pooled data or the feature extracted data having the kernel size set same as just that of the pooling layer or more than it. The data is then generated and given to the output layer. The major drawback it is that it includes many parameters that require complex computations at its nodes especially in the training input dataset. So, the number of nodes and connections are tried to be reduced by the help of the dropout techniques, tremendously reducing the requirement of the computational resources for the same hardware complexity. For example, LeNet and AlexNet were designed to be a deep and wide network, maintaining the computational part to be costant in term sof the complexity.[6-12]

The most beneficial aspect of the CNNs is reducing the parameters count when compared to that of the ANN and the hardware complexity may seem to be of more requirements but it is not actually. The activation maps and the summary functions can be obtained by the help of the CNN for it to do the complex tasks of the feature extractions as and when the input propagates along the hidden layers.

2.1.4 HARDWARE THAT CAN BE USED

Graphical processing unit (GPU) is chip present in the computer which performs high speed mathematical operations, especially for the sake of the images to be processed. Earlier the central processing unit (CPU) did the processing of the image with inferior operating speed when compared to that of the GPU and it had degraded the overall model performance especially at the time of the training. GPU are made to be incorporated with CPU in the circuitry of the motherboard as a pixel card or a server such that the processing power can be increased by a manifold amount. NVIDIA, AMD, Intel and ARM are the major manufacturers of the GPU units. These units have the capability to process the video frames at a very high speed and it is mostly preferred especially in the gaming applications. [6-11]

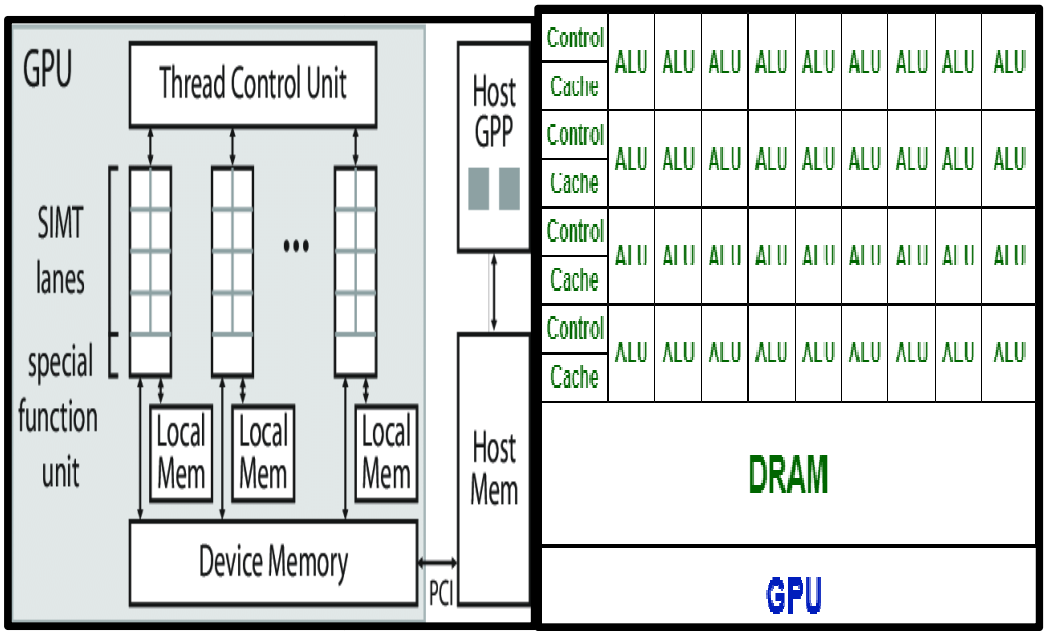


Figure 2.5: GPU Architecture.

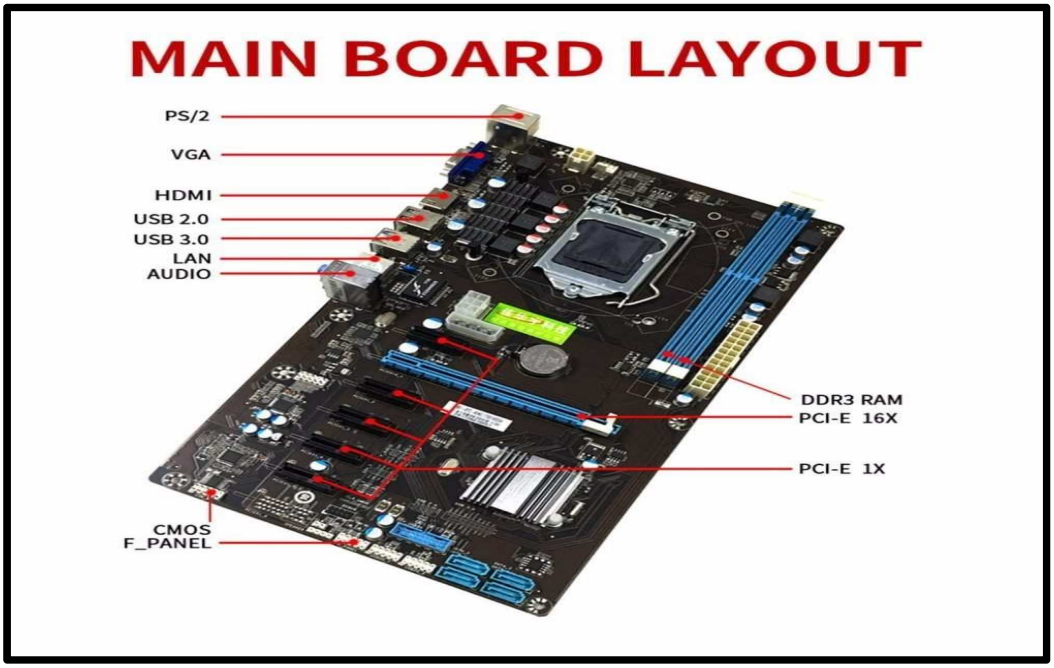


Figure 2.6: GPU NVIDIA Mother Board.

2..4.4.1 GPU VS CPU

GPU is capable of rendering images at a greater speed than the CPU due to its parallel processing architecture, allowing it carrying out many mathematical operations in same amount of time when considering both, optimizing the throughput by a large value. CPU do not possess this feature, despite having multicore processors available to perform these same calculations in parallel way of processing architecture. Even though the CPU additionally has better clock speed, but yet it is inferior to perform the tasks on the image processing applications when compared to that of the GPU. The cost function will not be having minimum value especially at the time of training and hence it may give inaccurate results despite having good values of the accuracy.

GPU is designed essentially for the sake of the processing of the statistics pertaining to the image which are to be estimated by its parallel architecture by performing the same operation at the single instant of time unlike the multicore CPU which supports for the parallel processing, but each core can perform different operations at the same instant of the time.

Most CPU's have 4 or 8 cores usually or ranging up to 32 cores. Each core can process its own task or threads enabling the support for the multitasking or multithreading. Some processors support the multithreading due to which they can be divided into halves in terms of the count of the cores, where each core computes 2 threads at once making it to be beneficial for the video analysis and its transcoding whereas the GPUs can process about 4 to 10 threads at once, keeping in synchronisation with that of the middle one.

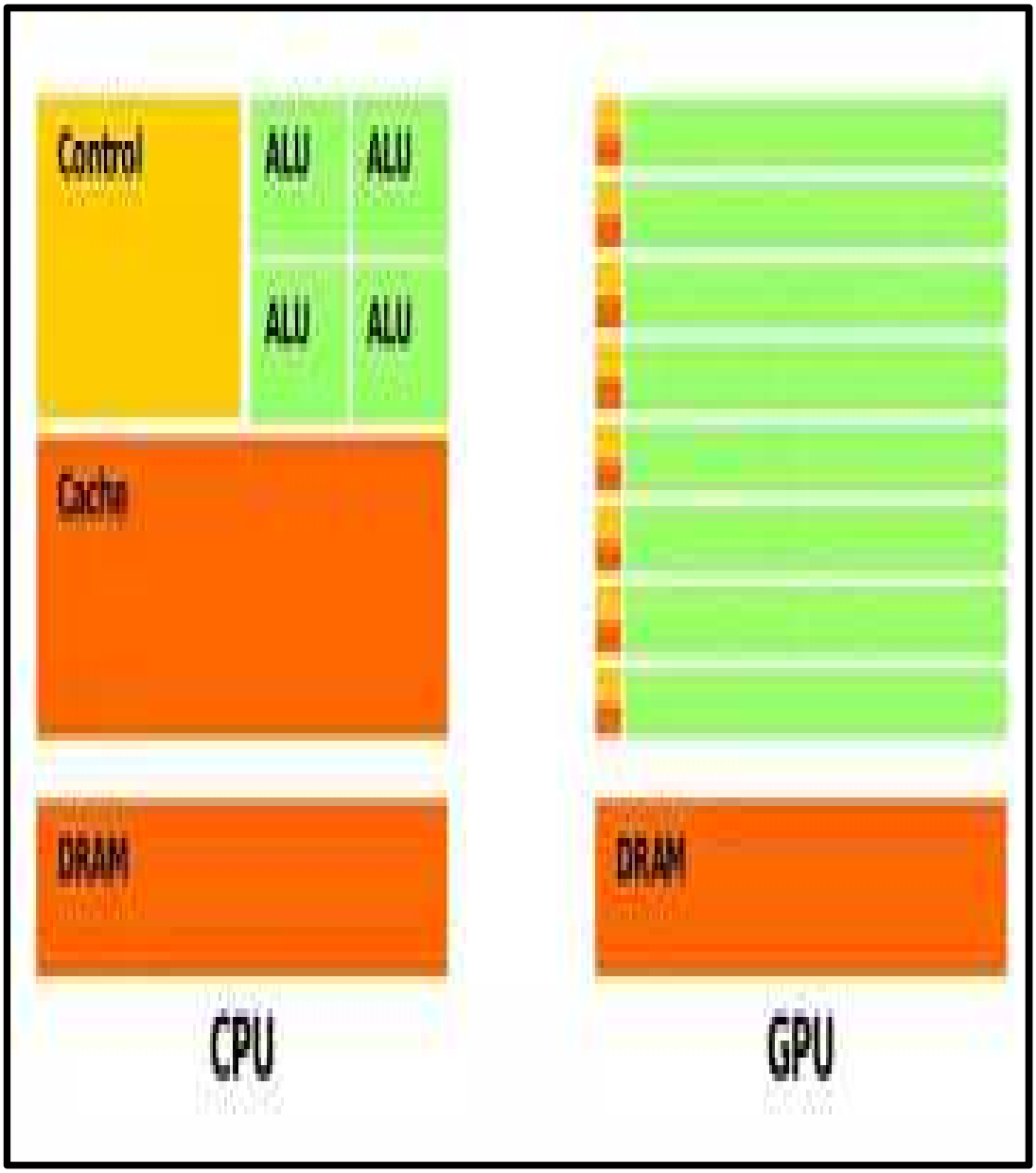


Figure 2.7: GPU vs CPU

	GPU / GPGPU	CPU
Memory Latency	• Context switching automatically done in HW (ZERO cycle latency) • GPU can switch to different thread until data returns from memory	• Context switching handled in SW (by OS) • Stalls if there is an L1/L2 cache miss. Creates bubbles during program execution on cache misses
Thread Management	• Thread scheduler and dispatch unit implemented in HW (manages large # threads with minimal overhead) • Solving problems with massively parallel data such as "sliding window" operations like FIRs, convolutions and other signal processing algorithms are more efficient on the GPU.	• Thread overhead and work-item creation done in SW (CPU needs to manage thread pool) • Maximum # threads is limited compared to GPU.
Parallelism	• HW designed to facilitate parallel operations on all shaders (coordination handled in HW).	• CPU/SW and multi-core CPU (inter-core) communication overhead exists • CPU thread overhead is "heavier" than equivalent GPU thread overhead.

Figure 2.8: GPU vs CPU-2.

CHAPTER-3

METHODOLOGY AND PROCEDURE

3.0 INTRODUCTION/PROBLEM STATEMENT

Of the available approaches, mostly are the 2 degree pipeline wherein the features are extracted by the help of the local binary patterns, SVM and the Multilayer perceptron. CNN on the other hand implement the task in the 2 step method wherein the extraction is done in order to learn the feature at the time of the extraction. These are region specific trainings used for extracting the desired characteristic features at an ease.

The age and gender estimation was done initially by the help of the height of the face and the wrinkles in the face, but these models were found to be quite inefficient as the face height can vary and the wrinkles as well. So, the determination by these methods were prevented, rather CNN was used to determine the age and gender by the help of the training done on the datasets like the UTKFace datasets etc., backend by the help of the GPU units. There are models also being used like the Deep CNN and the cascaded mixtures combinations, such that parameters can be determined, but they are way too complex and consume more hardware computational resources due to which the data handling and the processing of it consumes some amount of time and may result in the reduction of the throughput. Age estimation can be done by using the regression algorithm in general wherein the data is taken continuously but this algorithm may not be quite efficient as the data collected from the physical appearance can be tainted easily. The data taken from the physical data is to be categorized into any one of the predefined categories and due to which the determination of the age is not upto the mark as the data can be mapped to the corresponding feature and as a result it will be mapped to the wrong labels. It is because of the limited amount of the nodes for the labels being created in the network where the data is mapped to the most nearest features which has been extracted. There are upto 8 nodes which are ready to be mapped such that the data can be mapped according to the distance or the measurements

related to the eyes, nose and ear etc. are considered for the sake of the estimation. The data nowadays is estimated by the help of the facial features such that the data is estimated at a better accuracy value. The data is estimated based on the facial features like that of the beard and the moustache and in association with that of the features pertaining to eye, nose, ear etc. The localisation of the features is done in order to estimate the desired features. It was made essential to determine the age and gender at once as the process had been made easier for the estimation. The data for the gender is processed by the help of the autoencoder followed by the binary classifier or a normal classifier such that the encoded data is given to the classifier in order to determine the gender but it is not quite efficient enough to follow it forever, hence the data is given to a CNN network of 2 stages such that the data is given to it in order to process it.[12-34, 12-92]

Computer vision is the science that is used to apply related to the images and the video frames wherein the image processing applications and the video analytics is to be done, the data is to be taken for the sake of the filtering, detection techniques like the edge detection etc., histogram methods, multirate sampling techniques to the images such that the data is used for the other purposes. The use of the computer vision makes it easier for the sake of the analysis of the images for the sake of the colour parameter to be estimated. [35-58, 59-92]

Pattern recognization is achieved by the help of the neural networks and data used to train itself and then validate it. Pattern is usually depicting the type of the dress worn to be estimated like the shirt or the dress etc. Statistical parameters determine and govern the model efficiency. The use of the classifier is a must in order to classify the data as per the required labels available such that the data is given to a clustering algorithm if required in order to form the clusters based on the classification results, having clothes of the same kind to be represented to be a single cluster. The network must be having good number of dense layers for it to have good values of the accuracy. [59-92]

3.1 FLOW CHART

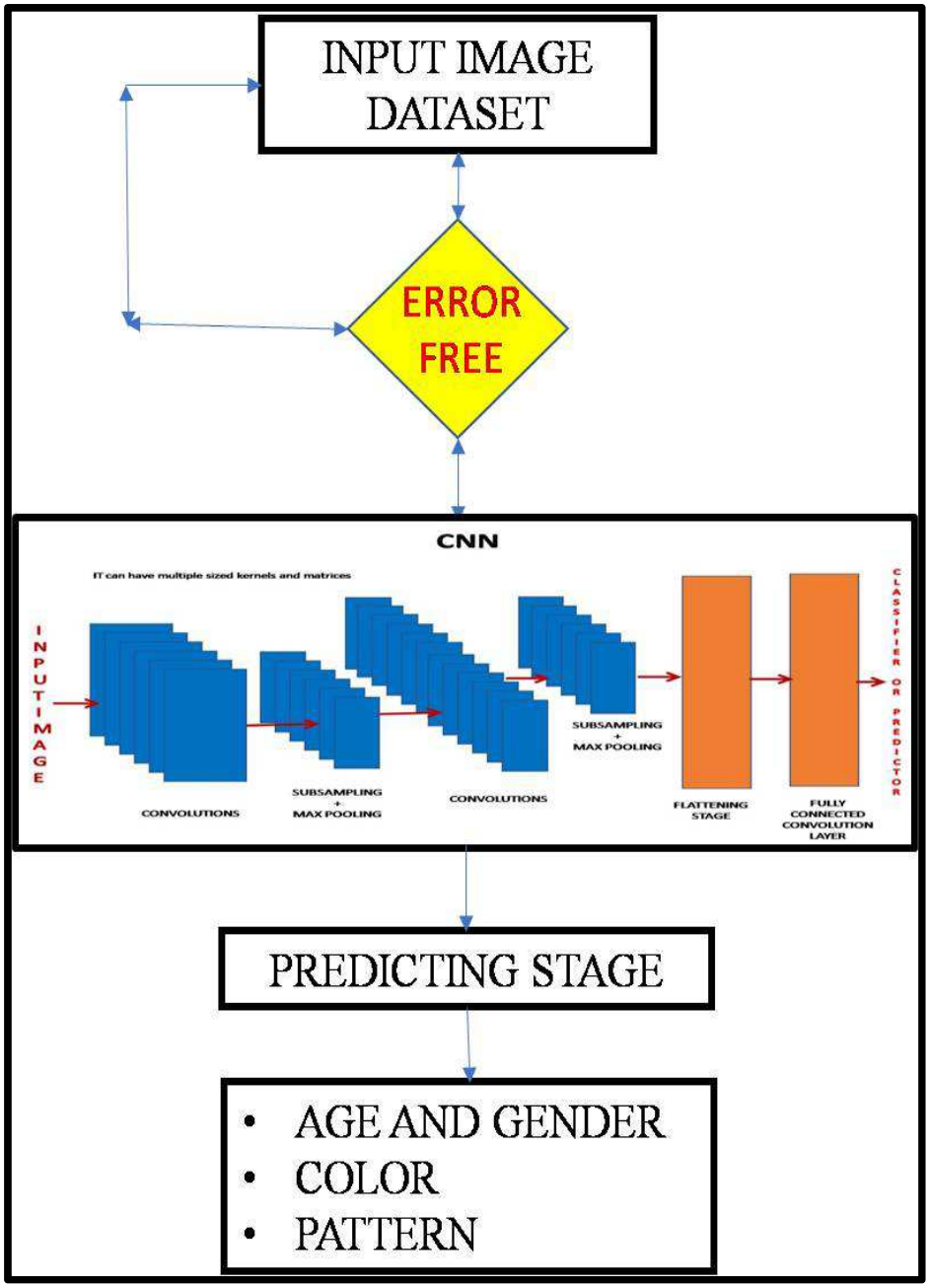


Figure 3.1: Basic Flow chart for the problem statement

The data is assumed to be pure and then the data is given to the CNN neural network for training itself and later validating itself at a good value of the accuracy. Based on the efficiency of the training, the data prediction can be done easily. The efficiency is governed by the intensity of the perceptron count and the functions being used in each layer of the network. The

accuracy also depends on the hardware being used for the sake of the computations to be done on the data.

3.2 TRAINING SET

Training set from the input dataset is used for training the model such that the network learns about what kind of data on which it will operate. It is essential to train the network in order to determine the testing of the network to be efficient. In order to pass the tests, the training must done at a good value pof the accuracy in order to minimize the value of the cost function. The data is to be error free especially at the time of the training for it to learn about the data perfectly. In CNN, the training can be done effectively by the help of the GPU units rather the CPU units such that the accuracy be more than 95%. In general the entire data is to be splitted into a set of 80:20 where the 8o% constitutes for the training set and the rest for the train set from the main downloaded dataset such the tests can be passed easily as the model has learnt more than what it is to be tested from. Overtraining and the under training must be reduced such that the overall accuracy of the network model can be a good value. [35-92]

3.3 TESTING SET

Testing set is used to test the trained neural network model such that the functionality of the network can be easily determined and evaluated accordingly. The data is to be tested and it comprises upto 20% of the original dataset such that the evaluation can be done in a fair manner and the data can be processed accordingly to pass the tests. The validation accuracy value must also be a good value such that the model can be used for the application giving better results. The testing dataset can be having the inputs taken from the training set or from a different dataset.[35-92]

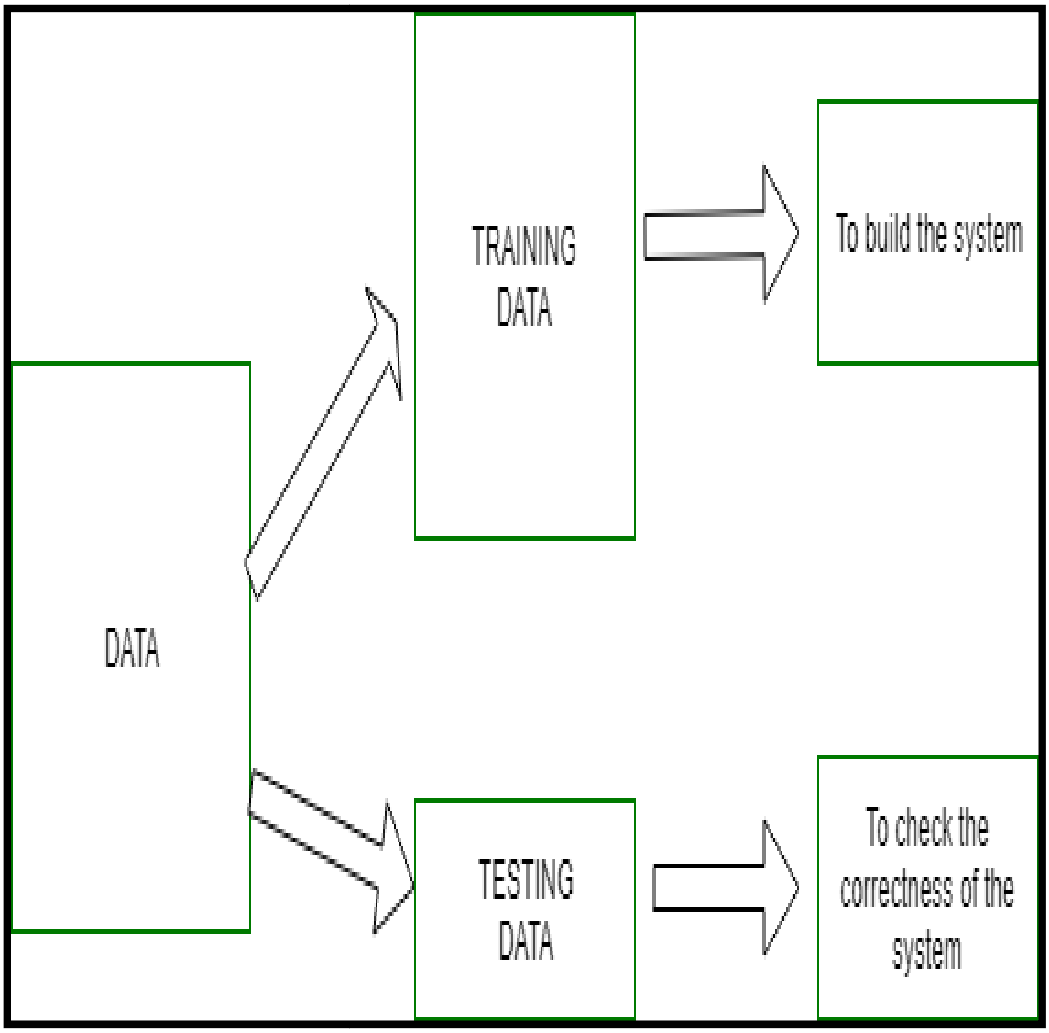


Figure 3.2: Evaluation of a model

3.4 INSTALLATION OF ANACONDA SOFTWARE

Python language can be downloaded from the python.orh or it can be used by the help of the anaconda navigator. The anaconda navigator is open source software wherein it is derived based on the python language and the data can be programmed accordingly in an enhanced version of the representation of the text editors and has support of very powerful tools. It has tools like the jupyter notebooks, spyder as code editors and other applications like the anaconda prompt where it supports for the command line arguments to be passed and all the applications share this environment. Python is an interpreted language and it has the vast support of the libraries but in case of the anaconda navigator, it supports more

libraries than what it had been made of. It supports for the R programming as well.

3.4.1 ADVANTAGES

There are various **ADVANTAGES** of using Anaconda Software which are as follows:

- Anaconda includes Python plus about 600 additional Python packages. These are all free to install.
- Anaconda installs without administrator privileges. When don't have the ability to install programs on a computer, we can still download and use Anaconda. The Anaconda distribution of Python will also allow us to install additional modules from the Python package index (PyPI.org) and conda-forge (conda-forge.org), the conda package index.
- Anaconda makes package management and virtual environments easier. Virtual environments and package handling might not seem to make a huge difference right now. If we just downloaded Anaconda for the first time, we are probably not dealing with package management and virtual environments yet. After writing a couple of Python programs and start downloading a couple of extra modules from PyPI or conda-forge, dealing with package management and virtual environments becomes more critical.

3.4.2 STEPS TO INSTALL ANACONDA NAVIGATOR

- Visit the website and go to Anaconda.com/downloads,
- Select for the Windows OS option,
- Download by clicking on the .exe installer of the anaconda navigator,
- After downloading it, run the .exe application installer
- Open the Anaconda Prompt and run some Python code

Figure 3.3: Anaconda Navigator Screen

1. Visit the Anaconda website and go to the downloads page: the link being Anaconda.com/downloads. The Anaconda Downloads Page looks like

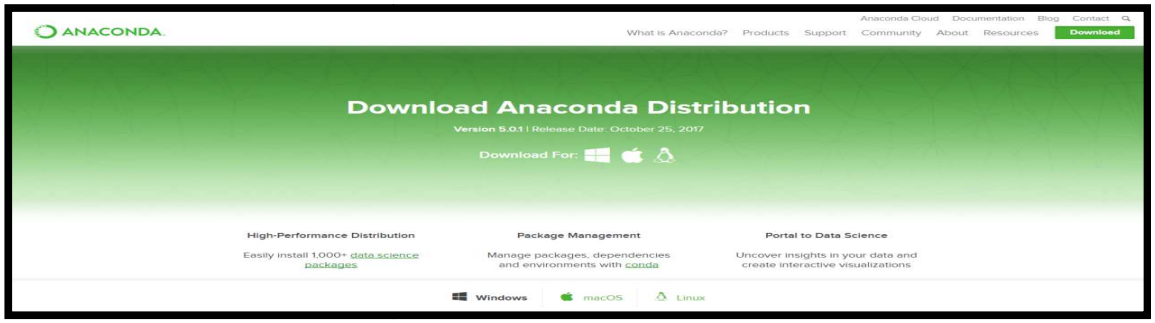


Figure 3.4: Anaconda Downloads webpage

2. Select Windows: Select the Windows options or any other required OS option which the system possesses from the listings. Wait till the entire version is downloaded. Preferrably download the latest version of about 3.6 to 3.7 .

Figure 3.5: Windows option for download on the webpage.

3. DOWNLOAD

The recent Python 3 versions released is to be download. Ensure the system is either 64 bit or 32 bit OS and download the correct version of the software. The download can also be done from the email, it is optional and depends on the user. Check whether the computer is running on a 64-bit or 32-bit version of Windows. The download can also be done form the email. The download can take some time as the file is large and sizes upto 500 MB.

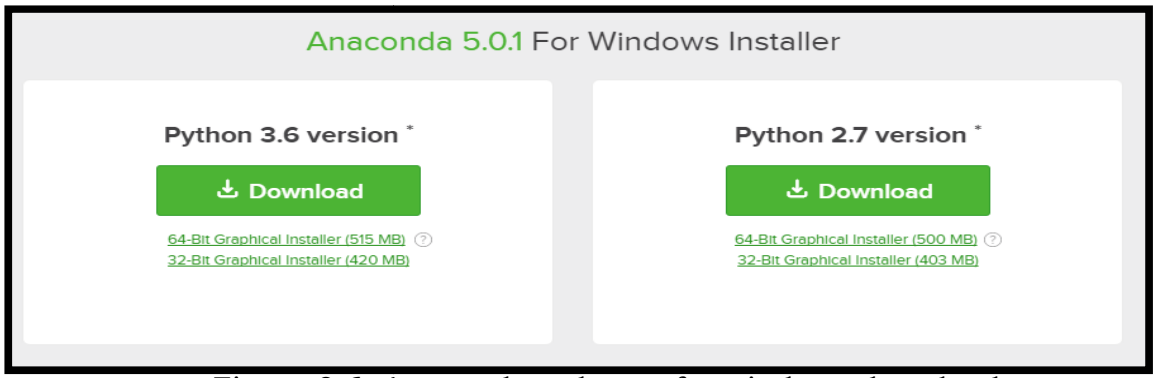


Figure 3.6: Anaconda webpage for windows download.

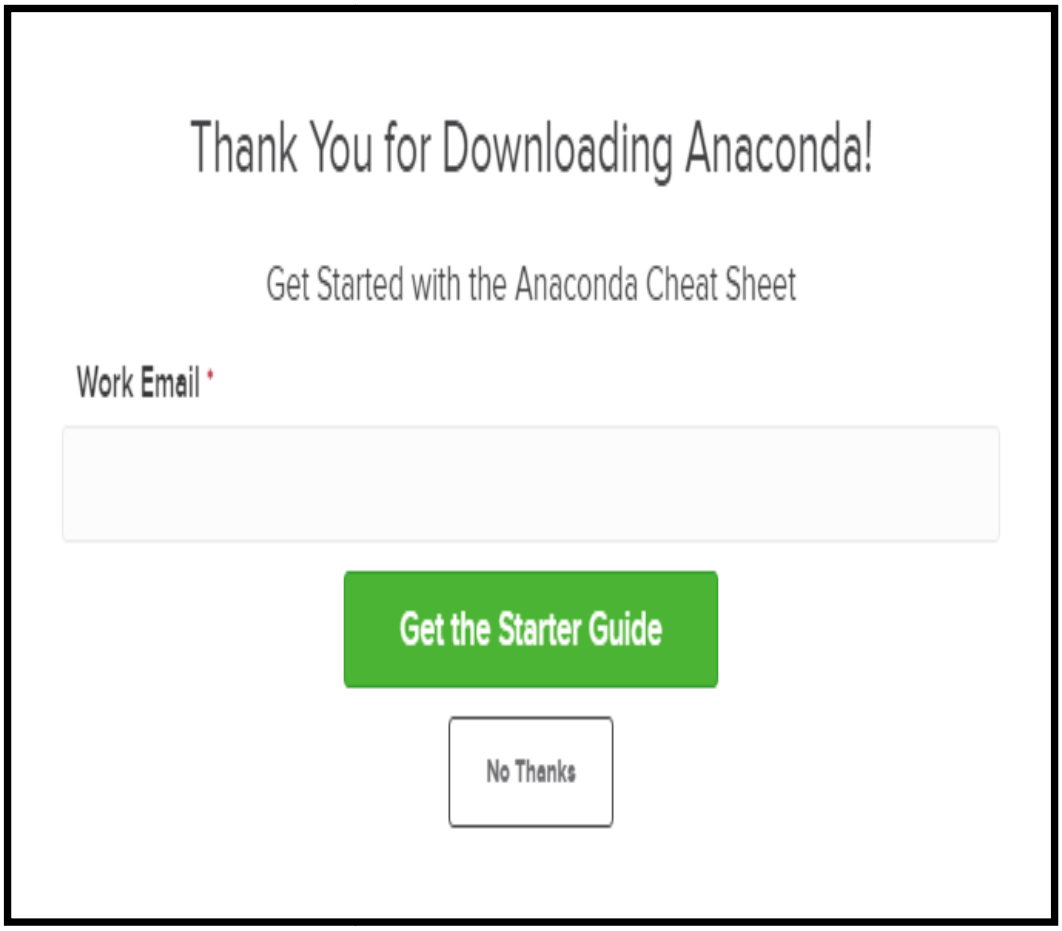


Figure 3.7: Anaconda webpage for email to be entered. (optional)

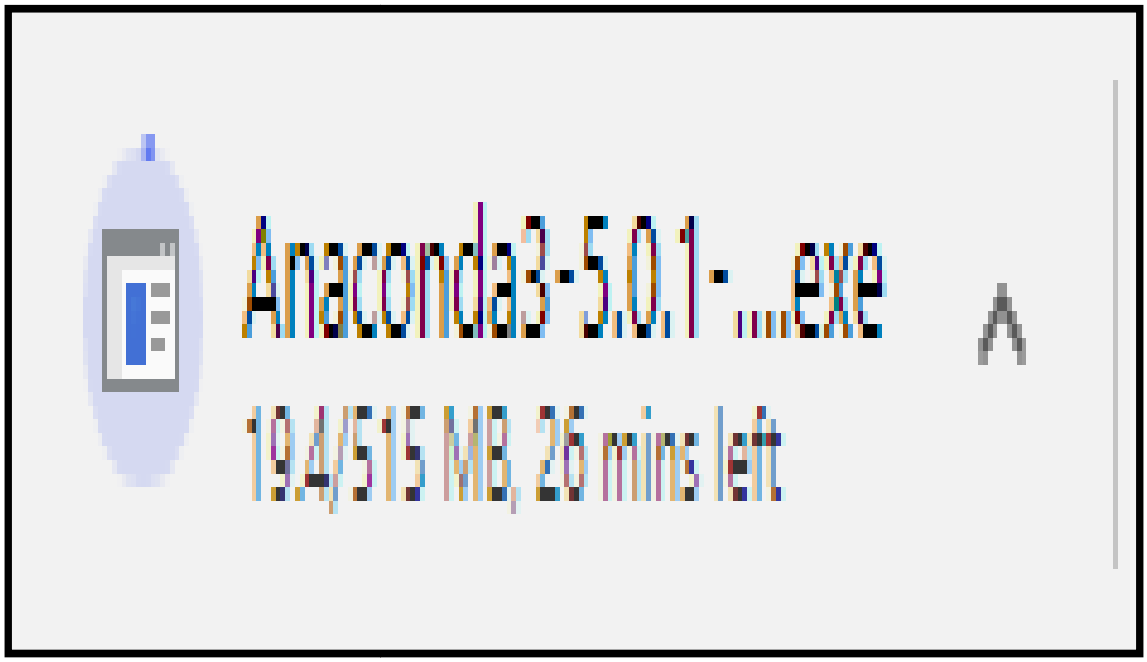


Figure 3.8: Download in progress.

4. Open and run the installer: Once the download is terminated, click on open and run the *.exe* installer file to start installing the downloaded file. Select the click Next button to proceed to the next step of the installation.

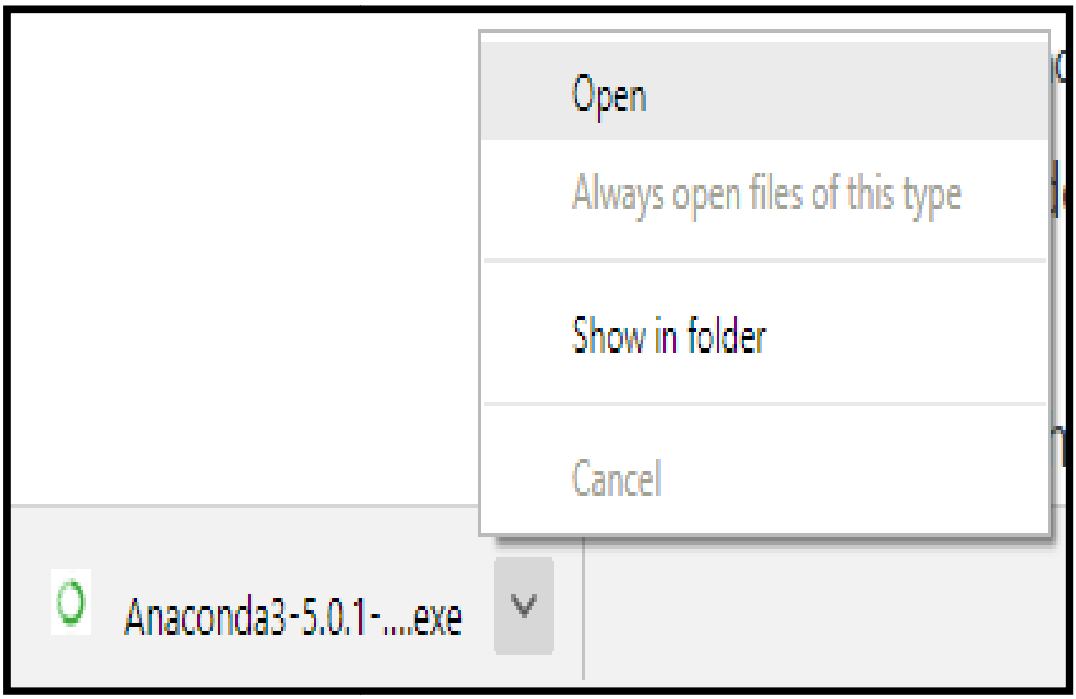


Figure 3.9: Downloading complete and open the file.

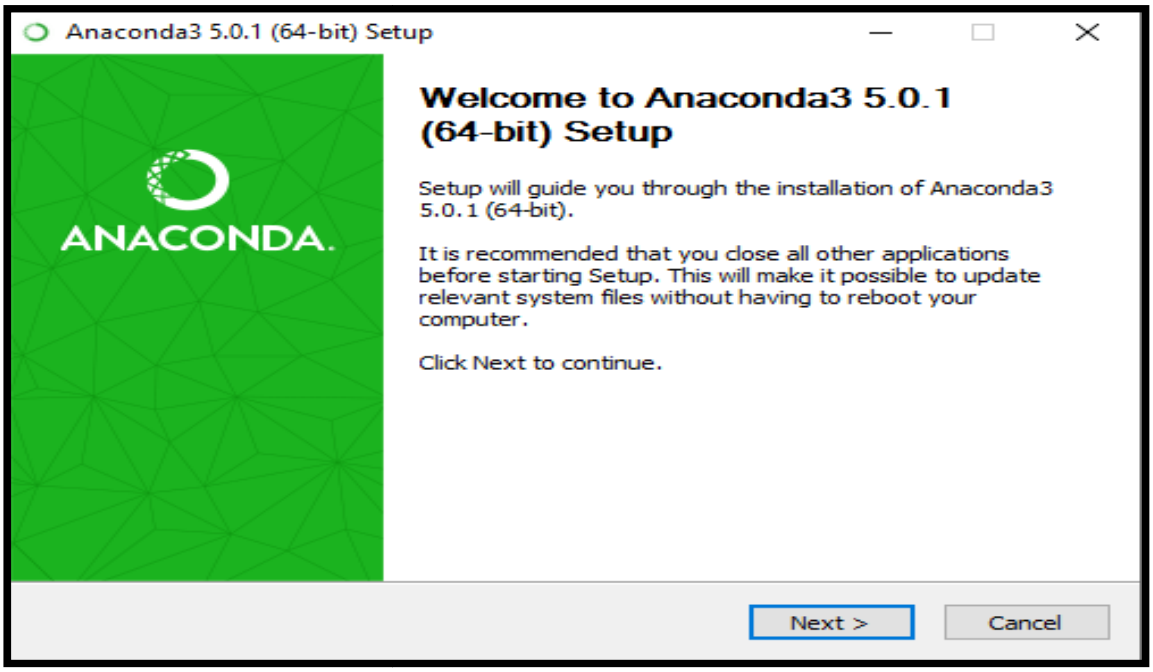


Figure 3.10: Click to install window of anaconda.

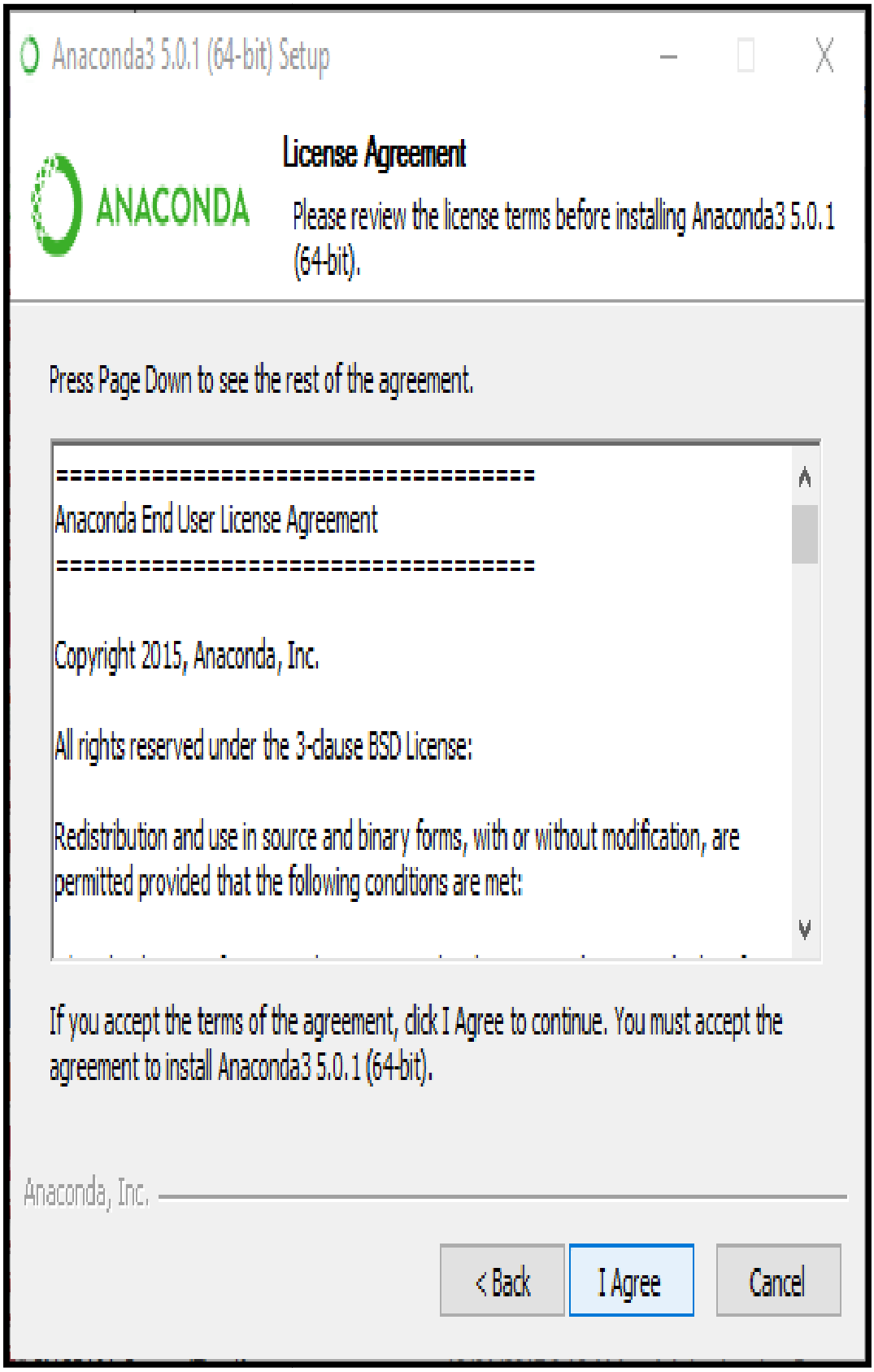


Figure 3.11: Window for anaconda license.

On the Advanced Installation screen, unselect "Add Anaconda to my path environment variable" and download the normal 3.6 version.

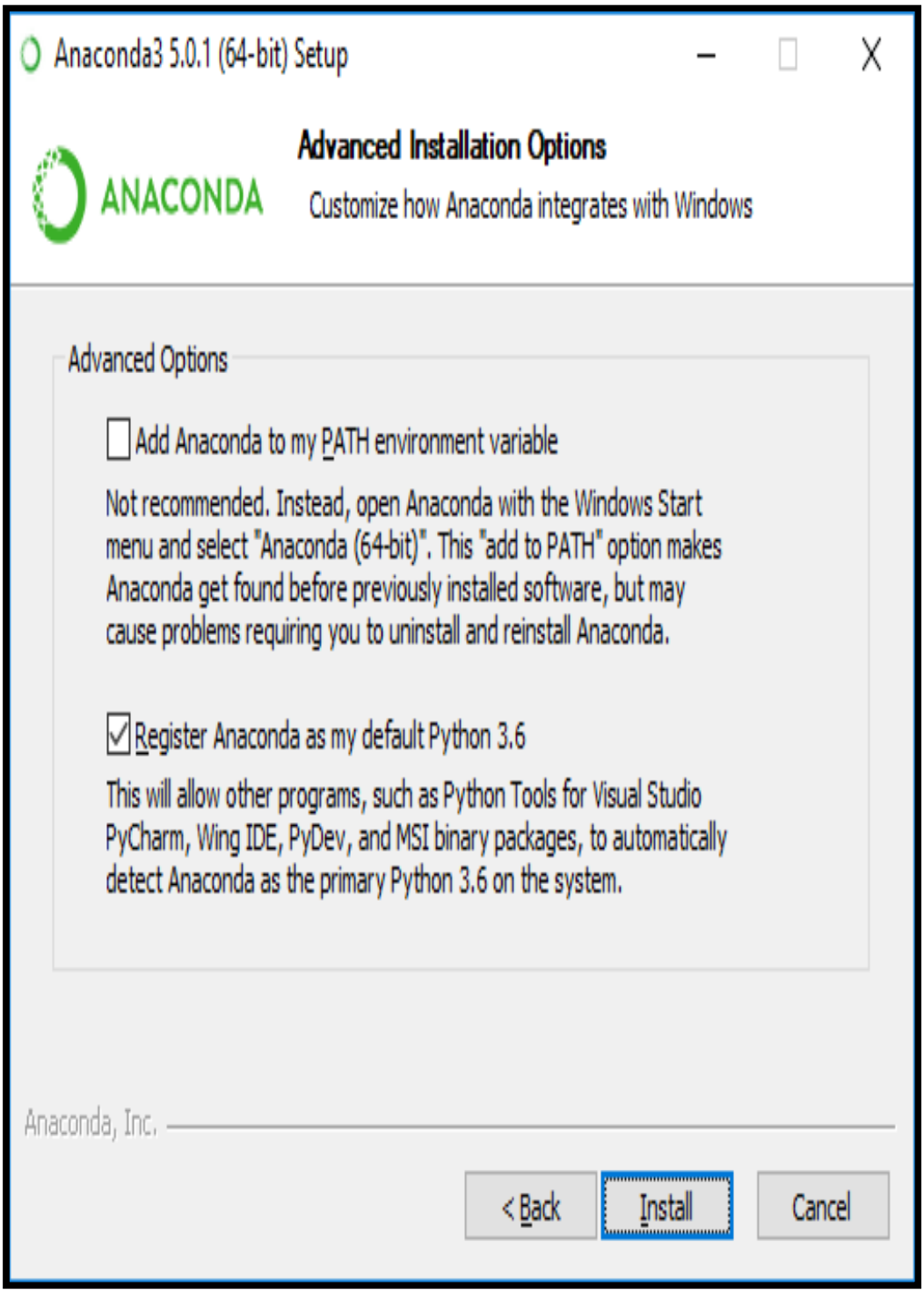


Figure 3.12: Advanced installation option window

5. Open the Anaconda Prompt from the Windows start menu: As the installation is complete, go to the Windows start menu and select Anaconda Prompt in order to install the required versions of the packages.

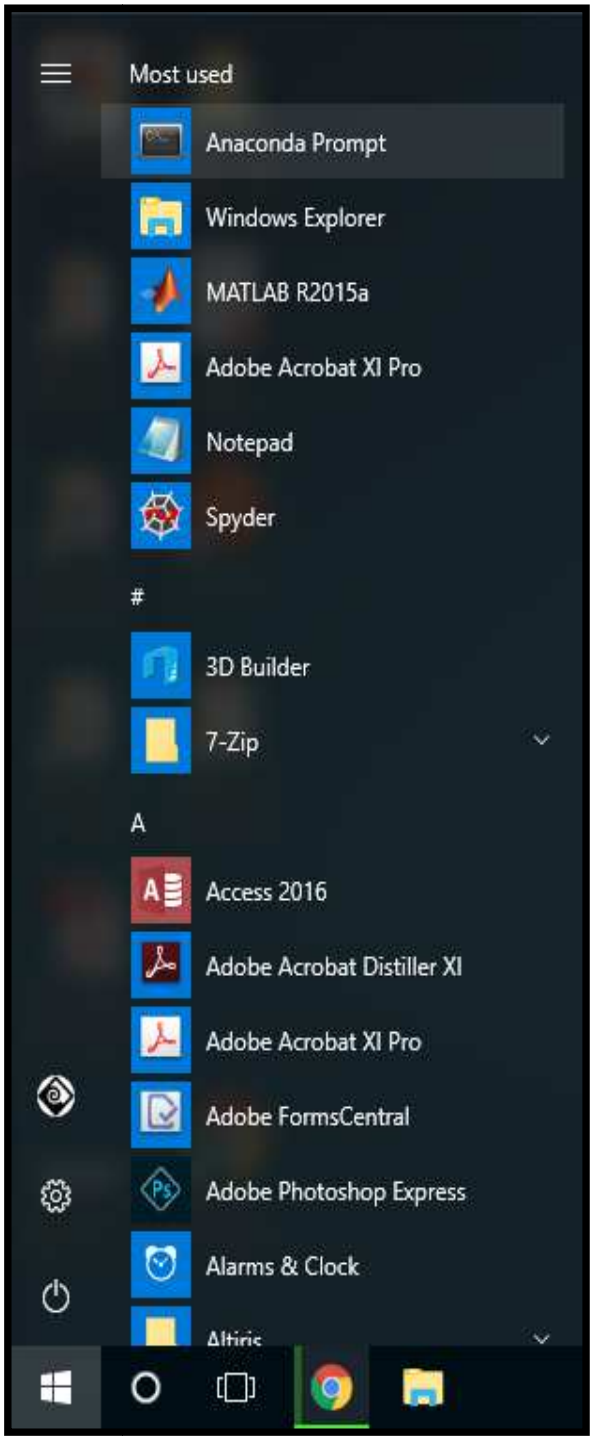


Figure 3.13: Search for anaconda prompt in the windows start menu.

The Anaconda Prompt is opened which is a black sceen command window, taking commands which are executable as per the defined standards of the downloaded software. The python command if entered, takes it to the start of the python interpreter and the respective functions of the python can be used in the form of commands. The opened interpreter is closed by the help of the exit() function.

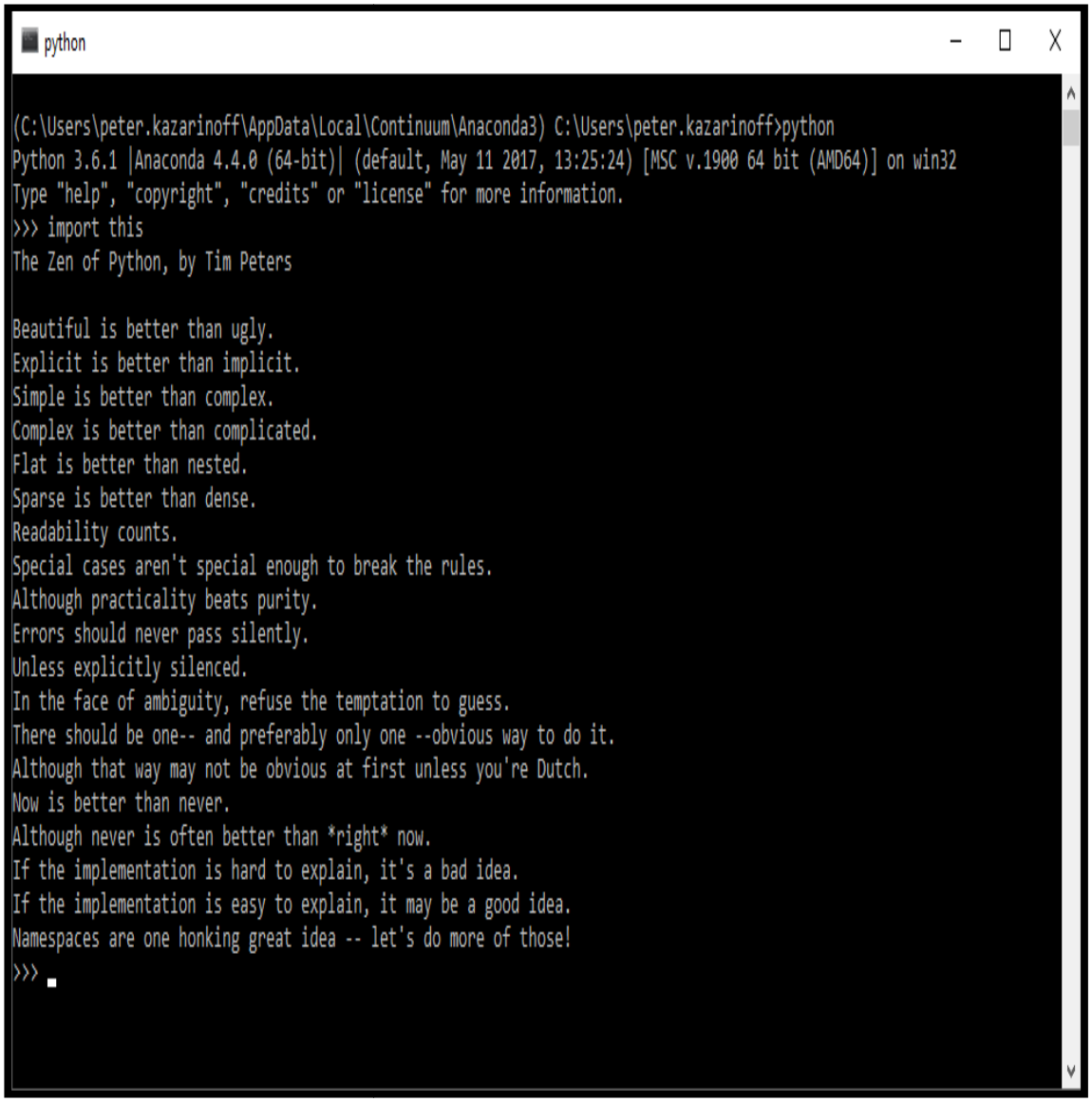


Figure 3.14: exit( ) command to exit the run mode.

One can easily perform the options related to the task by the help of the python programming language. They can be installed by the conda install or the pip install libraryname to install the particular package

which can be imported whenever required until and unless installed in the windows OS. The benefits of python are:

- **Increased productivity:** By some reports, Python can increase a programmer's productivity – how much work they can accomplish in a given time – by as much as ten times! It literally is faster than a speeding bullet!

- **Extensibility**: One great advantage of Python is the fact that it has a very extensive library of, well, libraries. A library is a set of existing code you can add-in to your program. These libraries cover things that are common features of a program and save you from having to write the code over and over again yourself. For example, instead of having to write a section of code to perform a complicated mathematical equation, you can supply use a library and save yourself a huge headache.

- **Python is easy to read:** One tough part of being a programmer is the fact that, sometimes, your code does not work. When that happens, you might find yourself re-reading your code – or worse, someone else's – to try and figure out why your program is not behaving as it should. Fortunately, Python is easy to read and most of the language makes sense at a glimpse. This makes finding issues a lot easier than more complicated languages.

- **Portability:** Python runs on many platforms and systems, meaning your programs can reach a wider audience.

- **Internet of Things (IoT):** The Internet of Things may sound like a magical world full of digital beasts, and in some ways, it is. The IoT consists of smart objects – light switches, doorknobs, toaster ovens, appliances – that you find in your everyday home. These household appliances are controllable by voice commands and mobile devices, making them more interactive than their primitive predecessors. I mean sure, your mom and dad yelled at the dishwasher all the time – but did it ever listen? Now, thanks to the IoT and languages like Python, it can! You still have to put your dishes inside of it, but still!

- **Python frameworks:** Frameworks are like skeletons for a program – they allow you to quickly set up the basics for certain types of applications without needing to code common elements that usually exist in the type of software you are developing. This saves programmers time and reduces the number of errors that can occur when you have to manually code. Python is supported by a large number of frameworks that can make launching a new program very rapid indeed.[111]

CHAPTER-4

PACKAGES AND DATASETS USED

4.1 INTRODUCTION TO THE PACKAGES USED

There many packages in python due to which the coding has been made very easy. A package is also called as library or a header file which contains the predefined methods/functions, fields, and variable that serves a particular purpose. Package It is a file with __init__.Py used to call it. It is present in every python path directory, having a record named as __init__.Py. The packages are imported by the help of the import keyword. They structure python's module namespace by the help of "dotted module names". They maintain well-organized data set hierarchy, making them easily accessible. They can store various amounts sub-packages and modules, and call them whenever required.

4.1.1 TENSORFLOW

TensorFlow is an open source library which is mainly used for the sake of the fast numerical mathematical computing especially in the neural networks. It is under the control of Google and also has the control by the Apache 2.0 open source license version. The API is developed only for the Python programming language, but it can also be used as the C++ API. They are very useful in the Deep Learning applications having like Theano, TensorFlow usage. They can run on single CPU structures, on the GPUs, cell devices and large scale disbursed machines with loads.TensorFlow is compatible to work with Python 2.7 and 3.0 versions. the GPU is employed especially in the linux by the help of the cuda toolkit. In the Anaconda Prompt window,

Installation: pip/conda install tensorflow.

Uninstall: conda uninstall tensorflow

4.1.2 KERAS

Keras is a python library used especially for the sake of deep learning applications. It runs by using Theano or TensorFlow as its base. They are very much feasible for the research and development and can be used mainly for the deep learning applications. It runs on Python 2.7 or versions above 3. They can be executed on the GPU's and CPU's but there must be underlying framework. MIT has its permissive license. Keras was introduced by François Chollet, Google engineer based on these standards:

- **Modularity**: They use discrete components/modules for the arbitrary approaches especially for the deep learning applications.
- **Minimalism**: It is enough to get final results, maximizing readability.
- **Extensibility**: New modules can be added easily and can be used for the smooth operation of the framework.
- **Python:** They are local to the Python.

They model neural networks. The main model is "Sequence" which is a linear stack of layers. So, sequence is called/created and can be added with layers to it. After compiling the model, by the underlying framework, the computations can be optimized by the underlying framework. The loss/cost function and optimizer are to be used in order to enhance model's accuracy. The model must be made to be fit to the input training and testing datasets. Once trained, the model can make predictions on the test data .The steps to proceed with the keras framework is:

- Define model. Create a "sequence" and add the required amount of the dense layers.
- Compile model. Specify and optimize loss functions and optimizers.
- Fit the model for the train data.
- Make predictions on test data. Use the newly made model or the trained model.

On the Anaconda Prompt window,

Installation: pip/conda install keras.

Uninstall: conda uninstall keras

4.1.3 OPENCV

OpenCV came into existence in 1999 by Gary Bradsky working in the Intel and its first version released in the 2000. OpenCV is used by variety of algorithms related to Computer Vision and Machine Learning. Currently OpenCV is used by the programming languages like C++, Python and Java etc. and is compatible with the Windows, Linux, Android, iOS etc. The interfaces are primarily dependent on CUDA. They are yet to improve for high-speed GPU operations. OpenCV-Python is a Python API of OpenCV in combination with the characteristics of OpenCV C++ API and Python language. Initially, the following packages are to be installed, before installing the opencv:

- Numpy.
- Matplotlib

Installation of OPENCV: conda/pip install opencv.
Unintallation of OPENCV: conda/pip uninstall opencv.

Install all of them at the default location, at the C drive and ensure that they are working properly. On the anaconda prompt window, the version of the opencv can be observed by the following commands in python interpreter.

>>> import cv2

>>> print cv2.__version__

Note that there is no 64 bit official numpy version, so we must be able create our own version on the same compiler or use the anaconda environment.

4.1.4 SCIKIT

Scikit-examine is an open source python library used for gadget scientific purposes. This library has the algorithms like KNN, XGBoost, random forest, SVM etc. It is a quite useful library. It is built with Numpy its

base, so the numpy and scipy must be installed first. Scikit-learn is used by kaggle and its competitors. Scikit-examine is useful preprocessing of data, dimensionality reduction, classification, regression, clustering etc. To download and Install scikit-learn:

Scikit-learn can be installed by the command: pip install -U scikit-learn

4.1.5 PANDAS

Pandas is one of the python libraries, known for using the reliable data structures for easy access of the data in a structured data (tabular, multidimensional, potentially heterogeneous) and time series data. It is useful for the practical, realtime data analysis in python. It is the most powerful data analysis/manipulation tool for the sake of organizing the data.

It saves the data in the form of the tabular records with different columns, just like SQL table or an Excel spreadsheet. Ordered or unordered time series data need not have been labeled in any aspect. It is easy to handle the missing data as it is represented as NaN in floating and non-floating point data. Columns insertion and deletion from Data Frame is very easy and alignment of the data can be done as well. The splitting of the dataset can be easily. Easily converts the differently indexed data and NumPy data structures into DataFrame objects. IT supports for the label-based slicing, indexing, and subsetting from data sets, reshaping datsets and pivoting of datasets are its additional functionalities. Robust IO tools to load data from flat files (CSV and delimited), Excel files, databases, and saving/loading data from the ultrafast HDF5 format

Time series data has specific functionality like the date range generation and frequency conversion, moving window statistics, date shifting and lagging with the respective time stamping. Data preprocessing is also done by it such that all errors are removed from the data,

analyzing/modeling it, organizing the results suitable for plotting/tabular display.

Pandas Library Architecture:

- pandas/core: Has data structures of the Pandas library.
- pandas/src: Has basic functionality of Pandas, usually written in C or Cython.
- pandas/io: Used to input and output files, data, etc
- pandas/tools: Codes and algorithms for various operations in Pandas.
- pandas/sparse: Has sparse versions, to handle missing values
- pandas/stats: Has functions related to statistics, like linear regression
- pandas/util: Consists testing tools and debugs the library.
- pandas/rpy: Interface to connect with R. Also called as R2Py.

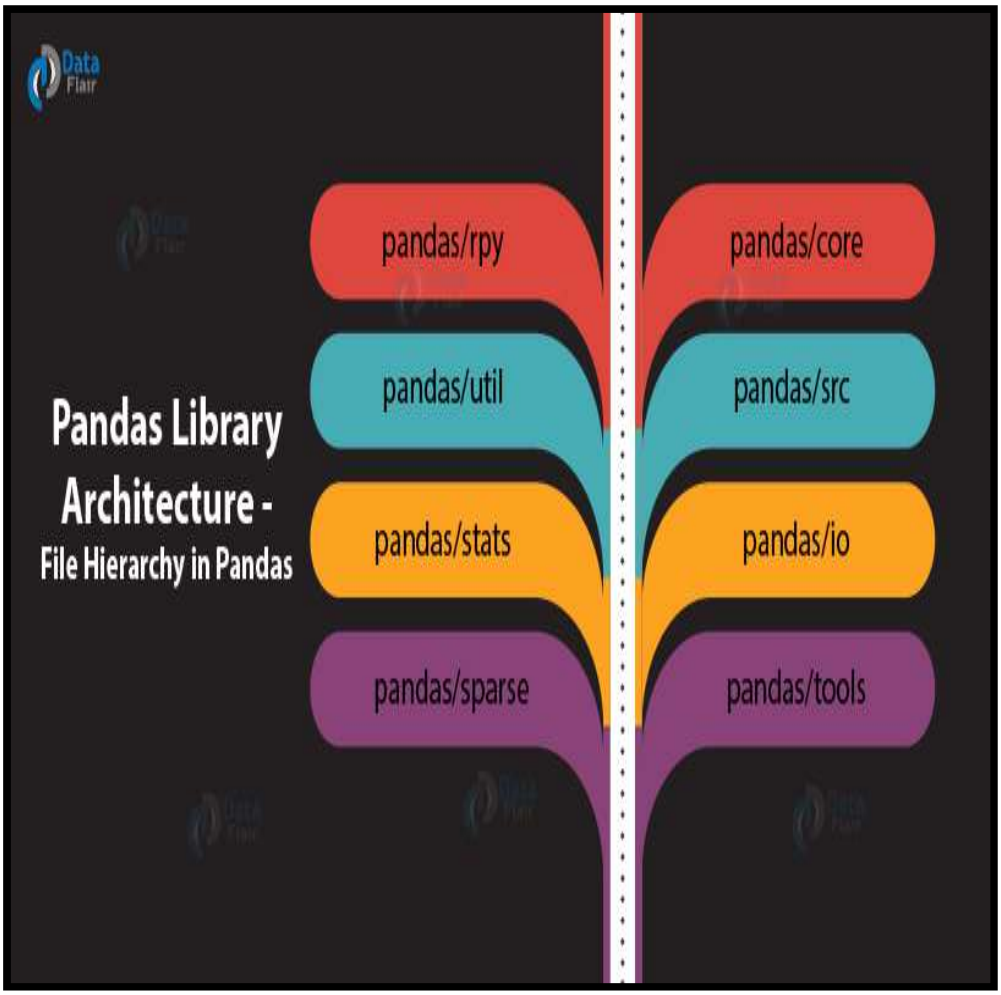


Figure 3.15: Pandas Library Architecture

In the Anaconda Prompt window,

Installation of the Pandas: conda/pip install pandas

Uninstallation of the Pandas: conda uninstall pandas.

4.1.5.1 APPLICATIONS OF PANDAS

- **Economics:** For facts analysis.
- **Recommendation Systems:** Recommend guidelines/alternatives.
- **Big Data:** To store the large amounts of data in a dataframe, for easy access.
- **Data Science:** Proper handling of the data and the data preprocessing can be done.

4.1.6 NUMPY

NumPy is required for the sake of the computing associated with the arrays and values stored in them. It includes:

- N-dimensional array and respective functions
- Integrating C/C++ & Fortran codes.
- Linear Algebra, Fourier transform, and Random functions.

It is used in the multi-dimensional statistics and integrate wide forms of the databases. On the anaconda prompt window,

Installation of numpy: pip /conda install numpy

Uninstallation of numpy: pip/conda uninstall numpy.

4.1.7 TKINTER PACKAGE

Tkinter is used as a wrapper around the Tcl interpreter embedded in Python interpreter. Tkinter functions convert them to the Tcl commands given later to the embedded interpreter, allowing combining the Python and Tcl into a single application. There are other GUI library alternatives like wxPython, PyQt (PySide), Pygame, Pyglet, and PyGTK.

- Window: Rectangular display screen used to display the required content and other details.

- ❖ Top Level Window: Independently existing window on screen. It is associated with resizable standard frame and has desktop manager controls.
- ❖ Frame: It organizes the complex layouts.
- ❖ Widget: Building blocks of an application for GUI.
- ❖ Core widgets/Containers.
- ❖ Child and parent relationship
- ❖ Buttons like the radiobutton, checkbutton (checkbox), menubutton (combobox).
- ❖ The text widgets displaying the desired texts.
- ❖ The entry widgets: scale, scroll, listbox, slider, spinbox, entry (singleline), text (multiline), and canvas (vector and pixel graphics).
- ❖ Extension widgets: tk_optionMenu, tk_dialog, tk_messageBox, tk_getOpenFile, tk_getSaveFile, tk_chooseColor, tk_chooseDirectory.

4.1.8 PYTHON IMAGING LIBRARY (PIL)

Python Imaging Library (PIL/Pillow) is a free library used to support for opening, manipulating, and saving various image file formats like the png, jpeg, gif, tiff, and bmp and new formats can also be created based on the programmer's requirement. Latest version of PIL is 1.1.7 and can run on python 2 and 3 versions.

Latest versions of the PIL are called as Pillow having forked the PIL repository with python 3.x. Pillow is used for image manipulation like:

- Pixel Manipulations,
- Masking And Transparency Handling,
- Filtering Of Image - blurring, contouring, smoothing, or edge finding,
- Enhancement of the Image - sharpening, adjusting brightness, contrast or color,
- Addition of text to images.

4.2 DATASETS

4.2.1 UTKFACE DATASET

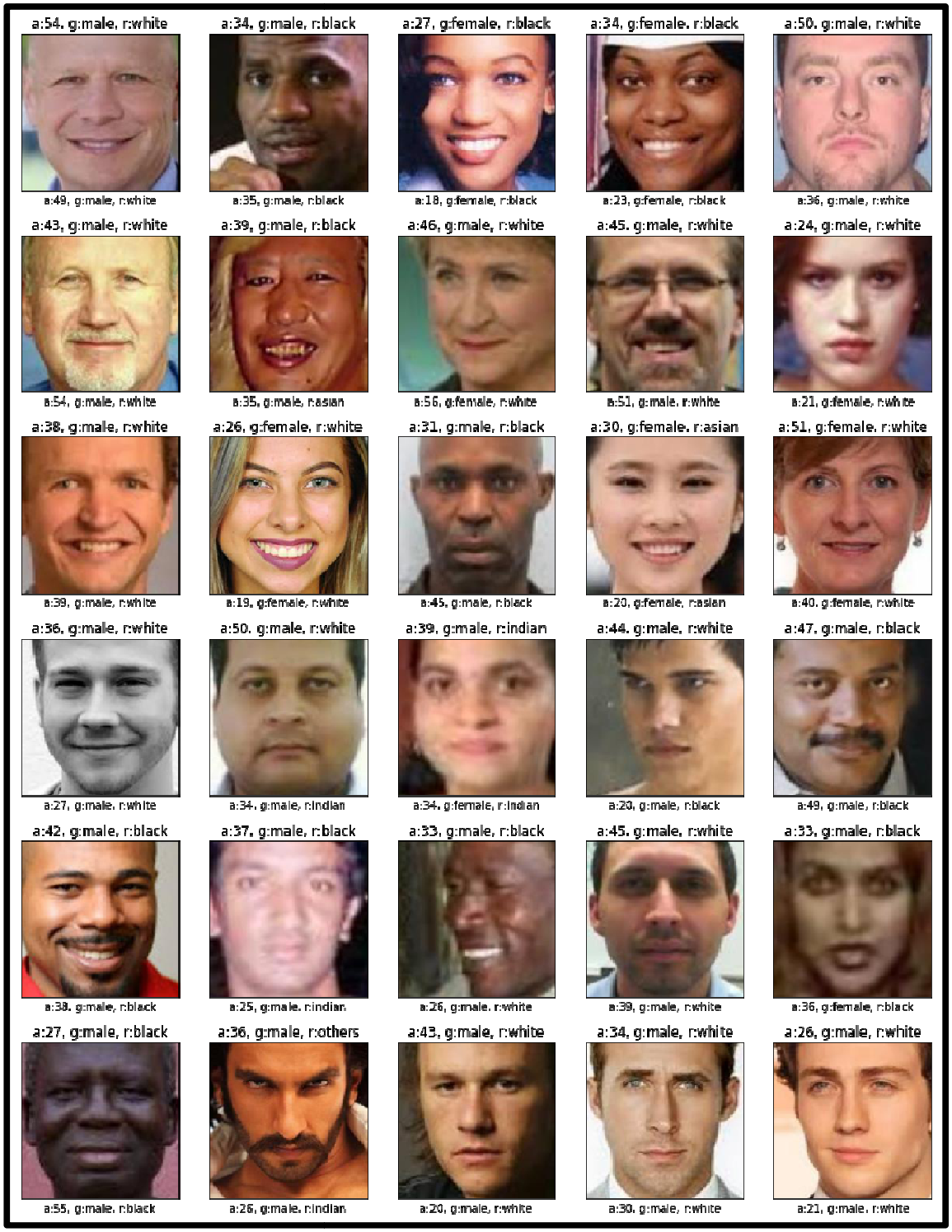


Figure 4.2:UTKFace Dataset.

UTKFace dataset is a dataset comprising of faces of people having ages in the range of 0 to 116 years. The dataset has more than 20,000 (24,107 precisely) face images with usage for the age, gender, and pattern. They cover huge variations in the Poses, Facial Expressions, Illumination/Brightness, Resolution etc. It is used for the face detection, age estimation, age progression/regression, landmark localization etc. Of all the images, it comprises of 52% of images of male and the rest is female. Each image is denoted by age, gender as 1 for female and 0 for male and the age number for the particular image. The dataset has 68 labels for each face. It can be used but the problem is that the dataset is not pure enough to e used. The labels are indicated in the file name as for each image the format of [age]_[gender]_[race]_[date&time].jpg.

- [age] -> integer->0 to 116 ---→Age
- [gender] is either 0 (male) or 1 (female)--→Gender
- [race]->integer->0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, Middle Eastern)---→Race.
- [date&time] is in the format of yyyymmdd__HHMMSSFFF, showing the date and time of an image collected for the UTKFace dataset.-→Date and time. [99]

4.2.2 FASHION MNIST

Fashion-MNIST is a grayscale dataset developed by the Zalando which improves the accuracy of the model, due to the images being grayscale when compared to that of the original MNIST dataset and it is the replacement of it but the images are of same dimensions taken. It has a training set of 60,000 input train images and a test set images of about 10,000. Each image is a grayscale image with dimensions of about 28x28 and each image is associated with a label and of about 10 classes.[98-99]

Here's an example how the data looks (each class takes three-rows):

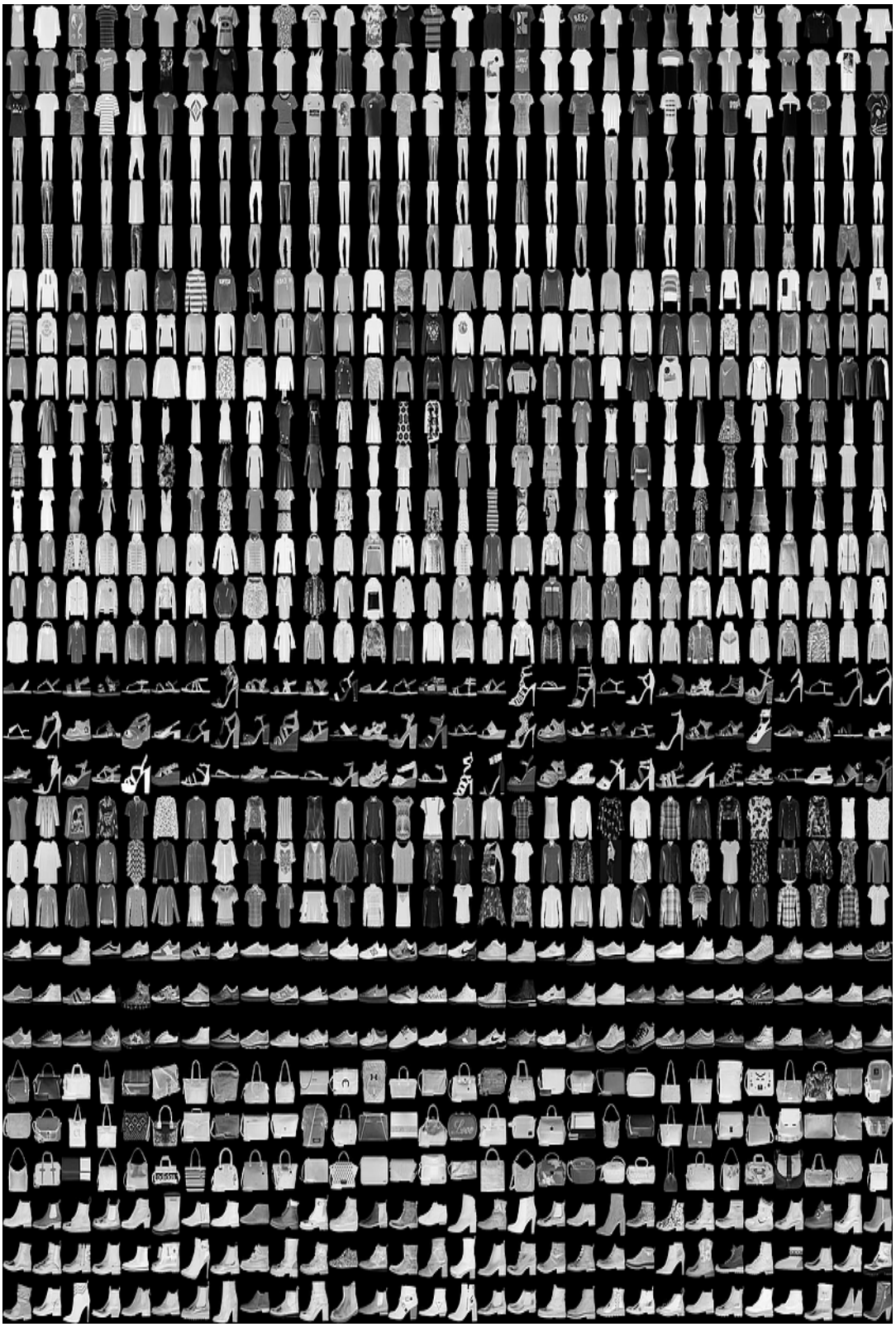


Figure 4.3: FashionMNIST Dataset.

As each image of the training and test sets are labeled for the better easier estimation of the predictions. The labels are shown as:

LABEL	DESCRIPTION
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Table-1: FashionMNIST Labels.

Figure 4.4: Aptness of detection from the dataset.

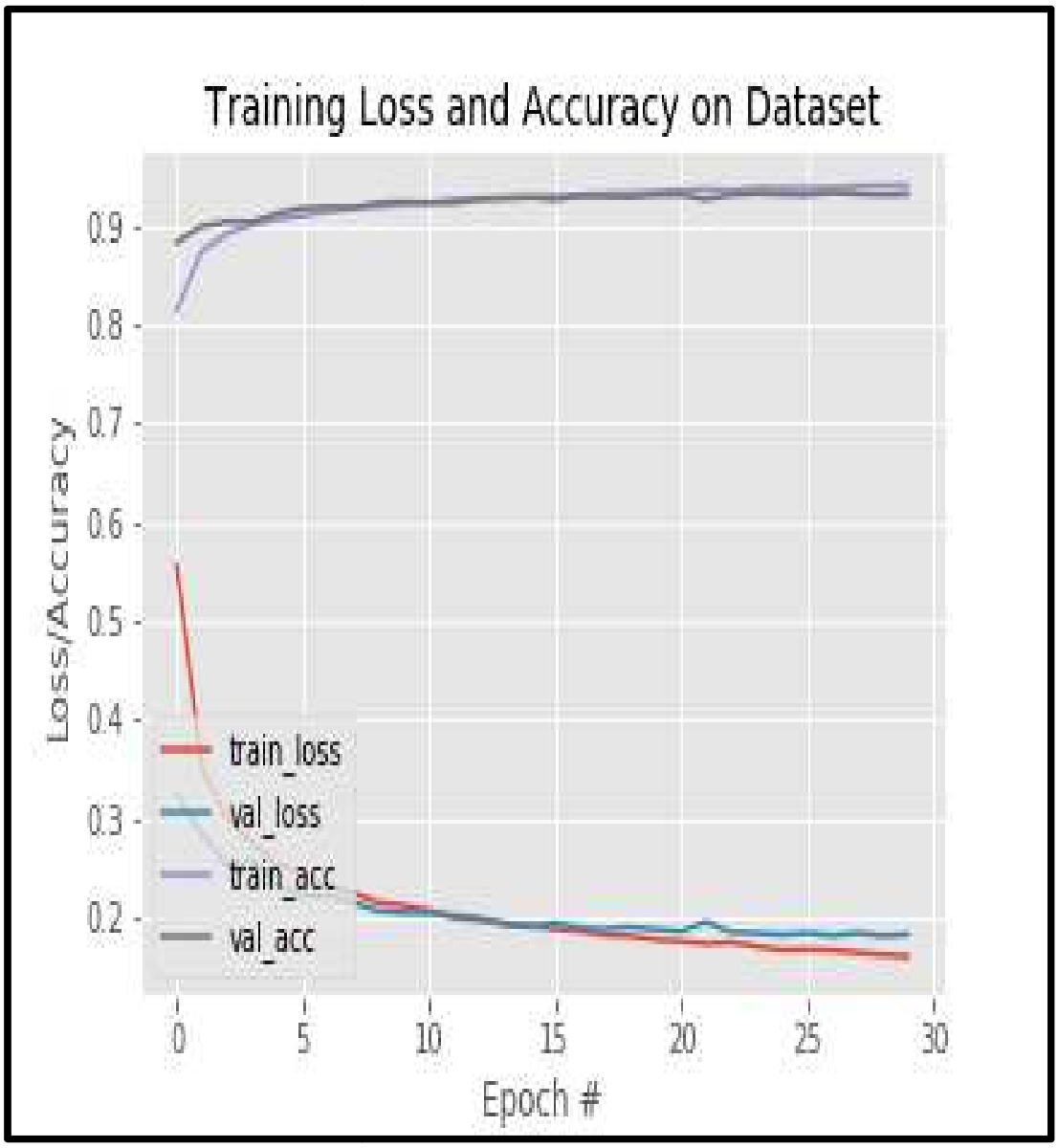


Figure 4.5 : Accuracy plot of detection.

4.2.3 IMDB-WIKI Dataset

IMDB-WIKI is the largest dataset available for the age and gender with labels publicly. It is used to evaluate DEX (Deep Expectation) method and it was introduced by Rothe. The dataset was automatically created by IMDb (Internet Movie Database) and Wikipedia together by taking the most popular actors on IMDb along with the date of birth and gender labelled onto the image. Each image has the date on which it was taken especially in the metadata. They have 20,284 actors and of about 460,723 labelled

images with 57% of it to male faces. Similarly for the Wikipedia data was taken and the metadata was maintained where this subset was having 62,328 images out of which 75% are male faces. The total number of images in this dataset is 523,051.[105]

It is the largest available dataset publicly having facial images with gender and age labels for the training set. [105]

Figure 4.6 : IMDB Datasets.

4.2.4 COLOR CSV FILE

The colors csv file is required in order to predefine the r, g, b values for every color code defined and based on the color code, the naming of the colors is done. The data is the basis for the estimation of the color of the dress the individual wears based on the distance computed for the r,g,b, values on the point in the image and the csv file. The least distance computed will be the color at the selected point of the image.

4.3 MODELS USED

4.3.1 CAFFE MODEL

Caffe model is a prebuilt in deep learning framework in consideration of the efficiency, speed, and accuracy of ease of usage. It was developed by Berkeley AI Research by Yangqing Jia and is released under the BSD 2-Clause license. Its architecture is innovatively modelled and optimized without any complex coding. Either CPU or GPU are used for the training of the model, preferably the GPU. In it's first 12 months of developemnt, around 1,000 developers had contributed to it. [100, 102-103]

The speed of the Caffe model makes it suitable for the experiments and industry deployment and researches. It processes over 60M pictures in a day by the use of the NVIDIA K40 GPU. It takes 1 ms/image for training and 4 ms/image for feature extractions. Caffe model is believed to be the i fastest convnet implementations at present. ImageNet training data is required for the training and validation. The training and validation inputs are described in train.txt and val.txt files as text listed along with the labels.[The images are resized to 256x256 dimensions for quicker processing using mapreduce.[102-103].

4.3.2 MINIVGGNET MODEL

VGGNET model is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the Oxford University. The model gives an accuracy of about 92.7% upto 94 % approximately in the ImageNet or the fashionMNIST. It has huge kernel-sized filters of about 11 and 5 in

the first dense layer and second dense layer respectively along with 3×3 kernel-sized filters one after the other. VGG16 was trained for many weeks using the NVIDIA Titan Black GPU's. [100]

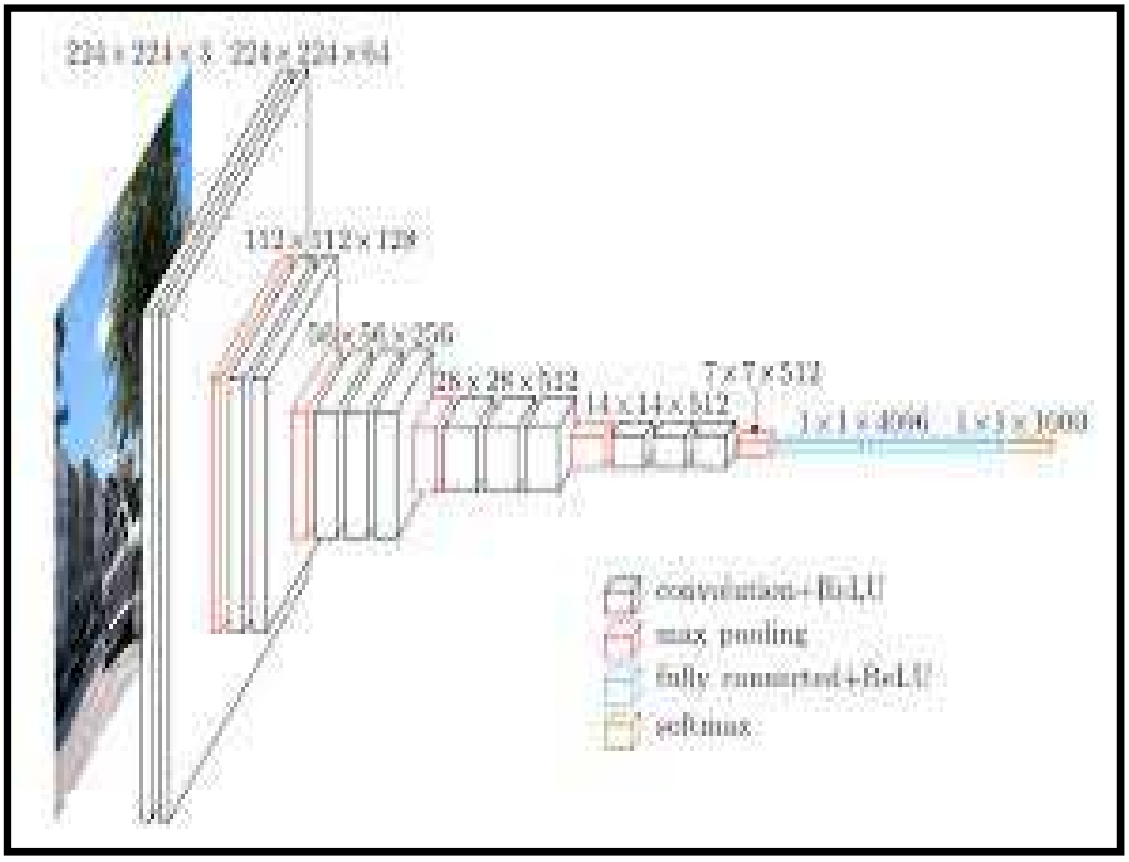


Figure 4.7: MiniVGGNET model.

The input to the conv1 layer is of fixed size if about 224 x 224 RGB picture, filtering the small receptive fields of about 3×3. Extra 1×1 convolution filters maybe used for linear transformation of input channels followed by non-linearity due to ReLu activation function. The convolution stride is set to 1 pixel; the spatial padding at conv. layer inputs preserves spatial resolution after convolution. Spatial pooling is done by 5 max-pooling layers complying with 2×2 pixel window, and a stride of 2. It has 3 Fully-Connected convolutional layers (FC) follow a stack of convolutional layers with different depth architectures. The first two dense layers have 4096 channels each. The output layer has softmax layer. [97-101]

CHAPTER-5

PROGRAM CODE

5.0 INTRODUCTION

The determination of the age, gender, color and the pattern is done in the python language by the use of the Anaconda Navigator's Spyder as its text editor and the data set is collected in advance like the fashionmnist and the UTKFace Dataset (if required). Packages like the pandas, numpy, opencv, tensorflow, keras and the PIL libararies are used along with the scikit, matlplotlib and tkinter library.

5.1 AGE AND GENDER CODE

```
import cv2

import math

import argparse # optional to use, used to take the input from the command prompt


def highlightFace(net, frame, conf_threshold=0.7):

    frameOpencvDnn=frame.copy()

    frameHeight=frameOpencvDnn.shape[0]

    frameWidth=frameOpencvDnn.shape[1]

    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)


    net.setInput(blob)
```

```
    detections=net.forward()
    faceBoxes=[]
    for i in range(detections.shape[2]):
        confidence=detections[0,0,i,2]
        if confidence>conf_threshold:
            x1=int(detections[0,0,i,3]*frameWidth)
            y1=int(detections[0,0,i,4]*frameHeight)
            x2=int(detections[0,0,i,5]*frameWidth)
            y2=int(detections[0,0,i,6]*frameHeight)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0),
int(round(frameHeight/150)), 8)
    return frameOpencvDnn,faceBoxes
parser=argparse.ArgumentParser()
parser.add_argument('--image')
args=parser.parse_args()
#img="female30to35.jpg"
faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
genderProto="gender_deploy.prototxt"
genderModel="gender_net.caffemodel"
```

```
MODEL_MEAN_VALUES=(78.4263377603,87.7689143744, 114.895847746)
ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList=['Male','Female']
faceNet=cv2.dnn.readNet(faceModel,faceProto)
ageNet=cv2.dnn.readNet(ageModel,ageProto)
genderNet=cv2.dnn.readNet(genderModel,genderProto)
video=cv2.VideoCapture(0)
padding=20
while cv2.waitKey(1)<0:
    hasFrame,frame=video.read()
    if not hasFrame:
        cv2.waitKey()
        break
    resultImg,faceBoxes=highlightFace(faceNet,frame)
    if not faceBoxes:
        print("No face detected")
    for faceBox in faceBoxes:
        face=frame[max(0,faceBox[1]-padding):
          min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
                :min(faceBox[2]+padding, frame.shape[1]-1)]
```

```
blob=cv2.dnn.blobFromImage(face,1.0,(227,227),MODEL_MEAN_VALUES,
swapRB=False)

        genderNet.setInput(blob)

        genderPreds=genderNet.forward()

        gender=genderList[genderPreds[0].argmax()]

        print(f'Gender: {gender}')

        ageNet.setInput(blob)

        agePreds=ageNet.forward()

        age=ageList[agePreds[0].argmax()]

        print(f'Age: {age[1:-1]} years')

        cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE_AA)

        cv2.imshow("Detecting age and gender", resultImg)
```

5.2 COLOR DETERMINATION CODE

```
# import the necessary packages

import numpy as np

import pandas as pd

import cv2

x=0

y=0

w=0

h=0
```

```
index=["color","color_name","hex","R","G","B"]
csv = pd.read_csv('colors.csv', names=index, header=None)
def getColorName(R,G,B):
    minimum = 10000
    for i in range(len(csv)):
        d = abs(R- int(csv.loc[i,"R"])) + abs(G- int(csv.loc[i,"G"]))+ abs(B- int(csv.loc[i,"B"]))
        if(d<=minimum):
            minimum = d
            cname = csv.loc[i,"color_name"]
    return cname
# initialize the HOG descriptor/person detector
#def pdet():
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
cv2.startWindowThread()
# open webcam video stream
cap = cv2.VideoCapture(0)
# the output will be written to output.avi
out = cv2.VideoWriter(
    'output.avi',
    cv2.VideoWriter_fourcc(*'MJPG'), 15.,  (640,480))
```

```
while(True):
    # Capture frame-by-frame ret,
frame = cap.read()
    # resizing for faster detection
    frame = cv2.resize(frame, (640, 480))
    # using a greyscale picture, also for faster detection
    gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    # detect people in the image
    # returns the bounding boxes for the detected objects
    boxes, weights = hog.detectMultiScale(frame, winStride=(8,8) )
    print(boxes)
    boxes = np.array([[x, y, x + w, y + h] for (x, y, w, h) in boxes])
    for (xA, yA, xB, yB) in boxes:
        print ("xA", xA)
        print ("yA", yA)
        print ("x+wA", xB)
        print ("y+hA", yB)
        # display the detected boxes in the colour picture
        cv2.rectangle(frame, (xA, yA), (xB, yB),  (0, 255, 0), 2)
        roi=frame[yA:yB,xA:xB]
        cv2.imwrite("boxes.png",roi)
```

```
    #Selecting the midpoint in the video
    point = ((yB - yA) // 2 + yA, (xA - xB) // 2 + xA)
    b, g, r = frame[point]
        text = getColorName(r,g,b) + ' R='+ str(r) +  ' G='+ str(g) +  ' B='+ str(b)
    #cv2.putText(img,text,start,font(0-7),fontScale,color,thickness,lineType )
    cv2.putText(frame, text,(50,50),2,0.8,(255,255,255),2,cv2.LINE_AA)


    # Write the output video
    out.write(frame.astype('uint8'))
    # Display the resulting frame
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# When everything done, release the capture
cap.release()
# and release the output
out.release()
# finally, close the window
cv2.destroyAllWindows()
cv2.waitKey(1)
#    return frame
```

4.3 PATTERN DETERMINATION CODE

5.3.1 TRAINING CODE

5.3.1.1 MAIN CODE

```
# -*- coding: utf-8 -*-

"""

Created on Thu Mar 19 17:21:20 2020

@author: User

"""

# set the matplotlib backend so figures can be saved in the background

import matplotlib

matplotlib.use("Agg")

# import the necessary packages

from pyimagesearch.minivggnet import MiniVGGNet

from sklearn.metrics import classification_report

from keras.optimizers import SGD

from keras.datasets import fashion_mnist

from keras.utils import np_utils

from keras import backend as K

from imutils import build_montages

import matplotlib.pyplot as plt

import numpy as np

import cv2
```

```
# initialize the number of epochs to train for, base learning rate,
# and batch size
NUM_EPOCHS = 30
INIT_LR = 1e-2
BS = 32
# grab the Fashion MNIST dataset (if this is your first time running
# this the dataset will be automatically downloaded)
print("[INFO] loading Fashion MNIST...")
((trainX, trainY), (testX, testY)) = fashion_mnist.load_data()
# if we are using "channels first" ordering, then reshape the design
# matrix such that the matrix is:
#       num_samples x depth x rows x columns
if K.image_data_format() == "channels_first":
        trainX = trainX.reshape((trainX.shape[0], 1, 28, 28))
        testX = testX.reshape((testX.shape[0], 1, 28, 28))

# otherwise, we are using "channels last" ordering, so the design
# matrix shape should be: num_samples x rows x columns x depth
else:
        trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
        testX = testX.reshape((testX.shape[0], 28, 28, 1))
# scale data to the range of [0, 1]
```

```
trainX = trainX.astype("float32") / 255.0
testX = testX.astype("float32") / 255.0
# one-hot encode the training and testing labels
trainY = np_utils.to_categorical(trainY, 10)
testY = np_utils.to_categorical(testY, 10)
# initialize the label names
labelNames = ["top", "trouser", "pullover", "dress", "coat",
        "sandal", "shirt", "sneaker", "bag", "ankle boot"]
# initialize the optimizer and model
print("[INFO] compiling model...")
opt = SGD(lr=INIT_LR, momentum=0.9, decay=INIT_LR / NUM_EPOCHS)
model = MiniVGGNet.build(width=28, height=28, depth=1, classes=10)
model.compile(loss="categorical_crossentropy", optimizer=opt,
        metrics=["accuracy"])
# train the network
print("[INFO] training model...")
H = model.fit(trainX, trainY,
        validation_data=(testX, testY),
        batch_size=BS, epochs=NUM_EPOCHS)
model.save("pattern_classifier_3.h5")
preds = model.predict(testX)
# show a nicely formatted classification report
```

```
print("[INFO] evaluating network...")
print(classification_report(testY.argmax(axis=1), preds.argmax(axis=1),
	target_names=labelNames))
#model.save("pattern_classifier_3.h5")
# plot the training loss and accuracy
N = NUM_EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), H.history["acc"], label="train_acc")
plt.plot(np.arange(0, N), H.history["val_acc"], label="val_acc")
plt.title("Training Loss and Accuracy on Dataset")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="lower left")
plt.savefig("plot.png")

# initialize our list of output images
images = []
# randomly select a few testing fashion items
for i in np.random.choice(np.arange(0, len(testY)), size=(16,)):
```

```
        # classify the clothing
        probs = model.predict(testX[np.newaxis, i])
        prediction = probs.argmax(axis=1)
        label = labelNames[prediction[0]]

        # extract the image from the testData if using "channels_first"
        # ordering
        if K.image_data_format() == "channels_first":
                image = (testX[i][0] * 255).astype("uint8")

        # otherwise we are using "channels_last" ordering
        else:
                image = (testX[i] * 255).astype("uint8")

# initialize the text label color as green (correct)
        color = (0, 255, 0)
        # otherwise, the class label prediction is incorrect
        if prediction[0] != np.argmax(testY[i]):
                color = (0, 0, 255)

        # merge the channels into one image and resize the image from
        # 28x28 to 96x96 so we can better see it and then draw the
```

```
        # predicted label on the image
        image = cv2.merge([image] * 3)
        image = cv2.resize(image, (96, 96), interpolation=cv2.INTER_LINEAR)
        cv2.putText(image, label, (5, 20), cv2.FONT_HERSHEY_SIMPLEX,
0.75, color, 2)
        # add the image to our list of output images
        images.append(image)
# construct the montage for the images
montage = build_montages(images, (96, 96), (4, 4))[0]
# show the output montage
cv2.imshow("Fashion MNIST", montage)
cv2.waitKey(0)
```

5.3.1.2 MINIVIGGNET CODE

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 19 17:18:35 2020

@author: User
"""
# import the necessary packages
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
```

```
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation
from keras.layers.core import Flatten
from keras.layers.core import Dropout
from keras.layers.core import Dense
from keras import backend as K
class MiniVGGNet:
	@staticmethod
	def build(width, height, depth, classes):
		# initialize the model along with the input shape to be
		# "channels last" and the channels dimension itself
		model = Sequential()
		inputShape = (height, width, depth)
		chanDim = -1
		# if we are using "channels first", update the input shape
		# and channels dimension
		if K.image_data_format() == "channels_first":
			inputShape = (depth, height, width)
			chanDim = 1
	# first CONV => RELU => CONV => RELU => POOL layer set
		model.add(Conv2D(32, (3, 3), padding="same",
			input_shape=inputShape))
```

```
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(32, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
# second CONV => RELU => CONV => RELU => POOL layer set
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
# first (and only) set of FC => RELU layers
model.add(Flatten())
model.add(Dense(512))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.5))
```

```
        # softmax classifier
        model.add(Dense(classes))
        model.add(Activation("softmax"))
        # return the constructed network architecture
        return model
```

5.3.2 PREDICTION CODE

```
# -*- coding: utf-8 -*-
"""
Created on Thu March  2 11:45:59 2019

@author: user
"""
import numpy as np
from keras.preprocessing import image
from tkinter import *
from PIL import ImageTk, Image
from tkinter import filedialog
import os
import cv2
from keras.models import load_model

classifier = load_model('pattern_classifier_1.h5')
```

```
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

root = Tk()

root.geometry("550x300+300+150")

root.resizable(width=True, height=True)


def openfn():

    filename = filedialog.askopenfilename(title='open')

    return filename

def open_img():

    x = openfn()

    test_image = image.load_img(x, target_size = (28, 28))

    test_image = cv2.cvtColor(np.float32(test_image), cv2.COLOR_BGR2GRAY)

    test_image = image.img_to_array(test_image)

    test_image = np.expand_dims(test_image, axis = 0)

    result = classifier.predict_classes(test_image)

    print(result)

    index=["top", "trouser", "pullover", "dress", "coat",

        "sandal", "shirt", "sneaker", "bag", "ankle boot"]

    label = Label( root, text="Prediction : "+index[result[0]-1])

    label.pack()

    img = Image.open(x)

    img = img.resize((250, 250), Image.ANTIALIAS)
```

```
    img = ImageTk.PhotoImage(img)
    panel = Label(root, image=img)
    panel.image = img
    panel.pack()
    return result

btn = Button(root, text='open image', command=open_img).pack()

root.mainloop()
```

5.4 MERGED/ COMBINED PARAMETERS CODE

```
# import the necessary packages
import numpy as np
import pandas as pd
import math
import cv2
import numpy as np
from keras.preprocessing import image
from tkinter import *
from PIL import ImageTk, Image
from keras.models import load_model
x=0
y=0
```

```
w=0

h=0

xA=0

yA=0

xB=0

yB=0

global padding


classifier = load_model('pattern_classifier_3.h5')

classifier.compile(optimizer = 'adam',

loss = 'categorical_crossentropy', metrics = ['accuracy'])

index=["color","color_name","hex","R","G","B"]

csv = pd.read_csv('colors.csv', names=index, header=None)

def getColorName(R,G,B):

    minimum = 10000

    for i in range(len(csv)):

        d = abs(R- int(csv.loc[i,"R"])) + abs(G- int(csv.loc[i,"G"]))+ abs(B- int(csv.loc[i,"B"]))

        if(d<=minimum):

            minimum = d

            cname = csv.loc[i,"color_name"]

    return cname
```

```
def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn=frame.copy()
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)
    net.setInput(blob)
    detections=net.forward()
    faceBoxes=[]
    for i in range(detections.shape[2]):
        confidence=detections[0,0,i,2]
        if confidence>conf_threshold:
            x1=int(detections[0,0,i,3]*frameWidth)
            y1=int(detections[0,0,i,4]*frameHeight)
            x2=int(detections[0,0,i,5]*frameWidth)
            y2=int(detections[0,0,i,6]*frameHeight)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0), int(round(frameHeight/150)), 8)
    return frameOpencvDnn,faceBoxes
faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
ageProto="age_deploy.prototxt"
```

```
ageModel="age_net.caffemodel"

genderProto="gender_deploy.prototxt"

genderModel="gender_net.caffemodel"


MODEL_MEAN_VALUES=(78.4263377603,87.7689143744, 114.895847746)

ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']

genderList=['Male','Female']

faceNet=cv2.dnn.readNet(faceModel,faceProto)

ageNet=cv2.dnn.readNet(ageModel,ageProto)

genderNet=cv2.dnn.readNet(genderModel,genderProto)


# initialize the HOG descriptor/person detector

#def pdet():

hog = cv2.HOGDescriptor()

hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

cv2.startWindowThread()


# open webcam video stream

cap = cv2.VideoCapture(0)


# the output will be written to output.avi

out = cv2.VideoWriter(
```

```
    'output.avi',
    cv2.VideoWriter_fourcc(*'MJPG'),
    15.,
    (640,480))
padding=20
while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()
    # resizing for faster detection
    frame = cv2.resize(frame, (640, 480))
    # using a greyscale picture, also for faster detection
    gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    # detect people in the image
    # returns the bounding boxes for the detected objects
    boxes, weights = hog.detectMultiScale(frame, winStride=(8,8) )
    print(boxes)
    boxes = np.array([[x, y, x + w, y + h] for (x, y, w, h) in boxes])
    for (xA, yA, xB, yB) in boxes:
        print ("xA", xA)
        print ("yA", yA)
        print ("x+wA", xB)
        print ("y+hA", yB)
```

```
    # display the detected boxes in the colour picture
    cv2.rectangle(frame, (xA, yA), (xB, yB),
                    (0, 255, 0), 2)
    roi=frame[yA:yB,xA:xB]
    cv2.imwrite("boxes.png",roi)
#Selecting the midpoint in the video
point = ((yB - yA) // 2 + yA, (xA - xB) // 2 + xA)
b, g, r = frame[point]
text = getColorName(r,g,b) + ' R='+ str(r) +  ' G='+ str(g) +  ' B='+ str(b)
#cv2.putText(img,text,start,font(0-7),fontScale,color,thickness,lineType )
cv2.putText(frame, text,(50,50),2,0.8,(255,255,255),2,cv2.LINE_AA)
test_image = image.load_img('boxes.png', target_size = (28, 28))
test_image = cv2.cvtColor(np.float32(test_image), cv2.COLOR_BGR2GRAY)
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict_classes(test_image)
print(result)
index=["top", "trouser", "pullover", "dress", "coat",
     "sandal", "shirt", "sneaker", "bag", "ankle boot"]
```

```
    cv2.putText(frame,str(index[result[0]-
1]),(xA,yA),2,0.8,(255,255,255),2,cv2.LINE_AA)


    frame,faceBoxes=highlightFace(faceNet,frame)
    if not faceBoxes:
        print("No face detected")


    for faceBox in faceBoxes:
        face=frame[max(0,faceBox[1]-padding):
                   min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-
padding)
                   :min(faceBox[2]+padding, frame.shape[1]-1)]


        blob=cv2.dnn.blobFromImage(face, 1.0, (227,227),
MODEL_MEAN_VALUES, swapRB=False)
        genderNet.setInput(blob)
        genderPreds=genderNet.forward()
        gender=genderList[genderPreds[0].argmax()]
        print(f'Gender: {gender}')


        ageNet.setInput(blob)
        agePreds=ageNet.forward()
        age=ageList[agePreds[0].argmax()]
```

```
        print(f'Age: {age[1:-1]} years')


        cv2.putText(frame, f'{gender}, {age}', (faceBox[0], faceBox[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE_AA)
        #cv2.imshow("Detecting age and gender", frame)


    # Write the output video
    out.write(frame.astype('uint8'))
    # Display the resulting frame


    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break


# When everything done, release the capture
cap.release()
# and release the output
#out.release()
# finally, close the window
cv2.destroyAllWindows()
cv2.waitKey(1)
#    return frame
```

CHAPTER-6

RESULTS

6.1 RESULTS OF AGE AND GENDER

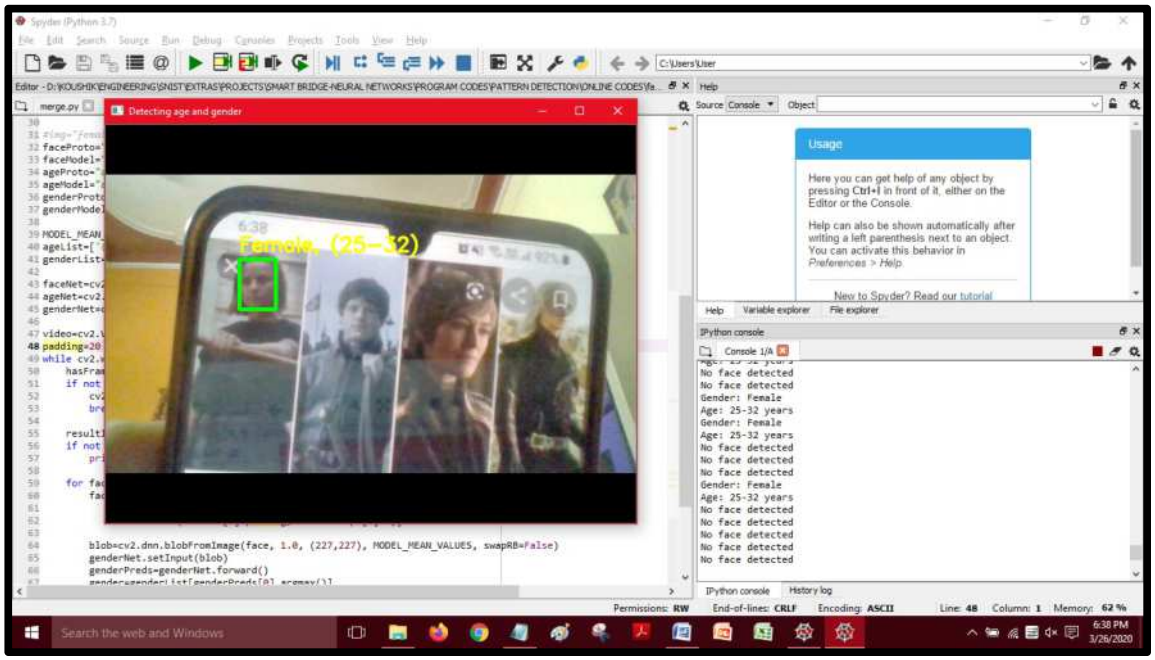


Figure: 5.1 Age and Gender output-1

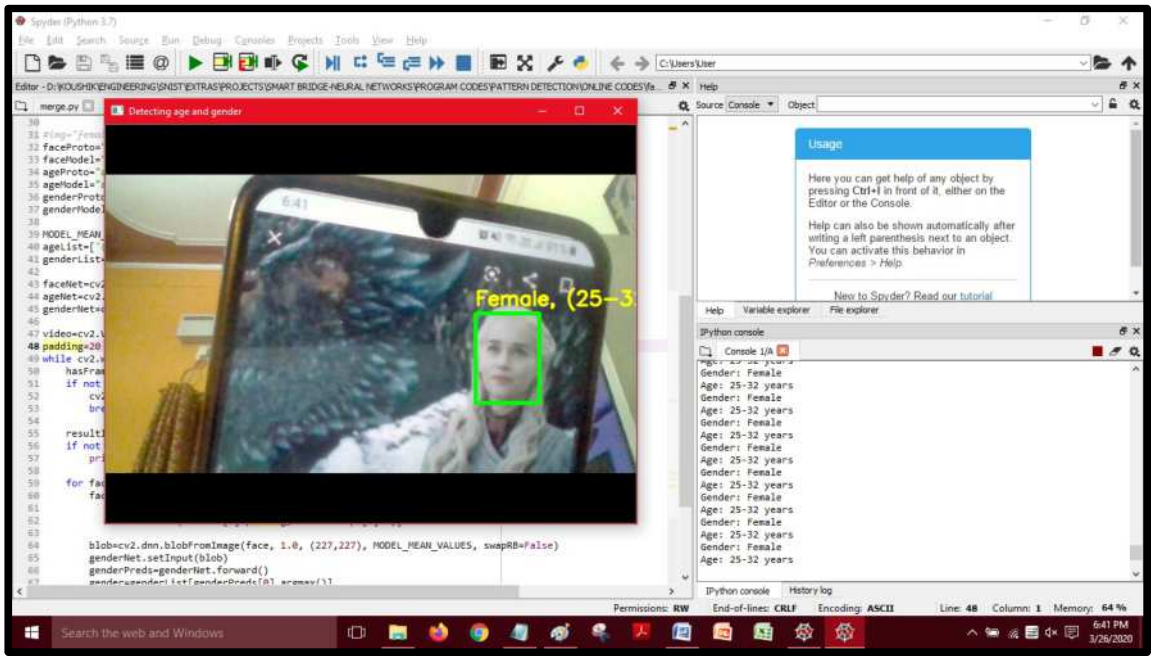


Figure 5.2 Age and Gender Ouput-2

6.2 RESULTS OF COLOR DETERMINATION

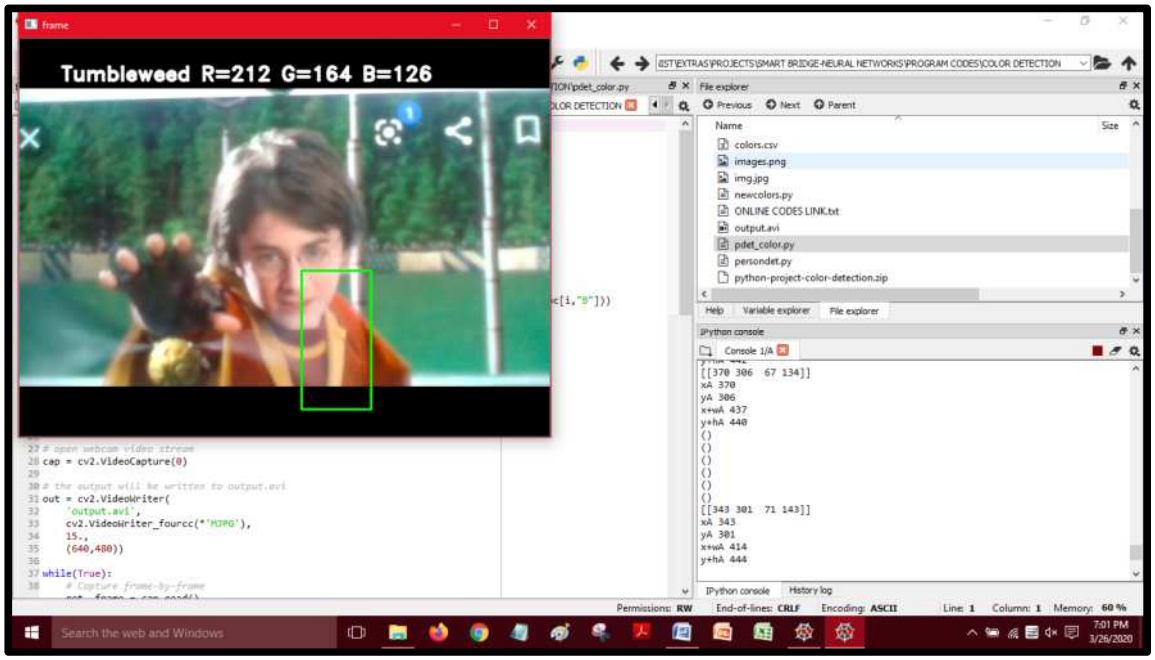


Figure 5.3: Colour Determination Ouput-1.

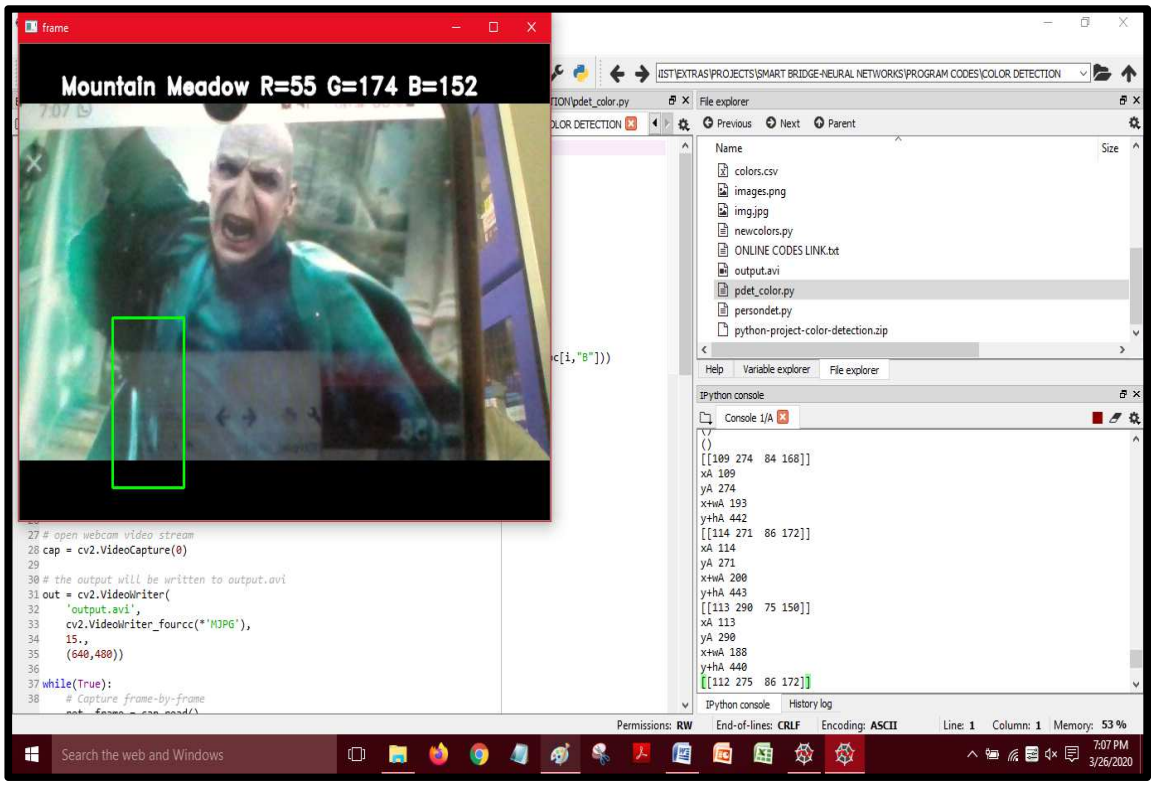


Figure 5.4: Color Determination output-2

6.3 RESULTS OF PATTERN DETERMINATION CODE

Figure 5.5: Pattern Input Image-1

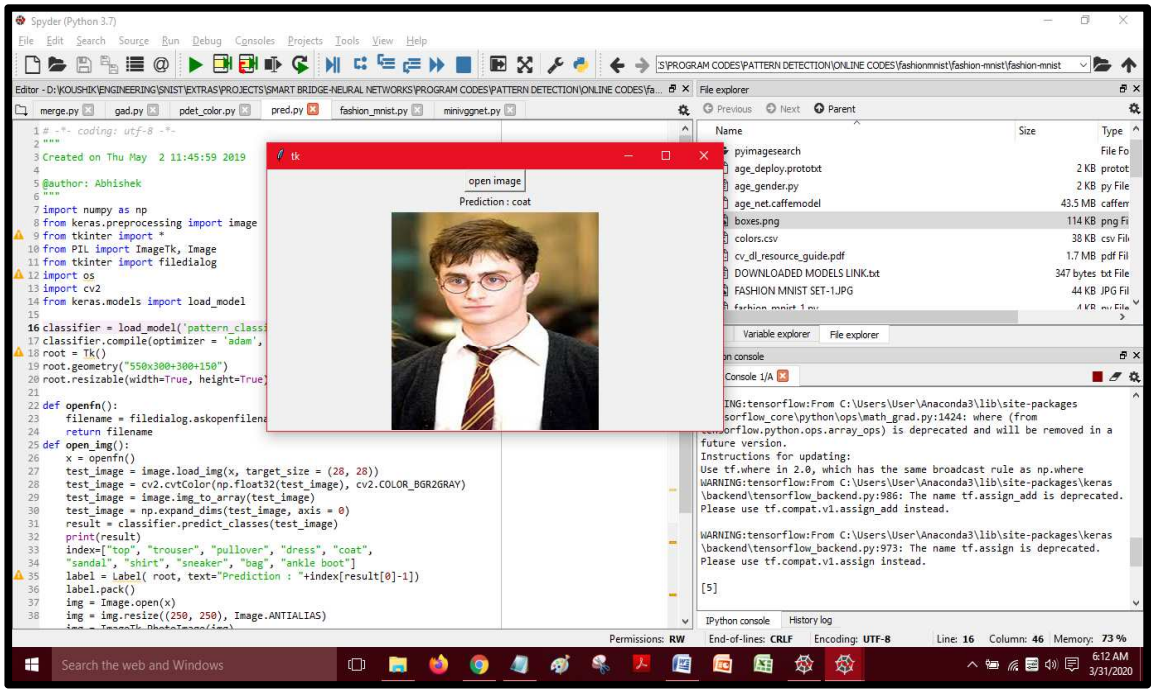


Figure 5.6: Pattern Determination Output-2

Figure 5.7: Pattern Input Image-2

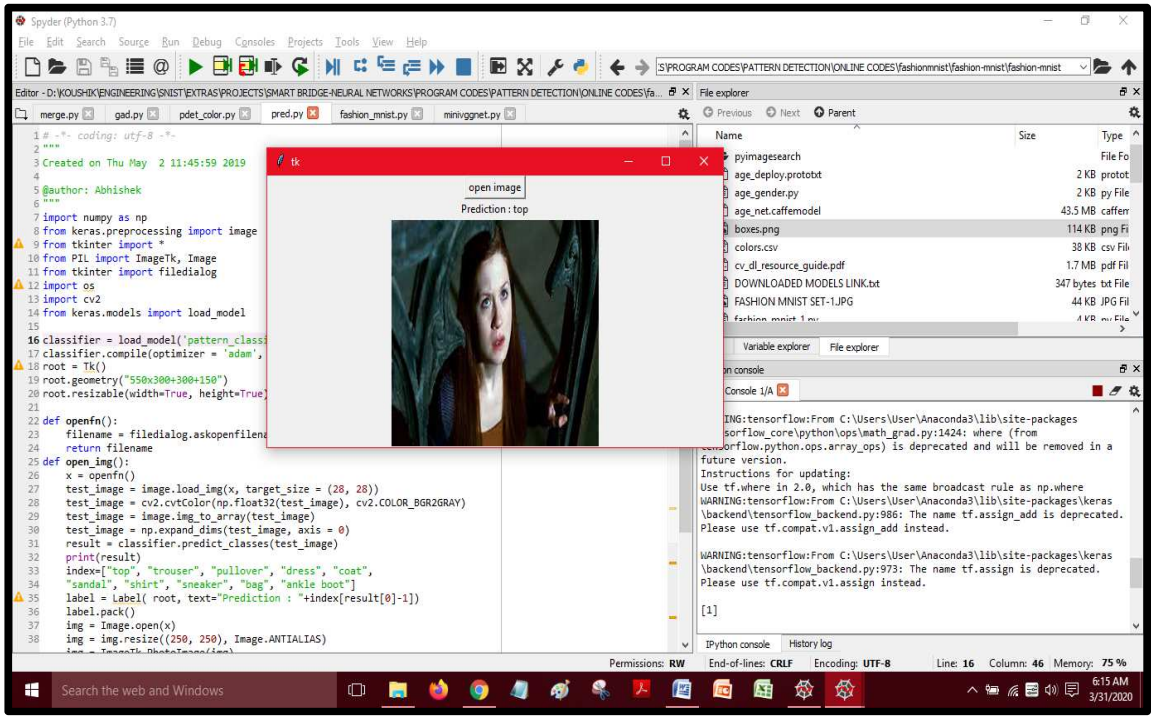


Figure 5.8: Pattern Determination Output-2

6.4 MERGED/ COMBINED PARAMETERS OUTPUTS

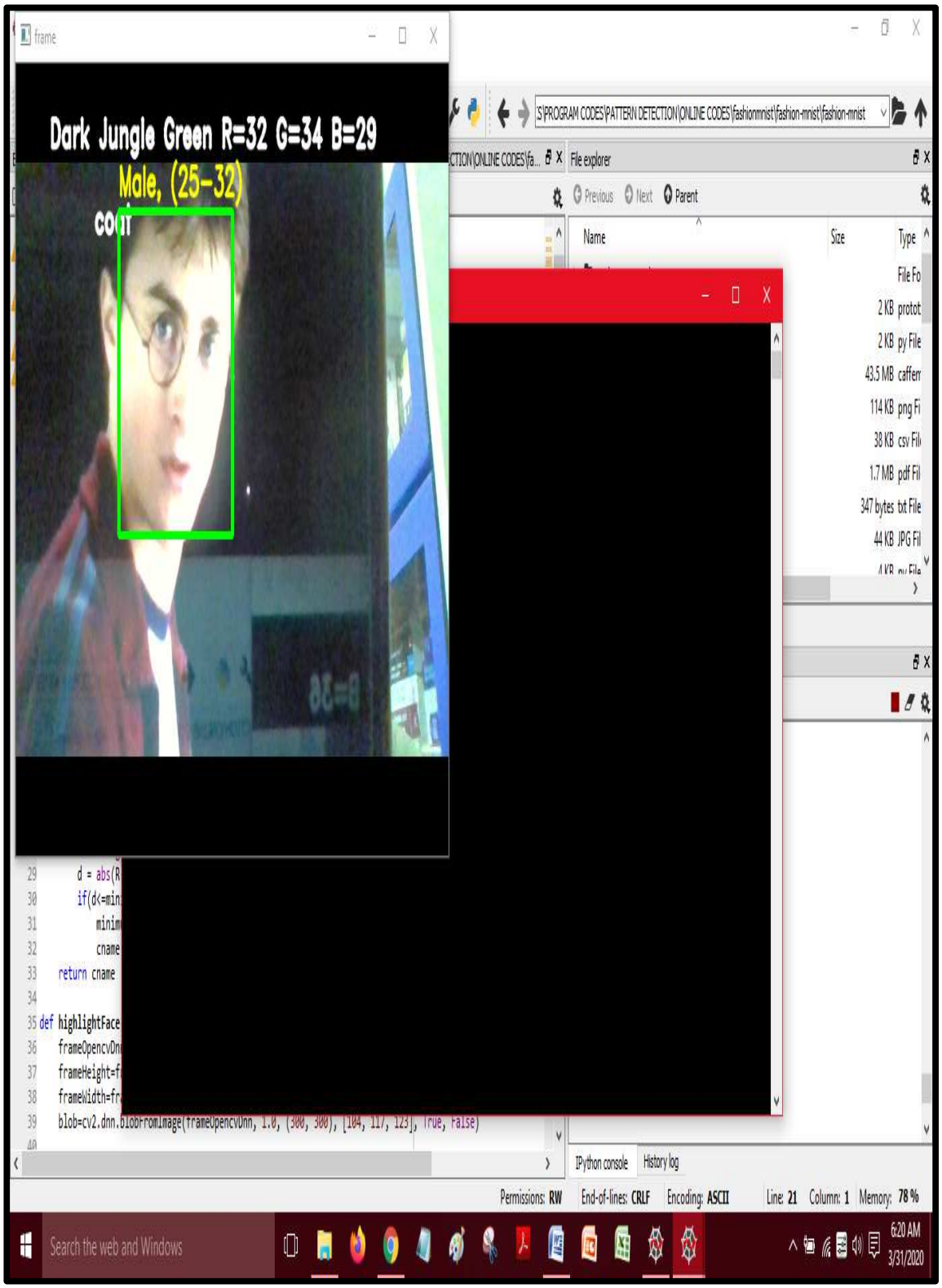


Figure 5.9: Combined Output-1

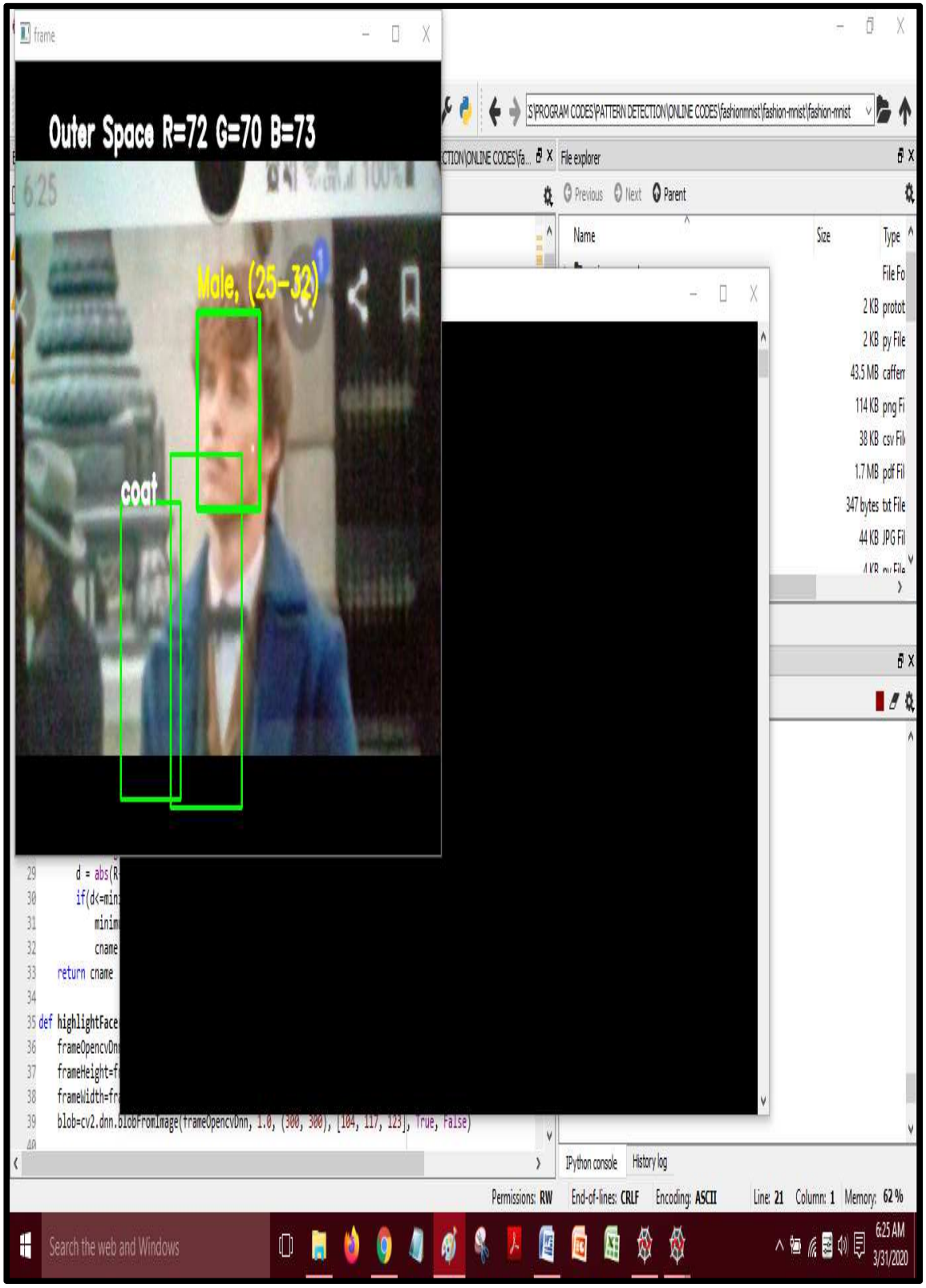


Figure 5.10: Combined Output-2

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The Gender Classification and Age detection algorithms may be applied with an improved range of facial image records set. This will boom the accuracy degree of the output. The Proposed Models had been educated and examined on information units using Linear Kernel. However the similar evaluation can be done the usage of different kernels of SVM Classifier like 'rbf kernel', 'quadratic kernel', and so on. Gender Classification is executed based at the extracted feature' lip'. The same set of rules is can be implemented on different function(s) like eyes, nose or combination of more than one function in human facial image.

7.2 FUTURE SCOPE

It presents a brand new shade image segmentation algorithm primarily based on colour similarity in real-time colouration image segmentation for cyber bodily structures. We first determine the dominant colour. And then, we use a mathematical model known as SIMULATION, which takes the hue and the brightness under consideration at the same time to calculate the colouration similarity inside the RGB colour space. After that, we integrate the proposed methods of calculation of the photo colour additives to shape a shade map. Furthermore, pixels are divided based totally on colour map and the segmentation is finished. Since the past 30 years, pattern recognition had tremendous considerable growth with applications like the character, target detections, medical diagnosis, biomedical signal and image analysis, remote sensing etc.

CHAPTER-8

REFERENCES

[1] Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In Proc. Conf. Comput. Vision Pattern Recognition, pages 1725–1732. IEEE, 2014

[2] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2014)

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.

[4] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, in: NIPS, 2015, pp. 2017–2025.

[5] Mingxing Duan , Kenli Li , Canqun Yang , Keqin Li, A hybrid deep learning CNN–ELM for age and gender classification, M. Duan et al. / Neurocomputing 275 (2018) 448–461, http://dx.doi.org/10.1016/j.neucom.2017.08.062 0925-2312/© 2017 Elsevier B.V.

[6] E. Eidinger , R. Enbar , T. Hassner , Age and gender estimation of unfiltered faces, IEEE Trans. Inf. Forensics Secur. 9 (12) (2014) 2170–2179 .

[7] G. Levi , T. Hassncer , Age and gender classification using convolutional neural networks, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 34–42 .

[8] Y.H. Kwon, N. da Vitoria Lobo, Age classification from facial images, in: Pro- ceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. CVPR'94, 1994, pp. 762–767, doi: 10.1109/CVPR. 1994.323894 .

[9] T.R. Kalansuriya , A.T. Dharmaratne , Facial image classification based on age and gender, in: Proceedings of the 2013 International

Conference on Advances in ICT for Emerging Regions (ICTer), 2013, pp. 44–50 .

[10] X.X. Niu , C.Y. Suen , A novel hybrid CNN–SVM classifier for recognizing hand- written digits, Pattern Recogn. 45 (4) (2012) 1318–1325 .

[11] S.B. Kim , K.S. Han , H.C. Rim , S.H. Myaeng , Some effective techniques for naive Bayes text classification, IEEE Trans. Knowl. Data Eng. 18 (11) (2006) 1457–1466 .

[12] G.B. Huang , Q.Y. Zhu , C.K. Siew , Extreme learning machine: theory and appli- cations, Neurocomputing 70 (1–3) (2006) 489–501 . [9] F.S. Khan, J. van de Weijer, R.M. Anwer, M. Felsberg, C. Gatta, Semantic pyra- mids for gender and action recognition, IEEE Trans. Image Process. 23 (8) (2014) 3633–3645, doi: 10.1109/TIP.2014.2331759 .

[13] H. Guang-Bin , C. Lei , S. Chee-Kheong , Universal approximation using incremen- tal constructive feedforward networks with random hidden nodes., IEEE Trans. Neural Netw. 17 (4) (2006) 879–892 .

[14] S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back, Face recognition: a convolutional neural-network approach, IEEE Trans. Neural Netw. 8 (1) (1997) 98–113, doi: 10. 1109/72.554195 .

[15] F. Liu, G. Lin, C. Shen, {CRF} learning with {CNN} features for image segmen- tation, Pattern Recogn. 48 (10) (2015) 2983–2992. http://dx.doi.org/10.1016/j. patcog.2015.04.019 . DiscriminativeFeatureLearningfromBigDataforVisual Recognition 460 *M. Duan et al. / Neurocomputing 275 (2018) 448–461.*

[16] G.S. Xie, X.Y. Zhang, S. Yan, C.L. Liu, Hybrid CNN and dictionary-based mod- els for scene recognition and domain adaptation, IEEE Trans. Circ. Syst. Video Technol. PP (99) (2015) 1, doi: 10.1109/TCSVT.2015.2511543 .

[17] J. Tang, C. Deng, G.B. Huang, B. Zhao, Compressed-domain ship detection on spaceborne optical image using deep neural network

and extreme learning machine, IEEE Trans. Geosci. Remote Sens. 53 (3) (2015) 1174–1185, doi: 10. 1109/TGRS.2014.2335751

[18] S. Fu, H. He, Z.G. Hou, Learning race from face: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (12) (2014) 2483–2509, doi: 10.1109/TPAMI.2014.2321570 . [16] Y. Fu, G. Guo, T.S. Huang, Age synthesis and estimation via faces: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 1955–1976, doi: 10.1109/TPAMI. 2010.36 .

[19] N. Ramanathan, R. Chellappa, Modeling age progression in young faces, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vi- sion and Pattern Recognition, 1, 2006, pp. 387–394, doi: 10.1109/CVPR.2006. 187 . [18] X. Geng, Z.H. Zhou, K. Smith-Miles, Automatic age estimation based on facial aging patterns, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2234–2240, doi: 10.1109/TPAMI.2007.70733 .

[20] G. Guo, Y. Fu, C.R. Dyer, T.S. Huang, Image-based human age estimation by manifold learning and locally adjusted robust regression, IEEE Trans. Image Process. 17 (7) (2008) 1178–1188, doi: 10.1109/TIP.2008.924280 . [20] K.Y. Chang, C.S. Chen, A learning framework for age rank estimation based on face images with scattering transform, IEEE Trans. Image Process. 24 (3) (2015) 785–798, doi: 10.1109/TIP.2014.2387379 .

[21] C. Li, Q. Liu, W. Dong, X. Zhu, J. Liu, H. Lu, Human age estimation based on locality and ordinal information, IEEE Trans. Cybern. 45 (11) (2015) 2522–2534, doi: 10.1109/TCYB.2014.2376517 .

[22] A. Lanitis, The FG-net aging database, 2002 http://www-prima.inrialpes.fr/ FGnet/html/benchmarks.html .

[23] K. Ricanek, T. Tesafaye, Morph: a longitudinal image database of normal adult age-progression, in: Proceedings of the Seventh International Conference on Automatic Face and Gesture Recognition, 2006. FGR 2006, 2006, pp. 341–345, doi: 10.1109/FGR.2006.78 .

[24] G. Guo, X. Wang, A study on human age estimation under facial expression changes, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2547–2553, doi: 10.1109/CVPR.2012. 6247972 .

[25] E. Makinen, R. Raisamo, Evaluation of gender classification methods with au- tomatically detected and aligned faces, IEEE Trans. Pattern Anal. Mach. Intell. 30 (3) (2008) 541–547, doi: 10.1109/TPAMI.2007.70800 .

[26] J. Bekios-Calfa, J.M. Buenaposada, L. Baumela, Revisiting linear discriminant techniques in gender recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (4) (2011) 858–864, doi: 10.1109/TPAMI.2010.208 .

[27] H. Chen, A. Gallagher, B. Girod, Face modeling with first name attributes, IEEE Trans. Pattern Anal. Mach. Intell. 36 (9) (2014) 1860–1873, doi: 10.1109/TPAMI. 2014.2302443 .

[28] H. Han, C. Otto, X. Liu, A.K. Jain, Demographic estimation from face images: human vs. machine performance, IEEE Trans. Pattern Anal. Mach. Intell. 37 (6) (2015) 1148–1161, doi: 10.1109/TPAMI.2014.2362759 .

[29] Y. Lcun , L. Bottou , Y. Bengio , P. Haffner , Gradient-based learning applied to doc- ument recognition, Proc. IEEE 86 (11) (1998) 2278–2324 . [35] F. Jialue , X. Wei , W. Ying , G. Yihong , Human tracking using convolutional neu- ral networks, IEEE Trans. Neural Netw. 21 (10) (2010) 1610–1623 .

[30] Y. Cao , Y. Chen , D. Khosla ,Spiking deep convolutional neural networks for energy-efficient object recognition, Int. J. Comput. Vis. 113 (1) (2015) 54–66 .

[31] M. Duan, K. Li, X. Liao, K. Li, A parallel multiclassification algorithm for big data using an extreme learning machine, IEEE Trans. Neural Netw. Learn. Syst. PP (99) (2017) 1–15, doi: 10.1109/TNNLS.2017.2654357 .

[32] B. Zuo , G.B. Huang , D. Wang , W. Han , M.B. Westover , Sparse extreme learning machine for classification, IEEE Trans.

Cybern. 44 (10) (2014) 1858–1870 . [39] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regres- sion and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2) (2012) 513–529, doi: 10.1109/TSMCB.2011.2168604 .

[33] Y. Yang , Q.M. Wu , Y. Wang , K.M. Zeeshan , X. Lin , X. Yuan , Data partition learn- ing with multiple extreme learning machines, IEEE Trans. Cybern. 45 (6) (2014) 1463–1475 .

[34] J. Luo , C.M. Vong , P.K. Wong , Sparse Bayesian extreme learning machine for multi-classification., IEEE Trans. Neural Netw. Learn. Syst. 25 (4) (2014) 836–843 .

[35] Pedro D. Marrero Fernández, Fidel A. Guerrero Peña, Tsang Ing Ren, Jorge J.G. Leandro , Fast and Robust Multiple ColorChecker Detection using Deep Convolutional Neural Networks. Imavis (2018), https://doi.org/10.1016/j.imavis.2018.11.001

[36] A. Ernst, A. Papst, T. Ruf, J.-U. Garbas, Check My Chart: A Robust Color Chart Tracker for Colorimetric Camera Calibration, in: 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications, 2013.

[37] T. Tajbakhsh, R.-R. Grigat, Semiautomatic Color Checker Detection in Distorted Images, in: 5th IASTED International Conference on Signal Processing, Pattern Recognition and Applications, 2008, pp. 347–352.

[38] D. Kapusi, P. Prinke, R. Jahn, D. Vehar, R. Nestler, K.-H. Franke, Simultaneous geometric and colorimetric camera calibration, in: Tagungsband 16. Workshop Farbbildverarbeitung, 2010.

[39] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, 2008.

[40] S. Bianco, C. Cusano, Color Target Localization Under Varying Illumination Conditions, in: 3rd International Workshop on Computational Color Imaging, 2011, pp. 245–255.

[41] C. Schwarz, J. Teich, A. Vainshtein, E. Welzl, B. L. Evans, Minimal enclosing parallelogram with application, in: 11th Annual Symposium on Computational Geometry, 1995, pp. 434–435.

[42] A. Kordecki, H. Palus, Automatic detection of colour charts in images, Przeglad Elektrotechniczny 90 (9) (2014) 197–202.

[43] T. A. Le, A. G. Baydin, R. Zinkov, F. Wood, Using synthetic data to train neural networks is model-based reasoning, in: International Joint Conference on Neural Networks, (IJCNN 2017), 2017, pp. 3514–3521.

[44] P. Y. Simard, D. Steinkraus, J. C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: 7th International Conference on Document Analysis and Recognition, 2003, pp. 958–963.

[45] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[46] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition, arXiv:1406.2227v4 [cs.CV] (2014) 1–10.

[47] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading Text in the Wild with Convolutional Neural Networks, International Journal of Computer Vision 116 (1) (2016) 1–20.

[48] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic Data for Text Localisation in Natural Images, in: IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2016), 2016, pp. 2315–2324.

[49] I. Goodfellow, Y. Bulatov, J. Ibarz, Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks, arXiv:1312.6082v4 [cs.CV] (2014) 1–13.

[50] L. V. Ahn, B. Maurer, C. Mcmillen, D. Abraham, M. Blum, re-CAPTCHA: Human-Based Character Recognition via Web Security Measures, Science 321 (12) (2008) 1465–1468.

[51] F. Stark, R. Triebel, D. Cremers, CAPTCHA Recognition with Active Deep Learning, in: 37th German Conference on Pattern Recognition, 2015.

[52] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, T. S. Huang, DeepFont: Identify Your Font from An Image, in: 23rd ACM International Conference on Multimedia, 2015, pp. 451–459.

[53] S. J. Gershman, N. D. Goodman, Amortized Inference in Probabilistic Reasoning, in: 36th Annual Conference of the Cognitive Science Society, (CogSci 2014), 2014, pp. 517–522.

[54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2015), 2015, pp. 1–9.

[55] U. Ramer, An Iterative Procedure for the Polygonal Approximation of Plane Curves, Computer Graphics and Image Processing 1 (3) (1972) 244–256.

[56] D. H. Douglas, T. K. Peucker, Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 10 (2) (1973) 112–122.

[57] R. J. Gil-Garcia, J. M. Badia-Contelles, A. Pons-Porrata, A general framework for agglomerative hierarchical clustering algorithms, in: IEEE 18th International Conference on Pattern Recognition, (ICPR 2006), 2006, pp. 569–572.

[58] P. V. Gehler, C. Rother, A. Blake, T. Minka, T. Sharp, Bayesian color constancy revisited, in: IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2008), 2008, pp. 1–8.

[59] Sarwo, Yaya Heryadi, Widodo Budiharto, and Edi Abdurachman, Logo and Brand Recognition from Imbalanced Dataset Using MiniGoogLeNet and MiniVGGNet Models, © Springer Nature Switzerland AG 2019 N. T. Nguyen et al. (Eds.): ACIIDS 2019, LNAI 11431, pp. 385–393, 2019. https://doi.org/10.1007/978-3-030-14799-0_33.

[60] Samuel, A.L.: Some studies in machine learning using the game of checkers. IBM J. Res. Dev. 3(3), 210–229 (1959).

[61] Hinton, G.E., Osindero, S., Teh, Y.-W.: Communicated by Yann Le Cun A fast learning algorithm for deep belief nets 500 units 500 units. Neural Comput. 18, 1527–1554 (2006).

[62] Bianco, S., Buzzelli, M., Mazzini, D., Schettini, R.: Deep learning for logo recognition. Neurocomputing 245, 23–30 (2017)

[63] Es-Sabry, M., El Akkad, N., Merras, M., Saaidi, A., Satori, K.: Grayscale image encryption using shift bits operations. In: 2018 International Conference on Intelligent System Computer Vision, ISCV 2018, no. Figure 3, May 2018.

[64] Allawi, S.T., Al-A'Meri, J.H.: Image encryption based on linear feedback shift register method. In: Al-Sadiq International Conference on multidisciplinary IT Communication Technical Science and Application, AIC-MITCSA 2016, pp. 16–19 (2016).

[65] Wadi, S.M., Zainal, N., Abdulgader, A.: Grey scale image hiding method based on decomposition operation. In: Proceeding - 2013 IEEE Student Conference on Research and Development, SCOReD 2013, pp. 315–318, December 2015.

[66] Mohanty, M., Asghar, M.R., Russello, G.: 2DCrypt: image scaling and cropping in encrypted domains. IEEE Trans. Inf. Forensics Secur. 11(11), 2542–2555 (2016) .

[67] Owen, C.B., Makedon, F.: High quality alias free image rotation. In: Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 115–119 (1997)

[68] Barnett, S.M., Zambrini, R.: Resolution in rotation measurements. J. Mod. Opt. 53(5–6), 613–625 (2006).

[69] Su, H., Zhu, X., Gong, S.: Deep Learning Logo Detection with Data Expansion by Synthesising Context. CoRR, vol. abs/1612.0 (2016).

[70] Kalantidis, Y., Pueyo, L.G., Trevisiol, M., van Zwol, R., Avrithis, Y.: Scalable triangulation based logo recognition (Flickr 27). In: Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR 2011), pp. 20:1–20:7 (2011)

[71] Romberg, S., Clara, S.: Scalable logo recognition in real-world Images categories and subject descriptors.

[72] Revaud, J., Schmid, C.: Correlation-Based Burstiness for Logo Retrieval Categories and Subject Descriptors: Keywords, pp. 965–968 (2012).

[73] Revaud, J., et al.: DeepMatching: Hierarchical Deformable Dense Matching To cite this version: DeepMatching: Hierarchical Deformable Dense Matching (2015)

[74] Romberg, S., Lienhart, R.: Bundle min-hashing for logo recognition. ACM (2013)

[75] Boia, R., Bandrabur, A., Florea, C.: Local description using multi-scale complete rank transform for improved logo recognition, pp. 4–7 (2014).

[76] Li, K., Chen, S., Su, S.: Logo detection with extendibility and discrimination (2013).

[77] Zhang, Y., et al.: Deep learning for logo recognition. In: International Conference on Intelligent Systems Design and Applications, ISDA, vol. 245, no. 36, pp. 2051–2054 (2017)

[78] Arivazhagan, N.: Logo Recognition. pp. 1–2

[79] Su, H., Zhu, X., Gong, S.: Deep learning logo detection with data expansion by synthesizing context. In: Proceedings of the 2017 IEEE Winter Conference Applications of Computer Vision (WACV 2017), pp. 530–539 (2017).

[80] Pan, C., Yan, Z., Xu, X., Sun, M., Shao, J., Wu, D.: Learning architecture in video

surveillance. pp. 123–126. IEEE (2013).

[81] Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1–9, 07–12 June 2015.

[82] Simonyan, K. Zisserman, A.: Very deep convolutional networks for large-scale image

recognition. In: International Conference on Learning Representations, pp. 1–14 (2015)

[83] He, Z.: Deep residual learning for image recognition. arXiv.org e-Print Arch 7(3), 171–180 (2015)

[84] He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe,

[85] B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. ECCV 2016. LNCS, vol. 9908. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38.

[86] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, January 2017, pp. 5987–5995 (2017)

[87] Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift (2015).

[88] Pyimagesearch: Deep Learning for computer Vision using Python. Book (2017)

[89] Liu, W., et al.: SSD: single shot multibox detector. arXiv, pp. 1–15 (2016)

[90] Girshick, R.: Fast R-CNN. arXiv.org e-Print Arch. (2015)

[91] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. Nips, pp. 1–10 (2015)

[92] Impiombato, D., et al.: You only look once: unified, real-time object detection. Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip. 794, 185–192 (2015)

[93] Soft Computing; Researchers from Fukuoka Institute of Technology Report New Studies and Findings in the Area of Soft Computing (A Vegetable

Category Recognition System: a Comparison Study for Caffe and Chainer Dnn Frameworks)[J]. Computers, Networks & Communications,2019.

[94] Caffe, a deep learning framework developed by the Berkeley Vision and Learning Center (BVLC)[OL].http: //caffe.berkeleyvision.org/.

[95] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). *Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET).* doi:10.1109/icengtechnol.2017.8308186

[96] Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., … Lungren, M. P. (2018). *Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. Artificial Intelligence in Medicine.* doi:10.1016/j.artmed.2018.11.004

[97] Nain, N., Vipparthi, S. K., & Raman, B. (Eds.). (2020). *Computer Vision and Image Processing. Communications in Computer and Information Science.* doi:10.1007/978-981-15-4018-9

[98] Kayed, M., Anter, A., & Mohamed, H. (2020). *Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture. 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE).* doi:10.1109/itce48509.2020.9047776

[99] Seo, Y., & Shin, K. (2018). Hierarchical Convolutional Neural Networks for Fashion Image Classification. Expert Systems with Applications. doi:10.1016/j.eswa.2018.09.022

[93] Soft Computing; Researchers from Fukuoka Institute of Technology Report New Studies and Findings in the Area of Soft Computing (A Vegetable Category Recognition System: a Comparison Study for Caffe and Chainer Dnn Frameworks)[J]. Computers, Networks & Communications,2019.

[94] Caffe, a deep learning framework developed by the Berkeley Vision and Learning Center (BVLC)[OL].http: //caffe.berkeleyvision.org/.

[95] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). *Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET).* doi:10.1109/icengtechnol.2017.8308186

[96] Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., … Lungren, M. P. (2018). *Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. Artificial Intelligence in Medicine.* doi:10.1016/j.artmed.2018.11.004

[97] Nain, N., Vipparthi, S. K., & Raman, B. (Eds.). (2020). *Computer Vision and Image Processing. Communications in Computer and Information Science.* doi:10.1007/978-981-15-4018-9

[98] Kayed, M., Anter, A., & Mohamed, H. (2020). *Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture. 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE).* doi:10.1109/itce48509.2020.9047776

[99] Seo, Y., & Shin, K. (2018). Hierarchical Convolutional Neural Networks for Fashion Image Classification. Expert Systems with Applications. doi:10.1016/j.eswa.2018.09.022

[100] Sarwo, Heryadi, Y., Budiharto, W., & Abdurachman, E. (2019). *Logo and Brand Recognition from Imbalanced Dataset Using MiniGoogLeNet and MiniVGGNet Models. Lecture Notes in Computer Science, 385–393.* doi:10.1007/978-3-030-14799-0_33

[101] Abel Kahsay Gebreslassie Maarig Aregawi Hagos, Achim Ibenthal ,Eneyew Adugna Etsub DOI https://doi.org/10.1007/978-3-030-26630-1_5

[102] Caffe, a deep learning framework developed by the Berkeley Vision and Learning Center (BVLC)[OL].http: //caffe.berkeleyvision.org/.

[103] Soft Computing; Researchers from Fukuoka Institute of Technology Report New Studies and Findings in the Area of Soft Computing (A Vegetable

Category Recognition System: a Comparison Study for Caffe and Chainer Dnn Frameworks)[J]. Computers, Networks & Communications,2019.

[104] Ziyang Han, He Huang, and Jianlin Wang School of Electronics and Information Engineering, Soochow University, Suzhou 215006, P. R. China https://doi.org/10.1109/ICCSS.2018.8572342

[105] Di Tommaso, G., Faralli, S., Stilo, G., & Velardi, P. (2018). *Wiki-MID: A Very Large Multi-domain Interests Dataset of Twitter Users with Mappings to Wikipedia. The Semantic Web – ISWC 2018, 36–52.* doi:10.1007/978-3-030-00668-6_3

[106] Xiaocong Lian , Member, IEEE, Zhenyu Liu, Member, IEEE, Zhourui Song, Jiwu Dai, Wei Zhou , Member, IEEE, and Xiangyang Ji , Member, IEEE High-Performance FPGA-Based CNN Accelerator With Block-Floating-Point Arithmetic

[107] Rohit Gandikota , Deepak Mishra Indian Institute of Space science and Technology .How You See Me: Understanding Convolutional Neural Networks

[108] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing, 115, 213–237. doi:10.1016/j.ymssp.2018.05.050

[109] Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2018). Deep learning for sensor-based activity recognition: A Survey. Pattern Recognition Letters. doi:10.1016/j.patrec.2018.02.010

[110] Zhang, C., Patras, P., & Haddadi, H. (2019). Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Communications Surveys & Tutorials, 1–1. doi:10.1109/comst.2019.2904897

[111] Payne, J. R. (2019). Introduction to Computer Programming and Python. Python for *Teenagers, 1–16.* doi:10.1007/978-1-4842-4550-7_1

PP-2-PROJECT REPORT WITHOUT CODE

ORIGINALITY REPORT

12%	6%	3%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Technology, Madras Student Paper	1%
2	www.geeksforgeeks.org Internet Source	1%
3	en.wikipedia.org Internet Source	1%
4	aicip.eecs.utk.edu Internet Source	1%
5	Submitted to Fordham University Student Paper	1%
6	Submitted to Delhi Technological University Student Paper	<1%
7	pypi.org Internet Source	<1%
8	Submitted to University of Sydney Student Paper	<1%
9	Submitted to CSU, San Jose State University	

	Student Paper	<1%
10	Submitted to University of Lancaster Student Paper	<1%
11	Submitted to University College London Student Paper	<1%
12	Submitted to General Sir John Kotelawala Defence University Student Paper	<1%
13	link.springer.com Internet Source	<1%
14	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1%
15	Rajeev Kumar, Ritu Vijay, Surjit Singh. "chapter 20 Designing and Analysis of Antenna Using Back Propagation Network", IGI Global, 2019 Publication	<1%
16	Submitted to ABV-Indian Institute of Information Technology and Management Gwalior Student Paper	<1%
17	Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi. "Understanding of a convolutional neural network", 2017 International Conference on Engineering and Technology (ICET), 2017 Publication	<1%

18	Submitted to Shri Guru Gobind Singhji Institute of Engineering and Technology Student Paper	<1%
19	Submitted to University of Nottingham Student Paper	<1%
20	Submitted to National Research University Higher School of Economics Student Paper	<1%
21	arxiv.org Internet Source	<1%
22	zdoc.site Internet Source	<1%
23	Submitted to University of Limerick Student Paper	<1%
24	mc.ai Internet Source	<1%
25	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
26	neurohive.io Internet Source	<1%
27	Submitted to University of Rajshahi Student Paper	<1%
28	Submitted to University of Bedfordshire Student Paper	<1%

29	"Image and Graphics", Springer Science and Business Media LLC, 2019 Publication	<1%
30	Submitted to University of Stellenbosch, South Africa Student Paper	<1%
31	www.datamation.com Internet Source	<1%
32	www.auslogics.com Internet Source	<1%
33	Submitted to Asian Institute of Technology Student Paper	<1%
34	cogprints.org Internet Source	<1%
35	Lecture Notes in Computer Science, 2015. Publication	<1%
36	tim.klingt.org Internet Source	<1%
37	Haibin Liao, Yuchen Yan, Wenhua Dai, Ping Fan. "Age Estimation of Face Images Based on CNN and Divide-and-Rule Strategy", Mathematical Problems in Engineering, 2018 Publication	<1%
38	www.worldlink.com.cn Internet Source	<1%

39	Submitted to Colorado State University, Global Campus Student Paper	<1%
40	Submitted to National Institute of Technical Teachers' Training and Research, Bhopal Student Paper	<1%
41	"Computer Vision – ECCV 2016", Springer Science and Business Media LLC, 2016 Publication	<1%
42	Submitted to University of Surrey Student Paper	<1%
43	Submitted to National University Of Science and Technology Student Paper	<1%
44	Submitted to University of Birmingham Student Paper	<1%
45	ijesr.in Internet Source	<1%
46	tudr.thapar.edu:8080 Internet Source	<1%
47	"Video Analytics. Face and Facial Expression Recognition and Audience Measurement", Springer Nature, 2017 Publication	<1%
48	Submitted to Botswana Accountancy College	

	Student Paper	<1%
49	Submitted to National Institute of Technology, Kurukshetra Student Paper	<1%
50	Submitted to SUNY, Empire State College Student Paper	<1%
51	gnu.inflibnet.ac.in Internet Source	<1%
52	Submitted to Higher Education Commission Pakistan Student Paper	<1%
53	www.ai.rug.nl Internet Source	<1%
54	"Computer Vision", Springer Science and Business Media LLC, 2017 Publication	<1%
55	Submitted to Chulalongkorn University Student Paper	<1%
56	Proceedings in Adaptation Learning and Optimization, 2015. Publication	<1%
57	Submitted to University of London External System Student Paper	<1%

58	Submitted to UNITEC Institute of Technology Student Paper	<1%
59	Lobna M. Abou El-Maged, Ashraf Darwish, Aboul Ella Hassanien. "Chapter 1 Artificial Intelligence-Based Plant's Diseases Classification", Springer Science and Business Media LLC, 2020 Publication	<1%
60	zfu.beatahasznik.com Internet Source	<1%
61	pt.scribd.com Internet Source	<1%
62	"Intelligence Science and Big Data Engineering", Springer Science and Business Media LLC, 2017 Publication	<1%
63	Submitted to University of Westminster Student Paper	<1%
64	Submitted to University of Duhok Student Paper	<1%
65	Submitted to The University of Manchester Student Paper	<1%
66	Submitted to University of Utah Student Paper	<1%
	Submitted to University of Bath	

67	Student Paper	<1%
68	www.ijitee.org Internet Source	<1%
69	eprints-phd.biblio.unitn.it Internet Source	<1%
70	Submitted to Tuskegee University Student Paper	<1%
71	www.datasetlist.com Internet Source	<1%
72	repository.ntu.edu.sg Internet Source	<1%
73	Submitted to Sreenidhi International School Student Paper	<1%
74	garuda.ristekdikti.go.id Internet Source	<1%
75	"Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)", Springer Science and Business Media LLC, 2020 Publication	<1%
76	Submitted to IIT Delhi Student Paper	<1%
77	www.intsci.ac.cn	

	Internet Source	<1%
78	"Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications", Springer Science and Business Media LLC, 2018 Publication	<1%
79	kezewang.com Internet Source	<1%
80	"Embedded Systems and Artificial Intelligence", Springer Science and Business Media LLC, 2020 Publication	<1%
81	Submitted to Birla Institute of Technology and Science Pilani Student Paper	<1%
82	Tuan Anh Le, Atilim Giines Baydin, Robert Zinkov, Frank Wood. "Using synthetic data to train neural networks is model-based reasoning", 2017 International Joint Conference on Neural Networks (IJCNN), 2017 Publication	<1%
83	"Neural Information Processing", Springer Science and Business Media LLC, 2017 Publication	<1%
84	ex-epsilon.slu.se	

	Internet Source	<1%
85	"Advances in Computational Intelligence Systems", Springer Science and Business Media LLC, 2020 Publication	<1%
86	Submitted to University of Edinburgh Student Paper	<1%
87	Submitted to ABES Engineering College Student Paper	<1%
88	Submitted to Laureate Higher Education Group Student Paper	<1%
89	"Biometric Recognition", Springer Science and Business Media LLC, 2018 Publication	<1%
90	"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 Publication	<1%
91	Agrawal, Shikha, and Jitendra Agrawal. "Neural Network Techniques for Cancer Prediction: A Survey", Procedia Computer Science, 2015. Publication	<1%
92	Submitted to British University in Egypt Student Paper	<1%
93	Submitted to Pathfinder Enterprises	
	Student Paper	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		

Determination of Age, Gender, Dress Color and Type of a Person by Convolutional Neural Network (CNN)

CSN Koushik, Shruti Bhargava Choubey, Abhishek Choubey, D. Naresh, N. Bhanu Prakash Reddy

Sreenidhi Institute of Science and Technology, Yamnampet, Ghatksear, Hyderabad-501301, India.

Abstract:

Artificial Intelligence is a soft computing technique that mimics the human intelligence which interprets and percepts based on the inputs given to it, for the output generation or prediction. The input image is processed by the neural network along with the help of the algorithms of Machine Learning making it advantageous to process inputs with reduced use of hardware and the complexities of implementation irrespective of number of models are needed to be developed and deployed. Determining a person's age, gender and color and pattern of the dress is done by the help of the convolutional neural network and Computer Vision for video analytics such that they can be determined individually and in a combined manner by it at a good value of accuracy in training and validation. The existing works could determine only one but not with the combination of the parameters, so pre-trained models like the Caffe, minivggnet models are used to determine them along with the use of openCV. The coding is done in python language, wherein packages like tensorflow, keras, scikit and openCV are used. Opencv is used for the video analytics such it can be used for the trend analysis or the surveillance purposes.

Keywords- Artificial Intelligence, Machine Learning, Deep Learning, Neural Network, CNN, Python programming , Activation functions, OpenCV, Caffe Model, Minivggnet Model.

1. Introduction

Data is the characteristic parameter describing a particular attribute or an entity by value. Irrespective of the type of the data which is being used, huge amounts of data are being produced daily which is to be handled irrespective of the format in which it has been produced. The data scientists apply the principles of mathematics, statistics, science and programming languages etc. over the big data in their own algorithms which enhanced accuracy and efficiency and can store them in the various databases or storage devices.[Van der Aalst, W. (2016). Sanchez-Pinto LN, Luo Y, Churpek MM, (2018)]

Artificial Intelligence (AI) is a type of soft computing technique which is used to mimic the human intelligence for the every action which is to be performed. The actions can only be automated by the AI algorithms with the support of the Machine Learning (ML) and Deep Learning (DL) algorithms. In order to make the decisions/predictions, the model needs to be trained and then validated to determine the outputs. Testing is done to validate over what it has learnt at the training and verify the functionality.

Based on input data, the neural network can use the algorithms of machine learning to improve accuracy. Machine learning algorithms like Regression, Classification for Supervised Learning and Clustering for unsupervised learning etc. can be used which help to improve the model's efficiency and accuracy as a supporting algorithm for the output prediction to the main model being developed. The output prediction depends on the present inputs for those algorithms.[Van der Aalst, W. (2016). Sanchez-Pinto LN, Luo Y, Churpek MM, (2018)] Deep Learning (DL) improves the overall performance and the efficiency of the model which has to detect characteristics of the person like age, gender, color and pattern of the individual's dress by developing a neural network. The model being developed can be used for the trend analysis or for surveillance purposes. Deep learning's neural networks forms the basis for the entire model and then entire decision making process is done by the neurons of the neural network. [Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019), Salehi, H., & Burgueño, R. (2018)]

The training of the model is done by the help of the Graphical Processing Units (GPU) due to their good processing rate and are mostly preferred for the processing of the images. There are various models available online like the caffe model, minivggnet model and coco model etc. which were trained by the help of the GPU units, and possess good accuracy values in order to determine the particular desired parameter. These models extract the features from the input data and then classify or predict the desired outcomes. The feature extraction is done by the Convolutional Neural Network (CNN) for an input image or a video frame being given. [Sanchez-Pinto LN, Luo Y, Churpek MM (2018), Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019), Salehi, H., & Burgueño, R. (2018)]

The main objective of the paper is determine the parameters like the age, gender and the color and the type of the dress worn by the person by using the model being developed. It is makes it easier for the sake of the video analytics and the trend analysis and the surveillance purposes and can be used mostly in the shopping mall for trend analysis or for the surveillance purposes and it can be achieved by the use of the computer vision in association of it with the pretrained models like the caffe and the minivggnet model. These models have good values of accuracy and are found good to be used. The paper has been organized into the following sections namely the review of literature, methodology, results and discussion and conclusion wherein the respective sections describe about the CNN and neural networks briefly, methodology section for the methodology being implemented by us using CNN and the results and discussion section depicting the results got by us respectively.

2. Review of Literature

There are various neural networks available which can be used as per the requirement or inputs being given. They have 3 main layers namely the input layer, hidden layer(s) and the output layer. Each layer has large number of neurons where each is associated with a certain value of weights. The values of the weights are updated at the time of forward and backward propagations, along with the help of an activation function at every layer/neuron in order to activate them. Updating the weights, governs the overall accuracy of the neural network model, as the cost/loss (L) function is reduced to a minimum value, at a certain point in the gradient descent. The model can use various activation functions, but based on the input data, ReLu' is selected for the input and hidden layers whereas the output layer has a softmax function. [Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019), Salehi, H., & Burgueño, R. (2018)]

Artificial Neural Network (ANN) which is used to process the images are known as the Convolution Neural Networks (CNN). It is used for the features to be extracted every time when the convolutions are done. From the input image (i), a particular region is selected and then convolutions are done upon the intensity values of the pixels when the image is segmented. The convolutions are done in a matrix, wherein matrices of same dimensions are used for the convolutions across rows and columns on same input dataset with some dimensions (n_i). As the convolutions are completed in the convolutional layer with some kernel size (kr), the data is given to the max pool layers to reduce the dimensions of the matrix so as to be able to do the computations on the large set of values. The data is sub sampled initially and after the max pooling by the help of strides (st), optimizing the neurons connections or by zero padding (pd). [Shang, R., He, J., Wang, J., Xu, K., Jiao, L., & Stolkin, R. (2020). Song, S., Huang, H., & Ruan, T. (2018), Duan, M., Li, K., Yang, C., & Li, K. (2018).]

$$\mathbf{CNN_{output}} = \mathbf{i} * \mathbf{tr_{func}} = \sum_{k=0}^{N} \mathbf{wt_k x\ i_k\ x\ tr_{func_k}} \quad (1)$$

$$\mathbf{nout} = ((\mathbf{n_i} - \mathbf{2pd} + \mathbf{kr})/\mathbf{st}) + \mathbf{1} \quad (2)$$

$$\mathbf{err_{grdes}} = \left(\frac{1}{N}\right) \sum_{i=0}^{N} \nabla_i \mathbf{xL}\left(\mathbf{i_i}, \mathbf{CNN_{output_i}}, \mathbf{wt_i}\right) \quad (3)$$

The data after the max pooling stage is then flattened and given to a fully connected convolutional layer to complete the task. Whenever the convolutions are done, the particular features are extracted and they are then either given to the predictor or to classifiers available. The network efficiency depends on the number of dense layers, input training dataset and the optimizers used. The optimizer preferred is "adam". The activation functions and the matrix dimensions can be changed to improve the accuracy of the model. The matrix dimensions can be in the order of the 28x28, 64x64, and 128x128 but as the size increases, many features can be extracted. The functions like the relu and the softmax are given to the input and hidden dense layers and to the output layers respectively. [Shang, R., He, J., Wang, J., Xu, K., Jiao, L., & Stolkin, R. (2020). Song, S., Huang, H., & Ruan, T. (2018), Duan, M., Li, K., Yang, C., & Li, K. (2018).]

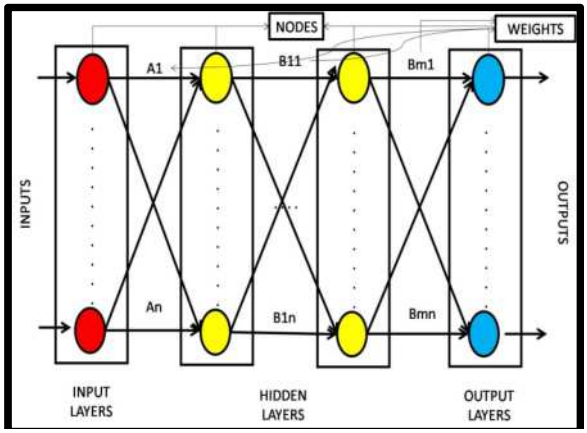


Fig. 1: A Basic Neural Network.

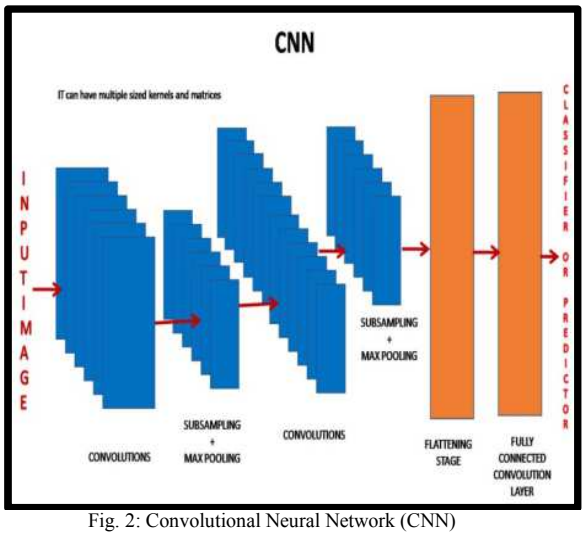


Fig. 2: Convolutional Neural Network (CNN)

To develop the network by the python language, the use of keras library framework along with the tensorflow is done in association with multiple built in packages. Various libraries can be used like the scikit, opencv and the matplotlib library along with the readily available built in models like the minivggnet, coco and the caffe models makes it easier to implement the network at a greater value of the accuracy. Earlier the use of the height of the face and the wrinkles in the face were used for the determination of the age, so they were not quite efficient enough whereas gender was estimated based on the hear structure in the 3 dimensional. Color determination can be done by the help of the K-Means clustering as well, it may be efficient but the time taken to form the clusters is more than determining the color name. To determine the pattern of the dress worn, it was done by the help of CNN and opencv and by the use of the CCNY dataset implemented by the Hough line transformation method as well; these were 2 methods available for the determination. [Nagpal, A., & Gabrani, G. (2019), Anderson, J. A., Glaser, J., & Glotzer, S. C. (2019), Sarwo, Heryadi Y., Budiharto W., Abdurachman E. (2019) , Eran Eidinger, Roee Enbar, and Tal Hassner (2014)]

3.Methodology

In order to determine the parameters like the age, gender and color and dress type an individual wears, a model has to be built based on the neural networks. In order to build a CNN model, initially we need to collect the datasets like the fashionmnist dataset for the sake of the dress pattern classifier determination and dataset like UTKFace dataset can be collected and used for the sake of the age and gender determination. The determination of the color is done by r, g, and b values by using the computer vision opencv package of the python. The use of opencv is done for the sake of the video streaming analytics. One must import the packages of python like the opencv, pil library, numpy and pandas, tensorflow and the keras, tkinter and the math packages. Initially for the above parameters to be defined, the detection of the person is to be done by the help of the method named as the histogram of gradient method and a bounding box is used to locate the respective characteristic.

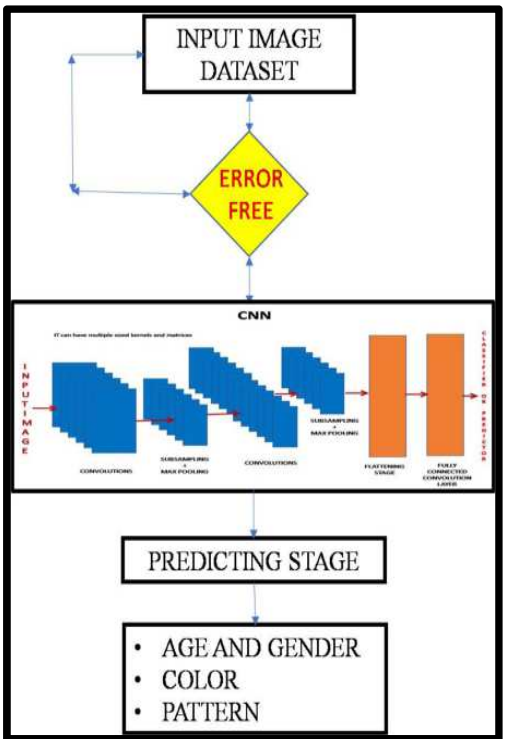


Fig. 3: Flow Chart of the process.

The age and gender determination is done by the use of the caffe model. For the age and gender determination, the model is given the output of the face detection method as its input to determine the age and gender. There are pre-required files which are required; they are the proto.txt files and the model files in which there is the usage of the batch normalization. These pre-required files are used for the sake of defining the regions of the face detection and the type of the activation function required, like the relu for input and hidden layers and softmax for the output respectively. The caffe model gives an accuracy of about 45 to 64 % for age and 74 to 90% accuracy for gender determination. The age groups are defined as 0-2, 4-6, 8-12, 15-20, 25-32, 38-43, 48-53, 60-100 and the gender is defined for male and female. [Duan, M., Li, K., Yang, C., & Li, K. (2018). Gil Levi and Tal Hassner (2015)]

The color is determined by the help of the r, g, and b values. A csv file is used to predefine all the r, g, b value combinations and their respective combination name of the color and the code for the color. The r, g, b values are taken from the particular Cartesian point in the image or the video frame and then the distance is calculated from each value combination. The minimum value of the distance from all the computations is said to be desired color description. Hence the computed color name will be displayed on the video screen as soon as the person is detected. The use of the opencv made it easier to find the desired color from the input video frame or the image.[Hosseini, S. M., Mohhamad-Djafari, A., Mohammadpour, A., Mohammadpour, S., & Nadi, M. (2019), Fernández, P. D. M., Peña, F. A. G., Ren, T. I., & Leandro, J. J. (2019), Li, C., Song, D., Tong, R., & Tang, M. (2019).]

In order to determine the dress pattern classifier output, a minivggnet model is used which provides an accuracy of about 94%, it predicts for about predefined 10 classes namely the shirt, top, ankle boots, coat, pullover, dress, trousers, bags, sandals and sneakers. The model is trained using the fashionmnist dataset as its input at the time of the training. The use of batch normalization makes it easier to understand the bulk dataset in an easier manner, as it is made defined for the black and white values upon the normalization. The data is given to a 2 layer minivggnet CNN model. The model is validated and predicts the output in response to the dataset used for the training. The fashionmnist dataset is the dataset which contains all the images of the dress types in grayscale such that irrespective of the color, the features are extracted easily in each layer at a good value of the accuracy.[Eran Eidinger, Roee Enbar, and Tal Hassner (2014), T. Hussain, M. Ahmad, S. Ali, S. Khan, A. Rahman and A. Haider, (2019), S.Prasanna, N.Priyadharshini and M.Arul Pugazhendhi, (2017), Sarwo, Yaya Heryadi, Widodo Budiharto, and Edi Abdurachman, (2019)]

4. Results and Discussion

All the parameters are merged and can be made to be displayed onto a single video screen initially depicting the person. As video analytics is being done, the video frames are stored in the form of the list wherein the frames alter at every respective time frame due to the live video capture by the camera. Irrespective of the camera resolution being used, the outputs are determined and these can be used for the trend analysis and surveillance purposes. The accuracy values of each model used play a major role in order to determine output at the time of the prediction stages. The results are also dependent on the input data purity and the efficiency with which it has trained itself.

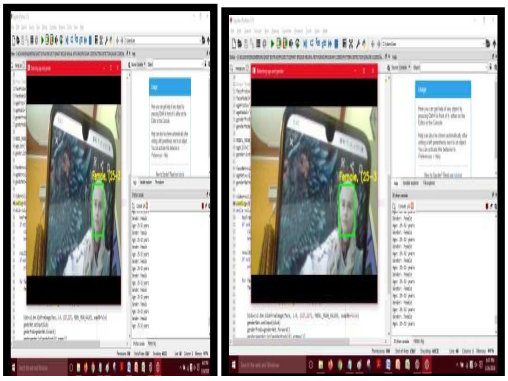


Fig. 4: (a) Age and Gender output 1on video screen 1. (b) Age and Gender output 2 on video screen.

Figure 5: (a) Color detection output (b) Colour detection output 2

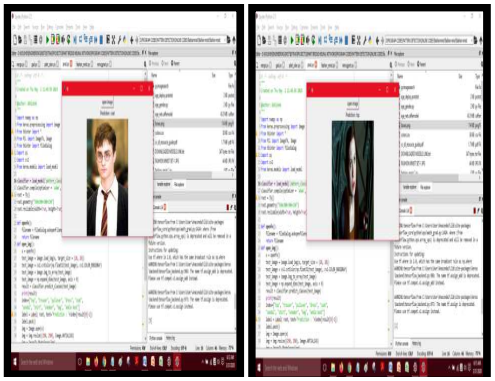


Fig. 6: (a) Pattern determination output 1. (b) Pattern determination output 2.

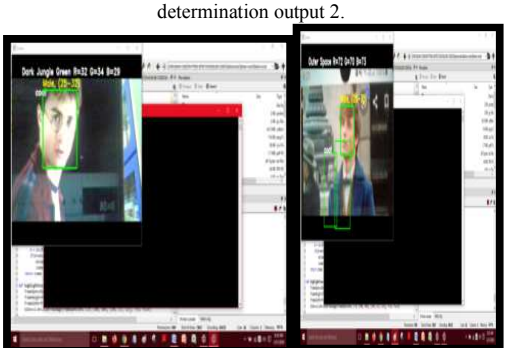


Fig. 7: (a) All parameters determination output 1. (b) All parameters determination output 2

Table 1: Model Efficiency Results. [Duan, M., Li, K., Yang, C., & Li, K. (2018). Gil Levi and Tal Hassner (2015)]

SNO	PARAMETER	METHOD AND ACCURACY RANGE
1	Age and Gender Determination	**Cafee Model:** Age: 45-73%, Gender: 75-93%
2	Color Determination	**Opencv:** Color: 90%
3	Pattern Determination	**Minivggnet Model:** Pattern: 94.39%

The above table depicts the accuracy values of the models which are used. The highest training values which can be got to the model when used at the time of the training for each parameter individually, overall the entire model when combined into a single model, it gave an accuracy value of about 88%.

Conclusion

The determination of the parameters lie the age, gender and the dress color and pattern is found to be essential for the sake of the fashion trend analysis and the for the sake of the surveillance purposes due to the use of the openCV. The data is given to the models which are trained initially like the caffe models and the minivggnet models, such that it can make the predictions from the input data which is being given to it immediately at very good values of accuracy ranging in between the 40%-70% and 75% to 90% for caffe model and around 90% for the color and the minivggnet model. Hence it is very useful to predict the trends for the people; existing and being followed by them. It will be having much application in the shopping malls wherein the data will be taken in a live video stream from a camera for surveillance and trend analysis. The data can also be used for the surveillance purposes also and thus it makes very useful to use it in the real time scenarios. So, by the help of the CNN model developed, all the hidden features are extracted from the particular region of interest from the video streaming and thereby easily determining the output for the desired parameters easily and effectively.

6. References

Van der Aalst, W. (2016). Data Science in Action. Process Mining, 3–23. doi:10.1007/978-3-662-49851-4_1 .

Sanchez-Pinto LN, Luo Y, Churpek MM, Big Data and Data Science in Critical Care, CHEST (2018), doi: 10.1016/j.chest.2018.04.037

Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of Big Data – evolution, challenges and research agenda. International Journal of Information Management, 48, 63–71. doi:10.1016/j.ijinfomgt.2019.01.021

Salehi, H., & Burgueño, R. (2018). Emerging artificial intelligence methods in structural engineering. Engineering Structures, 171, 170–189. doi:10.1016/j.engstruct.2018.05.084.

Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications. 2019 Amity International Conference on Artificial Intelligence

(AICAI). 978-1-5386-9346-9/19/$31.00 ©2019 IEEE ,doi:10.1109/aicai.2019.8701341

Anderson, J. A., Glaser, J., & Glotzer, S. C. (2019). HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations.

Shang, R., He, J., Wang, J., Xu, K., Jiao, L., & Stolkin, R. (2020). Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification. *Knowledge-Based Systems*, 105542.

Song, S., Huang, H., & Ruan, T. (2018). Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools and Applications. doi:10.1007/s11042-018-5749-3

Duan, M., Li, K., Yang, C., & Li, K. (2018). A hybrid deep learning CNN–ELM for age and gender classification. Neurocomputing, 275, 448–461. doi:10.1016/j.neucom.2017.08.062.

Gil Levi and Tal Hassner, Age and Gender Classification using Convolutional Neural Networks, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2015, pp. 34-42.

Hosseini, S. M., Mohhamad-Djafari, A., Mohammadpour, A., Mohammadpour, S., & Nadi, M. (2019). Carpets Color and Pattern Detection Based on Their Images. In *Multidisciplinary Digital Publishing Institute Proceedings* (Vol. 33, No. 1, p. 28).

Fernández, P. D. M., Peña, F. A. G., Ren, T. I., & Leandro, J. J. (2019). Fast and robust multiple ColorChecker detection using deep convolutional neural networks. *Image and Vision Computing*, *81*, 15-24.

Li, C., Song, D., Tong, R., & Tang, M. (2019). Illumination-aware faster R-CNN for robust multispectral pedestrian detection. *Pattern Recognition*, *85*, 161-171.

Seo, Y., & Shin, K. S. (2019). Hierarchical convolutional neural networks for fashion image classification. *Expert Systems with Applications*, *116*, 328-339.

Heryadi, Y., Budiharto, W., & Abdurachman, E. (2019, April). Logo and Brand Recognition from Imbalanced Dataset Using MiniGoogLeNet and MiniVGGNet Models. In *Asian Conference on Intelligent Information and Database Systems* (pp. 385-393). Springer, Cham.

Eidinger, E., Enbar, R., & Hassner, T. (2014). Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, *9*(12), 2170-2179.

Hussain, T., Ahmad, M., Ali, S., Khan, S., Rahman, A., & Haider, A. (2019, January). An Intelligent Dress Uniform Identification System. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-4). IEEE.

Prasanna, S., Priyadharshini, N., & Pugazhendhi, M. A. (2017). Textile robot for matching and pick up clothes based on color recognition. *Asian Journal of Applied Science and Technology (AJAST)*, *1*(3), 62-65.

Sarwo, Yaya Heryadi, Widodo Budiharto, and Edi Abdurachman, Logo and Brand Recognition from Imbalanced Dataset Using MiniGoogLeNet and MiniVGGNet Models, © Springer Nature Switzerland AG 2019 LNAI 11431, pp. 385–393, 2019. https://doi.org/10.1007/978-3-030-14799-0_33.