Analysis of Crowd Dynamics

Project report submitted in partial fulfilment of the requirement
for the degree of Bachelor of Technology

in

Computer Science and Engineering

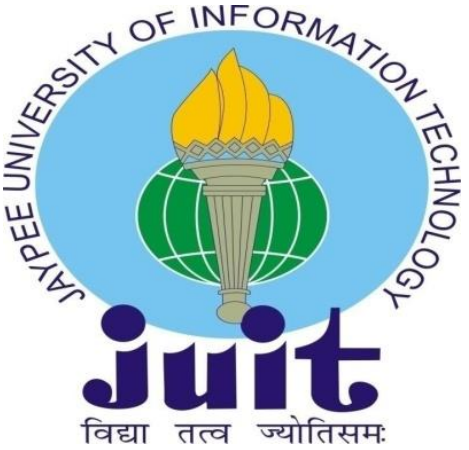
By

Sanyam Saxena [191214]

Under The Supervision of

Dr. Ruchi Verma

to



Department of Computer Science & Engineering and
Information Technology

**Jaypee University of Information Technology,
Waknaghat, 173234, Himachal Pradesh, INDIA**

Certificate

I hereby declare that the work presented in this report entitled **"Analysis of Crowd Dynamics"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Dr. Ruchi Verma** (Assistant Professor (SG), Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat).
I also authenticate that I have carried out the above mentioned project work under the proficiency stream **Information Security**.
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sanyam Saxena [191214]

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Ruchi Verma
Assistant Professor (SG)
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 13 May 2023

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report ✓ | Paper

Name: SANYAM SAXENA Department: CSE Enrolment No 191214

Contact No. 9473919787 E-mail. SANYAMSAXENA2187@GMAIL.COM

Name of the Supervisor: Dr. RUCHI VERMA

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ANALYSIS OF CROWD DYNAMICS

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 45
- Total No. of Preliminary pages = 10
- Total No. of pages accommodate bibliography/references = 3

Sanyam

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 15 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
			Word Counts	
Report Generated on				
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Acknowledgement

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor **Dr. Ruchi Verma, Assistant Professor (SG),** Dept. of CSE & IT, Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of **Deep Learning** to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Ruchi Verma, Assistant Professor (SG),** Dept. of CSE & IT, for her kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Sanyam Saxena
191214

Table of Content

S. No.	Title	Page No.
1.	Certificate	I
2.	Plagiarism Certificate	II
3.	Acknowledgement	III
5.	List of Figures	VI
6.	List of Abbreviations	VII
7.	List of Tables	VIII
8.	Abstract	IX
9.	Chapter-1 (Introduction) 1.1 Introduction 1.2 Objective 1.3 Motivation 1.4 Methodology 1.4.1 Problem Definition 1.4.2 Problem Analysis 1.4.3 Solution 1.4.4 Date Set used in the Project 1.4.5 Types of Data Set 1.4.6 Description of the data set 1.4.7 Steps of the solution 1.5 Language Used 1.6 Technical Requirements (Hardware) 1.7 Deliverables/Outcomes 1.8 Data-Flow Diagram (DFD)	01 – 11
10.	Chapter-2 (Literature Survey)	12 – 14
11.	Chapter-3 (System Development) 3.1 Requirements 3.1.1 Functional Requirements 3.1.2 Non-Functional Requirements	15 – 19

	3.2 RSA Algorithm 3.3 VGG Model 3.4 Transfer Learning	
12.	Chapter-4 (Performance Analysis) 4.1 System Properties 4.2 Performance	20 - 21
13.	Chapter-5 (Results & Conclusions) 5.1 Discussion on the Results Achieved 5.2 Applications of the Project 5.3 Limitations of the Project 5.4 Future Work	22 - 23
14.	References	24 – 26
15.	Appendices	27 - 35

List of Figures

Figure No.	Figure Details	Page No.
1.	A depiction of the crowd scene (Google Images)	01
2.	Recognition of Criminals (Edited by me, from Google)	02
3.	An example of Live Recognition	02
4.	Encryption Image	03
5.	Dataset for face recognition	08
6.	Ndarray of an image	09
7.	Data flow diagram (Google Images)	11
8.	RSA Working (Google Images)	17
9.	VGG model (VGG Neural Network Architecture)	18
10.	Concept of Transfer Learning (Google Images)	19
11.	Flow chart of the whole recognition system (edited by me, from Google)	27
12.	Timeline for the project (Made by me on visme)	28
13.	Code for Cryptography	29
14.	Code for Cryptography	30
15.	Code for Encryption & Decryption	30
16.	When query picture is searched in a video	31
17.	When query video is searched in a video	31
18.	When query picture is searched in multiple pictures	32
19.	Real Time Face Recognition	32
20.	Program_1	33
21.	Program_2	33
22.	Identification	34
23.	Live Recognition	35

List of Abbreviations

Abbreviations	Full form
RSA	Rivest-Shamir-Adleman
VGG	Visual Geometry Group
LBSN	Location Based Social Network
FPS	Frames per Second
RGB	Red Green Blue
CMYK	Cyan, Magenta, Yellow, Key (black)
Ndarray	N-dimensional Array
LFW	Labelled Faces in the Wild
YTF	YouTube Faces in the Wild
CCTV	Closed Circuit Television
GUI	Graphical User Interface

List of Tables

Table Number	Table Details	Page No.
1.	Accuracy of few pre-trained models (Source: Official deepface repository 2020)	20
2.	Table 2: Various metrics of few pre-trained models (Source: Serengil 2020)	21

Abstract

Many times, criminals are spotted in the recordings of the surveillance systems but because of lack of technical advancement and lack of manpower, authorities don’t get to know that their surveillance systems spotted any wanted criminal. Because of this many wanted criminals are free and are left unpunished. To put these wanted criminals behind the bars I propose an idea to identify them and inform the authorities immediately. This will help us to identify criminals as soon as possible and at the same time will save a lot of time and manpower. With the help of technology, I want to solve this problem and help the authorities.

As the dataset is confidential, consisting of pictures of criminals, it is required to be accessed by the authorized user only. Thus, for the sake of security, I have used cryptography (encryption) to attain confidentiality and integrity.

Chapter 01: Introduction

1.1 Introduction

The main purpose of this project is to provide authorities [Cyber Cell] with a user-friendly video analytics platform to analyse the crowd dynamics using surveillance systems.

Data collected from these surveillance systems is analysed and reported in case of any identification or live recognition. Initial phase of this project majorly focuses on providing solutions to all the objectives stated below.

For video surveillance system, currently, I have created the dataset by myself so that I can test this platform, later I wish to take a dataset from cyber cell so as to practically implement this project and solve this real-world problem. The data-set will consist of footages from cameras installed in the society. The footages are pixelated due to the use of low resolution cameras.

Methodology for the whole the video surveillance system is, firstly, preprocessing takes place, query image is preprocessed. Then this pre-processed image is matched with the dataset using different prebuilt models and if any matching occurs, authorities are signalled so as to take the appropriate action.

This system would act as an automated monitor to identify all the wanted criminals.

Figure 1: A depiction of the crowd scene (Google Images)

Figure 2: Recognition of Criminals (Edited by me, from Google)

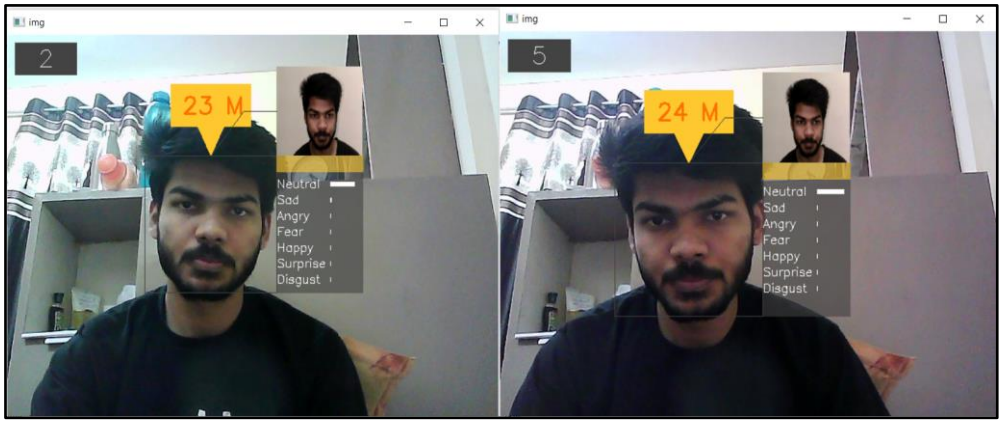


Figure 3: An example of Live Recognition

For encryption and decryption I have used RSA algorithm. RSA (Rivest–Shamir–Adleman) is a widely used public-key cryptosystem for secure communication. Ron Rivest, Adi Shamir, and Leonard Adleman created it in 1977. The RSA algorithm requires creating a pair of public and private keys, of which the private key is used by the intended receiver to decrypt the message and the public key is usable by anybody to encrypt messages.

The RSA algorithm's security is predicated on how challenging it is to factor the product of two huge prime numbers. The system is more secure the greater the prime numbers utilised to create the key pair. Numerous applications, like as secure email communication, online banking, and e-commerce transactions, require RSA. The RSA algorithm is still a crucial instrument for secure communication in the digital age and has grown to be one of the most utilised and researched encryption algorithms in the world.

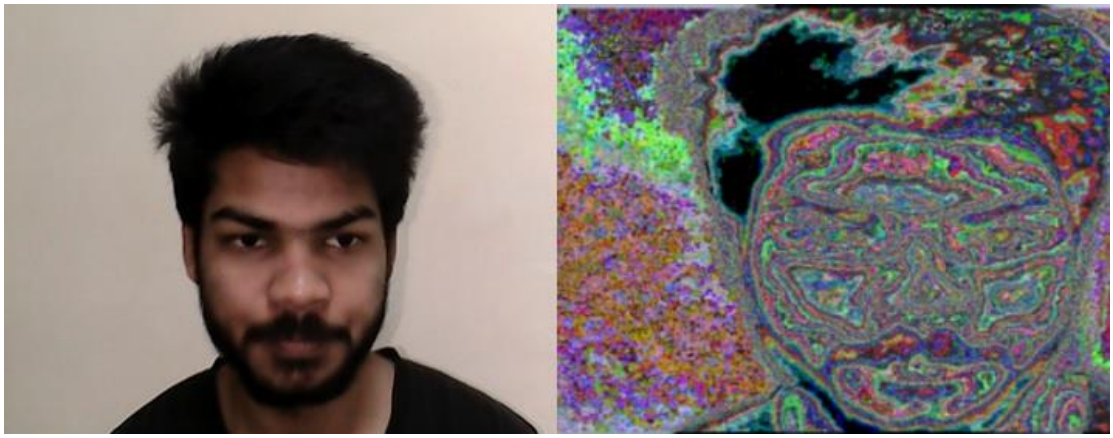


Figure 4: Encryption Image

1.2Objective

1. Identification.
    1.1. Identification of a person in a video recording using a picture. (Positive/negative outlook both)
    1.2. Identification of a person in a video recording using a small video clip/gif. (Positive/negative outlook both)
    1.3. Identification in a list of known criminals.
2. Live Recognition.

1.3 Motivation

Even after using high quality surveillance systems, authorities are not able to identify any wanted criminal even if they get spotted. The lack of technical advancement and lack of manpower led authorities in a helpless state. Because of this many wanted criminals are free and are left unpunished.

I want to help the authorities to get rid of this helplessness. This is the main motivation behind this idea. To help the authorities so that they can put these wanted criminals behind the bars Ipropose an idea to identify wanted criminals and inform the authorities immediately.

This will help us to identify criminals as soon as possible and at the same time will save a lot of time and manpower. This is the main inspiration behind the whole idea.

1.4 Methodology

1.4.1 Problem Definition

Analysis and identification of hidden crime patterns are the main problems for the police as there is a large amount of crime data. So I need some methods to help the investigating agency to solve the crimes.

Criminals are often discovered in the records of surveillance systems, but due to a lack of technological advances and manpower, authorities are unaware that their surveillance systems have detected a wanted criminal. Because of this, many wanted criminals go free and go unpunished. To put these wanted criminals behind bars Ipropose an idea to identify them and report them to the authorities immediately. This will help us identify criminals as quickly as possible while saving a lot of time and manpower.

1.4.2 Problem Analysis

CCTV cameras have been put in both public and private spaces as a result of advancements in security technology to provide surveillance. After a crime, no matter how big or small, has been committed, it is challenging for the criminal investigation department to assess all the video footage from public cameras. It becomes crucial to learn the criminal's past and movements. The objective is to complete this task quicker with ML than with any manual technique. Understanding the criminal activities occurring on and finding suspects depend on CCTV footage. Currently, this strategy involves spending time manually searching for these people on CCTV surveillance footage. Due to the low resolution of such CCTV cameras, the process takes a while. The suggested system is being created to loop over real-time surveillance photographs and identify

criminals based on reference criminal records as a solution to these issues. It is advantageous to use facial recognition technology to locate offenders. The real-time cropped image of the identified offender is saved when the closest match is found, and authorised personnel can access it to find and monitor the perpetrators or conduct additional investigations.

1.4.3 Solution

When a security expert inputs an image of the suspect into a facial recognition system for criminal identification, the system pre-processes the image to eliminate distracting components like noise. Following that, the algorithm categorises the photographs using cues like the separation between the eyes and the length of the chin line. The report is then displayed after the system searches the database for a precise match.

The success of this method depends on two factors. Detection and detection. One of the most crucial components of a face recognition system is face recognition, which can be broken down into four primary categories. Approaches based on knowledge, features, template matching, and look.

Machine learning approaches provide regression and classification techniques to help you achieve the goals.

In the planning phase, the system is planned. This phase also explains why and how the system is created. It is divided into two steps:

1. To gather facial photos to be used as a template for the project start system is the subject of a preliminary analysis.
2. Project Planning- Choose the best softwares and technology for detection.

Face recognition entails the use of automatic recognition followed by image or video-based person verification. Face recognition has

been explored extensively, yet there are still issues to be solved, such as:

- Misalignment
- Posture changes
- Lighting variations
- Expression variations

1.4.4 Date Set used in the Project

Currently, I have created the dataset by myself so that I can test this platform, later I wish to take a dataset from cyber cell so as to practically implement this project and solve this real-world problem. The data-set will consist of footages from cameras installed in the society (town). These footages are pixelated due to the use of low resolution cameras.

Some glimpse of the Data:

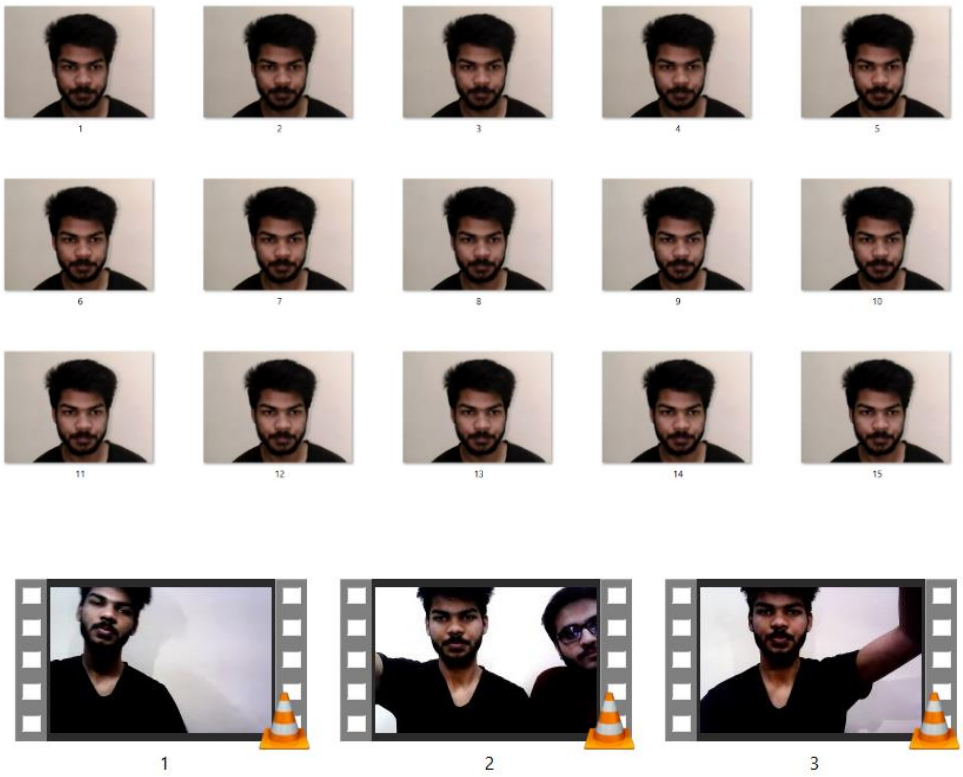


Figure 5: Dataset for face recognition

1.4.5 Types of Data Set

In this project, I have used images and videos as the data set, because I have to recognise criminals from the recorded videos and in real time as well.

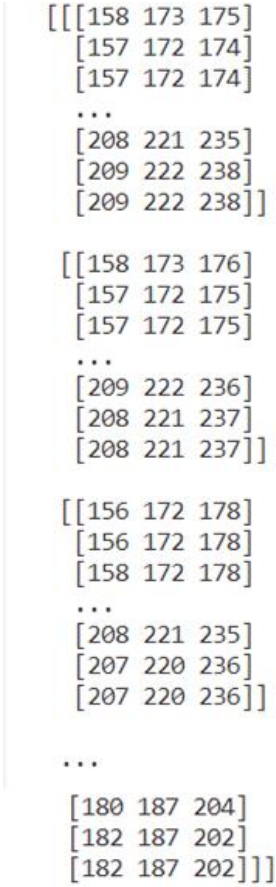


Figure 6: Ndarray of an image

1.4.6 Number of Attributes, fields, description of the data set

I have used RGB (coloured) images as accuracy of black & white images is very less. Also, I can use CMYK images or any coloured images. I have captured approximately 500 images of some individuals in different angles and made the dataset of several people. Also, I made some videos to test this system.

1.4.7 Steps of the solution for the project problem

1. Data Acquisition
2. Data Preprocessing
3. Feature Extraction
4. Model Building
5. Training
6. Testing
7. Deployment

1.5 Language Used

Python, a general purpose language, is used to make this project. Many packages of python are used to make this project a success. For Ex.: opencv, deepface, numpy, matplotlib, etc.

1.6 Technical Requirements (Hardware)

1. A high quality camera which can capture videos. [30FPS and 720p]
2. A computer with high processing power. [More the processing power, less will be the time for computation/results.]

1.7 Deliverables/Outcomes

1. Criminal Identification was successfully done in CCTV recordings with all types of media [video and picture] with a high accuracy.
2. Live Criminal Recognition was successfully done with a high accuracy.
3. Encryption and Decryption of media dataset was achieved.

1.8Data-Flow Diagram (DFD)

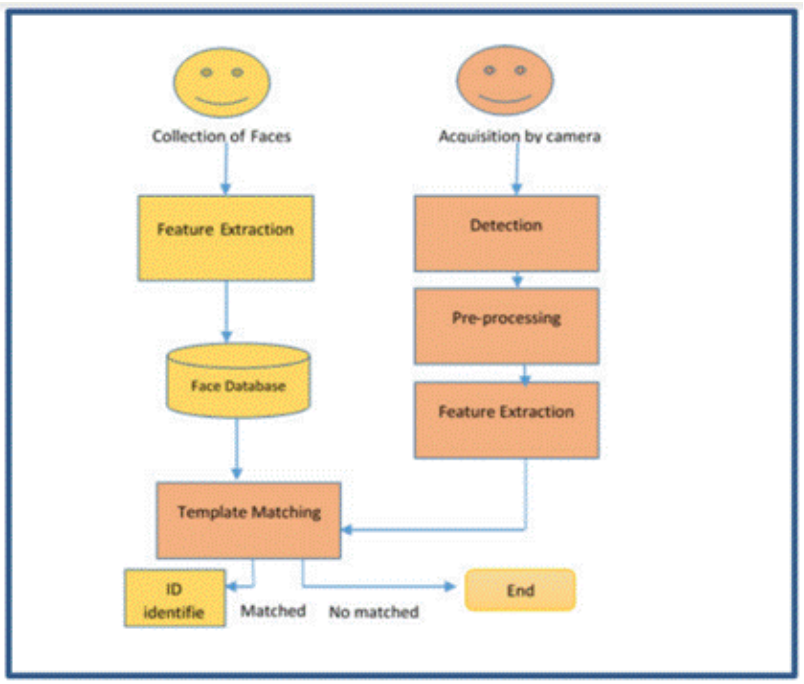


Figure 7: Data flow diagram (Google Images)

Chapter 02: Feasibility Study

Literature Survey

To learn more about the topic, several publications that have been produced in the disciplines of object, face, and facial recognition have been researched. Image processing, object recognition, and neural networks. The publications that have an impact on some of the suggested methodological designs and flows are summarised in this section.

[0] The use of prospective fingerprints might provide problems for the current methods of recognising fingerprints, which are straightforward and simple to use, and they might not be obtained from crime scenes. Criminals grow more intelligent and are often very cautious while leaving their fingerprints at a crime scene. For matching face feeds to faces in the database, the system had a face database and an image recognition algorithm.

The conventional pipeline for current face recognition [1] consists of four steps. Detection ⇒ Alignment ⇒ Display ⇒ Classification. Apply piecewise affine transformations using 3D explicit face modelling to derive facial expressions from a 9-layer deep neural network, repeating both alignment and rendering steps.

The most popular sub-technique for this method is face detection, but I also employed skin detection [2]. The Haarcascade classifier was the algorithm employed. An ellipse is used to model each face. Haar face detection is sped up using skin detection.

Each frame is examined during this multi-step procedure [3]. The Haar classifier made the initial detection. The Eigen face and Gabor algorithms were used to identify the face in the second stage, and decision-making and selection were performed in the third step.

A biometric procedure based on facial photos is used to recognise a person using face recognition [4].

Face recognition technology with computer assistance has been developed. Face recognition [5] is a technique that incorporates automated recognition followed by picture or video-based person verification.

To provide a consistent context, the author employed a technique-based approach [6]. To eliminate light interference in this piece, they employed two cameras. They employed a variety of techniques, including backdrop subtraction.

The value of a single basic attribute is used to categorise the majority of photos [7]. Features usually operate significantly quicker than pixels, hence it is always preferable to employ features rather than pixels. There are three intermediary phases in the algorithm:
A) The characteristics of the integration rectangle may be determined extremely rapidly by employing the intermediate version of the picture.
B) The Adaboost approach may be used to create a classifier that aids in separating the relevant feature from a huge collection of features. Use a variety of both favourable and adverse pictures to practise.
C) A series of various classifiers

To categorise target variables into several groups, classification techniques like KNeighbors Classification are utilised. Next, a neural network with input, high-density, and output layers may be used to increase the precision of predictions. The model forecasts criminal accounts like age, gender, and relationship with the victim based on these algorithms. As a result, it is anticipated that this approach would make it easier for police to handle criminal cases. [8]

The crime dataset was analysed using k means clustering by the authors of study [9]. Rapid mining software is used to create this model. Plotting the numbers across time allows for analysis of the clustered results. Thus, the model deduces from the study that there were fewer killings between 1990 and 2011.

The authors of [10] projected the places with a high likelihood of crime and depicted crime-prone zones. The Naive Bayes classifiers algorithm, which is a supervised learning and statistical approach for classification and has delivered 90% accuracy, was used by the authors to categorise the data.

On the Communities and Crime Dataset, Lawrence McClendon and Natarajan Meghanathan (2015) [11] employed a variety of prediction algorithms, including Linear Regression, Additive Regression, and Decision Stump methods utilising the same set of input (features). Comparing the three chosen methods, the linear regression approach produced the best results overall. The key benefit of the linear regression approach is that it can deal with some unpredictability in the test data while only introducing a 15% prediction error.

Using clustering algorithms, Rasoul Kiani, Siamak Mahdavi, et al. (2015) [12] established a framework for crime prediction. The tool RapidMiner is used to achieve this. GA (Genetic Algorithm) is used to identify outliers in the data in order to improve prediction accuracy. The accuracy of the results from this model is 91.64%.

A methodology was put up by Chirag Kansara, Rakhi Gupta, and colleagues in 2016 [13] that examines Twitter users' sentiments and forecasts whether they would represent a threat to a certain individual or society. Naive Bayes Classifier, used in this model, uses sentiment analysis to categorise the subjects.

Chapter 03: System Development

3.1 Requirements

In order to develop the suggested system, requirement analysis is the process of examining both functional and non-functional demands. Functional requirements outline what the system must be able to accomplish as well as how it must help users carry out and complete activities. The system must adhere to quality limits that are non-functional requirements as stated in the project contract. Non-functional requirements are also known as non-behavioural requirements.

3.1.1 Functional Requirements

This system is a user friendly tool which helps the users to identify and recognize people with respect to the dataset either from a video or a picture. Also, it is used for live recognition too. Further I'll add more features to this platform to increase its accuracy and make it more efficient and add different ways of data acquisition.

3.1.2 Non-Functional Requirements

This system will work on any device which supports python programming language but requires high computation power. It requires some time for processing and computing results based on the size of dataset. Higher the computing power is, less will be the time required.

3.2 RSA Algorithm

RSA encryption algorithm is a public-key encryption technique.

Public key encryption algorithm:
Public Key encryption algorithm is also called the Asymmetric algorithm. Asymmetric algorithms are ones in which the encryption and decryption keys used by the sender and the receiver are distinct. A set of keys is given to each sender:

- Public key
- Private key

Encryption is performed using the public key, while decryption is performed using the private key. A public key cannot be used for decryption. Although the two keys are connected, it is impossible to extract the private key from the public key. While the private key is kept private and only known by the key's owner, the public key is widely known. It implies that anyone can send the user a message using the user's public key. However, the communication can only be decrypted by the user using his private key.

RSA algorithm uses the following procedure to generate public and private keys:

1. Key Generation
2. Encryption
3. Decryption

Steps for encryption & decryption are in RSA are:

1. Select p and q as two separate prime numbers.
2. These prime numbers p and q should be randomly selected for security reasons, and they must have similar bit lengths.
3. Determine n = pq. The modulus for both the public and the private keys is 'n'. Its length, or key length, is represented in bits.
4. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1) = n - (p + q - 1)$, where φ is Euler's totient function.
5. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are co-prime. 'e' is the public key exponent.
6. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$, i.e. d is the multiplicative inverse of e (modulo $\varphi(n)$). Solve d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$
7. Encryption: $c \equiv m^e \pmod{n}$
8. Decryption: $m \equiv c^d \pmod{n}$

Some characteristics of the RSA algorithm:

- It is a well-liked exponentiation over integers, including prime numbers, in a finite field.
- Because this method uses sufficiently large integers, it is challenging to solve.
- This algorithm uses two sets of keys: a private key and a public key.

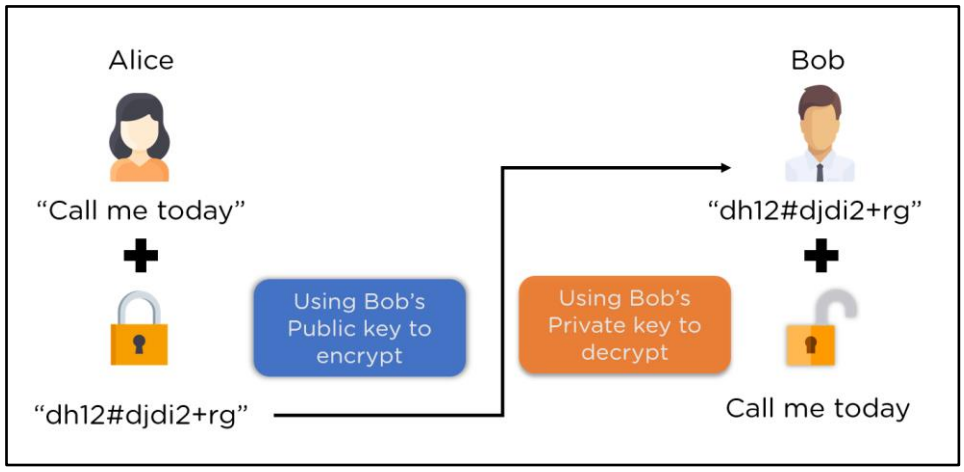


Figure 8: RSA Working (Google Images)

3.3 VGG Model

The VGG is an abbreviated form of the **Visual Geometry Group** from Oxford. Two models with 16 and 19 layers depth are there.

VGG is a convolution neural network (CNN) model supporting 16 and 19 layers. K. Simonyan and A. Zisserman from Oxford University proposed this model and published it in a paper called Very Deep Convolutional Networks for Large-Scale Image Recognition.

VGG16 is a CNN model which is used for image recognition. It is unique that it has only 16 layers which have weights, compared to relying on a large number of hyper-parameters. It is one of the best vision model architectures.

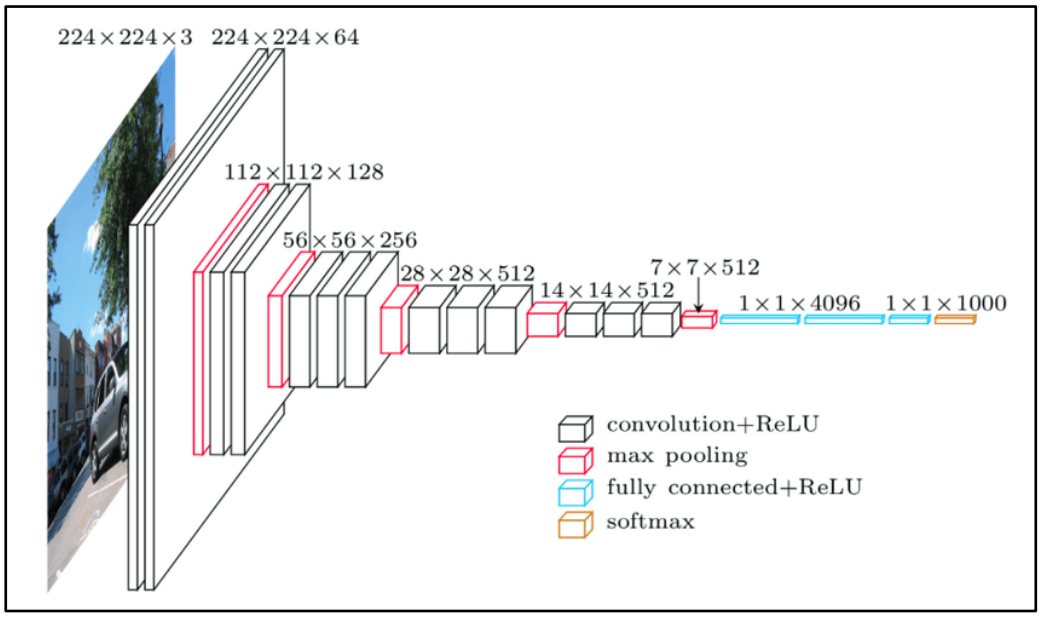


Figure 9: VGG model (VGG Neural Network Architecture)

3.4 Transfer Learning

Transfer learning is the process of employing a model that has already been trained to solve a new problem. It is presently particularly well-liked in deep learning because of its capacity to train deep neural networks with relatively minimal data. This is highly helpful because the majority of real-world problems frequently lack the millions of annotated data points needed to train such sophisticated models.

Using a machine learning model that has already been trained to address a different but related problem is referred to as "transfer learning." So the core idea behind transfer learning is to use what has been learned in one activity to enhance generalisation in another. The weights of a network are moved which were taken from "task-X" to a fresh "task-Y."

Transfer learning is typically used in computer vision and natural language processing tasks like face recognition, sentiment analysis, etc. due to the massive amount of CPU power required.

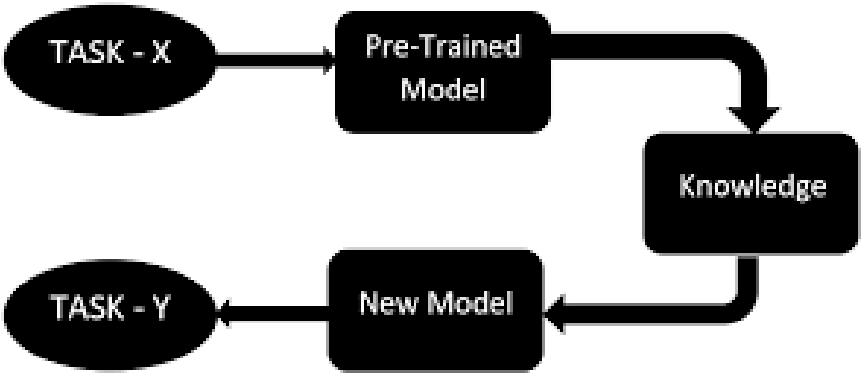


Figure 10: Concept of Transfer Learning (Google Images)

Chapter 04: Performance Analysis

4.1 System Properties

The system is modularized which means that all functions and files are separated as per their use case and functionality. This makes this system easily understandable, readable, follows certain standard and makes it scalable.

This system makes use of a simple, user-friendly, User Interface made by Python. The user interface can be greatly enhanced and made exciting and thrilling.

4.2 Performance

Based on experiments, these models are over performing models: FaceNet, VGG-Face, ArcFace and Dlib.

Following are the accuracies of these models on other datasets [Dataset used are: LFW and YTF] which are declared by its creators. This is the concept of transfer learning, in which an already existing model is used for a new application.

Table 1: Accuracy of few pre-trained models

(Source: Official deepface repository 2020)

Model	LFW Score	YTF Score
Facenet512	99.65%	-
ArcFace	99.41%	-
Dlib	99.38%	-
Facenet	99.20%	-
VGG-Face	98.78%	97.40%
Human-beings	97.53%	-
OpenFace	93.80%	-
DeepID	-	97.05%

Table 2: Various metrics of few pre-trained models

(Source: Serengil 2020)

Model Name	Cosine	Euclidean	Euclidean L2
VGGFace	Threshold: 0.31 Accuracy: 89.28 Precision: 97.41 Recall: 80.71 F1: 88.28	Threshold: 0.47 Accuracy: 81.42 Precision: 97.82 Recall: 64.28 F1: 77.58	Threshold: 0.79 Accuracy: 89.28 Precision: 97.41 Recall: 80.71 F1: 88.28
FaceNet	Threshold: 0.40 Accuracy: 98.21 Precision: 100 Recall: 96.42 F1: 98.18	Threshold: 11.26 Accuracy: 98.57 Precision: 100 Recall: 97.14 F1: 98.55	Threshold: 0.90 Accuracy: 98.21 Precision: 100 Recall: 96.42 F1: 98.18
OpenFace	Threshold: 0.11 Accuracy: 57.85 Precision: 95.83 Recall: 16.42 F1: 28.04	Threshold: 0.47 Accuracy: 57.85 Precision: 95.83 Recall: 16.42 F1: 28.04	Threshold: 0.47 Accuracy: 57.85 Precision: 95.83 Recall: 16.42 F1: 28.04
DeepFace	Threshold: 0.13 Accuracy: 54.64 Precision: 100 Recall: 9.28 F1: 16.99	Threshold: 42.21 Accuracy: 52.50 Precision: 100 Recall: 5.00 F1: 9.52	Threshold: 0.51 Accuracy: 54.64 Precision: 100 Recall: 9.28 F1: 16.99

Chapter 05: Results & Conclusions

5.1 Discussion on the Results Achieved

The ability to recognize a person from this system is tremendous as this system is more accurate than humans. This system is capable of recognizing people with more than 98% accuracy. This system is able to recognize faces in different orientations, with pixelated pictures and videos and is capable of providing quality recognition in normal surroundings.

This system might not work well in dim lighted areas, as Light intensity is an important factor for facial recognition. Also, low quality videos might not work with this recognition system.

The face recognition model, which makes use of a typical convolutional neural network, represents the face as a vector. Face pairings belonging to the same person should resemble one another more than face pairs belonging to different people. Cosine similarity is one of many metrics that may be used to determine similarity, Euclidean distance, and L2 shape.

5.2 Applications of the Project

- Criminal Identification.
- Face Recognition and all possible applications of Face Recognition.
- Analysis of Crowd Dynamics.
- Searching any missing person.

5.3 Limitations of the Project

- Setup of video surveillance system is costly as high quality cameras are required and high performance computers are required.
- Low quality videos might not work with this recognition system.
- Light intensity is an important factor, as pictures in dim light won't be recognized by this recognition system.

5.4 Future Work

- Integrating different methods of Data Acquisition like for ex. Location Based Social Networks (LBSN).
- Recognition of anomalous objects and activities.
- To recognize people who try to conceal their identity by any kind of makeover, etc.

References

[0] AIP Conference Proceedings 1891, 020002 (2017); https://doi.org/10.1063/1.5005335 Published Online: 03 October 2017

[1] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

[2] Abin, A. A., Fotouhi, M., & Kasaei, S. (2009, October). Realtime multiple face detection and tracking. In 2009 14th International CSI Computer Conference (pp. 379-384). IEEE.

[3] Tathe, S. V., Narote, A. S., & Narote, S. P. (2016, December). Face detection and recognition in videos. In 2016 IEEE Annual India Conference (INDICON) (pp. 1-6). IEEE.

[4] G. Thomson, “Facial Recognition,” Encyclopedia, 2005. [Online]. Available: https://www.encyclopedia.com/science/encyclopedias-almanacs-transcripts-and-maps/facialrecognition. [Accessed: 11-Oct-2018].

[5] M. Kafai, L. An, and B. Bhanu, “Reference face graph for face recognition,” IEEE Trans. Inf. Forensics Secur., vol. 9, no. 12, pp. 2132–2143, 2014.

[6] Lin, K., Chen, S. C., Chen, C. S., Lin, D. T., & Hung, Y. P. (2015). Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance. IEEE Transactions on Information Forensics and Security, 10(7), 1359-1370

[7] Chatrath, J., Gupta, P., Ahuja, P., Goel, A., & Arora, S. M. (2014, February). Real time human face detection and tracking. In 2014 international conference on signal processing and integrated networks (SPIN) (pp. 705-710). IEEE.

[8] Crime Data Analysis and Prediction of Perpetrator Identity using Machine Learning Approach.

[9] Agarwal, Jyoti, Renuka Nagpal, and Rajni Sehgal. "Crime analysis using K-means clustering." International Journal of Computer Applications 83.4 (2013).

[10] . Sathyadevan, Shiju, and Surya Gangadharan. "Crime analysis and prediction using data mining." Networks & Soft Computing (ICNSC), 2014 First International Conference on. IEEE, 2014

[11] McClendon, Lawrence, and Natarajan Meghanathan. "Using machine learning algorithms to analyze crime data." Machine Learning and Applications: An International Journal (MLAIJ) 2.1 (2015).

[12] Kiani, Rasoul, Siamak Mahdavi, and Amin Keshavarzi. "Analysis and prediction of crimes by clustering and classification." Analysis 4.8 (2015).

[13] Kansara, Chirag, et al. "Crime mitigation at Twitter using Big Data analytics and risk modelling." Recent Advances and Innovations in Engineering (ICRAIE), 2016 International Conference on. IEEE, 2016. [9]. Tsunoda, Masateru, Sousuk

[14] M. H. Yang, D. J. Kriegman & N. Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transaction on Pattern Analysis & Machine Intelligence, 24:1, (2002), pp. 34-58.

[15] Xu, Xiang & Liu, Wanquan & Li, Ling. (2014). Low Resolution Face Recognition in Surveillance Systems. Journal of Computer and Communications. 02. 70-77. 10.4236/jcc.2014.22013.

[16] A. Kumar and P. Mishra, "Efficient RSA algorithm for securing IoT devices," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 444-449, doi: 10.1109/CONFLUENCE53242.2021.9452789.

[17] S. Khan and R. Bhatnagar, "A Survey on Recent Developments in RSA Algorithm and Its Variants," 2021 International Conference on Information Technology (ICIT), Bhubaneswar, India, 2021, pp. 182-187, doi: 10.1109/ICIT51780.2021.9440653.

[18] S. Sengupta, R. Nandy and P. Sarkar, "An Improved RSA Cryptosystem using Efficient Modular Multiplication Algorithm," 2021 6th International Conference on Computing, Communication and Security (ICCCS), Rourkela, India, 2021, pp. 1-5, doi: 10.1109/ICCCS51706.2021.9444632.

Appendices

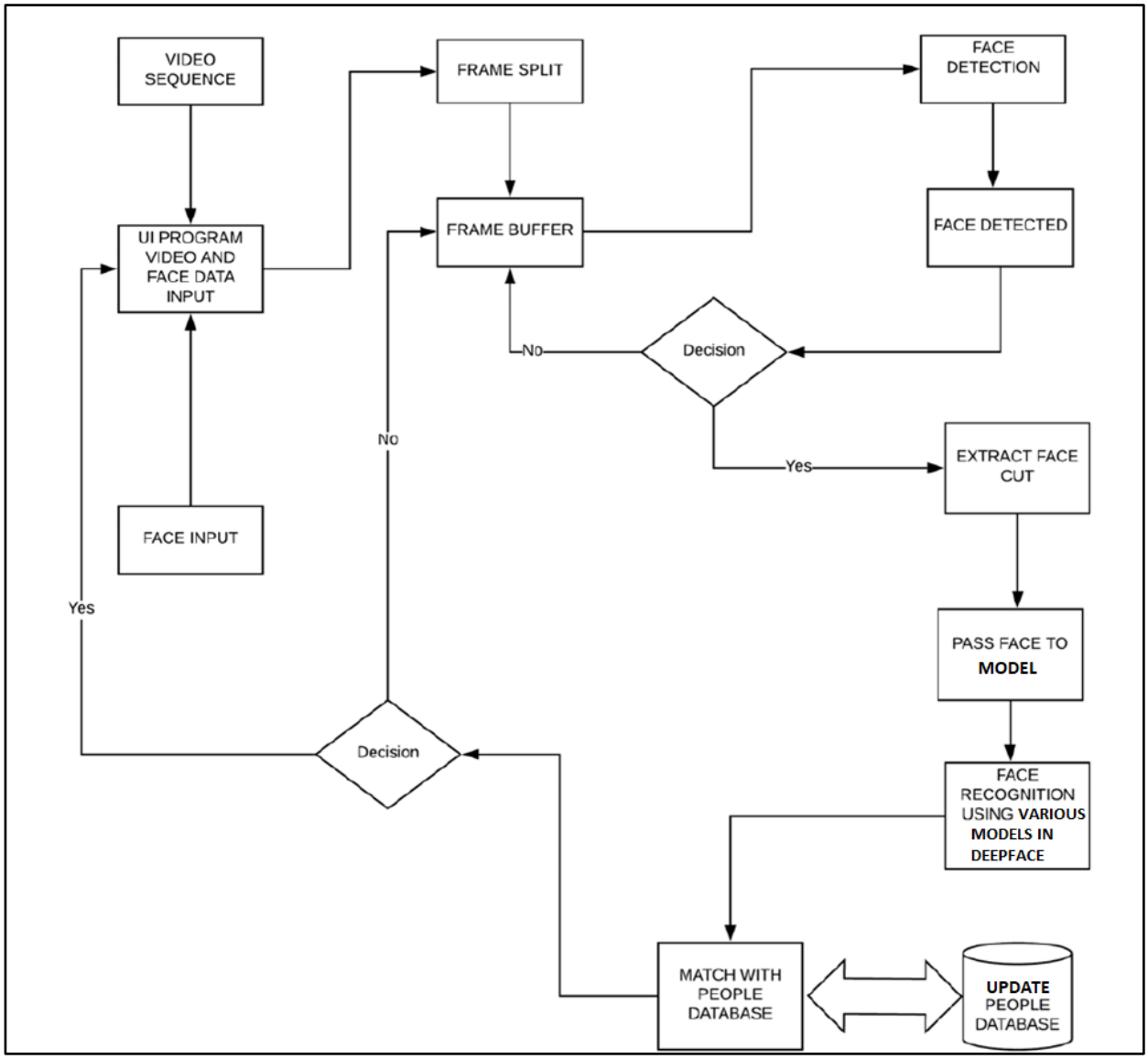


Figure 11: Flow-chart of the recognition system (edited by me, from Google)

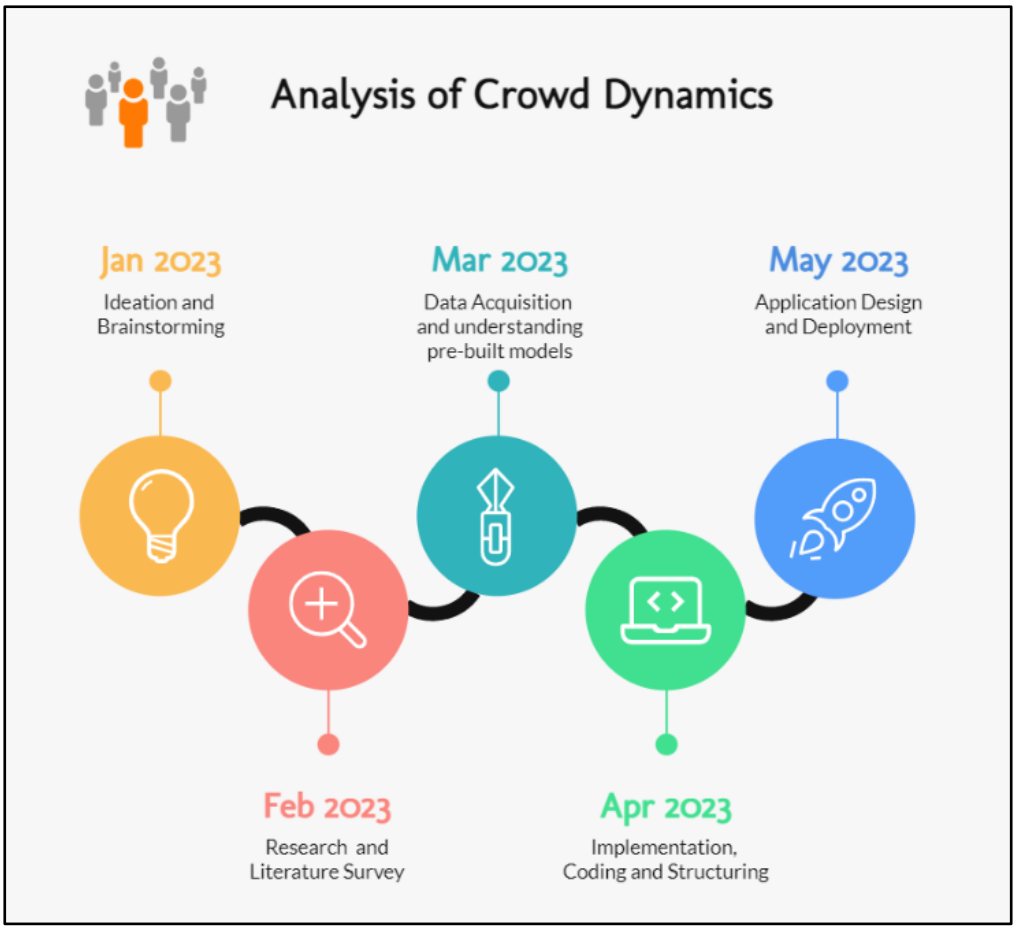


Figure 12: Timeline for the project (Made by me on visme)

Most important part of scripts:

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt
%matplotlib inline

my_img = cv2.imread('RSA.jpg')
# cv2_imshow(my_img)
plt.imshow(my_img, cmap="gray")


#RSA
# STEP 1: Generate Two Large Prime Numbers (p,q) randomly
from random import randrange, getrandbits

def power(a,d,n):
  ans=1;
  while d!=0:
    if d%2==1:
      ans=((ans%n)*(a%n))%n
    a=((a%n)*(a%n))%n
    d>>=1
  return ans;


def MillerRabin(N,d):
  a = randrange(2, N - 1)
  x=power(a,d,N);
  if x==1 or x==N-1:
    return True;
  else:
    while(d!=N-1):
      x=((x%N)*(x%N))%N;
      if x==1:
        return False;
      if x==N-1:
        return True;
      d<<=1;
  return False;

def is_prime(N,K):
  if N==3 or N==2:
    return True;
  if N<=1 or N%2==0:
    return False;

  #Find d such that d*(2^r)=X-1
  d=N-1
  while d%2!=0:
    d/=2;

  for _ in range(K):
    if not MillerRabin(N,d):
      return False;
  return True;


def generate_prime_candidate(length):
  # generate random bits
  p = getrandbits(length)
  # apply a mask to set MSB and LSB to 1
  # Set MSB to 1 to make sure we have a Number of 1024 bits.
  # Set LSB to 1 to make sure we get a Odd Number.
  p |= (1 << length - 1) | 1
  return p
```

Figure 13: Code for Cryptography

```
def generatePrimeNumber(length):
  A=4
  while not is_prime(A, 128):
        A = generate_prime_candidate(length)
  return A



length=5
P=generatePrimeNumber(length)
Q=generatePrimeNumber(length)

print(P)
print(Q)



#Step 2: Calculate N=P*Q and Euler Totient Function = (P-1)*(Q-1)
N=P*Q
eulerTotient=(P-1)*(Q-1)
print(N)
print(eulerTotient)



#Step 3: Find E such that GCD(E,eulerTotient)=1(i.e., e should be

def GCD(a,b):
  if a==0:
    return b;
  return GCD(b%a,a)

E=generatePrimeNumber(4)
while GCD(E,eulerTotient)!=1:
  E=generatePrimeNumber(4)
print(E)
```

Figure 14: Code for Cryptography

```
#Step 5: Encryption

for i in range(100,700):
  for j in range(100,1000):
    r,g,b=my_img[i,j]
    C1=power(r,E,N)
    C2=power(g,E,N)
    C3=power(b,E,N)
    enc[i][j]=[C1,C2,C3]
    C1=C1%256
    C2=C2%256
    C3=C3%256
    my_img[i,j]=[C1,C2,C3]


# plt.imshow(my_img, cmap="gray")
cv2_imshow(my_img)


#Step 6: Decryption
for i in range(100,700):
  for j in range(100,1000):
    r,g,b=enc[i][j]
    M1=power(r,D,N)
    M2=power(g,D,N)
    M3=power(b,D,N)
    my_img[i,j]=[M1,M2,M3]

cv2_imshow(my_img)
```

Figure 15: Code for Encryption & Decryption

```
for frame_num in range(0,len(frames)):
    detected_face = DeepFace.detectFace(img_path = frames[frame_num], target_size = (160,
    160), enforce_detection = False)
    result = DeepFace.verify(img1_path = input_image_path, img2_path = detected_face,
    model_name = 'VGG-Face', enforce_detection = False)

    if result['verified'] == True:
        flag = 1
        print(result, '\nInput image matched with the Frame number :', frame_num, 'of the
        video :', video)
        end_time = time.time()
        print(f'Time taken to detect and match the face : {end_time - start_time}')

        cv2.namedWindow("Uploaded")
        cv2.moveWindow('Uploaded', 400, 200)
        input_image_original=padding_size.padding(input_image_original)
        cv2.imshow('Uploaded', input_image_original)
        face=padding_size.padding(frames[frame_num])
        cv2.namedWindow("Matched in Video")
        cv2.moveWindow('Matched in Video', 750, 200)
        cv2.imshow('Matched in Video', face)
        cv2.waitKey(0)
        cv2. destroyAllWindows()
        break
```

Figure 16: When query picture is searched in a video

```
for frame_num in range(0,len(frames)):
    if flag == 1:
        break
    detected_face = DeepFace.detectFace(img_path = frames[frame_num], target_size = (160,160),
    enforce_detection = False)

    for input_video_frame in input_frames:
        result = DeepFace.verify(img1_path = input_video_frame, img2_path = detected_face, model_name =
        'Facenet512', enforce_detection = False)

        if result['verified'] == True:
            flag = 1
            print(result, '\nInput image matched with the Frame number :', frame_num, 'of the video :',
            video)
            end_time = time.time()
            print(f'Time taken to detect and match the face : {end_time - start_time}')

            img = frames[frame_num]
            cv2.namedWindow("Uploaded")
            cv2.moveWindow('Uploaded', 400, 200)
            input_image_original=padding_size.padding(input_video_frame)
            cv2.imshow('Uploaded', input_image_original)
            face=padding_size.padding(frames[frame_num])
            cv2.namedWindow("Matched in Video")
            cv2.moveWindow('Matched in Video', 750, 200)
            cv2.imshow('Matched in Video', face)
            cv2.waitKey(0)
            cv2. destroyAllWindows()
            break
```

Figure 17: When query video is searched in a video

```
img_path = _browser_.func()
input_image_original = cv2.imread(img_path)
df = DeepFace.find(img_path = img_path, db_path = d_path,enforce_detection=False)
if df.empty:
    print("No match!!!")
else:
    img = cv2.imread(df["identity"][0])
    cv2.namedWindow("Uploaded")
    cv2.moveWindow('Uploaded',400,200)
    input_image_original=padding_size.padding(input_image_original)
    cv2.imshow('Uploaded', input_image_original)
    face=padding_size.padding(img)
    cv2.namedWindow("In Database")
    cv2.moveWindow('In Database', 750, 200)
    cv2.imshow('In Database', face)
    cv2.waitKey(0)
    cv2. destroyAllWindows()
```

Figure 18: When query picture is searched in multiple pictures

```
import cv2
from deepface import DeepFace
#path where data is stored.
#use 'q' to exit the new window
path = "C:/Users/Sanyam Saxena/Desktop/Project/data/ds"
DeepFace.stream(db_path = path)
```

Figure 19: Real Time Face Recognition

Screen shots of the various stages of the Project

```
from deepface import DeepFace
import numpy as np
import cv2
from pip import main
import _browser_
import padding_size

path="C:/Users/Sanyam Saxena/Desktop/Project/data/Video/3.mp4"
vidcap = cv2.VideoCapture(path)
success,image = vidcap.read()
count = 0
success = True
frames = []
while success:
    #cv2.imshow('sasa',image)
    frames.append(image)
    # cv2.imwrite("frame%d.jpg" % count, image)     # save frame as JPEG file if you want
    success,image = vidcap.read()
    # print('Read a new frame: ', success)
    count += 1
    #cv2.waitKey(10)
print(count, " frames extracted")
frames = np.array(frames)
print("data shape =\t", frames.shape)
#cv2. destroyWindow('sasa')
frames.nbytes
```

Figure 20: Program_1

```
28  pth = _browser_.func()
29  input_image_original=cv2.imread(pth)
30  input_image = DeepFace.detectFace(img_path = pth,target_size=(160,160),
    detector_backend='retinaface')
31  for i in range(0,len(frames)):
32      face = DeepFace.detectFace(img_path = frames[i],target_size=(160,160),
        detector_backend='retinaface')
33      res = DeepFace.verify(img1_path = pth, img2_path = face,enforce_detection=False)
34      if res['verified']==True:
35          print(res)
36          cv2.namedWindow("Uploaded")
37          cv2.moveWindow('Uploaded',400,200)
38          input_image_original=padding_size.padding(input_image_original)
39          cv2.imshow('Uploaded',input_image_original)
40          face=padding_size.padding(frames[i])
41          cv2.namedWindow("Matched in Video")
42          cv2.moveWindow('Matched in Video',750,200)
43          cv2.imshow('Matched in Video',face)
44          cv2.waitKey(0)
45          cv2. destroyAllWindows()
46          break
```

Figure 21: Program_2

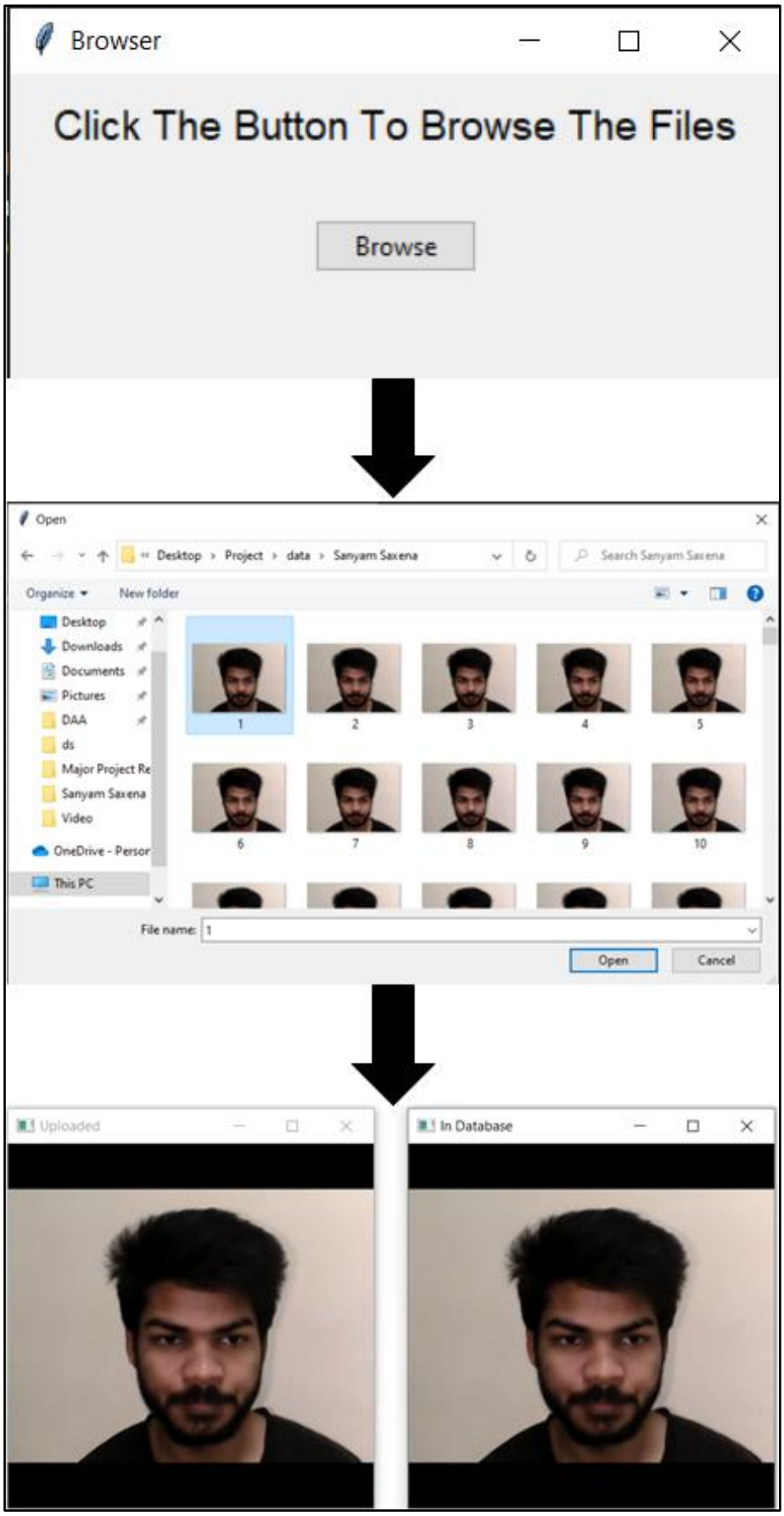


Figure 22: Identification

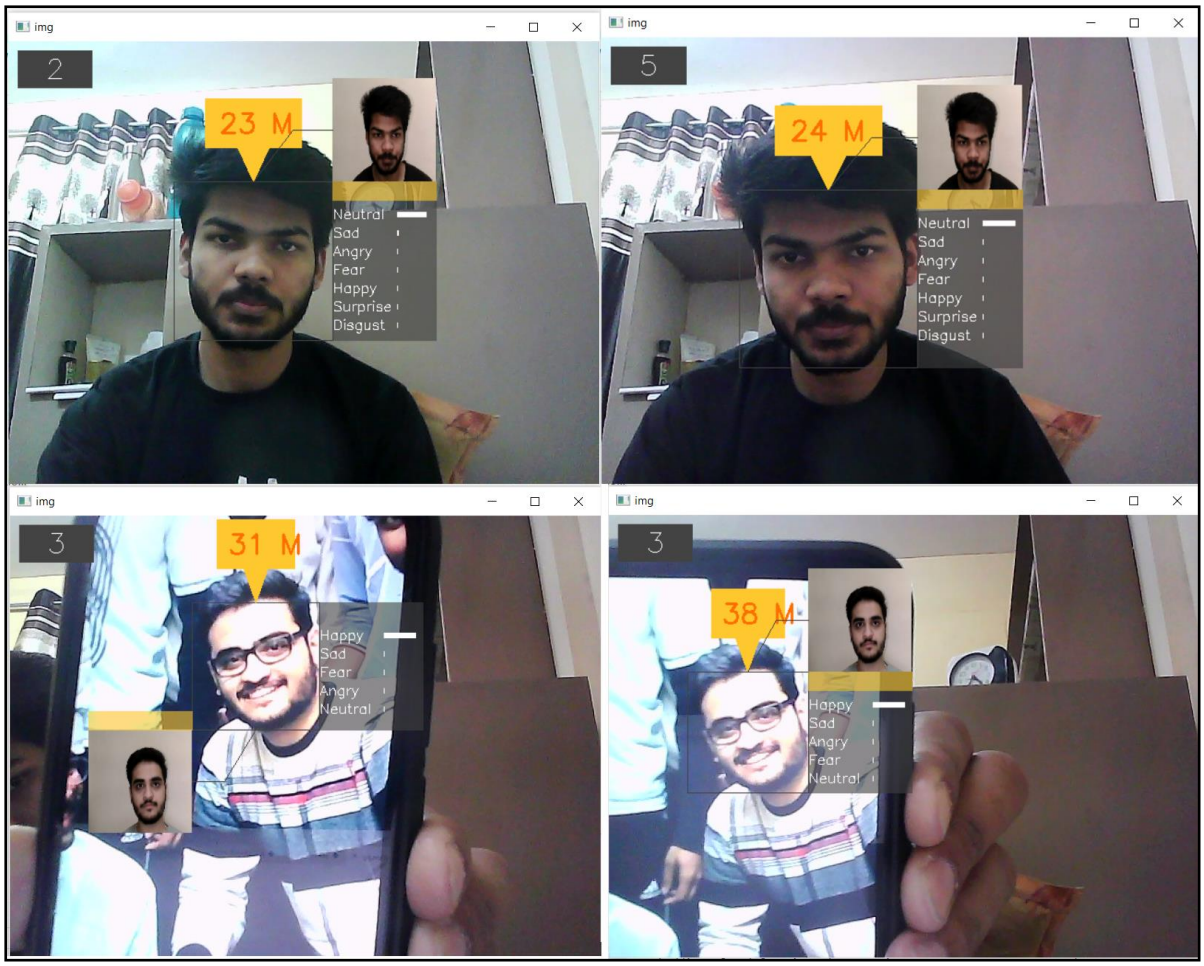


Figure 23: Live Recognition

****