A PROJECT SYNOPSIS

ON

"Real Time Face Detection and Recognition"

Submitted in the partial fulfillment of the requirements for

The degree of

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

1) NAVNATH AUTI
2) SAHIL NAZARE
3) BINITDEV PANDEY
4) ADNAN SHAIKH

UNDER THE GUIDANCE OF

Prof. Hemlata Gosavi

Department of Computer Engineering
Saraswati College of Engineering, Kharghar, Navi Mumbai
University of Mumbai
2022-23

Saraswati College of Engineering, Kharghar

Vision:

      To be universally accepted as autonomous center of learning in Engineering Education and Research.

Mission:

- ➢ To educate students to become responsible and quality technocrats to fulfil society and industry needs.
- ➢ To nurture student's creativity and skills for taking up challenges in all facets of life.

_____

Department of Computer Engineering

Vision:

      To be among renowned institution in Computer Engineering Education and Research by developing globally competent graduates.

Mission:

- ➢ To produce eminent Engineering graduates by imparting quality teaching learning and providing hands on experience in programming to fulfill society and industry needs.
- ➢ To encourage students to pursue research and learn new technologies in Computer Engineering and across interdisciplinary areas so that they participate in research, innovation and entrepreneurship.
- ➢ To provide stimulating learning ambience to enhance innovative ideas, problem solving ability, leadership qualities, team-spirit and ethical responsibilities.

(Approved by AICTE, recg. By Maharashtra Govt. DTE, Affiliated to Mumbai University)

PLOT NO. 46/46A, SECTOR NO 5, BEHIND MSEB SUBSTATION, KHARGHAR, NAVI MUMBAI-410210

Tel.: 022-27743706 to 11 * Fax: 022-27743712 * Website: www.sce.edu.in

CERTIFICATE

This is to certify that the requirements for the synopsis report entitled" **Real Time Face Detection and Recognition***" have been successfully completed by the following students:*

Roll numbers	Name
05	NAVNATH AUTI
46	SAHIL NAZARE
48	BINITDEV PANDEY
68	ADNAN SHAIKH

In partial fulfillment of Sem –VII, **Bachelor of Engineering of Mumbai University in Computer Engineering** of Saraswati college of Engineering, Kharghar during the academic year 2022-23.

Internal Guide   **External Examiner**

Prof. Hemlata Gosavi

Project Co-Ordinator   **Head of Department**

Dr. Anjali Dadhich   Prof. Sujata Bhairnallykar

Principal

Dr. Manjusha Deshmukh

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVE's

1. To embed a strong foundation of Computer Engineering fundamentals to identify, solve, analyze and design real time engineering problems as a professional or entrepreneur for the benefit of society.
2. To motivate and prepare students for lifelong learning & research to manifest global competitiveness.
3. To equip students with communication, teamwork and leadership skills to accept challenges in all the facts of life ethically.

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAM OUTCOMES

1. Apply the knowledge of Mathematics, Science and Engineering Fundamentals to solve complex Computer Engineering Problems.
2. Identify, formulate and analyze Computer Engineering Problems and derive conclusion using First Principle of Mathematics, Engineering Science and Computer Science.
3. Investigate Complex Computer engineering problems to find appropriate solution leading to valid conclusion.
4. Design a software System, components, Process to meet specified needs with appropriate attention to health and Safety Standards, Environmental and Societal Considerations.
5. Create, select and apply appropriate techniques, resources and advance Engineering software to analyze tools and design for Computer Engineering Problems.
6. Understand the Impact of Computer Engineering solution on society and environment for Sustainable development.
7. Understand Societal, health, Safety, cultural, Legal issues and Responsibilities relevant to Engineering Profession.
8. Apply Professional ethics, accountability and equity in Engineering Profession.
9. Work Effectively as a member and leader in multidisciplinary team for a common goal.
10. Communicate Effectively within a Profession and Society at large.
11. Appropriately incorporate principles of Management and Finance in one's own Work.
12. Identify educational needs and engage in lifelong learning in a Changing World of Technology.

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAMME SPECIFIC OUTCOME

- Formulate and analyze complex engineering problems in computer engineering (Networking/Big data/ Intelligent Systems/Cloud Computing/Real time systems).
- Plan and develop efficient, reliable and secure system and customized application software using cost effective emerging software tools ethically.

DECLARATION

      I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included. I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1) NAVNATH AUTI
2) SAHIL NAZARE
3) BINITDEV PANDEY
4) ADNAN SHAIKH

Date:

ACKNOWLEDGEMENT

After the completion of this work, words are not enough to express feelings about all those who helped us to reach goal.

It's a great pleasure and moment of immense satisfaction for us to express my profound gratitude to **Project Guide**, **Prof. Hemlata Gosavi**, whose constant encouragement enabled us to work enthusiastically. His perpetual motivation, patience and excellent expertise in discussion during progress of the project work have benefited us to an extent, which is beyond expression.

We would also like to give our sincere thanks to **Prof. Sujata Bhairnallykar, Head of Department**, and **Dr. Anjali Dadhich**, Project **Co-Ordinator** from Department of Computer Engineering, Saraswati college of Engineering, Kharghar, Navi Mumbai, for their guidance, encouragement, and support during a project.

I am thankful to **Dr. Manjusha Deshmukh, Principal,** Saraswati College of Engineering, Kharghar, Navi Mumbai for providing an outstanding academic environment, also for providing the adequate facilities.

Last but not the least we would also like to thank all the staffs of Saraswati college of Engineering (Computer Engineering Department) for their valuable guidance with their interest and valuable suggestions brightened us.

1) NAVNATH AUTI
2) SAHIL NAZARE
3) BINITDEV PANDEY
4) ADNAN SHAIKH

ABSTRACT

In today's fast-growing world everyone wants to save time as much as they can, one cannot relay dependent on older method if we want to recognize someone without asking them or verifying now days it can be only possible through a system where user authentication can be done through videos or images with the help of AI and ML.

Face detection is a popular application of object detection in computer vision. To detect faces in a digital image or video input requires the computer to utilize a series of algorithms or techniques. The technology of detecting faces has evolved proportional to its usage in various applications such as biometric security, autofocus in cameras, robotics, and social media applications. The aim of this project is to provide a description on the evolution of face detection and recognition.

It will be difficult for any organization to evaluate everyone one by one therefore it is necessary to have something at hand which can check or verify person if needed. Face detection is an important research direction in the field of target detection. For the input image, the position of the face is returned. In order to complete the task of face detection using deep learning, data input, feature extraction and face feature detection are three steps, among which feature extraction is the most important part. By studying the basic principles of current mainstream target detection algorithms, this paper compares the characteristics of Two-stage and one stage detection models and their application in face detection tasks.

INDEX

List of Figures

CHAPTER 1

INTRODUCTON

1.1 GENERAL

The goal of face detection is to determine if there are any faces in the image or video. If multiple faces are present, each face is enclosed by a bounding box and thus we know the location of the faces

Human faces are difficult to model as there are many variables that can change for example facial expression, orientation, lighting conditions, and partial occlusions such as sunglasses, scarfs, masks, etc. The result of the detection gives the face location parameters, and it could be required in various forms, for instance, a rectangle covering the central part of the face, eye centres or landmarks including eyes, nose and mouth corners, eyebrows, nostrils, etc.

Computer vision is an extended disciplinary field of computer graphics. The term computer vision describes the process, or the action done by the computer that mimics the human visual system. Another interpretation of computer vision is creating a self-sufficient system that can carry out some of tasks of the human visual cortex. Most scientists use computer vision to extract information from digital images or videos the same way a human being's visual receptors do.

Face detection is one of the more popular applications of object detection in computer vision. The computer uses a series of mathematical algorithms, pattern recognition and image processing to identify faces from an image or video input. Over the years, the technology of detecting faces has evolved proportional to its usage in various applications. One of the earliest documented usage of face detection was traced back to the work of Woodrow Wilson Bledsoe in the 1960s, where his research using manual measurements to recognize faces paved the way to the various systems developed.

The general concept of face detection is most commonly achieved with the following three steps; the first step is to examine the picture or video frame to determine the regions of interest. This is usually done via a sliding window. The second step is to acquire the extracted features or patterns from the region of interest. This is where the main aspects of the face detection algorithm lie such as using Haar-like features, Histogram of Oriented Gradients or deep learning methods such as convolutional neural networks. Finally, the third step is to classify if the detected regions of interest into faces and non-faces for recognition.

1.2 OBJECTIVE AND PROBLEM STATEMENT

This project aims to build a system that can recognize faces in a real time to help the organization in determining the identification of particular person by integrating face detection and recognition software with IP Camera.

Functionality:

- ➢ Extraction of frames from Ip camera video.
- ➢ Extraction faces from frames.
- ➢ Recognition of extracted faces from frames.
- ➢ If face matched with any faces in face repository, then storing it in Index DB with corresponding frame time and location of camera.

CHAPTER 2

LITERATURE REVIEW

Edwin Jose, Greeshma M, Mithun Haridas T. P, "Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2" 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS).

- Authors proposed system for face detection and recognition of human faces using embedded graphic card Jetson TX2 in IP camera, Models used by the authors are MTCNN and FaceNet.

Rehmat Ullah , Hassan Hayat, Afsah Abid Siddiqui, Uzma Abid Siddiqui, Jebran Khan, Farman Ullah, Shoaib Hassan, Laiq Hasan, Waleed Albattah , Muhammad Islam, and Ghulam Mohammad Karami, "A Real-Time Framework for Human Face Detection and Recognition in CCTV Images" 2022 Hindawi Mathematical Problems in Engineering.

- Authors proposed a CCTV based face recognition system which recognize single face from Image using PCA, CNN and KNN and it required average of 30 images for each face before classifying accurately.

Paul Viola, Michael J. Jones, "Robust Real-Time Face Detection" July 11, 2003 International Journal of Computer Vision.

- Authors proposed Machine Learning algorithm for face detection known as Cascade Classifiers, it trains the AdaBoost classifiers in Cascade manner and underlying Stumps classify Haar like features that is created using another intelligent algorithm, used to create Integral Image for quick

computation. Integral Image together with Cascade Classifier make computation efficient and robust. This algorithm is best suited for real-time face detection

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Senior Member, IEEE, and Yu Qiao, Senior Member, IEEE, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks" October 10, 2016 IEEE SIGNAL PROCESSING LETTERS, VOL. 23

- Authors proposed Deep learning algorithm for face detection on different alignment of face. Multitask Cascaded Convolutional Networks is enhanced version of Cascade Classifier uses three different layers of CNN for detection of faces. First layer uses P-Net or Proposal Network for obtaining candidate and their bounding box regression, second layer uses R-net or Refine Network which rejects large number of false candidates and perform calibration of bounding box regression and Third layer uses O-net or Overall Network for describing face in more detail and five facial landmarks position. Each layer uses 3 loss function, first loss function uses cross entropy-loss for face classification, second loss function uses Euclidean loss for bounding box regression and Third loss function also uses Euclidean loss for Facial landmark localization.

F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815-823. doi: 10.1109/CVPR.2015.7298682

- The authors deeply illustrated a method which directly learns an embedding into an Euclidean space for face verification. The FaceNet model maps a tight crop alignment which is around face area to a 128-byte Euclidean embedding (unique features of face). The authors took into consideration two deep network architectures to make this model. The authors showed precise

results of the two categories of this model on Labeled Faces in the Wild (LFW) dataset and the YouTube Face DB respectively. The paper also includes the future work of reducing the size of model, CPU requirements and the long training time of the model.

CHAPTER 3

PROBLEM STATEMENT AND SCOPE

3.1 PROBLEM STATEMENT

- This project aims to build a system that can recognize faces in a real time to help the organization in determining the identification of particular person by integrating face detection and recognition software with IP Camera.

3.2 SCOPE

- Preventing the frauds at ATMs in India. A database of all customers with ATM cards in India can be created and facial recognition systems can be installed. So, whenever user will enter in ATM his photograph will be taken to permit the access after it is being matched with stored photo from the database.

- Reporting duplicate voters.

- Passport and visa verification can also be done using this technology.

- Also, driving license verification can be done using the same approach.

- In defence ministry, airports, and all other important places the technology can be used to ensure better surveillance and security.

- It can also be used during examinations such as Civil Services Exam, SSC, IIT, MBBS, and others to identify the candidates.

- This system can be deployed for verification and attendance tracking at various government offices and corporates.

- For access control verification and identification of authentic users it can also be installed in bank lockers and vaults.

- For identification of criminals the system can be used by police force also.

CHAPTER 4

PROPOSED SYSTEM

4.1 ALGORITHM

Data Pre-processing:

Data pre-processing is a step of cleaning, transforming and aggregating data before it can work with algorithms. In this project we need to pre-process live video frames before they can be fed to a model using OpenCV we can detect and extract face(s) from frames. OpenCV provides three different algorithms for detecting faces in a stream of images viz., Cascade Classifier and MTCNN (Multitask Convolution Neural Network) using Haar Basis functions with AdaBoost as its core component first created by Viola-Jones and HOG (Histogram of Oriented Gradients) descriptor and object detector by Navneet Dalal and Bill Triggs. Cascade Classifier can process 25 images/second with precision of 95.24% and recall of 82.60%, MTCNN can process 3 images/second with precision of 98.02% and recall of 89.85% this reading is estimated using CPU it can be increased using better CPU or GPU. MTCNN can deal with scale and orientation of the face where Cascade Classifier cannot. For our purpose we will be using MTCNN for training face recognition algorithms and Cascade Classifier for real time face detection (MTCNN can also be used for real time face detection if GPU is in use).

Cascade Classifier:

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted

Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar like features (based on Haar Wavelet used for compression of wave further notion will like from Haar like features) shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.
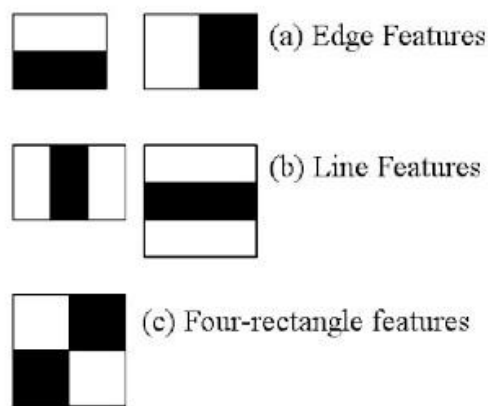

Figure 4.1.1. HAAR-like Features.

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image (cumulative sum of pixels located at left top of corresponding pixel). However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by Adaboost.
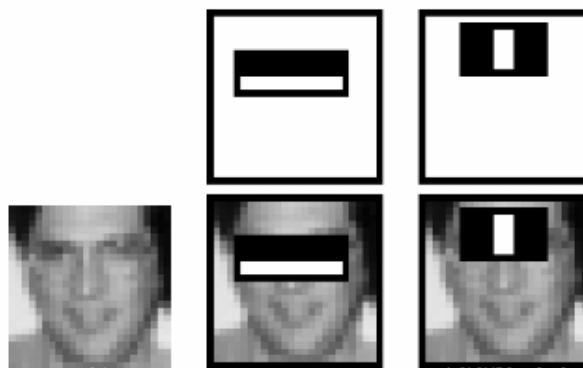

Figure 4.1.2. Example of HAAR-like features.

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found. There is a common trade-off between detection rate and precision for this classifier).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. Wow! Isn't it a little inefficient and time consuming? Yes, it is. The authors have a good solution for that.

In an image, most of the image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot (further cascade classifiers don't need to process that part again rather concentrating on more complex features), and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.

For this they introduced the concept of Cascade of Classifiers. Instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. How is that plan!

The authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages. (The two features in the above image are actually obtained as the best two features from Adaboost). According to the authors, on average 10 features out of 6000+ are evaluated per sub-window.

MTCNN:

MTCNN is a multitask neural network model for face detection. In order to take into account the performance and accuracy, and avoid the huge performance consumption caused by traditional ideas such as sliding window and classifier, it first uses small model to generate target region candidate box with certain possibility, and then uses more complex model for fine classification and higher precision region box regression, and makes this step recursive to form a three-layer network, namely p-net , R-Net, o-net, to achieve fast and efficient face detection. In the input layer, image pyramid is used to transform the scale of the initial image, and p-net is used to generate a large number of candidate target area frames. After that, R-Net is used for the first selection and border regression of these target area frames, and most of the negative examples are excluded. Then, the more complex and higher precision network o-net is used to discriminate and regress the remaining target area frames.

For face classification, MTCNN sets the learning objective as a binary classification problem, and uses cross entropy loss function for each sample x_i

$$L_i^{det} = -(y_i^{det}\log(p_i) + (1 - y_i^{det})(1 - \log(p_i)))$$

P_i is the sample x_i predicted by neural network Probability of belonging to face the label of $\{1, 0\}$. 0 is the real image indicates that there is no face in the image, and 1 indicates that there is a face in the image.

For the candidate frame regression task, the bounding box regression algorithm is used to make the target frame predicted by the network model close to or coincide with the real target frame. MTCNN algorithm uses a 4dimensions vector (x, y, w, h) to output the predicted window, which represents the upper left coordinate of the prediction window and the width and height of the window respectively. For each

human face candidate frame, the difference square loss function is used to predict the deviation between the candidate frame of network output and the nearest ground truth:

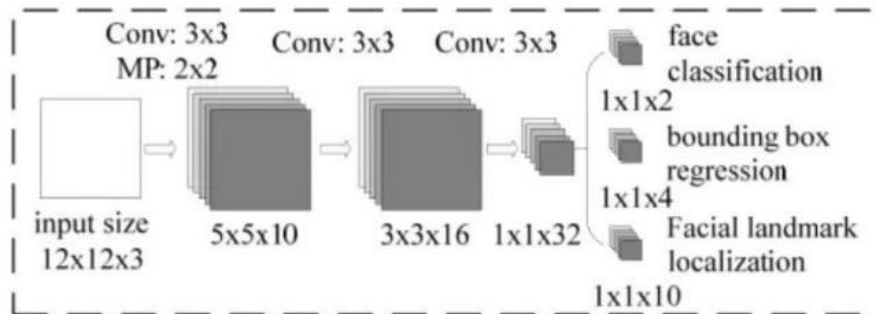$$L_i^{box} = \left\| y_i - y_i^{box} \right\|^2$$
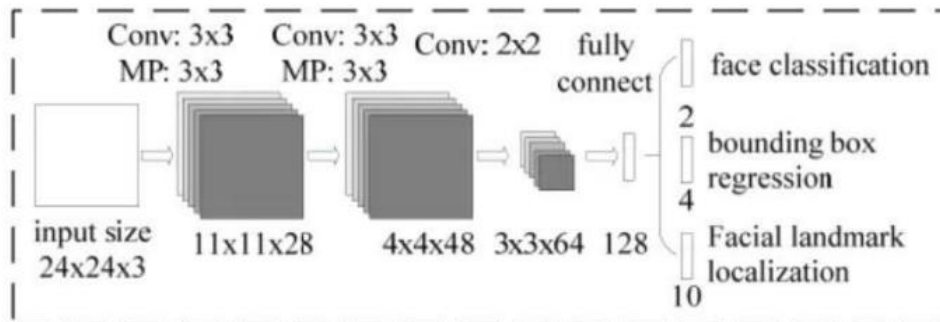

Figure 4.1.3. P-net Structure.

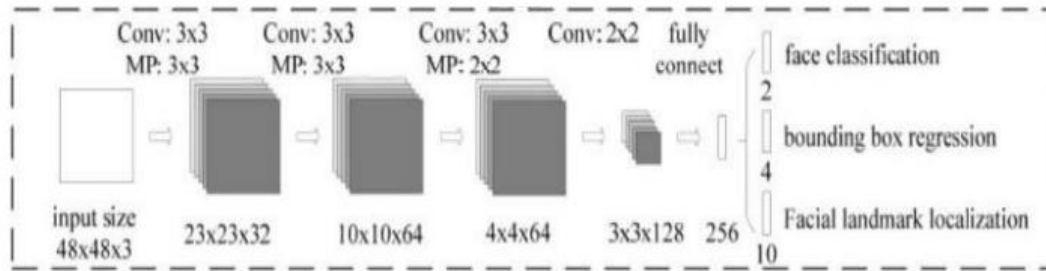
Figure 4.1.4. R-net Structure.

Figure 4.1.5. O-net Structure.

FaceNet Model:

FaceNet is a unified system for face verification, face recognition and clustering common faces. This system was developed by Google researchers that achieved FaceNet is a unified system for face verification, face recognition and clustering state-of-the-art results on variety of face recognition datasets. This system is based on extracting Euclidean embeddings from the detected faces which is used to verify a face. This model is made on taking into consideration two different deep convolutional network architecture. First architecture is based on Zeiler and Fergus Model which uses multiple interleaved layers of convolutions, non-linear activations, local response normalizations, max pooling layers and an additional several 1x1xd convolutional layers. Second architecture is based on Inception Model of Szegedy et al. This model can take input from a range of dimensions, i.e., 96x96 pixels to 224x224 pixels, also supports RGB or Grayscale images. The output is a 128-byte 1 dimensional embedding vector. The unique features of the face are all boiled-down into a 1x128 dimensional vector. This network consists of a batch input followed by a deep architecture and a L2 normalization, which generates the embedding, which is followed by a triplet loss function during training. Triplet Loss function basically reduces the distance between an anchor and a positive identity (similar face) and increases the distance between an

anchor and a negative identity (not so similar face). The first architecture has in total of 140 million parameters and 1.6 billion FLOPS. The second architecture has in total of 7.5 million parameter and 1.6 billion FLOPS. This architecture was built to run on mobile CPUs. FaceNet model accuracy is of 99.63% on Labelled Faces in the Wild (LFW) dataset and 95.12% on YouTube Faces DB. FaceNet is the best face recognition model till date as compared to Facebook's DeepFace Model.
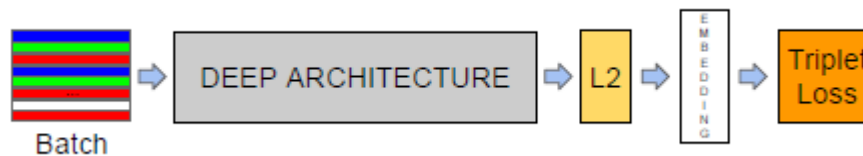

Figure 4.1.6. FaceNet Model Structure.

Introduction of Kivy

We are creating desktop application to demonstrate our project using Kivy. Kivy is open-source library or framework of python which freely available that allows us to develop multiplatform applications that run on Desktops like Windows, macOS, Linux and could also run-on iOS and android devices. Kivy is flexible that give freedom to us to write code once and runs as it is on different platform. The application development using this kivy library helps to develop interactive applications with a simple GUI that is easy to use. The main advantage of using the kivy platform is that the application is developed in any platform, and after that it can be used in any type of device to help reuse the code and be deployed easily. The UI given by the kivy framework helps to develop interactive application that are user friendly. Kivy supports various type of input devices that include keyboard, mouse etc. Application that build by using

kivy are very fast. It is completely based on python programming language that is very powerful and help to create the application in minimum time.

Desktop application

In our project we will be using Kivy to create desktop application which will be use by end-user. This desktop application work in offline mode (without internet) and because of that it is fast and more responsive compare to web application. This application is integrated with OpenCV model which detect the face. This application has an option to take an input as image as well as video. This input is stored in a local repository which is used to verify a face which is detected by the OpenCV model. If face matched with any faces in face repository, then storing it in Index CSV file with corresponding frame time and location of camera. The administrative user will be able see the required contents of CSV file.

4.2 DESIGN DETAILS

Realtime video will recorded through IP camera. Extraction of frames from IP camera video. Here it will process video using OpenCV as OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features. Captured video frames will go to haar cascade classifier which is an Object Detection Algorithm used to identify faces in an image or a real time video. Extraction faces from frames. Then extracted faces will go to FaceNet model for further process. Recognition of extracted faces from frames. If face matched with any faces in face repository which is loaded using MTCNN, then storing it in Index DB with corresponding frame time and location of camera.
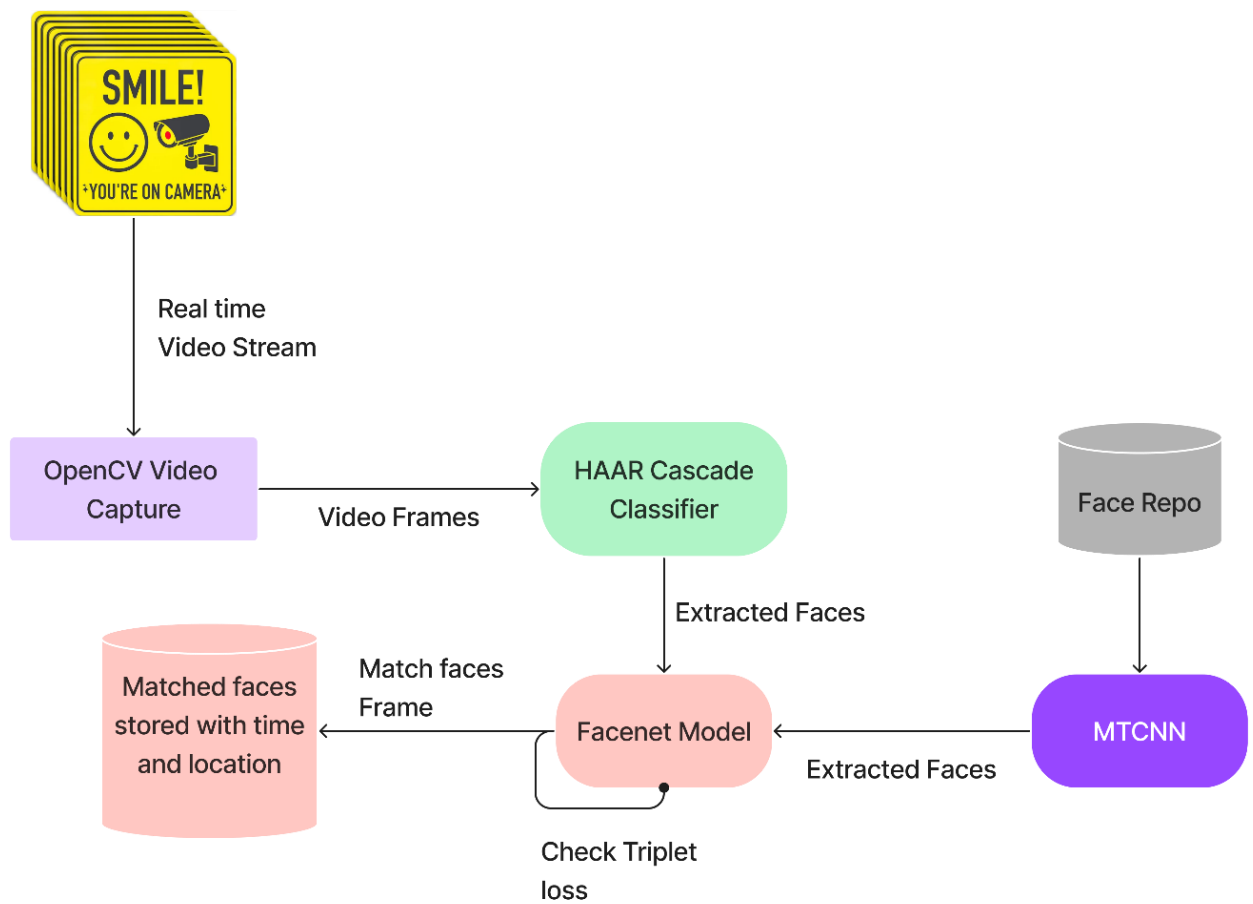
4.3 FLOWCHART

Figure 4.3.1. Flow diagram

CHAPTER 5

REQUIREMENTS

5.1 HARDWARE AND SOFTWARE SPECIFICATIONS

5.1.1 HARDWARE:

1. RAM: Minimum 4GB

2. CPU: Intel Core i3 or higher

3. Hard Drive: 40 GB Free Space

4. IP Camera

5.1.2 SOFTWARE:

1. TensorFlow

2. Numpy

3. OpenCV

4. Python IDE

5. Kivy

6. Jupyter Notebook

CHAPTER 6

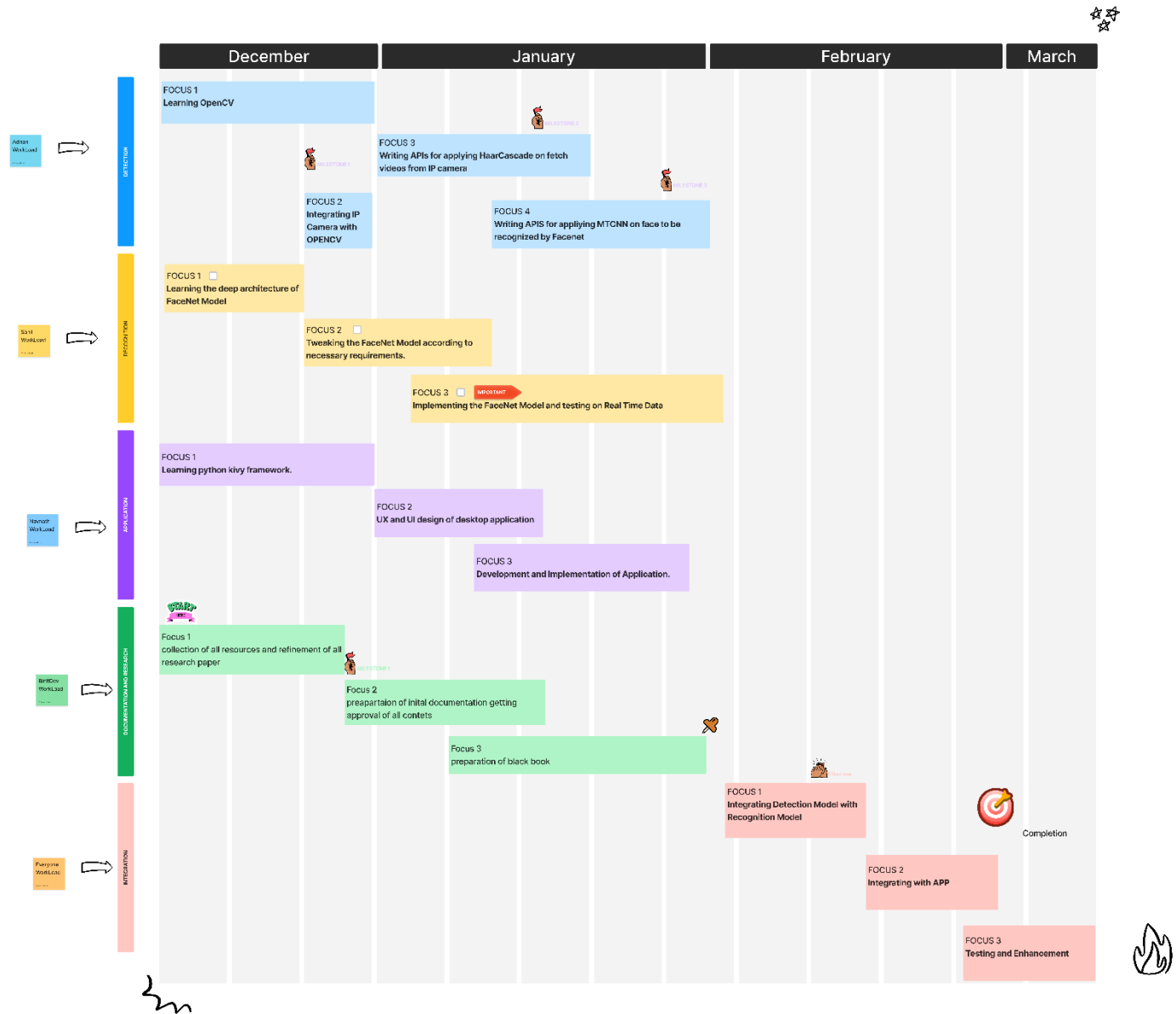# IMPLEMENTATION PLAN FOR NEXT SEMESTER

Fig 6.1: Gantt Chart

CHAPTER 7

CONCLUSION

This project will demonstrate a system for face recognition from surveillance video. The system will demonstrate to be robust to changes in illumination, scale, facial expression and reasonably robust to occlusions and changes in pose. Attention was given to performance considerations, and the system can operate in real time. All these features make this approach suitable for a video surveillance application. This work is a fully-fledged surveillance and suspect tracking system, using facial recognition and marking the suspects positions on the maps for more real time analysis. This system is a complete offline method and hence providing an inherent and enhanced security for our data.

REFERENCES

[1] FaceNet and MTCNN on Jetson TX2" 2019 5th International Conference on Advanced Computing & Edwin Jose, Greeshma M, Mithun Haridas T. P, "Face Recognition based Surveillance System Using Communication Systems (ICACCS).

[2] Rehmat Ullah , Hassan Hayat, Afsah Abid Siddiqui, Uzma Abid Siddiqui, Jebran Khan, Farman Ullah, Shoaib Hassan, Laiq Hasan, Waleed Albattah , Muhammad Islam, and Ghulam Mohammad Karami, "A Real-Time Framework for Human Face Detection and Recognition in CCTV Images" 2022 Hindawi Mathematical Problems in Engineering.

[3] Paul Viola, Michael J. Jones, "Robust Real-Time Face Detection" July 11, 2003 International Journal of Computer Vision.

[4] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Senior Member, IEEE, and Yu Qiao, Senior Member, IEEE, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks" October 10, 2016 IEEE SIGNAL PROCESSING LETTERS, VOL. 23

[5] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815-823. doi: 10.1109/CVPR.2015.7298682

[6] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. CoRR, abs/1412.1265, 2014. 1, 2, 5, 8

[7] 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, 8

[8] https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

[9] https://numpy.org/numpy-tutorials/

[10] https://github.com/davidsandberg/facenet/tree/master/src/align