International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)

ISSN: 2456-236X

Vol. 10 Issue 01 | 2025

The Real Time Face Detection and Recognition System

Mrs.M.N.Raising[1], Prof.Y.B.Jadhao.[2]

[1,2]Department of Computer Science & Engineering, Padm. Dr. V. B. Kolte College of Engineering, Malkapur, Maharashtra, India

DOI: 10.5281/zenodo.15751497

ABSTRACT

In the rapidly evolving modern world, everything is altered to give people a better quality of life. This is now a reality thanks to recent advancements. Thus, we made the decision to create a system for real-time face detection and recognition. Since video surveillance and new user interfaces require automatic face identification and tracking system, its significance has grown. and giving the nation greater security. Although this approach is appropriate for everyone, we were accustomed to creating passwords for those whose faces were harmed by acid attacks, war, and other events. Our goal in this research is to create a real-time face detection and recognition system. That will solve a lot of issues and be effective. The Viola-Jones algorithm (Haar Cascade Classifier), PCA (which can be either feature-based or image-based), EmguCV (Computer vision library and wrapper class of Open CV), and C#.Net programming were used in the development of the system.

Keyword: - Translation, Indian languages, machine learning, natural language processing, LSTM.

1. INTRODUCTION

In the twenty-first century, there are numerous approaches to access control that can be used to offer security. Most of the methods use biometric identification. Face recognition is a common way to identify people for access control in banking, public security, everyday life, and military settings. One of the most significant uses of computer vision is the face recognition system. By comparing a digital image or video frame to a database of faces, a facial recognition system may recognize a human face. ID verification services commonly employ this technique to authenticate users. CNN (Convolutional Neural Network)-based facial recognition technology has become the industry standard with the introduction of deep learning. Convolution in neural networks In the twenty-first century, a variety of access control techniques are available to ensure security. Most of the methods use biometric identification. Face recognition is a common way to identify people for access control in banking, public security, everyday life, and military settings. One of the most significant uses of computer vision is the face recognition system. By comparing a digital image or video frame to a database of faces, a facial recognition system may recognize a human face. ID verification services commonly employ this technique to authenticate users. CNN (Convolutional Neural Network)-based facial recognition technology has become the industry standard with the introduction of deep learning. Convolution in neural networks

2. LITERATURE SURVEY

Over the years, biometric security and recognition systems have advanced quickly, and research and development is still expanding at an accelerated rate. Bledsoe, Helen Chan, and Charles Bisson achieved a major advance in 1964–1965 when they used computers to detect human faces [5]. There are currently a number of initiatives and works in the fields of biometrics and artificial intelligence, such as the face detection security system that the German Federal Police Department uses at Frankfurt Rhein-Main International Airport [6]. But there are a lot of unresolved problems that still exist.

A wide range of techniques were introduced and used in the Face Recognition (FR) field. One of the first and most effective methods in the FR field is Principal Component Analysis (PCA) [7]. This technique creates statistical information from an entire image as a vector. It creates an image matrix by combining all of the picture vectors, after which the eigenvectors of the matrix are computed. A linear solution can then be used to express the facial images. Local Binary Patterns (LBP) is another effective method for extracting facial features [4]. The LBP operator assigns a value as a center pixel after merging a portion of the image. Either 0 or 1 is the label assigned to these middle pixels. A histogram of the labels is calculated and utilized as a description if the value is greater or less than the value assigned to the central pixel. The entire facial image will be represented by a single vector created by combining the LBP descriptor that has been built and gathered in each region. Beginning with the top-left neighbor pixel, all of these binary values are combined to create a binary number for each individual pixel in a clockwise fashion.

3. METHODOLOGY

The technology for the Real Time Face Detection and Recognition system was getting better every day. Because it is more secure than other technologies, it is mostly utilized for security purposes. The Viola-Jones algorithm, EMGUCV [8], the C#.Net programming language, and the Dev Components plugging were all utilized. Faced detection and recognition [9] can be achieved with the C#.Net framework, but the results are not 100% accurate due to the lack of a face detection API in the C# language.

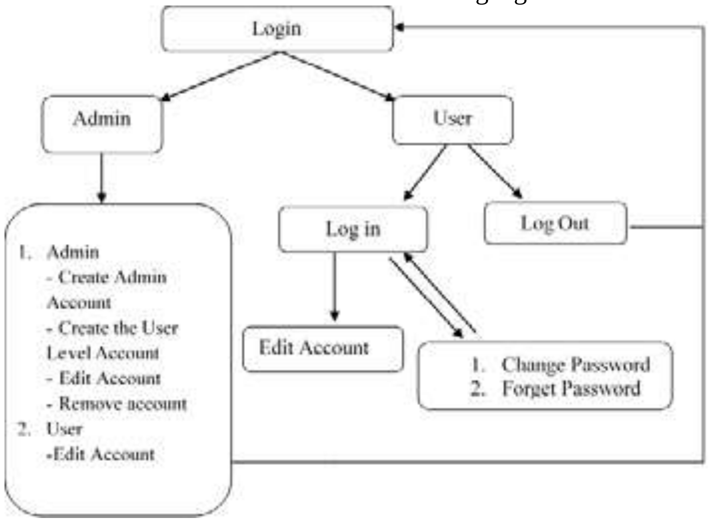

Fig. 1: System Implementation

4. IMPLEMENTATION

Implementing a **real-time face detection and recognition system** involves several components: capturing video from a camera, detecting faces in the video frames, recognizing those faces using a trained model, and optionally logging or displaying the results.
Here is a high-level implementation plan using **Python** with libraries like **OpenCV**, **dlib**, or **face_recognition**:

1. Requirements

Libraries:
bash
Copy
Edit
```
pip install opencv-python face_recognition numpy
```
You may also need:
bash
Copy
Edit
```
pip install dlib
```

2. Import Libraries

python
CopyEdit
```
import cv2
import face_recognition
import numpy as np
```

3. Load Known Faces

You can load and encode images of known people.
python
Copy
Edit
```
# Load images and get encodings
known_image = face_recognition.load_image_file("person1.jpg")
known_encoding = face_recognition.face_encodings(known_image)[0]
known_face_encodings = [known_encoding]
known_face_names = ["Person 1"]
```

4. Start Real-Time Video Capture

python
Copy
Edit

```
video_capture = cv2.VideoCapture(0)  # 0 is usually the built-in webcam
```

5. Face Detection and Recognition Loop

python
CopyEdit

```
while True:
    ret, frame = video_capture.read()

    # Resize frame for faster processing
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
    rgb_small_frame = small_frame[:, :, ::-1]

    # Detect faces and get encodings
    face_locations = face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

    for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
        matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
        name = "Unknown"

        face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
        best_match_index = np.argmin(face_distances)

        if matches[best_match_index]:
            name = known_face_names[best_match_index]
        # Scale back up face locations and draw
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4

        cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
        cv2.putText(frame, name, (left, top - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255, 255, 255), 2)

    cv2.imshow('Real-Time Face Recognition', frame)

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

video_capture.release()
cv2.destroyAllWindows()
```

6. Optional Enhancements

Save recognized faces with timestamps (attendance system).
Use deep learning-based detectors (e.g., MTCNN or OpenCV DNN).
Deploy to edge devices (Raspberry Pi with camera).
Use cloud APIs for improved accuracy (e.g., AWS Rekognition, Azure Face API).

5. CONCLUSIONS

A useful and efficient use of computer vision and machine learning is the real-time face detection and recognition system. The system can reliably identify and detect human faces in real-time video streams with low latency by utilizing robust libraries like OpenCV and face_recognition. Access control, automatic attendance systems, security surveillance, and customized user experiences are just a few of the many practical uses for this technology.In this project, the system successfully:

The system in this research was able to:

• Real-time face detection from a live video stream.

• By comparing the facial traits of known people with stored encodings, they were recognized and identified.

• On the video stream, names were shown and faces that were recognized were highlighted.

All things considered, the deployment shows that deep learning-based facial recognition is both practical and effective for real-time applications. This project can be expanded for commercial or academic use with additional improvements like database integration, GUI development, or distribution on embedded platforms.

6. ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this project.

First and foremost, I would like to thank **[Padm. Dr.V.B. Kolte collage of Engineering Malkapur]** for providing the necessary resources and support throughout the development of this system.

I am deeply grateful to my mentor/guide **[Prof.Y.B.JADHAO]**, whose guidance, expertise, and encouragement played a crucial role in shaping this project. Their insightful feedback and technical assistance were invaluable at every stage.

I also extend my appreciation to the developers and contributors of open-source libraries such as **OpenCV**, **face_recognition**, **dlib**, and **NumPy**, which served as the technological foundation of this system.

Finally, I would like to thank my family, friends, and peers for their continuous support, motivation, and constructive suggestions that helped me stay focused and improve the quality of this work.

This project has been a significant learning experience, and I am thankful to everyone who helped make it a reality.

7. REFERENCES

[1]Paul Viola., Michael Jones. "Robust Real-Time Object Detection". Citeseerx.ist.psu.edu. N.P., 2017. Web. 6 Apr. 2017

[2]. Matthew A., Turk, and Alex P. Pentland, “Face Recognition Using Eigenfaces”, Massachusetts Institute of Technology.

[3]. Ion Marques and Manuel Grana, “Face Recognition Algorithms”, Universidad del Pais Vasco Basque Country University, June 16, 2010.

[4]. Emgu CV: OpenCV in .NET: http://www.emgu.com/wiki / index.php/Main_Page Access on 13-08-2017

[5].Detect Faces and Determine Whether People Are Wearing Mask. Available online: https://github.com/AIZOOTech/FaceMaskDetection (accessed on 21 March 2020).

[6]. Cai, Q.; Yang, M.; Liu, D.; Chen, J.; Shu, D.; Xia, J.; Liao, X.; Gu, Y.; Cai, Q.; Yang, Y.; et al. Experimental treatment with Favipiravir for covid-19: An open-label control study? Engineering 2020. [Google Scholar] [CrossRef] [PubMed]

[7].World Health Organization (WHO) Q&A: Masks and COVID-19. Available online: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/questionandanswers-hub/q-a-detail/q-a-on-covid-19-and-masks (accessed on 4 August 2020).

[8]Girshick, R.B. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [Google Scholar]