A

Project Report on

Intelligent security and information management system using video analysis

in partial fulfilment of requirements for the award of the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

Submitted by:

Ajeet Kumar Yadav (1805231005)

Maneesh (1805231032)

Prashant Singh (1805231037)

Pranjal Mittal (1900520319003)

under guidance of

Dr. Rajiv Kumar Singh

Institute of Engineering and Technology, Lucknow

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

CANDIDATE DECLARATION

We hereby declare that the work being presented in this dissertation entitled “INTELLIGENT SECURITY AND INFORMATION MANAGEMENT SYSTEM USING VIDEO ANALYSIS," submitted in partial completion of the requirements for a Bachelor of Technology in Electronics and Communication Engineering, is an authentic record of my own work completed under the supervision of Dr. Rajiv Kumar Singh, Department of Electronics and Communication Engineering, IET Lucknow; who provided supervision and assistance.

The content of this dissertation, in full or in part, has not been submitted by me to any other university or institute for the award of any degree or diploma.

Date:

Place: Lucknow

Ajeet Kumar Yadav
1805231005

Maneesh
1805231032

Prashant Singh
180523103

Pranjal Mittal
1900520319003

CERTIFICATE

This is to certify that Project report entitled "INTELLIGENT SECURITY AND INFORMATION MANAGEMENT SYSTEM USING VIDEO ANALYSIS" which is submitted by Ajeet Kumar Yadav (1805231005), Maneesh (1805231032), Prashant Singh (180523103) & Pranjal Mittal (1900520319003) in partial fulfilment of the requirement for the award of Bachelor of Technology in Electronics and Communication engineering, in Department of Electronics Engineering of Institute of Engineering and Technology, during the academic year 2021-2022, is a record of the bonafide work carried out by the student during 8th semester under my supervision.

To the best of my knowledge, the work reported has not really been submitted for a degree elsewhere.

Date:

Dr. Neelam Srivastava	Dr. Rajiv Kumar Singh
H.O.D. ECD	Project Guide

ACKNOWLEDGEMENT

Success is a delectable fruit that everyone wishes to sample. To reach this goal, a lot of physical and mental effort is required. Every time we prepare this report, we realise how much we couldn't have done it without the aid of so many talented and devoted people. As a result, we'd want to convey our gratitude to those whose assistance has been invaluable.

We would really like to express our gratitude and appreciation to Dr. Rajiv K. Singh, Professor in the Department of Electronics Engineering, IET Lucknow, for suggesting the current topic and for providing us with inspiring guidance, constructive criticism, and useful recommendations throughout the project.

We deeply express sincere thanks to our Head of Department Dr. Neelam Srivastava for encouraging and allowing us to present the project on the topic “Intelligent security and information management system using video analysis” at our department premises for the partial fulfilment of the requirements leading to the award of Bachelor of Technology degree.

Last but not least, we would like to acknowledge all of our friends who have patiently offered varying levels of assistance to support us in completing this project.

Ajeet Kumar Yadav (1805231005)
Maneesh (1805231032)
Prashant Singh (1805231037)
Pranjal Mittal (1900520319003)

ABSTRACT

Nowadays, because computers are powerful enough to implement sophisticated algorithms, people use computers to run a variety of applications. Facial recognition is one of the most popular applications in this discipline. In truth, computers are capable of not only instantly recognising who a person is, but also of operating 24 hours a day, seven days a week, some things that people are not able to do. As a consequence, computers are increasingly being used to replace people in repetitive and real-time tasks. In this paper, we integrate facial recognition into a video surveillance system that checks the presence and frequency of actions of registered people.

The GUI (graphical user interface) on this system makes user-to-system and vice-versa interaction direct and straightforward. The faster-RCNN approach is at the heart of the system. There are five steps to this approach (e.g., Preprocessing of frames, detecting faces, removing non-frontal-view faces, recognizing, and database management).The recognition phase, in particular, is treated as kind of an open-set facial recognition problem, allowing the system to recognise people who have never registered in the database. We also improve system performance by exploiting the hardware resources of users' machines. Despite the fact that the system was designed to function with a low-resolution webcam, it performed reasonably well on our private dataset.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Short form	Full form
DL	Deep Learning
ML	Machine Learning
CNN	Convolutional Neural Networks
R-CNN	Region-based Convolutional Neural Network
MTCNN	Multi-Task Cascaded Convolutional Neural Networks
RPN	Region Proposal Network
SVC	Support Vector Classifier
SVM	Support Vector Machine
KNN	K-Nearest Neighbour
LBP	Local Binary Pattern
LFW	Labelled Faces in Wild
CV	Computer Vision

Short form	**Full form**
GUI	Graphic User Interface
ANN	Artificial Neural Network
NMS	Non-Maximum Suppression
ROI	Return Of Investment
GBM	Gradient Boosting Machine
API	Application Programming Index

PROBLEM STATEMENT

We encountered a need for an intelligent system that can be used for security and information management systems.

In a general/simple security system, cameras work the whole day. If any unknown person enters in any restricted area. We will not be able to flag that person with the help of a camera only. We will have to invest our money on a person, which can monitor the camera feed continuously

To tackle this need we used F-CNN and MTCNN to design this intelligent system which identify a selective object from the image and manages its information and alerts to admin according to the previously set condition

SOLUTION

The video Surveillance System used here will only store the data that shows any activity being performed on the screen that can be analysed and predicted on the basis of object and face detection.

In real life scenarios, systems like face recognition are used for a number of things and tasks, including tracking, workforce management, and locating information on celebrities. Face recognition systems can be designed in a variety of ways, but they are typically hampered by light, non-frontal faces, camera quality, and other factors. As a result, each method has its own set of difficulties. Face recognition includes two primary stages: face detection and face recognition. However, we want to limit the blur-free and frontal-view faces of users, thus our system comprises four stages: face detection, blur detection, face recognition, and landmark detection.

The aim of this project is to optimise the available space for memory allocation, recognise known/ unknown persons and alert the user.

Chapter-1

INTRODUCTION

Face recognition systems are used in a variety of situations in real life, including tracking, managing personnel, and locating information about celebrities. There are numerous techniques to employ video analysis to construct a security system, but these systems are usually impacted by light, non-frontal faces, camera resolution, and other factors. As a result, each method has its own set of difficulties. Face recognition is divided into two stages: face detection and face recognition.

1. ;Automatic video monitoring technologies' must be applied in the development of security and surveillance tools to assist the human operator recognise and respond to potential dangers.
2. This research uses a faster-Region convolutional neural network, which is a cutting-edge object detection network. The Faster R-CNN detector is made up of two parts: a deep fully convolutional network that proposes areas and a Fast R-CNN detector that employs those areas.
3. The entire system functions as a solitary, consolidated network for object detection.

The video surveillance system in use here will only save data that demonstrates any activity on the screen that can be evaluated and anticipated using object and face detection.

In real life, face recognition systems are used for a number of tasks, including tracking, workforce management, and locating information on celebrities. Face recognition systems can be designed in a variety of ways, but they are typically hampered by light, non-frontal faces, camera quality, and other factors. As a result, each method has its own set of difficulties. Face recognition includes two primary stages: face detection and face recognition. However, we want to limit the blur-free and frontal-view faces of users, thus our system comprises four stages: face detection, blur detection, face recognition, and landmark detection.

1.1 Face recognition system

1.1.1 Face detection

We must first detect faces in an image and bound the high-level regions to exclude effects such as hair and background before moving on to the face identification stage. Viola and Jones [1] proposed a face detector that trained cascaded classifiers using Haar-Like features and the AdaBoost algorithm, resulting in good performance and real-time efficiency.. However, a few studies [2],[3],[4] suggest that Haar-Like features, even with more complex features and more training photos, can fail significantly in real-world applications with larger visual differences of human faces. [5],[6],[7] propose deformable part models (DPM) for face detection in addition to the cascade model and get impressive results.They do, however, have a significant computational cost and may require expensive annotation during the training phase.

Convolutional neural networks (CNN) gain great accuracy in a variety of computer vision applications, particularly face detection tasks, as a result of the emergence of data models. Li et al. [8] use cascaded CNN for face detection, but this requires bounding box calibration in addition to face detection, which increases computational complexity and ignores the relationship between facial landmarks localization and bounding box regression. Face alignment is also very important. Regression-based methods [9],[10],[11] and template fitting methods [7],[12],[13] are two common approaches.. The majority of existing face identification and face alignment algorithms, on the other hand, overlook the relationship between these two tasks. Despite the fact that there are various works seeking to jointly solve them, these works still have limits.For example, Chen et al. [14] use pixel value difference features to illustrate random forest alignment and detection. The presence of handcrafted elements, on the other hand, limits its performance. Based on our previous experiments, we chose a new technique that integrates these two tasks by using a unified cascaded CNN via multi-task learning called Multi-task Convolutional Network, which is also a face detector used in the FaceNet model.

Figure 1.1: Face Detection

1.1.2 Landmark detection

We will control non-frontal faces for improved accuracy since the placements of the facial landmark points surrounding facial components (nose, eye centres, etc.) capture facial deformations owing to head motions and facial expressions. As a result, they're critical for a variety of facial analysis tasks. Over the years, numerous facial landmark identification algorithms have been created to automatically recognise those critical spots. In this study, we use the Dlib package, which is a powerful open source tool for detecting faces and facial features.

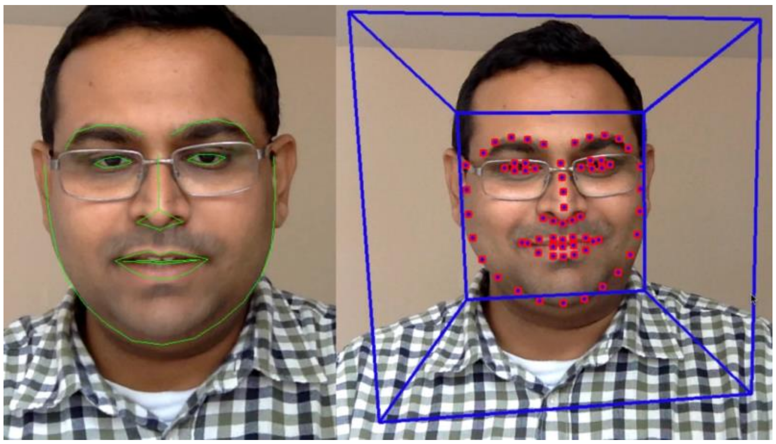

Figure 1.2: Landmark Detection

1.1.3 Face recognition

Feature vectors are produced from the areas of a face after face detection and alignment. Face recognition has historically used the Gabor feature as one of the two most prevalent features. Tudor Barbu [15] used the Gabor transform to extract features before applying K-Nearest Neighbour (K-NN) to determine the identification of a face based on clustering data. On the Yale Face Database B [22], this approach performed admirably, with a 90 percent accuracy rate. Local Binary Patterns (LBP) is an approach based on the Haar-Like feature released by OpenCV [16], an open-source library focusing on Computer Vision algorithms. LBP has a fantastic real-time efficiency in terms of speed, however it is not stable in terms of accuracy. This approach, in particular, cannot deal with noise, which is why LBP and Haar-Like features are rarely used in actual systems. Because of the limits of traditional features, deep learning-based techniques have gradually evolved in their place. Yi Sun, Xiaogang Wang, and Xiaoou Tang [17] developed a deep model called Deep hidden IDentity (DeepID) that extracted face features using a Convolutional Neural Network (CNN). This approach, though, had the advantage of being able to train with a little dataset, which was particularly valuable for applications that couldn't collect a large number of users but despite this, the DeepID model became extremely complicated, with multiple neural network branches for each person, in order to achieve high accuracy. Ten patches of a face were chosen from each image that contained interesting information, and they were then scaled using three figures in RGB and grey channel. The model used 60 separate networks to extract information from an image, then fed the retrieved features into a Joint Bayesian classifier. On Labelled Faces in the Wild (LFW), DeepID attained an excellent accuracy of 97.45% [23]. DeepID2 [18], an upgraded version of its predecessor, was introduced in 2014 by the inventors of DeepID. In the new version, interesting zones were built to eliminate redundant patches that couldn't provide high-level functionality. With a 99.63 percent boost in accuracy, this endeavour had a strong positive influence on accuracy. In 2015, Google Inc. [19] created the FaceNet model, which extracted features using a deep CNN Inception [20] and a triplet loss function.

Figure 1.3: Face Recognition

FaceNet features are robust in both face verification and face recognition because they use a hard triplet loss to segregate features for each person. The model's accuracy of 99.63 percent on LFW and 95.12 percent on Youtube Faces DB [24] was sufficient to depict the model's perfection.

1.2 Motivation

We were faced with the need to develop an intelligent security system that could monitor the video feed and assist us in detecting threats by analyzing the activities of individuals and only storing the data that was required to keep the system running, rather than storing the empty screen where no activity was taking place, as well as assist in database management.

1.3 Objective

Face Detection is, of course, the most basic duty in Face Recognition. You must first "catch" a face in order to recognise it when compared to a new face captured later.

The "Haar Cascade classifier" is the most prevalent method for detecting a face (or any other item).

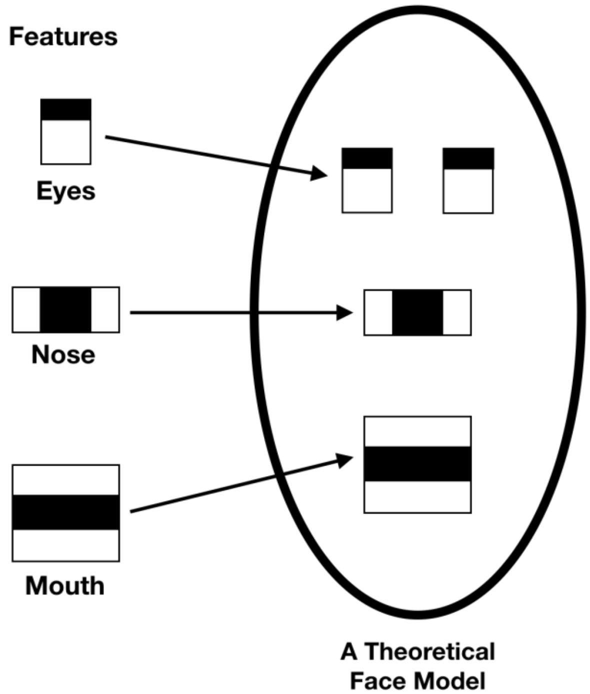

Figure 1.4: Haar Cascade Classifier

Paul Viola and Michael Jones presented an efficient object identification methodology based on Haar feature-
based cascade classifiers in their paper "Rapid Object Detection with a Boosted Cascade of Simple Features." It's a machine-
learning technique that involves learning a cascade function from a large number of positive and negative photographs. It's then used to locate things in other photographs.

Here, we'll be working on facial recognition. To train the classifier, a large lot of positive images (face images) and negative images are required (images without faces). Then we must extract characteristics from it. The fact of the matter is that OpenCV comes with a detector as well as a trainer.Cascade Classifier Training provides all the knowledge you need to train your own classifier for any object, such as vehicles, planes, and so on, using OpenCV.

1.4 Significance

Security systems and equipment are essential for protecting your house and family. Security systems and equipment protect your family members from intruders and robbers. Every commercial establishment needs security since no one wants to lose their assets. A home security system's major goals are to protect your home and your family. In addition to burglary, a security system may detect a range of threats, including smoke, fire, carbon monoxide poisoning, and water damage.

1.5 Idea

CNNs are a type of image processing architecture that allows for better and faster results. They've recently been employed in text processing, where the network's input is the embedding of the tokens rather than the pixels of the images.

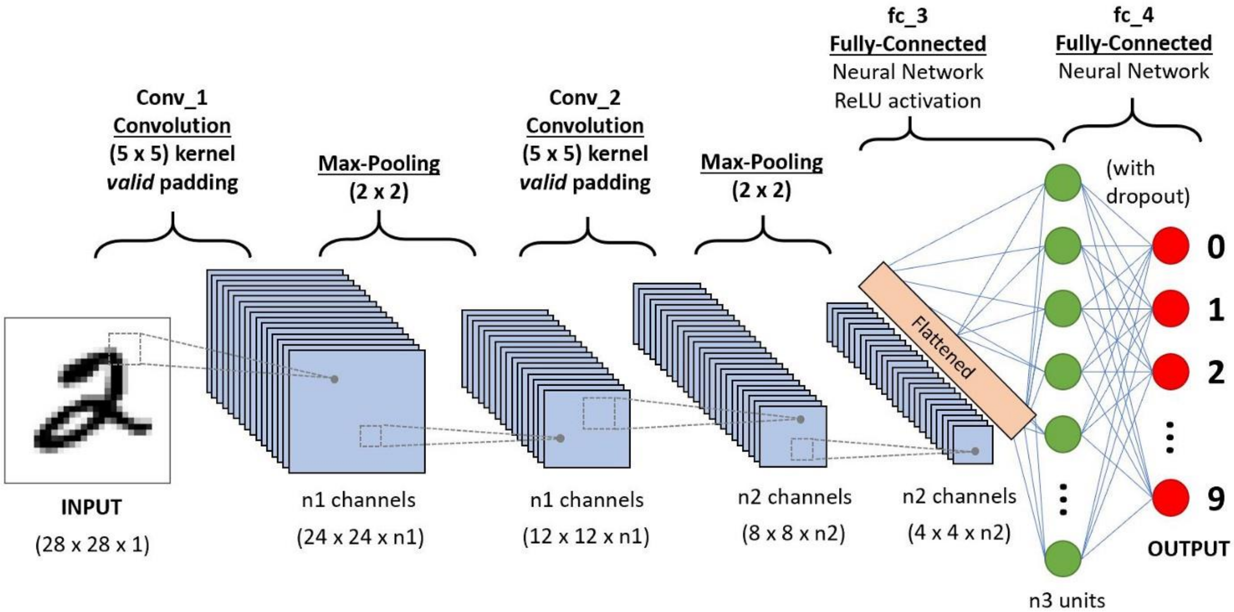

Figure 1.5: CNN Architecture

We'll use it to help us work more efficiently and move forward with our project, as well as other algorithms, tools, and frameworks. We'll also design a user interface to present our data and make it more accessible to users.

Chapter-2

RELATED WORKS

In the beginning, there has been a lot of progress, innovation, experimentation, and application. Smart home automation systems increasingly provide services beyond basic home management and environmental monitoring, thanks to advancements in IT and IoT technology. Object detection methods like template matching and rudimentary part-based models were utilised [e.g., Fischler and Elschlager (1973)]. To improve the intelligence responses in smart home automation systems, machine learning and deep learning algorithms have been used. The usage of Bluetooth, GSM, SMS, Zigbee, and other communication modes for control, which has the disadvantage of limited range coverage, is one of the research shortcomings noted. The usage of web-based design and SMS as a form of appliance control are two more research issues mentioned. Later, statistical classifier-based approaches (e.g., Neural Networks, SVM, Adaboost, Bayes, and others) were develope d [for example, Osuna et al. (1997), Rowley et al. (1998), Sung and Poggio (1998), Schneider man and Kanade (2000), Yang et al. (2000a,b), Fleuret and Geman (2001), Romdhani et al. F inally, only a few previous studies incorporated intelligent decisions into the systems. The va st bulk of scientific investigations in training and evaluation techniques, as well as categorizat ion methods and techniques, was formed on the basis set down by its first functional family o f object detectors, which were all subjected to statistical classifiers.

Face detection is the most popular use of object detection since it is an essential capability for any technology that interacts with humans. Later, statistical classifier-based approaches like SVM, Neural Networks, Bayes, Adaboost, and many others like them were developed [for example, Osuna et al. (1997), Rowley et al. (1998), Sung and Poggio (19 98), Schneiderman and Kanade (2000), Yang et al. (2000a,b), Fleuret and Geman (2001), Ro mdhani et al. The vast bulk of scientific investigations in training and evaluation techniques, as well as categorization methods and techniques, was formed on the basis set down by its fir st functional family of object detectors, which were all subjected to statistical classifiers.The bulk of situations are related to objects that people deal with on a regular basis, such as other people.

In order to detect the things showing in the image at multiple scales and positions, most objec t detection systems use the same basic strategy, known as sliding window: an exhaustive sear ch is used to detect the objects appearing in the image at different scales and places. This sear ch employs a classifier, which is the detector's fundamental component and determines wheth er a particular image patch relates to the item or not. Because the classifier only works at a ce rtain scale and patch size, several downscaled versions of the input image are generated, and t he classifier is used to classify all possible patches of the given size for each of the downscale d versions.

Sliding windows come in three different designs. The first one is predicated on bag-of-words (Weinland et al., 2011; Tsai, 2012), a method for validating an item's existence that can be used effectively in specific settings by periodically enhancing the visual region containing the object [e.g., Lampert et al. (2009)]. The second method [for example, Prati et al. (2012)] looks for portions of the image in which the entity is likely to turn up by evaluating patches and looking for them iteratively. By foregoing a thorough search of all photo patches, you can save time and effort. These two approaches try to reduce the number of image patches where categorization is conducted. The third method [for example, Azzopardi and Petkov (2013)] finds critical spots and then matches them to do detection. These approaches may not necessarily ensure that all instances of an object will be detected.

Chapter-3

ROADMAP

A system is built in order to manage the appearance of a person in front of a camera. Face detection, landmark detection, and recognition are usually the first three phases in a facial recognition system. However, to provide high-quality input frames for the underlying algorithms, we add an early filter (Input preprocessing step) that can filter out blur frames captured by persons moving in front of a webcam. Clean frames are then delivered to the Face detection block, which counts how many faces are present in them.

Only one face per frame is allowed to be handled in our specific limitation. Input preprocessing, face detection/person detection, face recognition, database administration, and alarm system are the five stages. This means that it is more difficult for our system to enter frames from the camera when they pass through it. As a result, the best frames will be analysed more frequently, potentially enhancing recognition accuracy. Those with more than one face are not accepted. Following that, Landmark detection is used to locate statistically significant places in the face in order to determine whether the face is in the camera's frontal view. This frontal-view check will aid in improving the face recognition algorithm's accuracy.Finally, in the Face identification step, key features are extracted from the blur-clean, one-face, and frontal-view frames from the previous stages, and a classification job is performed to identify which category the input most likely belongs to. If the individual is in our database, it will simply count the frequency; but, if the person is not in our database, it will first send an alert to the owner before keeping the frequency record.

We also built a user-friendly GUI(graphic user interface) to ensure that anyone who wants to use the system to manage (owner) or check (home visitor) presence may do so without any special knowledge. We meticulously evaluate the distribution outlier of attributes characterising registered accounts to make the system more resilient. As a result, the algorithm can recognise users who have never registered for the app before, which is similar to the open-set problem in facial recognition.

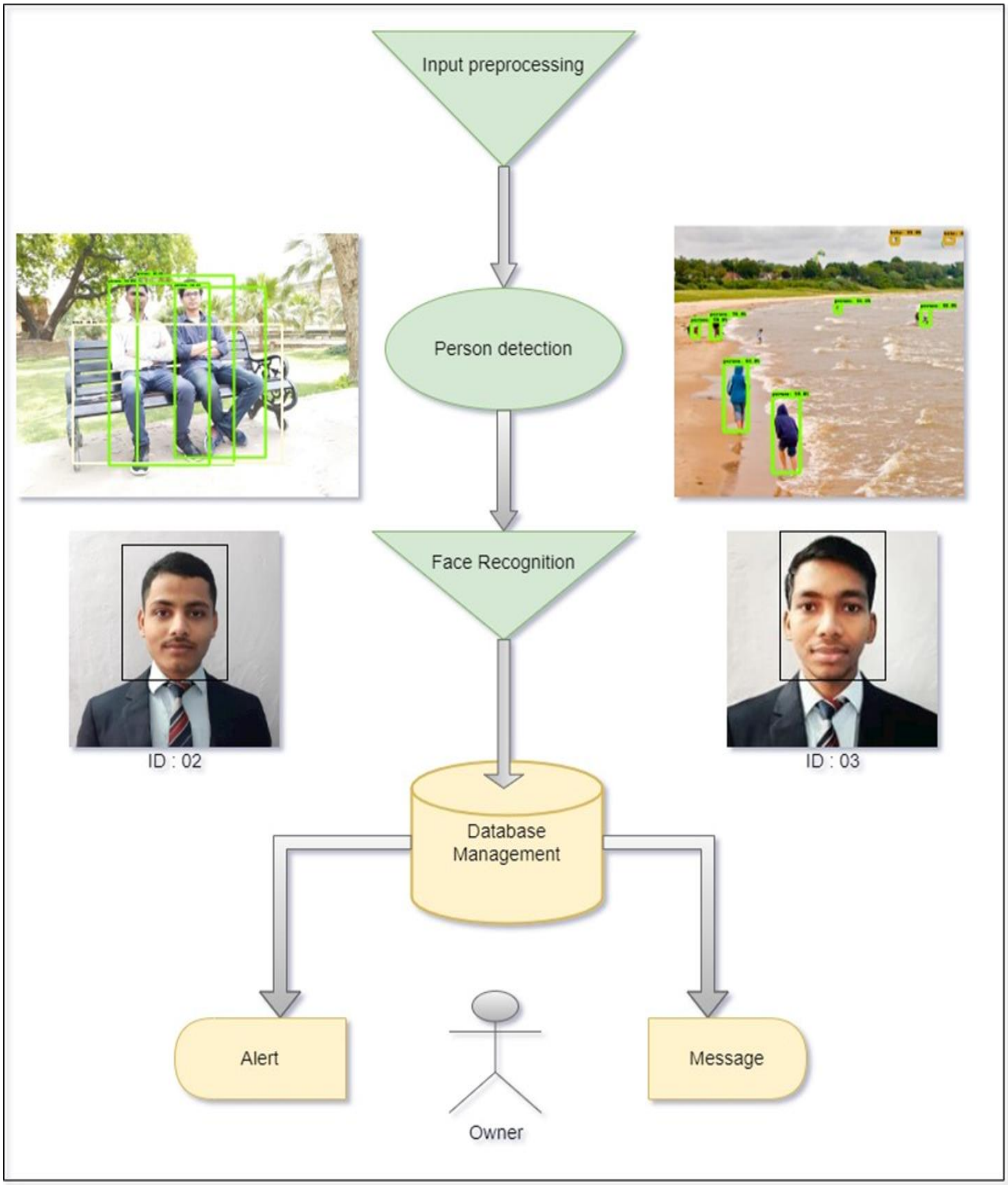

Figure 3.1: Project Workflow

Chapter-4

DL and ML Algorithms

4.1 Neural Network

4.1.1 Biological Neuron

“A neuron is a type of biological cell that processes data. It consists of a cell body (soma) and two tree-like branches (dendrites and axon)”. Dendrites are used to receive signals from other neurons, also known as a receiver, and transmits any signal created by a cell body through the axon, also known as a transmitter, to another neuron. These neurons are present in the human brains, all the human activity is performed with the help of these neurons. Every neuron in the human mind is connected with 103-104 other neurons; the human mind comprises 1014 to 1015 interconnections.

“A biological architecture of neurons is depicted in this diagram.”

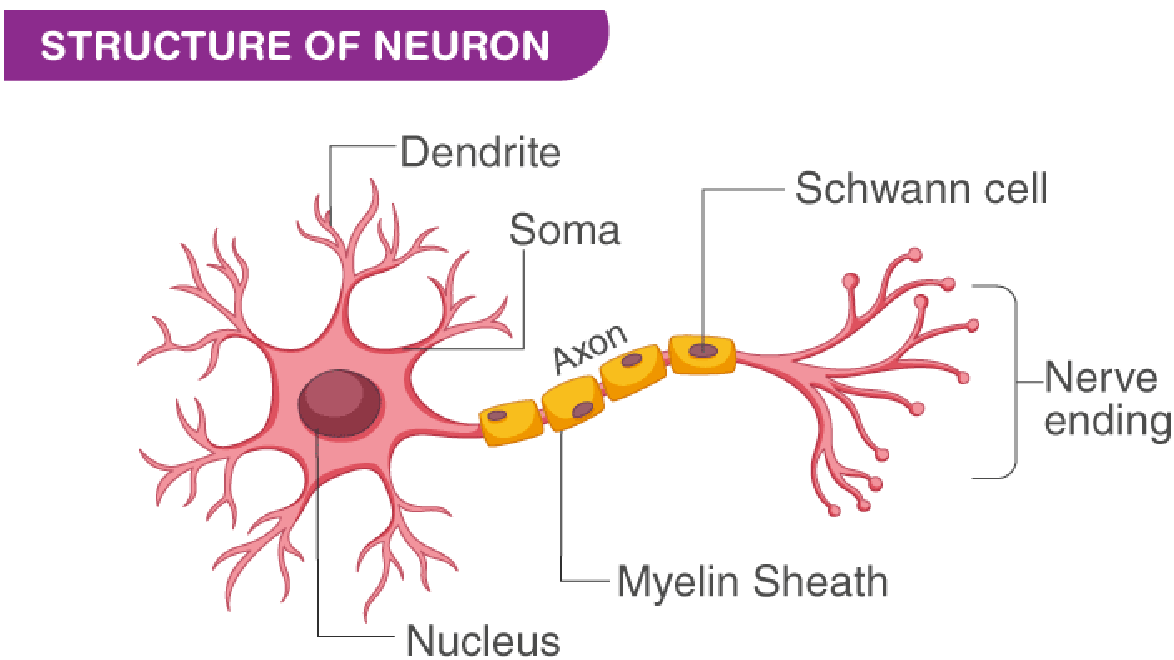

Figure 4.1: Biological Neuron

4.1.2 Artificial neural Network / Feed forward Neural network

The ANN algorithm is a powerful regression and classification Machine Learning tool. In a neural network, there are three different sorts of layers.

Input Layer : This is a neural network's first layer. The shape of the input data determines the size of the input layer (No. of features in the dataset). In the neural network, there will be only one input layer.

Hidden Layer : These layers are used to process the input according to the activation. There are multiple hidden layers present in a neural network.

We can say :

$$Number\ of\ hidden\ layers\ \propto\ Complexity\ of\ the\ model$$

$$\boldsymbol{Number\ of\ hidden\ layers\ \propto\ Accuracy\ of\ the\ model}$$

In the hidden layer, mostly we use relu as an activation function.

Output Layer : The problem statement decides the number of neurons in the output layer; for binary classification, only one neuron is used; for multi-class classification, many neurons are employed.

We always utilise softmax or sigmoid as an activation function in the output layer.

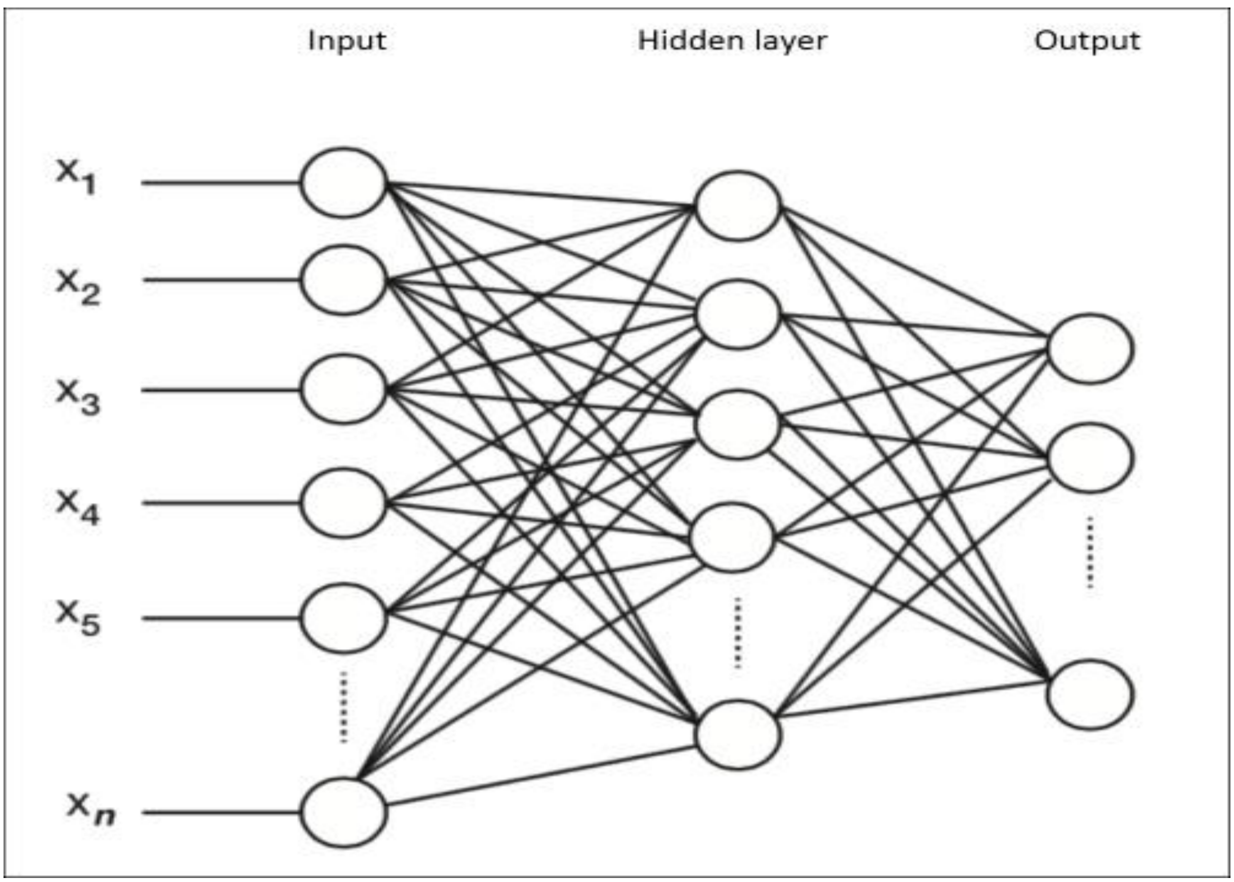

Figure 4.2: Artificial neural network

4.1.3 Working of Artificial neural network :

We have n features, and dataset has m training example

Input vector shape -> (n, 1)

We will randomly initialise the weight vector of size (n, 1)

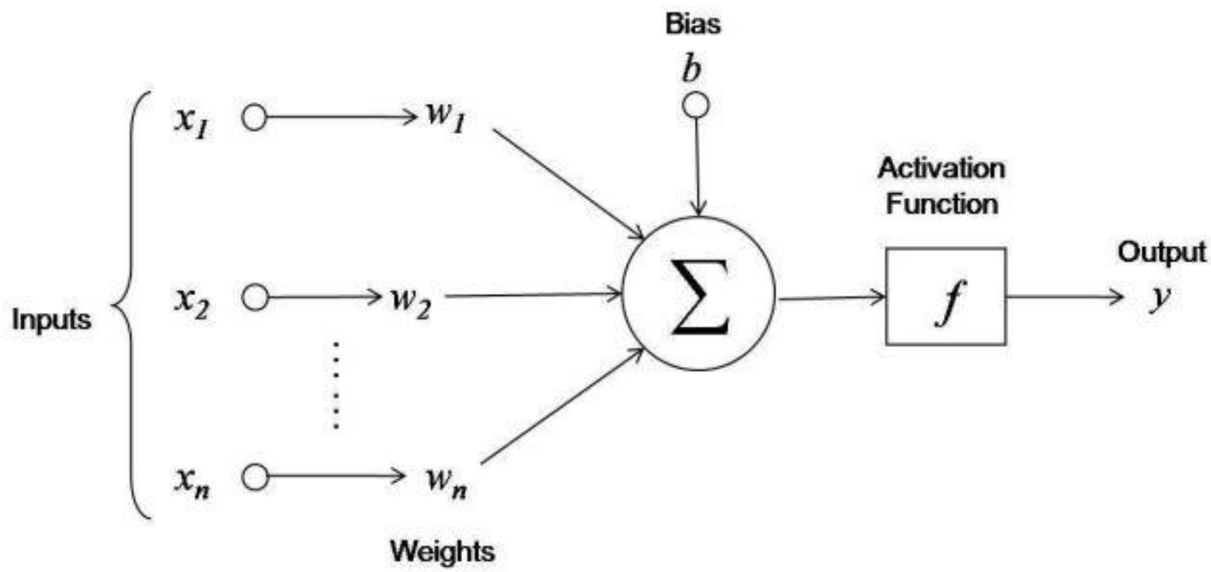

Figure 4.3: Working of Artificial Neural network

4.1.3.1 Equation :

$$z_k = \Sigma W_i * x_i$$
$$y_i = f(z_k)$$

Where f is a sigmoid activation function

4.1.3.2 In Matrix form:

$$z_k = [W_0 \;\; W_1 \;\; W_2 \ldots\ldots W_n][x_0 \;\; x_1 \;\; x_2 \ldots\ldots x_n]^T$$

4.1.3.3 Type of Activation functions

1) Piecewise - linear activation function

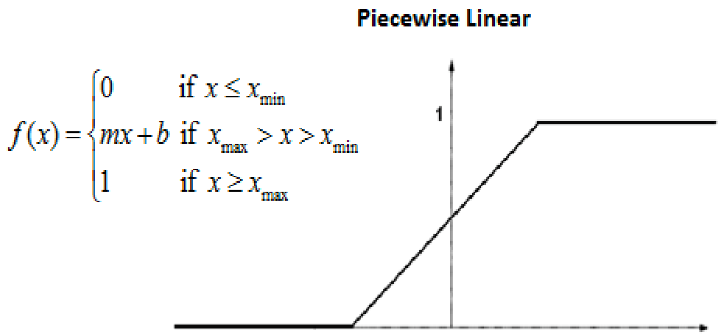

Figure 4.4: Piecewise - linear activation function

2) Sigmoid (Logistic) activation function :

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x).$$

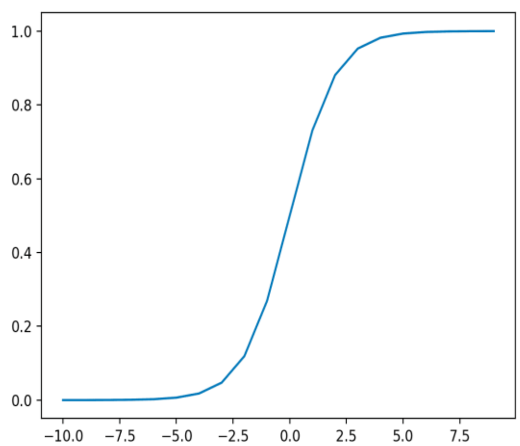

Figure 4.5: Sigmoid activation function

3) Softmax activation function :

$$\mathrm{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for j = 1,...,K}$$

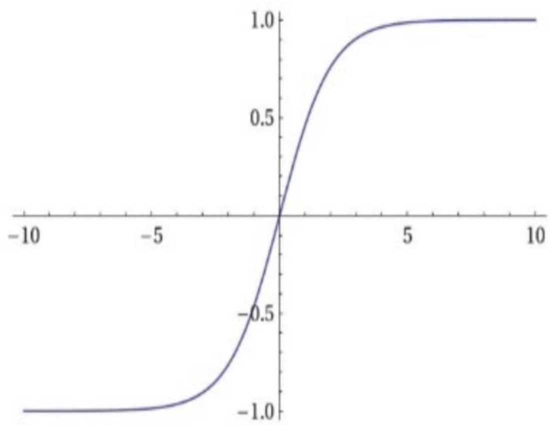

Figure 4.6: Softmax activation function

4) Relu activation function

$$\boldsymbol{R(z) = max(0, z)}$$

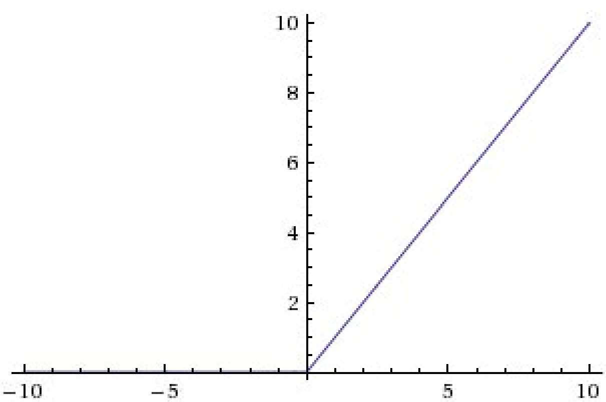

Figure 4.7: Relu activation function

Leaky-Relu activation function

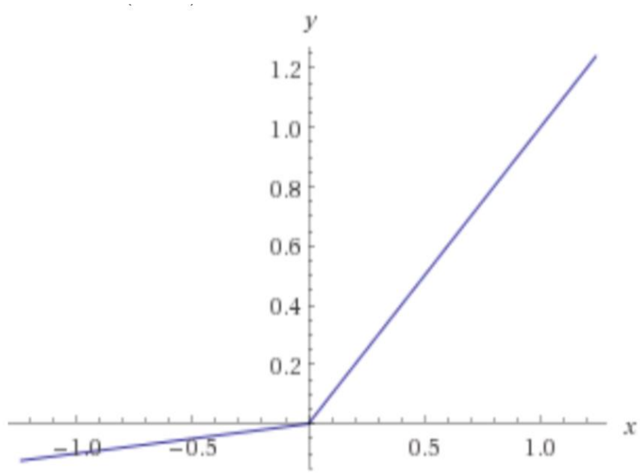

Figure 4.8: Leaky Relu activation function

5) Hyperbolic tangent activation function :

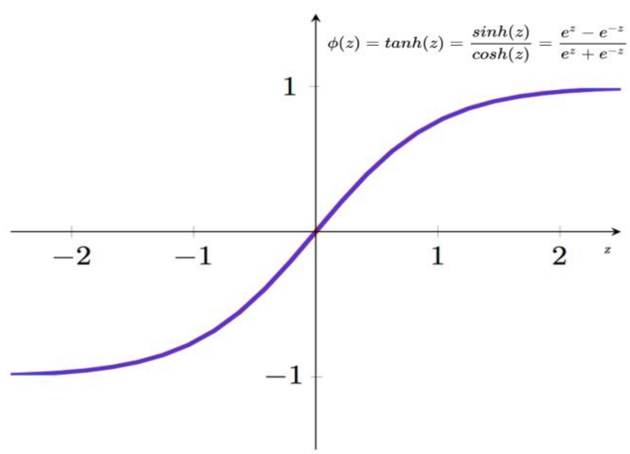

Figure 4.9: Hyperbolic tangent activation function

4.2 Convolution Neural Network :

Convolution layers are used in the neural network in CNN. The Convolution Neural Network's general architecture is made up of alternating layers of convolutional layers with relu activation function, as well as pooling and dropout layers. The convolution network's output is flattened before being sent into the Dense (completely connected) layers.

The outputs of convolution are referred to as feature maps as they contain features, which are extracted from the input matrix by the convolution kernel/filters.

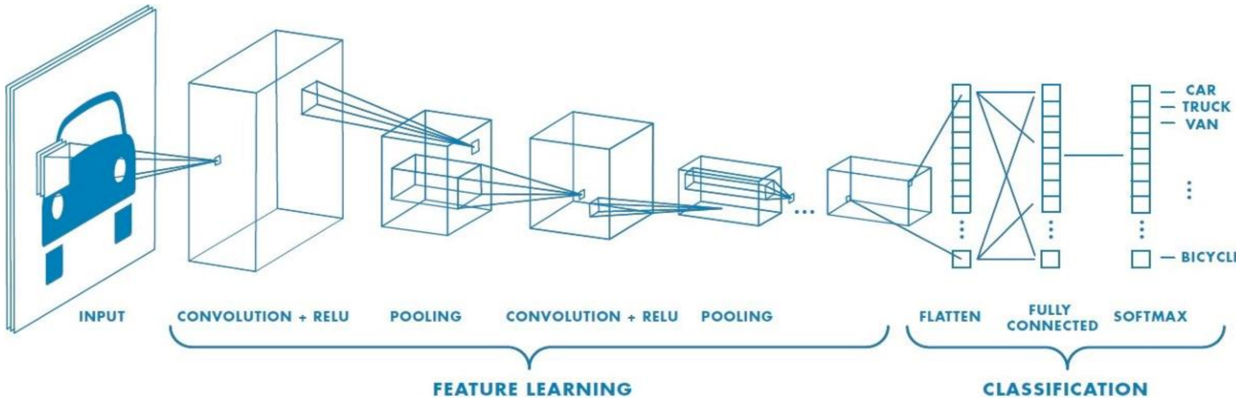

Figure 4.10: Convolution Neural Network

4.3 Faster R-CNN

4.3.1 About

Faster R-CNN was proposed in 2015 by Google to solve real time object detection in an efficient manner. This is a combination of previous object detection model Fast R-CNN and region propranolol neural network[RPN]. RPN works on the favourable regions present inside the pictures. It generates a set of bounding boxes of different sizes. For better results while training classification loss and bounding box regression loss are evaluated and optimization takes place.Fast R-CNN uses several convolutional and maximum pooling layers to generate feature maps in order to detect objects in specific regions. A series of densely integrated two branching output layers is created from the feature vector. First layer produces a softmax probability to estimate the classes and the second produces the real value class.

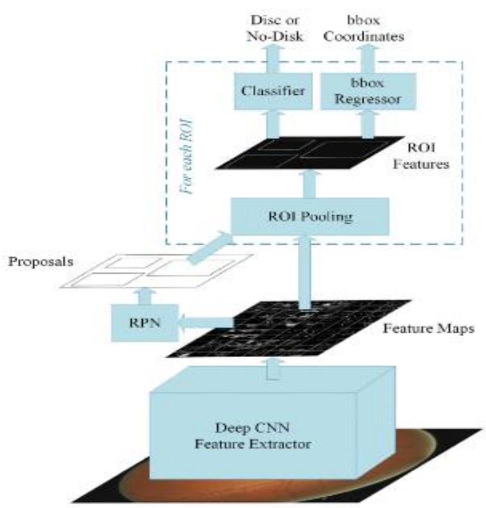

Figure 4.11: Internal components of Faster R-CNN

4.3.2 Architecture

The original architecture is properly explained in the original research paper of Faster R-CNN[1]. We mostly followed Girshick's code structure from his github repository[1], which is one of the publishers of Faster R-CNN. In our project we have used two kinds of base net: VGG16 and ResNet101. Initially we generate the bounding box for each pixel, and RPN try to extract foreground boxes and to adjust into shape. To eliminate the greatly overlapping boxes we use non-maximum suppression (NMS). It proposed bounding boxes to eliminate non anchor numbers.

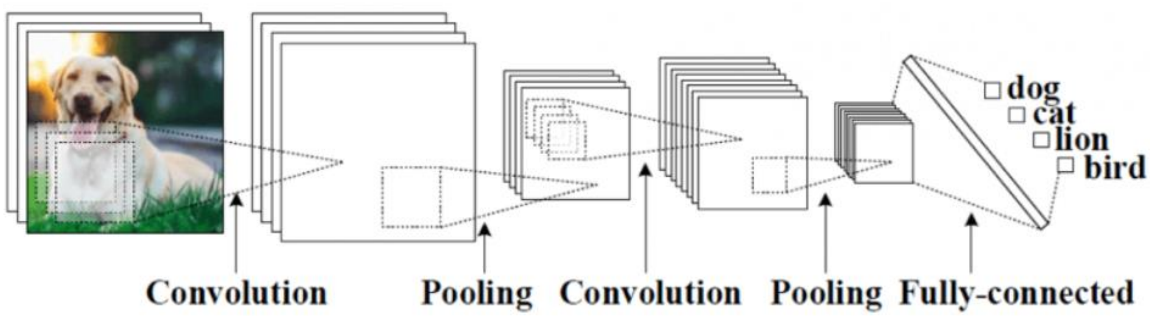

Figure 4.12: General CNN Architecture

The network, on the other hand, recommended RoI bounding boxes before identifying the classes for the associated NMS suppressed boxes. To extract the feature map into desired shape we use crop pooling. Crop pooling reduces the number of pixels from an image, it crops the RoI from the image. Crop pooling reduces the time complexity and runtime. The bounding box regression layer and the classifier are both linear densely integrated networks.

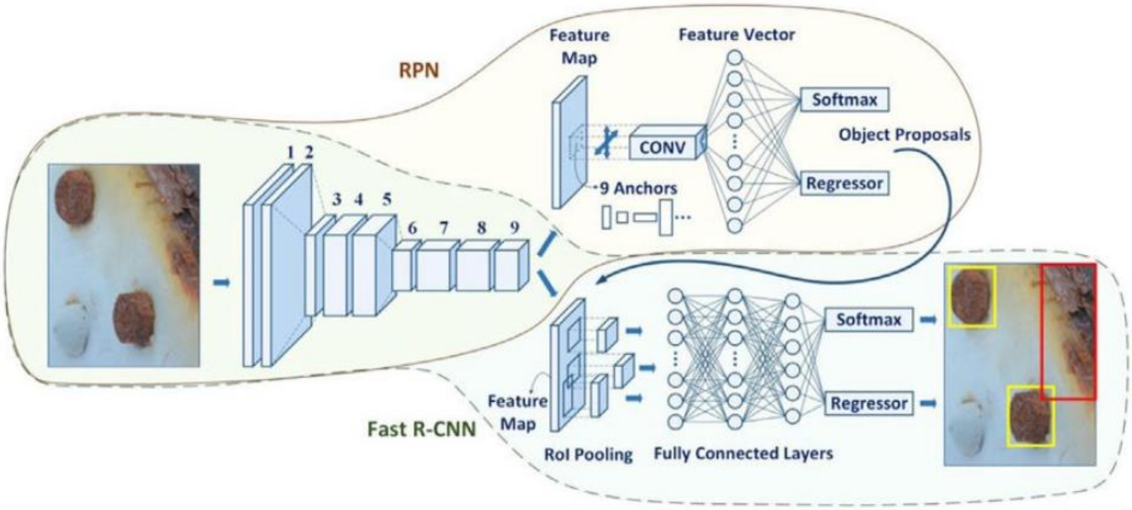

Figure 4.13: Faster R-CNN Architecture

4.3.3 Region Proposal Network (RPN)

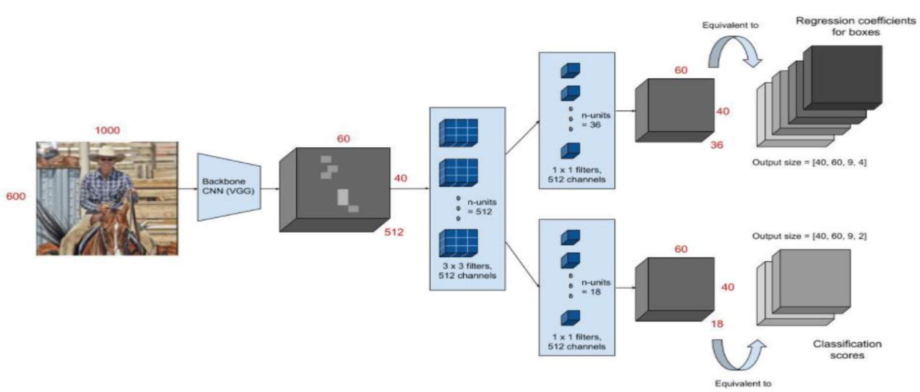

Figure 1: The architecture of the region proposal network or RPN

Figure 4.14: Region Proposal Network

In addition to the refers to how consistently score, the region proposal network is a deep convolutional system that uses restrictions and the objectiveness score. Trained RPN generates end-to-end proposals with a wide range of scales and aspect ratio. RPN and Fast R-CNN are merged together for object detection. It takes an image as input and returns a set of rectangular boxes with RoI objects in the image. We slide a network over the feature vector by the last convolution layer. Final output comes in the form of classification scores and regression coefficients for boxes.

4.3.4 Fast R-CNN

We feed input photos to CNN instead of region proposals to produce the feature map. Every time we need to feed 2000 region proposals, and hence it is faster than normal RCNN. We match only features that are generated from it only once. Fast R-CNN uses CPU based region proposal algorithms, Eg- the Selective search algorithm which takes around 2 seconds per image and runs on CPU computation. It has 17 convolution layers and 4 pooling layers. Convolution layer sizes vary in stages.

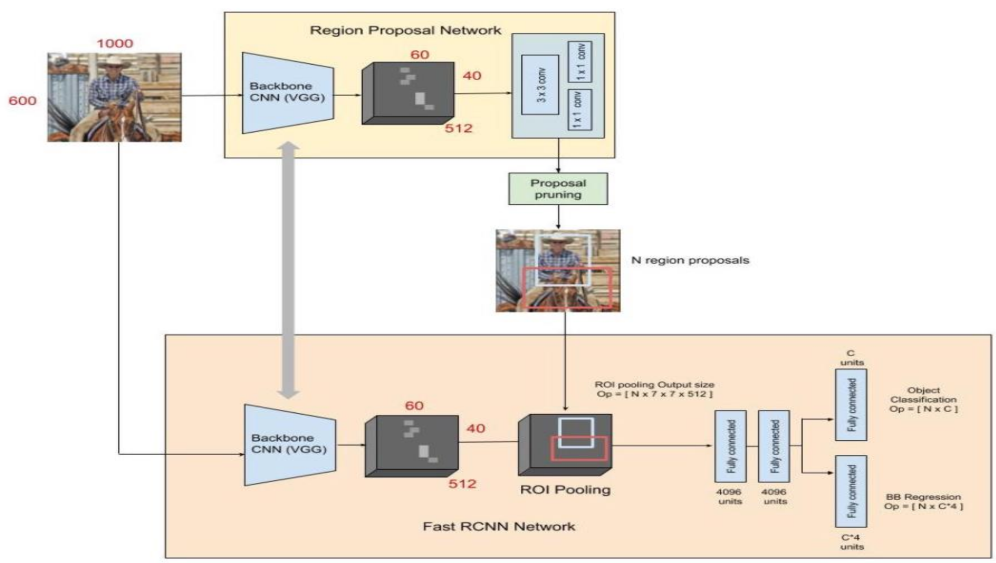

Figure 4.15: Fast R-CNN

Criteria	**R-CNN**	**Fast R-CNN**	**Faster R-CNN**
Region proposals method	Selective search	Selective search	Region proposal network
Prediction timing (sec)	40-50	2	0.2
Computation Time	High	High	Low
The mAP on Pascal VOC 2007 test dataset(%)	58.5	66.9 (only VOC 2007) 70.0 (both VOC 2007 and 2012)	69.9 (only VOC 2007)

Table 4.1 : Comparison of different Neural Networks

4.4 Machine Learning Algorithms

Machine Learning allows a machine to acquire new skills without having to be programmed. Machine learning algorithms are divided into two categories.

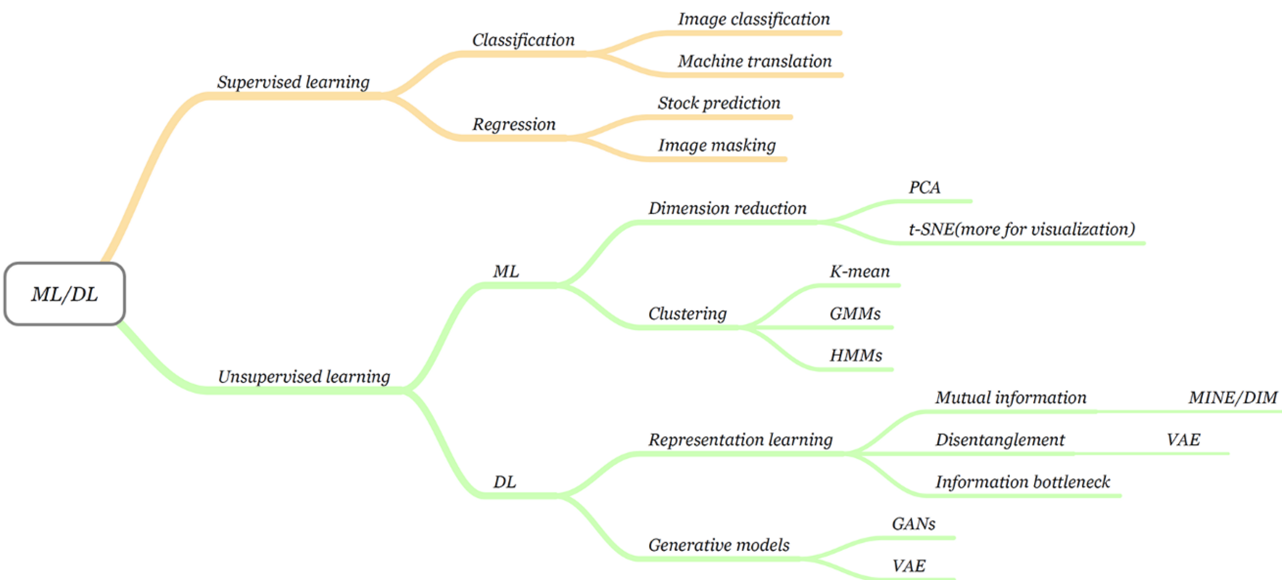

Figure 4.16: ML/DL Algorithms

4.4.1 Supervised Machine Learning Algorithms

In supervised learning, we use well-labelled data to train a machine learning model (data which have target variables).

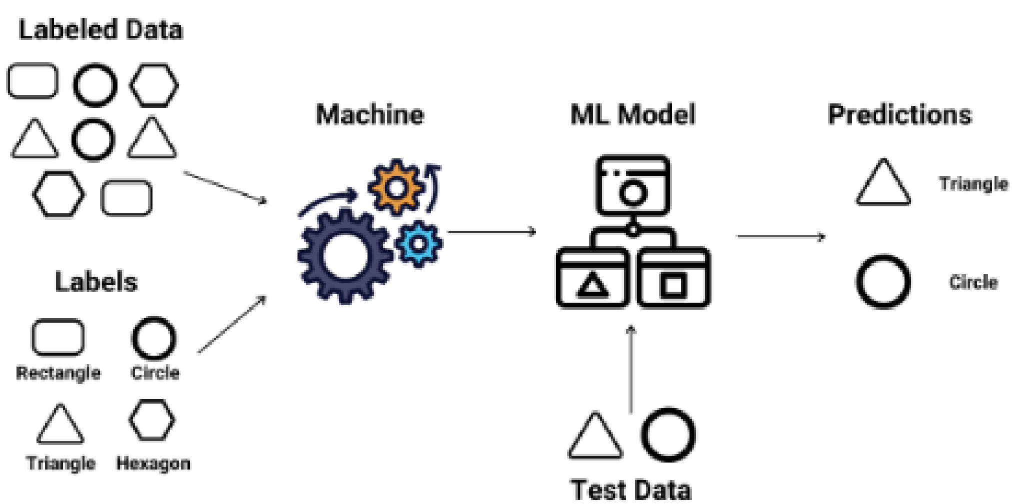

Figure 4.17: Supervised learning

1. Regression Algorithms : In Regression analysis, the target variable is continuous. There are a variety of algorithms that can be used as regression algorithms.
    - Linear Regression
    - Random Forest Regressor
    - XGboost Regressor
    - Light GBM Regressor
    - Decision Tree Regressor
    - Support Vector Regressor
2. Classification Algorithms : In classification analysis, the target variable the fixed, which may be binary classification or multi class classification. There are various classification algorithms.
    - Neural Network
    - Logistic Regression
    - Random Forest Classifier
    - K-nearest neighbour (KNN)
    - SVC
    - XGboost Classifier
    - Decision Tree Classifier
    - Naive bayes Classifier

4.4.2 Unsupervised Machine Learning Algorithms

The training dataset for unsupervised machine learning methods lacked target variables. Without any prior knowledge, the algorithms categorise unsorted data according to similarities, patterns, and differences.

- Principal Component Analysis
- K-means clustering
- Hierarchical clustering

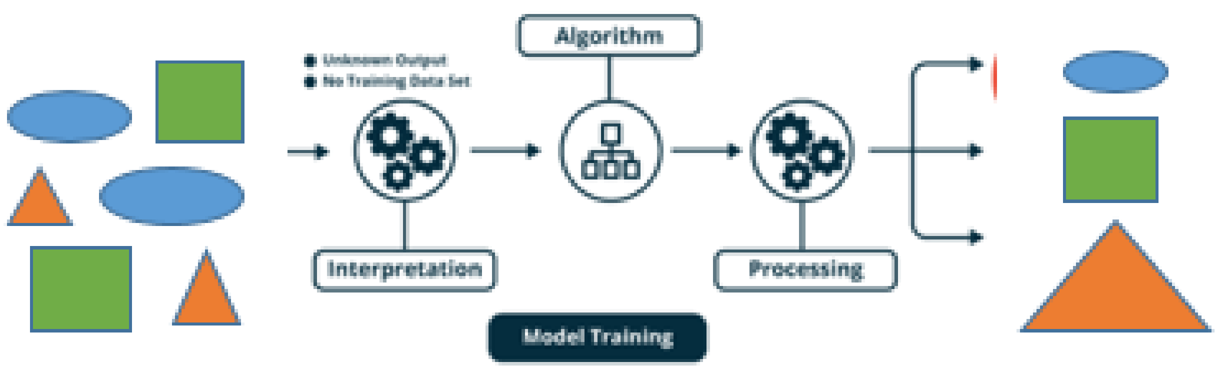

Figure 4.18: Unsupervised Learning

Here for face classification we are using Support vector Classifier algorithms.

Supervised Learning	Unsupervised Learning
Using labelled data, supervised learning algorithms are taught.	Unsupervised learning algorithms are taught on data that hasn't been labelled.
The supervised learning model uses direct feedback to determine whether it is correctly anticipating output.	There is no feedback in the unsupervised learning paradigm.
It predicts the output.	It finds the hidden patterns in data.
In this, input data is provided to the model along with the output.	In this, only input data is provided to the model.
The purpose of supervised learning is to train the model to predict the outcome when fresh data is provided.	Unsupervised learning aims to uncover hidden patterns and meaningful insights in an unknown dataset.
This needs supervision to train the model.	This does not need any supervision to train the model.

Table 4.2 : Difference Between Supervised & Unsupervised ML Learning

4.4.3 Support Vector Classifier (SVC)

This is a strategy that can be used to contribute to solving problems. SVM is primarily used to address classification problems. By the value of each element, which is the value of a certain combination, in the SVM approach, we turn each data object into a point in n-dimensional space (where n is the number of features you have). The splitting is then completed by locating a hyper-plane that effectively separates the two portions. Support Vectors are nothing more than linkages to certain places.

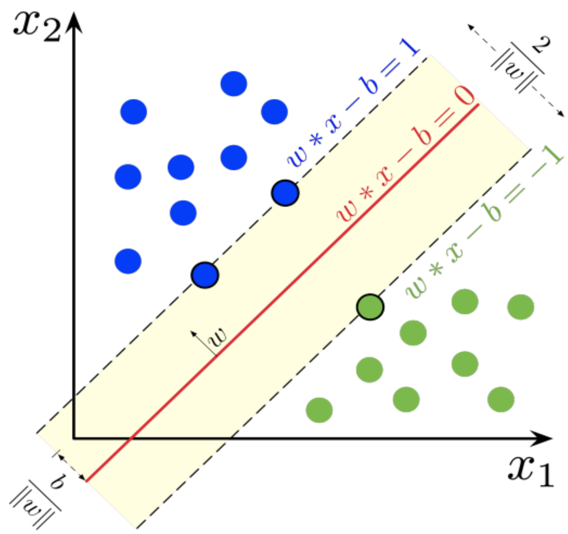

Figure 4.19: Support Vector Classifier

4.4.3.1 Hard-Margin SVM :

$$\begin{array}{ll} \underset{\mathbf{w}}{\arg\min} & \frac{1}{2}\mathbf{w}^\top \mathbf{w} \\ \text{s.t.} & y_i(\mathbf{w}_i^\top \mathbf{x}_i + b_i) \geq 1, \quad \forall i = 1, \dots, M \end{array}$$

4.4.3.2 Non-Linear SVM :

$$\begin{array}{ll} \underset{\boldsymbol{\alpha}}{\arg\min} & \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_i\alpha_j y_i y_j \mathbf{z}_i^\top \mathbf{z}_j + \sum_{i=1}^{M}\alpha_i \\ \text{s.t.} & \alpha_i \geq 0, \quad \forall i = 1, \dots, M \\ & \sum_{i=1}^{M} y_i\alpha_i = 0 \end{array}$$

4.4.3.3 Soft-Margin SVM :

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_{i=1}^{M} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}_i^\top \mathbf{x}_i + b_i) \geq 1 - \xi_i, \quad && \forall i = 1, \ldots, M \\ & \xi_i \geq 0, && \forall i = 1, \ldots, M \end{aligned}$$

4.4.3.4 Least Square SVM :

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \gamma \sum_{m=1}^{M} \xi_m^2 \\ \text{s.t.} \quad & y_m\left(\mathbf{w}^\top \phi(\mathbf{x}_m) + b\right) = 1 - \xi_m, \quad \forall m = 1, \ldots, M \end{aligned}$$

Chapter-5

MODEL TRAINING

Firstly we extracted frames from the real time video, we used the MTCNN (multi-task cascading convolution neural network) model for face detection in particular frames. If the person is present in the frame then move forward that frame for face recognition otherwise that frame will be rejected and will work on the next frame .

5.1 Face detection & extraction

```
# extract a single face from a given photograph
def extract_face(image, required_size=(160, 160)):
  pixels = np.asarray(image)
  detector = MTCNN()
  results = detector.detect_faces(pixels)
  x1, y1, width, height = results[0]['box']
  x1, y1 = abs(x1), abs(y1)
  x2, y2 = x1 + width, y1 + height
  face = pixels[y1:y2, x1:x2]
  image = Image.fromarray(face)
  image = image.resize(required_size)
  face_array = asarray(image)
  return (face_array, results)
```

Figure 5.1: Face detection & extraction

5.2 Load Model & Dataset

We use a pre-trained keras model. In the training dataset each person has 667 images, which are captured & extracted from webcam video.

The shape of training dataset is 3335 (667 * 5 classes). We created a .npz file for our training dataset.

Target Class in our dataset

- Ajeet (Class 0)
- Maneesh (Class 1)
- Prashant (Class 2)

- Pranjal (Class 3)
- Raj (Class 4)

```
model = load_model('/content/drive/MyDrive/Training Images/facenet/facenet_keras.h5')
print('Loaded Model')

data1 = load('/content/drive/MyDrive/Training Images/faces-dataset1.npz')
# load face embeddings
data = load('/content/drive/MyDrive/Training Images/faces-embeddings.npz')
```

Figure 5.2: Load model & Dataset

5.3 Embedding from extracted face

To generate the embedding of the face, first we normalise the value of the face matrix. Then use pre-trained keras models to generate embedding vectors for the images.

```
def get_embedding(model, face_pixels):
    # scale pixel values
    face_pixels = face_pixels.astype('float32')
    # standardize pixel values across channels (global)
    mean, std = face_pixels.mean(), face_pixels.std()
    face_pixels = (face_pixels - mean) / std
    # transform face into one sample
    samples = numpy.expand_dims(face_pixels, axis=0)
    # make prediction to get embedding
    yhat = model.predict(samples)
    return yhat[0]
```

Figure 5.3: Embedding from extracted face

5.4 Support Vector Classifier

We have a face embedding vector for a face. Now with the help of support vector classifier we have classified different vector in 5 classes

- Ajeet
- Maneesh

- Prashant
- Pranjal
- Raj

```
testX_faces = data1['arr_2']
trainX, trainy, testX, testy = data['arr_0'], data['arr_1'], data['arr_2'],
data['arr_3']
in_encoder = Normalizer(norm='l2')
trainX = in_encoder.transform(trainX)
testX = in_encoder.transform(testX)
      # label encode targets
out_encoder = LabelEncoder()
out_encoder.fit(trainy)
trainy = out_encoder.transform(trainy)
testy = out_encoder.transform(testy)
      # fit model
model1 = SVC(kernel='linear', probability=True)
model1.fit(trainX, trainy)
```

Figure 5.4: Support Vector Classifier

Chapter-6

MODEL EVALUATION

For model evaluation, we used 220 images per class. So there are a total of 1100 images.

6.1 Confusion Matrix :

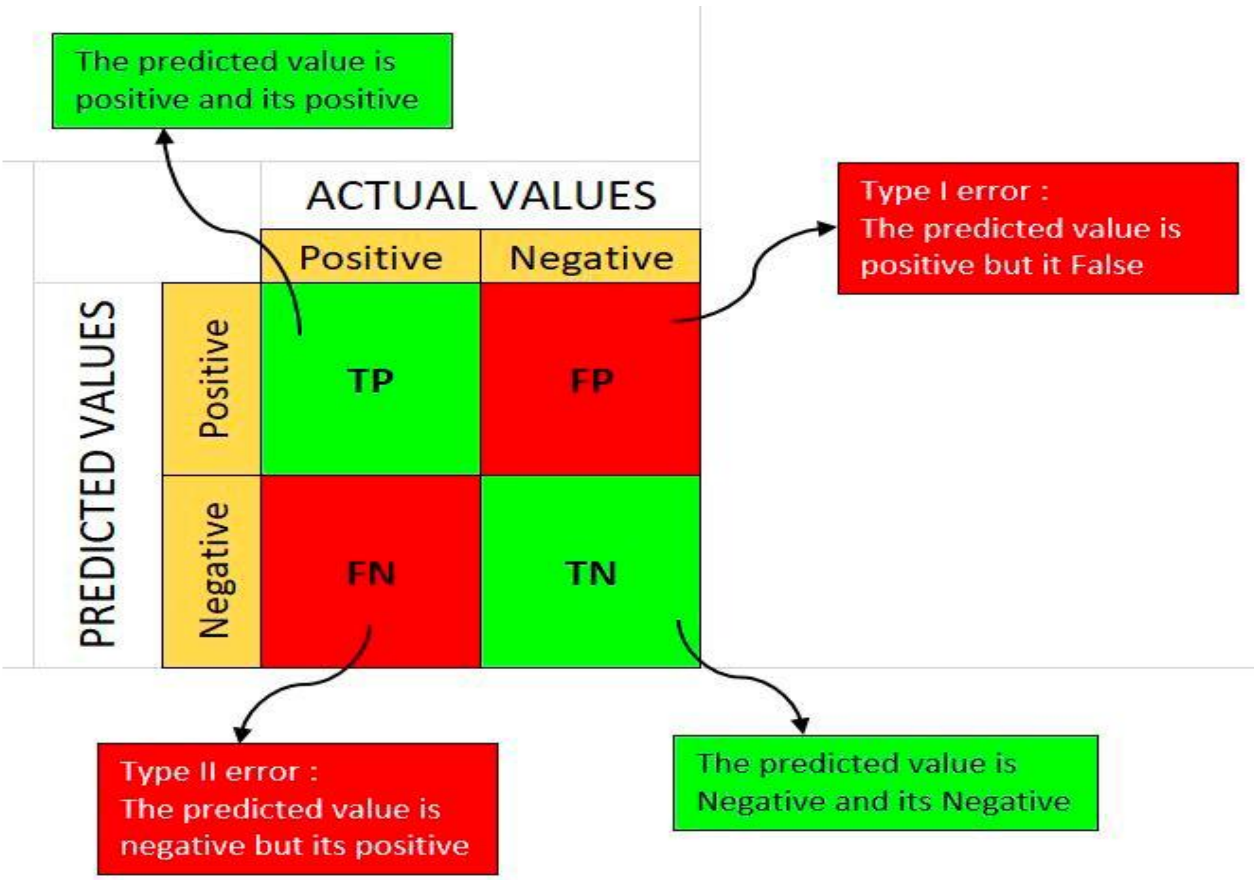

Figure 6.1: Confusion Matrix

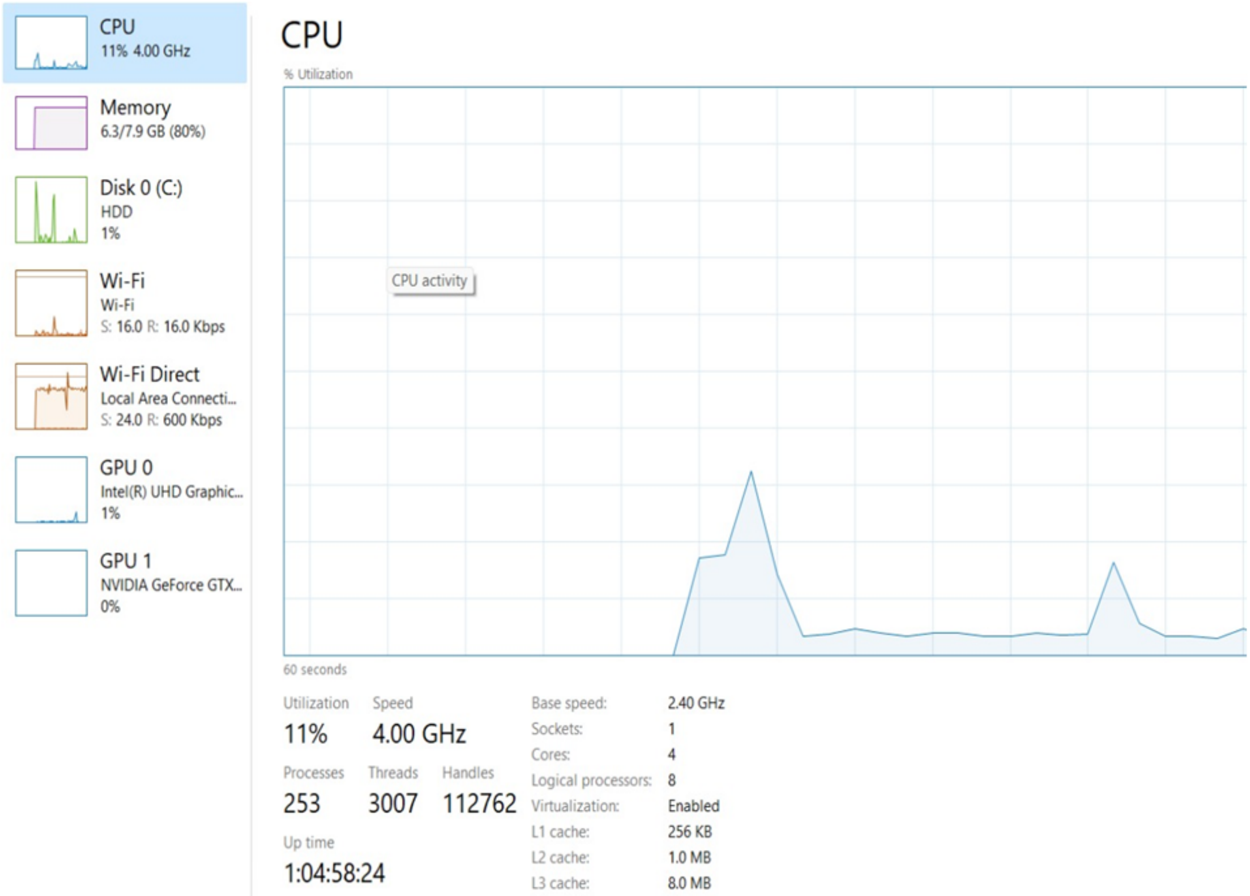

Figure 6.2: CPU Utilisation during Training

Validation Accuracy of Vgg19 model is 80.64%

```
#Train the network
model.fit(trainx, trainy, validation_data = (valx, valy), epochs=10, batch_size=32)     #
```

```
Epoch 1/10
105/105 [==============================] - 848s 8s/step - loss: 1.2230 - accuracy: 0.5061 - val_loss: 1.0677 - val_accuracy: 0.6709
Epoch 2/10
105/105 [==============================] - 789s 8s/step - loss: 0.4612 - accuracy: 0.8523 - val_loss: 2.0291 - val_accuracy: 0.7582
Epoch 3/10
105/105 [==============================] - 788s 8s/step - loss: 0.1906 - accuracy: 0.9410 - val_loss: 2.1925 - val_accuracy: 0.7318
Epoch 4/10
105/105 [==============================] - 787s 8s/step - loss: 0.1662 - accuracy: 0.9521 - val_loss: 1.5678 - val_accuracy: 0.8064
Epoch 5/10
 56/105 [==============>.............] - ETA: 5:39 - loss: 0.1073 - accuracy: 0.9738
```

Figure 6.3: Accuracy score of VGG-19

Validation Accuracy of Faster R-CNN model is 66.72%

```
6960/6960 [==============================] - 7s 957us/step - loss: 0.4348 - accuracy: 0.8563 - val_loss: 1.5258 - val_accuracy: 0.6599
Epoch 74/80
6960/6960 [==============================] - 7s 958us/step - loss: 0.4577 - accuracy: 0.8453 - val_loss: 1.4661 - val_accuracy: 0.6713
Epoch 75/80
6960/6960 [==============================] - 7s 958us/step - loss: 0.4427 - accuracy: 0.8586 - val_loss: 1.5110 - val_accuracy: 0.6615
Epoch 76/80
6960/6960 [==============================] - 7s 958us/step - loss: 0.4381 - accuracy: 0.8629 - val_loss: 1.4484 - val_accuracy: 0.6802
Epoch 77/80
6960/6960 [==============================] - 7s 960us/step - loss: 0.4016 - accuracy: 0.8657 - val_loss: 1.5276 - val_accuracy: 0.6599
Epoch 78/80
6960/6960 [==============================] - 7s 959us/step - loss: 0.4433 - accuracy: 0.8536 - val_loss: 1.5364 - val_accuracy: 0.6583
Epoch 79/80
6960/6960 [==============================] - 7s 958us/step - loss: 0.4168 - accuracy: 0.8678 - val_loss: 1.5727 - val_accuracy: 0.6477
Epoch 80/80
6960/6960 [==============================] - 7s 959us/step - loss: 0.3999 - accuracy: 0.8691 - val_loss: 1.5617 - val_accuracy: 0.6672
<keras.callbacks.callbacks.History at 0x7fcf626d1be0>
```

Figure 6.4: Accuracy score of Faster R-CNN

6.2 Model Output

After running these scripts, the webcam will be opened and the model will analyse real time video and recognise a person from a video. If the person is unknown it will produce the sound from the buzzer and send the mail to the user.

```
# start streaming video from webcam
thres = 65
url = "https://9cbb-2409-4063-6e80-b006-4425-6741-3eb4-f9bf.in.ngrok.io"

video_stream()
# label for video
label_html = 'Capturing...'
# initialze bounding box to empty
bbox = ''
count = 0
while True:
    js_reply = video_frame(label_html, bbox)
    if not js_reply:
        break

    # convert JS response to OpenCV Image
    frame = js_to_image(js_reply["img"])
    #img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    im_pil = Image.fromarray(frame)
    # load the photo and extract the face
    try:
      pixels, results = extract_face(im_pil)
      yhat = get_embedding(model, pixels)


      newTestX = list()
      newTestX.append(yhat)
      newTestX = numpy.asarray(newTestX)


      # normalize input vectors
      in_encoder = Normalizer(norm='l2')
      trainX = in_encoder.transform(newTestX)

      yhat_class = model1.predict(trainX)
      yhat_prob = model1.predict_proba(trainX)


      asli_name = yhat_class[0]
      class_probability = yhat_prob[0,asli_name] * 100
      if(class_probability > thres):
        predict_names = out_encoder.inverse_transform(yhat_class)
        print('Predicted: %s (%.3f)' % (predict_names[0], class_probability))
        requests.post(f"{url}/user",{'first_name':predict_names[0],'last_name':''})
      else:
        print('Unknown')
        display(IPython.display.Audio(url="https://www.soundjay.com/buttons/sounds/beep-01a.mp3", autoplay=True))
        sleep(2)
        display(IPython.display.Audio(url="https://www.soundjay.com/buttons/sounds/beep-01a.mp3", autoplay=True))

        requests.post(f"{url}/user",{'first_name':"Unknown",'last_name':''})
      #print('Expected: %s' % random_face_name[0])
      #im_pil.show()
    except:
      pass
```

Figure 6.5: Model Output

6.3 Output from real time video (Webcam)

This is the output of the model, which is predicting the person's name on the real time video. Here

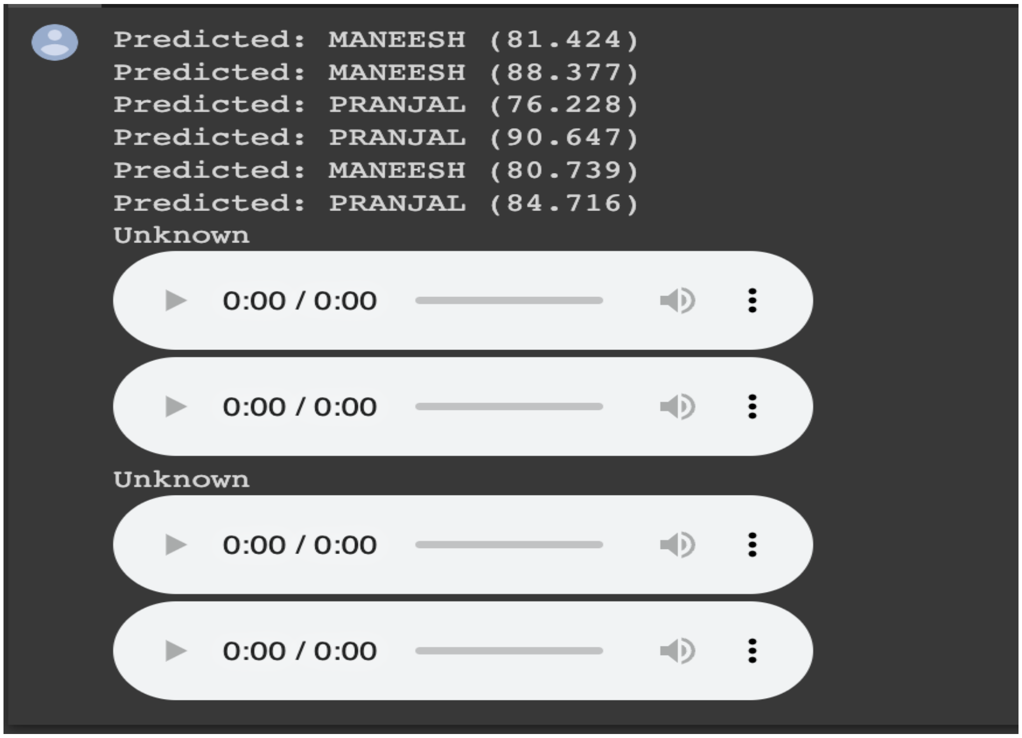

Figure 6.6: Output from real time video (Webcam)

Chapter-7

DJANGO

7.1 Django framework

Django is a Python-based website building framework. Using Django Developer creates more complex data based websites easily. For quick development, Django emphasises the reusability of components. As a result, we can concentrate on the speedy development of our project. Django employs the MVT (model, view, template) model.

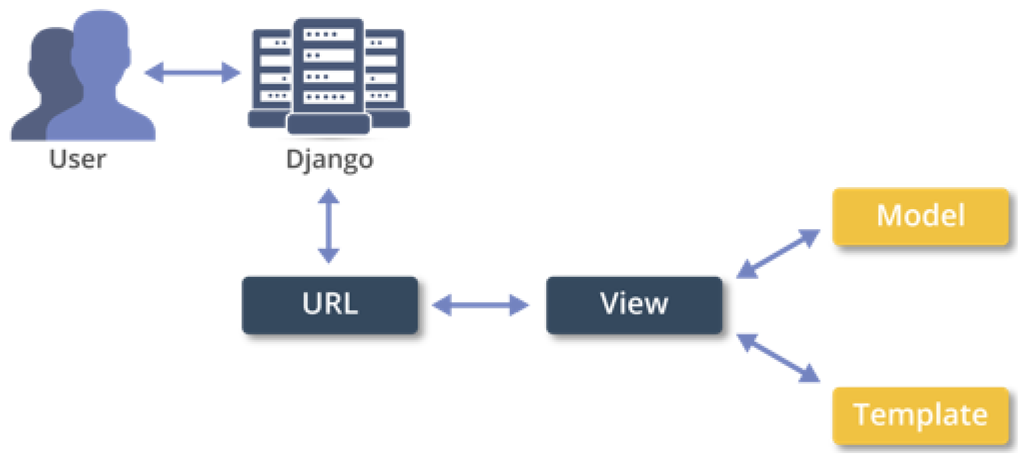

Figure 7.1: Django Framework

It is free and open source, with extensive documentation and a large user community.

7.1.1 Model

The Django model is a structure of the data, It contains information, relation and its behaviour about the data. The Django model creates a table in the database by default it uses sqlite3 but we can choose any database.

Attributes defined in the model represent fields in the database table. Django provides an API(application programming interface) for users.

7.1.2 View

A view in Django is a function (Class) that handles web requests and responds to them. Requests can be to GET,POST,PUT or DELETE and we return the response according to that.

Web response can be some data, web page, HTML, XML, image or some error messages.

Here we operate on data. To link the view function to the URL we can use urlpatterns that are given by django or we can use routers given by django rest framework.

7.1.3 Template

Django provides developers to generate HTML Dynamically. We can generate form HTML using django.forms. We can directly put data in form from our model directly and show it to the user.

It also provides advanced techniques to extend HTML pages to reuse HTML code like other programming languages.

We can add some links or any HTML modification using the mark_save() method provided in django templates.

Chapter-8

DataBase Management & User Interface

Our result of the user interface is that we store the data coming from the Machine learning model and show it to the end user.

In the database we store the person's information when it comes in front of the camera. And our user can see all the data later and it can also see the live stream of the camera feed.

8.1 Create the Database table

To begin, we must first establish a database for our system. All of the project's tables feature information about a known person, as well as the date and time of their visit.

Initially we store some information about the person but most of the datafield are updated dynamically.

Most databases, including postgreSQL, sqlite3, oracle, and mysql, are supported by Django. In this project we use sqlite3. For this we only need to tell Django that we are going to use sqlite 3 for our database.

DATABASE_ENGINE = ’sqlite3’

Next we create a person and person visit table in our database. For this, django provides a class method named ‘model’. We derive this class and make our own model class Person and PersonVisit that store personal information and person_visit information. Here is an example:

The below code creates a table called person which stores information about a person. The table has five columns: Name corresponding to the name of the class.

```
class Person(models.Model):
    id = models.BigAutoField(primary_key=True)
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    image = models.ImageField(upload_to = 'photos/%y')
    roll_no = models.IntegerField()
    captured_on = models.DateTimeField(auto_now_add=True)

    @property
    def short_description(self):
        return truncatechars(self.description,20)

    def admin_photo(self):
        return mark_safe('<img src="{}" width="200" />'.format(self.image.url))
    admin_photo.short_description = "Image"
    admin_photo.allow_tags = True

    def __str__(self) -> str:
        return self.first_name+self.last_name
```

Figure 8.1: Person class

This table stores the information about when a known person comes into the camera frame.

This table is referenced from the Person table and store date and time. When we delete a person from a model then the information stored about that person in this table is also delete.Because person field is a foreignkey in this table.

```python
class PersonVisit(models.Model):
    id = models.BigAutoField(primary_key=True)
    person = models.ForeignKey(Person,on_delete=models.CASCADE)
    captured_onn = models.DateTimeField()

    def __str__(self) -> str:
        return str(self.id)
```

Figure 8.2: PersonVisit class

This table simply stores images of unknown person and the admin can see these images and can delete them.

```python
class Unknown(models.Model):
    id = models.BigAutoField(primary_key=True)
    captured_on = models.DateTimeField(auto_now_add=True)
    image = models.ImageField(upload_to = 'unknown/%y')
    @property
    def short_description(self):
        return truncatechars(self.description,20)

    def unknown_photo(self):
        return mark_safe('<img src="{}" width="100" height="50"/>'.format(self.image.url))
    unknown_photo.short_description = "Image"
    unknown_photo.allow_tags = True
```

Figure 8.3: Unknown person class

This is the function based api view function that provides HTTP GET and HTTP POST method. User list function returns the information about known person with the help of personseralizer class.

When the api hit endpoint /user with post request with this will create a person id that not present in table if person present in table then it create a instance in personvisit table.

```python
@api_view(['GET','POST'])
def user_list(request):

    if(request.method == 'GET'):
        user = Person.objects.all()
        seralizer = PersonSeralizer(user,many=True)
        return Response(seralizer.data)

    elif request.method == 'POST':
        data = request.data
        person,createdy = Person.objects.get_or_create(
            first_name = data["first_name"],
            last_name = data["last_name"],
            roll_no = data["roll_no"],
            image = data["image"],
        )

        PersonVisit.objects.create(person=person,captured_onn=datetime.datetime.now())
        data = json.dumps(data)
        return Response(data,status=status.HTTP_201_CREATED)
```

Figure 8.4: API point to send and receive user data

This is the code for the admin page where I show the person and unknown list. In the person list we can see the full information about the person and can see when a person comes to the camera frame.

In an unknown model we only show the images of an unknown person with name with date and time.

```python
class PersonVisitInline(admin.TabularInline):
    model = PersonVisit
    list_display = ("person","captured_onn","id")
    readonly_fields = ("person","captured_onn","id")

@admin.register(Person)
class PersonAdmin(admin.ModelAdmin):
    inlines = [PersonVisitInline]
    list_display= ("first_name","last_name","id","captured_on","roll_no",'admin_photo')
    readonly_fields=("id","captured_on")

@admin.register(Unknown)
class Unknown(admin.ModelAdmin):
    list_display = ("id","captured_on","unknown_photo")
    readonly_fields = ("id","captured_on","unknown_photo")
```

Figure 8.5 Admin panel

Serializer class helps us to decode and encode data that we get from requests and also convert it to json format to which browsers understand.

```python
class PersonSeralizer(serializers.ModelSerializer):
    class Meta:
        model = Person
        fields = ['id','first_name','last_name','captured_on','roll_no','admin_photo']

class PersonVisitSeralizer(serializers.ModelSerializer):
    class Meta:
        model = PersonVisit
        fields = ['id','person','captured_onn']

class UnknownVisitSeralizer(serializers.ModelSerializer):
    class Meta:
        model = Unknown
        fields = ['id','captured_onn',"image"]
```

Figure 8.6: Seralizer functions

8.2 Admin User Interface

With the help of admin user interface, **Admin** can see the details of any person (timestamp and frequency of a person) with their image.

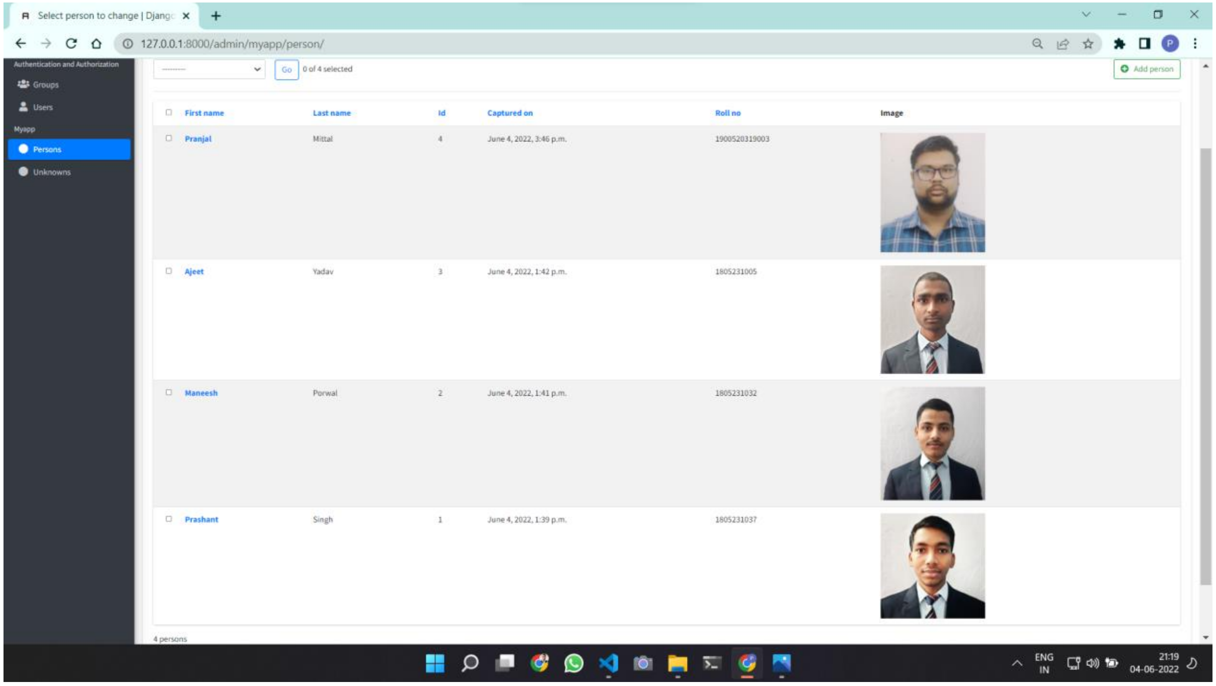

Figure 8.7: Admin panel of person

8.3 About Us

This is our about us page, in which we have our mentor and team details and our team vision & mission.

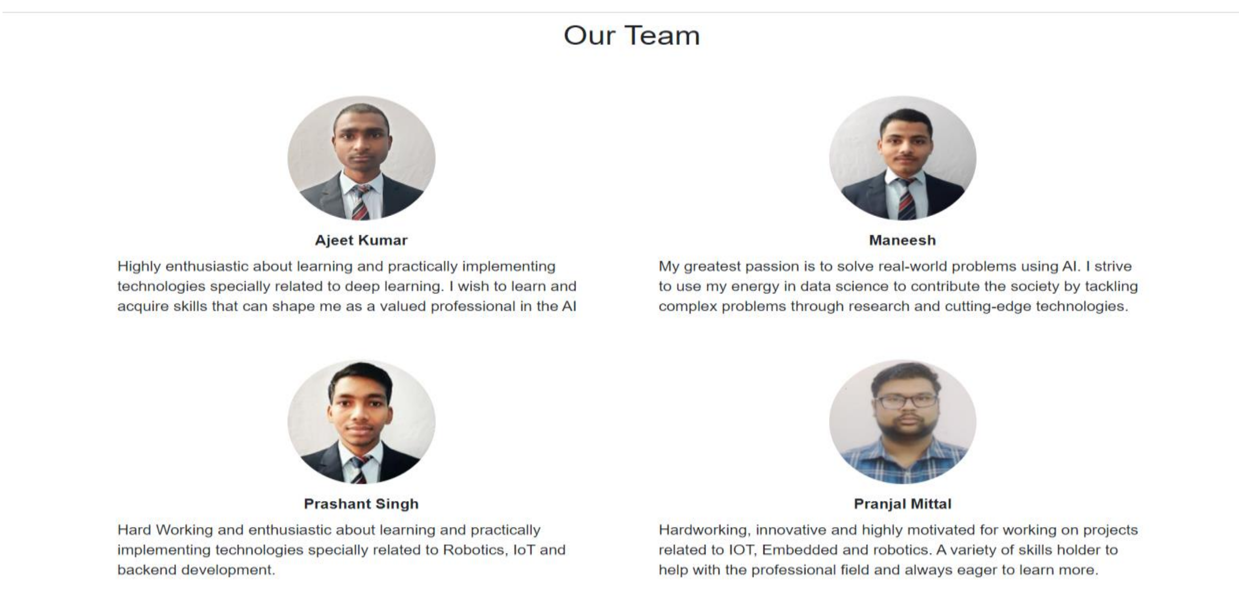

Figure 8.8: About_us page

8.4 Contact us

This is our contact us page, where we have the address, email and phone number of our team. If a person has any query/message can send us via this form.

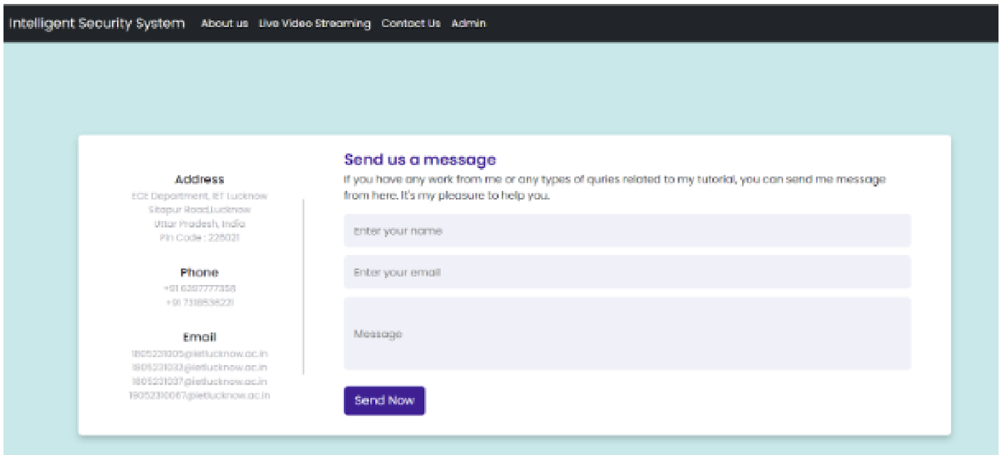

Figure 8.9: Contact_us page

Chapter-9

APPLICATION AREAS

Surveillance systems are used to explore environments that are difficult for human beings to explore. Some of the application areas are discussed below:

1. Remote Video Monitoring
2. Traffic Monitoring
3. Public Safety
4. Employee Safety
5. Loss Prevention
6. Facility Protection
7. Monitor Operations
8. Physical access to work environments
9. Aids in the search for children who have gone missing.
10. Allow customers to understand the current process and to have their credit/debit cards authenticated.
11. Allow students to log their attendance and gain access to labs.
12. Control access to a construction site's specific location.
13. Provides access to university facilities such as residence halls, common areas, and cafeterias, among other things.
14. Control and track who has access to certain areas, including employees, visitors, vendors, and maintenance personnel.
15. At airports, paperless travel is possible.
16. Process for controlling vehicle admission and exit.
17. Multiplex cinema access

Face recognition and intelligent surveillance have a wide range of applications. Applications can be divided into two categories: blacklist and whitelist. Security and surveillance applications, as well as criminal identification applications, are examples of blacklist applications. All other applications are categorised as whitelist apps, such as access control, attendance tracking and so on.

Chapter-10

OVERVIEW

The data management problem is concerned with building an intelligent system that can identify and separate wanted & required data from the whole dataset in an unknown environment, while incrementally rejecting the unwanted data to store in the dataset. It has been formulated and solved by the Artificial Intelligence community and several algorithms (for example,MTCNN) already exist.

We made use of deep facial detection and recognition algorithms to tackle the problem of person knowledge checking in this project. The system has a four-stage pipeline (e.g., motion blur detection, face detection, landmark detection, and face recognition). Furthermore, the system is equipped with a user-friendly interface that makes it simple for both administrators and users to engage with it. Despite the low-resolution webcam on most laptops, the application performs appropriately on our private dataset. This illustrates that our core algorithm is capable of dealing with the problem of low-quality input. In the future, we plan to expand our dataset so that it is asymptotic to real-world applications.More methods will also be examined in order to increase the algorithm's ability to discern feature distributions of output classes.

Most computer and robot vision systems rely heavily on object detection. Although significant progress has been made in recent years, and some real applications have been integrated into many electronics products (for example, face detection for auto-focus in smartphones) or assistant driving technological advances, we are indeed a long way from human-level performance, notably in open-world learning. It's important to note that object detection isn't frequently used in many areas where it may be quite valuable. As mobile robots and autonomous machines in general become increasingly common, object detection systems are becoming more crucial (e.g., quadcopters, drones, and eventually service robots).Finally, object detection systems will be required for nano-robots or robots that will start exploring previously unexplored areas, such as deep water depths or other planets, and these monitoring systems will be expected to learn new item classes as they would be encountered. The ability to learn in a real-time open-world environment would be critical in such situations.

Chapter-11

CONCLUSION

The terrorist activities carried out worldwide, majorly including the events of September 11th, 2001, as well as the subsequent Boston marathon incident, have enhanced public awareness of security challenges and the need of intelligent automated video surveillance systems used in public, commercial, law enforcement, and military settings.

Computerised video analysis of this study is commonly associated with analysis of video streams recorded by surveillance systems in a variety of fields, including various businesses, governmental institutions, utility plants, municipalities, agencies, financial firms, educational institutions, urban transport hubs, medical centres, and industries. It uses video real-time analysis to detect occurrences of interest in real time, as well as forensic video analysis to extract events and data from pre-recorded video.

As use of such video surveillance systems spreads and the portion of recorded video increases, so does the need to explore through video feed data and extract specific segments and events of interest. In many situations, time is of the essence, and the evaluation must be completed efficiently and quickly. This is a very time consuming, inefficient, and tiresome operation when done manually by human operators. Given that an average human operator tasked with observing video screens in an analogue video surveillance system cannot stay aware and sensitive for more than twenty minutes, and also that the operator's capacity to supervise the video as well as successfully react to situations is significantly harmed as time passes, operators prove to be an expensive commodity with limited alertness and attention. As a consequence, sophisticated automated video surveillance systems that monitor cameras and alert users to potentially dangerous situations are becoming increasingly enticing and popular.

When a video surveillance system is combined with sophisticated video analysis, the value of the system skyrockets. Surveillance systems are more practical and effective with intelligent automated video surveillance systems. Intelligent automated video surveillance systems have a wide range of applications: Security and intrusion management applications ensure perimeter control for sensitive or restricted areas such as permit parking and limited-access buildings, where security officers are alerted to automatically detect and track burglary or a suspicious individual loitering in the parking lot; urban surveillance applications detects highway accidents, measure traffic flow and monitors pedestrian congestion in public spaces; crowd management, in which it compiles consumer demographics in shopping malls and amusement parks, monitors for the build-up of crowds in public areas based on occupancy measurements, ensuring that customers are not impacted by uncomfortable and potentially unsafe crowds; industry applications, in which it performs quality control and counts the number of products on production lines; measuring refugee flows in troubled areas, Patrolling national borders, providing secure perimeters around bases and embassies, battlefield surveillance applications, real-time moving object detection and tracking from stationary and moving camera platforms, recognition of objects such as humans and different types of vehicles, human activity recognition, human gait analysis, vehicle tracking and counting, airborne surveillance, and other military applications are just a few examples.

The numerous advantages of automated intelligent video surveillance systems have led to an increase in the use of surveillance systems in metropolitan areas, schools, districts, workplaces, apartment complexes, storage facilities, retail locations, financial institutions, food outlets, and housing projects to monitor movement, detect illegal activity, and protect the public. According to a survey, only 13 urban police agencies in the United States used CCTV video surveillance systems in 1997. These systems were largely used to monitor pedestrian traffic and catch cars running red lights. This figure skyrocketed after September 11, 2001.Private surveillance cameras monitor public areas such as sidewalks, parking lots, freeways, and traffic lights in every city in America today. Many of these cameras are connected to the Internet, allowing real-time film of people in public places to be seen online. Thousands of Web cameras spread photographs of street scenes and renowned monuments around the world, allowing the Internet observer to decide who, what, and how closely the intended subject is seen. Singapore, Canada, the United States, Australia, China, the United Kingdom, and other European countries use public video surveillance to monitor population movements in order to prevent crime and possible terrorism. According to one figure, the average British citizen gets photographed by 300 different cameras in a single day, making the United Kingdom the world's leading country in terms of CCTV video monitoring. All of the positive aspects of video surveillance systems raise other questions and concerns, such as privacy issues, conceptual or theoretical factors, strategy questions concerning video surveillance, legal and social measurements, expense, return on capital, quality standards, and so forth (Nieto, Johnston-Dodds, & Simmons, 2002; Bharucha, London, Barnard, Wactlar, Dew, & Reynolds, 2006; Collins, et al., 2000; Scherr, 2007). A sophisticated automated video surveillance system records continuously without being distracted. Digitally recorded videos and images are saved eternally and can be seen at any time, from any location, and as many times as required. As a result, this information can be modified and used for reasons other than those intended by the image collector or the subject. As a result, the general public, some government leaders, and a variety of organisations are becoming increasingly worried about individual privacy. Continuous video monitoring was regarded by courts and legislators for three decades, until the late 1990s, as a type of "passive" surveillance that did not infringe on an individual's privacy and did not pose a substantial legal hurdle. The fast adoption of sophisticated video surveillance systems in recent years has had a considerable impact on the appropriate use of video and associated surveillance technologies, as well as their consequences for civil liberties and privacy rights. The rapid advancement of video surveillance technology has hampering judicial and legislative institutions' ability to create and implement legislation that protect the public's security and right to privacy. Global use of growing surveillance technology, law enforcement, national security interests, and citizen privacy rights are among the newly raised questions that will undoubtedly be imposed. Physical privacy, decisional privacy, and informational privacy are the three types of privacy covered by the law, and all three are handled by video surveillance systems. According to some polls, Americans are willing to give up some privacy in exchange for greater security, but they are concerned about video surveillance data being misused by government agencies, enterprises, or employers, as well as the chance of mistaken identification due to computer faults. The ethical concerns of detection systems on citizens' privacy, as well as a rational approach to the central moral dilemma about confidentiality and privacy safeguards, important considerations of the informed consent process, and the issue of finding the right balance among law enforcement, public safety, and personal rights, is well beyond the scope of this text and should be addressed by professionals in fields such as sociologists, legislators, and lawyers, are beyond even the scope of this text and should be addressed by professionals in fields such as sociologists, legislators, and lawyers. Video Surveillance Methods and Approaches introduces a number of unique methods and algorithms for analysing video autonomously with the goal of detecting, tracking, and recognising moving objects based on image sequences. Anthropometric measurements, military aviation radar, ISAR image processing, and face and iris biometric techniques are also included. Although the text is aimed primarily at graduate scholars and practitioners active in the sector of signal and image processing with applications in static/moving detection and recognition, tracking, and identification, I hope that it will also be useful to those working in the field of video surveillance and security. The instructional and informational material in the book will benefit the intended readers, while the general public will gain a knowledge of the algorithms and intelligent software that form the core of intelligent automated video surveillance systems. Video surveillance, military applications, homeland security, quality control, healthcare, industry, and safety are just a few of the areas and applications where the book's practical applicability shines. The presented mathematical algorithms are utilised in a variety of security video surveillance systems, including some with top importance and priority, such as homeland security and national defence. These algorithms could be tweaked to perform additional tasks like identification, behaviour analysis, and other sorts of context - aware applications, which will be the subject of my future study.

Chapter-12

FUTURE SCOPE

Video Surveillancing or intelligent security based on face recognition technology is being used all over the world, and it is having a positive impact. Why is India being left out? This technology has a lot of potential in India, and it can help the country progress in a variety of ways. The technology and its applications can be used in a variety of industries across the country.

- In India, preventing ATM fraud is a priority. A database of all ATM cardholders in India may be built, and facial recognition technology might be used. As a result, every time a user approaches an ATM, his photo will be taken and compared to a database photo before granting access.
- In India, duplicate voters can be reported.
- Passports and visas can also be verified using this technology.
- The same method can also be used to verify a driver's licence.
- Technology can be utilised to improve surveillance and security in the defence department, airports, and other key locations.
- It can also be used to identify candidates during examinations such as the Civil Services Exam, SSC, IIT, MBBS, and others.
- This method can be used to verify and track attendance in a variety of government and corporate settings.
- It can also be implemented in bank lockers and vaults for access control verification and authentication of authentic users.
- The technique can potentially be utilised by police forces to identify offenders.

Facial recognition technology has a promising future. Forecasters predict that this technology will grow at a rapid pace and create significant money in the next few years. The primary categories that will be heavily influenced are security and surveillance. Private companies, public structures, and schools are among the other domains that are now welcoming it with open arms. It is expected that businesses and banking institutions would embrace it in the future years to prevent fraud in debit/credit card purchases and payments, particularly those made online.

This method would close the gaps in the widely used yet insecure password system. Robots with facial recognition technologies may potentially make an appearance in the future. They can assist in the completion of tasks that are inconvenient or difficult for humans to do.

REFERENCES

[1] R. Girshick, Fast R-CNN, arXiv:1504.08083v2, 27 Sep 2015.

[2] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv:1506.01497v3, 6 Jan 2016.

[3] T. Karmarkar(2018, Aug 19), Region Proposal Network (RPN) — Backbone of Faster R-CNN,[Online].Available:

https://medium.com/egen/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9

[4] T. Ren, K. He, R. Girshick, and J. Sun, Fast R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv:1506.01497v3, 10 Feb 2018

[5] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10):1499–1503

[6] Florian Schroff, Dmitry Kalenichenko, James Philbin FaceNet: A Unified Embedding for Face Recognition and Clustering ICLR 2015

[7] Karen Simonyan ∗ & Andrew Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition ICLR 2015

[8] Bishop, C.M. (1995) Neural Networks for Pattern Recognition, Oxford: Oxford University Press.

[9] Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. MIT Press.

[10] Steve Schulman of Wall Street Software Digest, Software Reviews, Traders Catalog & Resource Guide, Vol. I, No. 7, October, 1993.

[11] Carlton F. Vogt Jr., BrainMaker: This Will Teach Your Computer to Think,Design News, December 3, 1990.

[12] Caffe: Convolutional Architecture for Fast Feature Embedding

[13] Templates[Online].Available:https://docs.djangoproject.com/en/4.0/topics/templates/ [Accessed: 25- May- 2022].

[14] Writingviews[Online].Available:https://docs.djangoproject.com/en/4.0/topics/http/views/ [Accessed: 25-May-2022]

[15] Yongli Zhang, Support Vector Machine Classification Algorithm and Its Application, Hebei United University, China, ICICA 2012

[16] Mustamo P. (2018) Object detection in sports: TensorFlow Object Detection API case study. University of Oulu, Degree Programme in Mathematical Sciences. Bachelor's Thesis, 43 p.

[17] Z. Zhu, P. Luo, X. Wang, and X. Tang. Recover canonicalview faces in the wild with deep neural networks. CoRR, abs/1404.3543, 2014. 2

[18] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In IEEE Conf. on CVPR, 2011. 5

[19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In IEEE Conf. on CVPR, 2014. 1, 2, 5, 7, 8, 9

[20] Y. Sun, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. CoRR, abs/1406.4773, 2014. 1, 2, 3

[21] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In Proc. ECCV, 2014. 7

[22] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12:2121–2159, July 2011. 4

[23] M. Lin, Q. Chen, and S. Yan. Network in network. CoRR, abs/1312.4400, 2013. 2, 4, 6

PLAGIARISM REPORT

Final_year_report_plag

ORIGINALITY REPORT

19%	11%	11%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www-mechanicalengineering.blogspot.com Internet Source	4%
2	www.coursehero.com Internet Source	1%
3	Submitted to University of Hertfordshire Student Paper	1%
4	Pranjali Singh, Amritpal Singh. "Chapter 43 Unmasking the Masked Face Using Zero-Shot Learning", Springer Science and Business Media LLC, 2022 Publication	1%
5	Submitted to Landmark University Student Paper	1%
6	Submitted to The University of Wolverhampton Student Paper	1%
7	Olutosin Taiwo, Absalom E. Ezugwu, Olaide N. Oyelade, Mubarak S. Almutairi. "Enhanced Intelligent Smart Home Control and Security System Based on Deep Learning Model",	1%

Wireless Communications and Mobile Computing, 2022
Publication

8	www.frontiersin.org Internet Source	<1 %
9	www.ncbi.nlm.nih.gov Internet Source	<1 %
10	Min Yoon, Yeboon Yun, Hirotaka Nakayama. "Sequential Approximate Multiobjective Optimization Using Computational Intelligence", Springer Science and Business Media LLC, 2009 Publication	<1 %
11	Submitted to Babes-Bolyai University Student Paper	<1 %
12	Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Yu Qiao. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks", IEEE Signal Processing Letters, 2016 Publication	<1 %
13	Submitted to Laguna State Polytechnic University Student Paper	<1 %
14	atmeye.com Internet Source	<1 %

15	matheo.uliege.be Internet Source	<1%
16	archives.univ-biskra.dz Internet Source	<1%
17	towardsdatascience.com Internet Source	<1%
18	www.jetir.org Internet Source	<1%
19	Submitted to Higher Education Commission Pakistan Student Paper	<1%
20	Submitted to King's College Student Paper	<1%
21	idr.mnit.ac.in Internet Source	<1%
22	Submitted to SUNY, Empire State College Student Paper	<1%
23	ethesis.nitrkl.ac.in Internet Source	<1%
24	unbscholar.lib.unb.ca Internet Source	<1%
25	www.aiktcdspace.org:8080 Internet Source	<1%

26	Submitted to Malaviya National Institute of Technology Student Paper	<1%
27	Submitted to Henry M. Jackson High School Student Paper	<1%
28	Submitted to Kamla Nehru Institute of Technology Sultanpur Student Paper	<1%
29	Dušan Teodorović, Katarina Vukadinović. "Traffic Control and Transport Planning:", Springer Science and Business Media LLC, 1998 Publication	<1%
30	Submitted to Liverpool John Moores University Student Paper	<1%
31	Submitted to Queen Mary and Westfield College Student Paper	<1%
32	www.h2kinfosys.com Internet Source	<1%
33	Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, Xindong Wu. "Object Detection With Deep Learning: A Review", IEEE Transactions on Neural Networks and Learning Systems, 2019 Publication	<1%

34	Submitted to Citizens' High School Student Paper	<1 %
35	Submitted to University of Sydney Student Paper	<1 %
36	www.mdpi.com Internet Source	<1 %
37	"Disciplinary Convergence in Systems Engineering Research", Springer Science and Business Media LLC, 2018 Publication	<1 %
38	cxymm.net Internet Source	<1 %
39	pdfcoffee.com Internet Source	<1 %
40	pub.epsilon.slu.se Internet Source	<1 %
41	"Big Data Analysis and Deep Learning Applications", Springer Science and Business Media LLC, 2019 Publication	<1 %
42	oregonsigmanu.com Internet Source	<1 %
43	webthesis.biblio.polito.it Internet Source	<1 %

www.webindustry.com

44	Internet Source	<1 %
45	Submitted to Aberystwyth University Student Paper	<1 %
46	Submitted to The University of Memphis Student Paper	<1 %
47	analyticsindiamag.com Internet Source	<1 %
48	cse.anits.edu.in Internet Source	<1 %
49	dokumen.pub Internet Source	<1 %
50	downloads.hindawi.com Internet Source	<1 %
51	hdl.handle.net Internet Source	<1 %
52	tel.archives-ouvertes.fr Internet Source	<1 %
53	www.repo.uni-hannover.de Internet Source	<1 %
54	www.sciencepub.net Internet Source	<1 %
55	www2.eecs.berkeley.edu Internet Source	<1 %

56	"Biometric Recognition", Springer Nature, 2016 Publication	<1 %
57	K. Kranthi Kumar, Y. Kasiviswanadham, D.V.S.N.V. Indira, Pushpa Priyanka palesetti, Ch.V. Bhargavi. "Criminal face identification system using deep learning algorithm multi-task cascade neural network (MTCNN)", Materials Today: Proceedings, 2021 Publication	<1 %
58	Saúl Cano-Ortiz, Pablo Pascual-Muñoz, Daniel Castro-Fresno. "Machine learning algorithms for monitoring pavement performance", Automation in Construction, 2022 Publication	<1 %
59	Guanglu Liu. "Research on Personalized Minority Tourist Route Recommendation Algorithm Based on Deep Learning", Scientific Programming, 2022 Publication	<1 %
60	Xiaobing Han, Yanfei Zhong, Liangpei Zhang. "An Efficient and Robust Integrated Geospatial Object Detection Framework for High Spatial Resolution Remote Sensing Imagery", Remote Sensing, 2017 Publication	<1 %

61	"Computer Vision and Image Processing", Springer Science and Business Media LLC, 2020 Publication	<1%
62	en.wikipedia.org Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		