REAL-TIME FACE RECOGNITION

Technical Report

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF COMPUTER SCIENCE AND ENGINEERING

BACHELOR OF TECHNOLOGY

Submitted by:
Jaskirat Singh
2302566

Submitted to:
Er. Satinderpal Singh
Assistant Professor

Dept. of Computer Science Engineering
Guru Nanak Dev Engineering College,
Ludhiana, 141006

Abstract

This report details the development of a real-time face_recognition system utilizing Python, OpenCV, and the face_recognition library. The primary goal of this project was to create a robust and efficient system capable of detecting and recognizing multiple faces in a webcam video feed simultaneously. The system processes video frames to identify known individuals by comparing live face encodings with pre-stored facial data. One of the key features of this system is its ability to handle multiple faces within a single frame and to automatically save frames that contain unknown faces for further analysis.

The system operates in real-time, offering high accuracy and reliability and when multiple individuals are present. It is designed with a user-friendly interface that displays recognized faces with labels and provides visual feedback on unidentified individuals. The project integrates advanced face detection algorithms and encoding techniques to ensure seamless operation and effective identification.

Through iterative development and testing, the system has proven to be both effective and practical, addressing the growing need for automated and accurate face recognition in security and identification applications. Future enhancements may include improved recognition algorithms, better handling of diverse environmental conditions, and integration with existing systems for comprehensive solutions in security and attendance management.

Acknowledgement

We are highly grateful to the Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the major project work at the college.

The constant guidance and encouragement received from Dr. Kiran Jyoti H.O.D. CSE Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Er. Satinderpal Singh, without his wise counsel and able guidance, it would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of computer science and engineering department of GNDEC for their intellectual support throughout the course of this work. Finally, we are indebted to all whosoever have contributed in this report work.

Jaskirat Singh

Contents

Chapter 1

Introduction

1.1 Introduction

This project explores the development of a real-time face recognition system using machine learning libraries, namely OpenCV and `face_recognition`. The project aims to detect, recognize, and classify faces, saving frames that contain unknown faces for further analysis. With growing security demands, face recognition technology provides an efficient solution for real-time identification.

Real-time face recognition combines computer vision and artificial intelligence to enhance security systems and create a seamless way of identifying individuals. This document outlines the methodology, applications, challenges, and future prospects of implementing this technology.

1.2 Overview of Face Recognition Systems

Face recognition is a biometric technology that uses facial features to identify individuals. It works by capturing an image of the face, analyzing its unique structure, and comparing it with known faces in the database. It has gained widespread use in areas such as security, surveillance, and access control systems.

Key Features of Face Recognition Systems

- **Facial Landmark Detection:** Identifies critical points on the face such as eyes, nose, and mouth.
- **Feature Extraction:** Extracts unique characteristics like distance between eyes or shape of cheekbones.
- **Database Comparison:** Matches the extracted features against stored facial templates in a database.

1.3 The Evolution of Face Recognition Technology

Face recognition technology has evolved significantly over the past few decades, transitioning from basic image processing to cutting-edge AI-driven systems.

Historical Perspective

- **Early Techniques:** Focused on geometric approaches, analyzing simple facial dimensions.
- **Advances in the 1990s:** Statistical methods such as Eigenfaces were introduced, laying the groundwork for modern systems.[6]
- **Deep Learning Revolution:** Today's systems use convolutional neural networks (CNNs) to detect complex patterns and improve recognition accuracy.

Key Milestones

- Development of algorithms capable of handling poor-quality images.
- Integration of multi-modal biometric systems (e.g., combining face recognition with iris or fingerprint recognition).
- Real-time recognition advancements due to improved hardware like GPUs.

1.4 AI in Face Recognition

AI plays a crucial role in enhancing the accuracy of face recognition systems. Machine learning models are trained on large datasets of facial images, learning to distinguish between different individuals.

Role of AI in Enhancing Capabilities

- **Deep Learning:** Models like ResNet and VGGNet are used to extract high-level features from facial images.
- **Data Augmentation:** Helps systems generalize better by simulating various conditions, such as lighting and occlusions.
- **Continuous Learning:** Systems adapt to new data over time, improving their accuracy.

Integration of AI with Real-Time Processing

AI algorithms enable the system to process live video streams, making face recognition fast and efficient. Applications such as video surveillance leverage this ability to detect individuals instantly.

1.5 Applications of Real-Time Face Recognition

The applications of face recognition span across multiple industries. This section highlights key use cases.

- **Security and Surveillance:** Identifying criminals in public spaces and enhancing security in airports and sensitive areas.
- **Access Control:** Biometric systems in offices, homes, and vehicles, replacing traditional authentication methods like passwords and keycards.
- **Retail:** Personalized recommendations based on customer preferences and seamless checkouts through face-based payment systems.
- **Healthcare:** Improving patient identification to prevent mix-ups and monitoring patient well-being through emotion recognition.
- **Education:** Automated attendance systems in schools and colleges, and assessing engagement levels during online classes.

1.6 Challenges and Ethical Considerations

While face recognition offers numerous benefits, it also presents challenges that need to be addressed.

Technical Challenges

- **Accuracy in Diverse Conditions:** Variations in lighting, angles, and partial occlusions affect performance. Systems need robust models to handle real-world scenarios.
- **Bias in Data:** Training datasets often underrepresent minorities, leading to potential biases. Addressing this requires more diverse and balanced datasets.
- **Scalability:** Real-time processing for large-scale applications demands significant computational resources.

Ethical Considerations

- **Privacy Concerns:** Potential misuse of facial data for unauthorized surveillance. Striking a balance between security and individual rights is crucial.
- **Regulation and Accountability:** Developing guidelines for ethical implementation and ensuring transparency in system operations.

1.7 Future of AI in Face Recognition

The future of AI in face recognition is poised to bring transformative changes across industries.

Upcoming Advancements

- **Improved Algorithms:** Use of advanced neural architectures like transformers to boost recognition accuracy and better handling of low-resolution and noisy images.
- **Privacy-Preserving Techniques:** Federated learning to enable data sharing without compromising privacy, and encryption techniques to secure facial templates in databases.
- **Augmented Capabilities:** Emotion recognition to gauge sentiments in real time and integration with augmented reality (AR) and virtual reality (VR).

Vision for a Responsible Future

- Encouraging collaborations between governments, tech companies, and civil society to create ethical frameworks.
- Leveraging AI to ensure inclusivity and fairness in face recognition technology.

Chapter 2

Literature Review

2.1 Introduction to Face Recognition Technologies

Face recognition technology has been a subject of extensive research since the 1960s. Early systems relied on geometric features such as the distance between the eyes or the shape of the nose. These methods, while innovative for their time, were limited by their inability to handle variations in facial expressions, lighting conditions, and angles. In the 1990s, researchers began using eigenfaces and principal component analysis (PCA), which allowed for a more robust representation of faces by capturing the variance in facial features. Recent systems, however, leverage machine learning algorithms to provide higher accuracy and adaptability. These advances have made face recognition a reliable tool in a wide array of applications, including security, law enforcement, and even personal devices like smartphones.

2.2 The Role of AI in Face Recognition

With the emergence of AI, face recognition systems have improved significantly. AI-based models like Convolutional Neural Networks (CNN) allow systems to process large amounts of data, learn complex patterns, and recognize faces in real-time, even in challenging conditions such as low lighting, occlusions, or aging. CNNs, a class of deep learning algorithms, are particularly effective in processing visual data, as they automatically learn to identify important features in an image without explicit programming. AI also enables systems to continually improve as they encounter more data, thus increasing their accuracy and adaptability over time. The integration of AI has transformed face recognition from a simple matching task to a sophisticated, dynamic technology capable of recognizing faces under various real-world conditions.

2.3 AI-Based Face Recognition: Theoretical Foundations

AI face recognition systems work by analyzing facial features and converting them into a numerical representation called face encoding. This encoding is then compared to known encodings stored in a database. The process typically involves several key steps:[2]

- **Face Detection:** The system first locates the face within an image or video stream, using algorithms such as the Viola-Jones algorithm or more modern methods like Single Shot Multibox Detector (SSD).
- **Face Alignment:** Once a face is detected, the system aligns the face to normalize variations in pose or angle.
- **Feature Extraction:** The system then extracts relevant features from the aligned face using deep learning techniques. These features can include distances between key points on the face, texture, and contours, among others.
- **Face Encoding:** These features are encoded into a numerical vector that uniquely represents the face. This encoding is often high-dimensional, capturing complex patterns in the face.
- **Face Matching:** Finally, the encoded face is compared to a database of stored encodings. If a match is found, the identity of the person is confirmed.

Modern systems use deep learning algorithms, particularly deep neural networks (DNNs), which can capture minute details of facial features, such as subtle variations in the curvature of the nose or the slope of the jawline, thus improving recognition accuracy. This increased sensitivity allows AI-based systems to be more robust, with the ability to function under a wider range of environmental and demographic conditions.

2.4 Challenges in AI-Based Face Recognition

Despite the significant advancements in face recognition, challenges remain in the field. One of the most significant challenges is dealing with variations in lighting, expression, and aging. These factors can affect the appearance of an individual's face and reduce the accuracy of recognition systems. Additionally, issues related to bias and fairness have emerged, as AI models may perform differently across different demographic groups, such as gender or ethnicity. To address these concerns, researchers are exploring techniques to increase the diversity of training data and enhance the fairness of models.

Another challenge involves security and privacy concerns. While face recognition can be a powerful tool for identification, its widespread use raises ethical questions about surveillance, consent, and the potential for misuse. As a result, there has been a growing demand for regulatory frameworks to govern the use of face recognition technologies, ensuring they are deployed responsibly and transparently.

2.5 Applications of Face Recognition Technology

The applications of face recognition technology are vast and continue to grow. In security, face recognition is widely used for surveillance and access control, where it can be integrated into systems like airport security checks, building entry points, and even smartphones. Law enforcement agencies utilize face recognition for criminal identification and finding missing persons.

In the commercial sector, retailers use face recognition for personalized shopping experiences, such as offering discounts or recommendations based on a customer's previous purchases. Social media platforms also rely on face recognition algorithms for automatic tagging of photos. Additionally, the healthcare industry has explored the use of face recognition for patient identification and monitoring, ensuring that medical services are delivered to the correct individuals.

2.6 Future Directions in Face Recognition

The future of face recognition technology is promising, with several key areas of research aimed at improving its effectiveness and applicability. One promising avenue is the development of multimodal systems that combine face recognition with other biometrics, such as fingerprint or voice recognition. This could enhance security and reliability, making it more difficult for unauthorized individuals to spoof the system.

Another area of research involves improving the explainability of AI models used in face recognition. As these systems become more complex, it is important to understand how they make decisions and ensure that their actions are transparent and interpretable. This could help address ethical concerns and build trust in AI-based systems.

Lastly, with the growing awareness of privacy concerns, there is a push for privacy-preserving face recognition systems. Techniques such as federated learning, which allows models to be trained on decentralized data without it leaving users' devices, are being explored as a way to maintain individual privacy while still benefiting from the technology.

2.7 Conclusion

In conclusion, face recognition technology has come a long way, from its early reliance on geometric features to its current use of deep learning algorithms. The integration of AI has significantly enhanced the accuracy and efficiency of these systems, enabling real-time face recognition in a variety of conditions. However, challenges remain, including issues related to fairness, security, and privacy, which must be addressed as the technology continues to evolve. As research progresses, the potential applications of face recognition are vast, and its integration into various sectors of society is likely to continue expanding, with future developments focusing on improving accuracy, transparency, and privacy.

Chapter 3

Objectives

3.1 Accurate Real-Time Face Recognition

The primary goal is to develop a system that can accurately recognize multiple faces in realtime using webcam footage.

3.2 Capture and Save Frames Containing Unknown Faces

The system should automatically capture and save frames where unknown faces are detected, allowing for further analysis or storage for future reference.

3.3 Multi-Face Recognition Capability

The system should efficiently detect and recognize multiple faces in a single frame, regardless of the position or number of faces.

3.4 Efficient and Scalable

The system should be scalable, allowing it to handle large datasets and expand to include more known faces in the future.

3.5 Integration of Security Features

A secondary goal is to ensure that all frames containing unknown faces are securely stored and managed, protecting sensitive data.

Chapter 4

Methodology

4.1 Research Design

The project follows an experimental research design, employing machine learning algorithms to design, develop, and test the face recognition system. The system uses Python libraries for real-time processing.[3]

4.2 Data Collection

The system uses pre-stored images of known faces for training and testing. Data is collected through the webcam in real-time and processed by the face_recognition library to extract facial encodings.

Stored image samples:

4.3 Data Preprocessing

Data preprocessing involves converting captured frames into RGB format and extracting face encodings from known and unknown faces. The encodings are then used for comparison with known faces.

4.4 Model Training

The face recognition model is trained using the pre-stored face encodings. OpenCV and dlib are used to detect and extract features from the facial images. The model compares the live face encodings with the known encodings to identify individuals.[4]

4.5 Evaluation and Testing

The system is evaluated based on its accuracy, speed, and ability to handle multiple faces. Several test cases are designed to assess its performance under various lighting conditions, angles, and backgrounds.
The system uses face detection algorithms provided by OpenCV[5] and face encoding/comparison algorithms from face_recognition. The algorithm is designed to process each frame, detect multiple faces, recognize them, and save frames with unknown faces. **T**est Cases

- **Test Case 1:** Ensure the system recognizes known faces.
- **Case 2:** Validate that unknown faces are detected and saved correctly.
- **Case 3:** Test the system's performance with multiple faces in a single frame.
- **Test Case 4:** Test the system under different lighting conditions with multiple faces.

4.6 Deployment

The final system is deployed on a local machine (HP 15s with Windows 11 Pro) for testing and real-time performance evaluation. It can be integrated into other security systems in the future.

4.7 Tools and Technologies

- **Languages:** Python[1]
- **Libraries:** OpenCV, face_recognition, numpy (1.26.4)
- **IDE:** Visual Studio
- **Tools:** Webcam for video capture

Chapter 5

Results and Discussions

5.1 User Interface Representation

The interface allows users to start/stop video capture, view recognized faces, and review saved frames of unknown individuals. It also displays indicators for multiple face detection and frame saving.

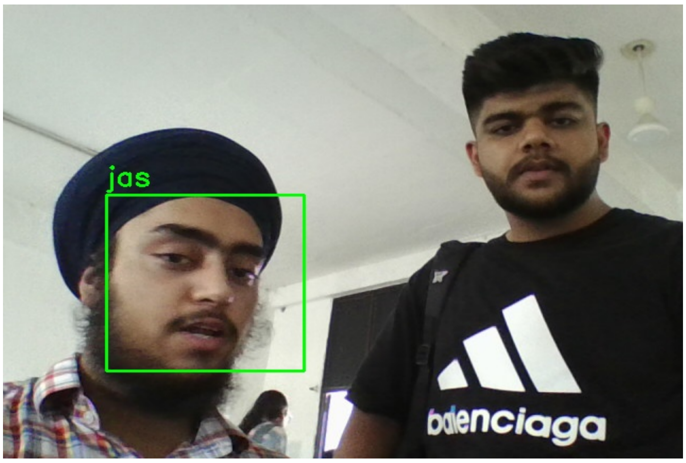
5.2 Snapshots and Discussion

Snapshots demonstrate the system identifying faces in real-time, with labels and bounding boxes around recognized faces. Multiple faces detected in a single frame are labeled and the process of saving unknown faces is illustrated.

5.3 Back-End Representation

The system uses a simple file-based approach for managing images and frames. Instead of a traditional database, frames containing unknown faces are saved in the same folder as the application. This approach involves:

- **Frame Storage:** Each frame with unknown faces is saved as a separate image file in the project directory.



- **Organization:** Images are named sequentially (e.g., frame1.jpg, frame2.jpg) to keep track of the frames. This allows for easy retrieval and review of frames containing unidentified individuals

Chapter 6

Conclusion

6.1 Conclusion

The project successfully developed a real-time face recognition system that meets the defined objectives. The system is efficient and accurate under various conditions, with the ability to handle multiple faces and save frames with unknown individuals for further analysis. This capability makes the system suitable for a range of practical applications, including security and attendance monitoring.

6.2 Future Scope

Future enhancements to the face recognition system could include the following:

- **Enhanced Accuracy:** Implementing advanced algorithms and machine learning techniques to improve the accuracy of face recognition, particularly in diverse and challenging conditions such as varying lighting and different facial expressions.
- **Integration with Attendance Systems:** The system could be integrated with attendance management systems to automate the process of marking attendance based on face recognition. This would streamline administrative tasks in educational institutions or workplaces, reducing manual effort and potential errors.
- **Multi-Modal Recognition:** Incorporating additional biometric methods, such as fingerprint recognition or voice authentication, to complement the face recognition system and enhance overall security and accuracy.
- **Real-Time Alerts:** Developing real-time alert mechanisms for scenarios where unauthorized or unknown individuals are detected, allowing for immediate action to be taken.
- **Cloud Integration:** Storing face encodings and attendance data on a cloud platform for remote access and analysis, facilitating scalability and easy data management.

- **User Experience Improvements:** Enhancing the user interface to provide more detailed information and analysis, such as historical data, trends, and actionable insights

Bibliography

[1] Python Software Foundation. Python programming language. The official documentation provides comprehensive details on Python's features, libraries, and usage. Accessed: 2024-11-19.

[2] Adam Geitgey. Face recognition patern. Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library.

[3] Adam Geitgey. face_recognition documentation. The `face_recognition` library offers a simple API for face recognition in Python, detailing its usage and functions. Accessed: 2024-11-19.

[4] Adam Geitgey. Tutorials on face_recognition api. Guides and tutorials on using the world's simplest Python-based facial recognition API. Accessed: 2024-11-19.

[5] OpenCV Developers. Opencv documentation. Accessed: 2024-11-19.

[6] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016.

Appendix

A : Source code

```
import cv2
import face\_recognition

\# Load and encode the known faces
names = ["jas", "jap", "harsh"]
data\_img = [face\_recognition.load\_image\_file(face + ".jpg") for face in names]
\# Encode the known faces
known\_face\_encodings = [face\_recognition.face\_encodings(face)[0] for face in
    data\_img]

\# Initialize the video capture object
vid = cv2.VideoCapture(0)
count = 0

while True:
    \# Capture video frame by frame
    ret, frame = vid.read()
    if not ret:
        break

    \# Convert the frame to RGB (face\_recognition expects RGB images)
    rgb\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

    \# Find all face locations and encodings in the current frame
    face\_locations = face\_recognition.face\_locations(rgb\_frame)
    face\_encodings = face\_recognition.face\_encodings(rgb\_frame, face\
        _locations)

    \# Process each face found in the frame
    for (top, right, bottom, left), face\_encoding in zip(face\_locations, face\
        _encodings):
        label = "Unknown" \# Default label if no match is found
        \# Compare each face encoding with known face encodings
        for i, known\_face\_encoding in enumerate(known\_face\_encodings):
            results = face\_recognition.compare\_faces([known\_face\_encoding],
                face\_encoding)
            if results[0]:
                label = names[i]
                break

        if label == "Unknown":
            cv2.imwrite(f"frame{count}.jpg", frame)
            count += 1
```

```

        \# Draw a rectangle around the face and label it
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
        cv2.putText(frame, label, (left, top - 10), cv2.FONT\_HERSHEY\_SIMPLEX,
            0.9, (0, 255, 0), 2)

    \# Display the resulting frame
    cv2.imshow('frame', frame)

    \# Exit on 'q' key press
    if cv2.waitKey(1) \& 0xFF == ord('q'):
        break

\# Release the video capture object and destroy all windows
vid.release()
cv2.destroyAllWindows()
```

B : Setup Commands

The following steps outline the installation and setup process required to run the face recognition system on Windows:

1. Install Python:
   o Ensure Python 3.x is installed. If not, download and install Python from Python.org.

2. Install Required Libraries:
   o Open the terminal (Command Prompt or PowerShell) and navigate to your project directory.
   o Install the necessary Python libraries using the following commands:

```
pip install opencv-python
pip install face_recognition
pip install numpy
```

3. Install dlib:
   o Download the appropriate dlib wheel file for your Python version (in this case, dlib-19.24.99-cp312-cp312-win_amd64.whl for Python 3.12 on Windows 64-bit) from a trusted source like PyPI or other repositories.
   o Once downloaded, install the dlib package by navigating to the folder where the .whl file is located and running:

```
pip install dlib-19.24.99-cp312-cp312-win_amd64.whl
```

4. Verify Installation:
   o Check if all libraries are installed successfully by running the following in the terminal:

```
pip list
```

   o This should display all installed packages, including opencv-python, face_recognition, numpy, and dlib. These commands ensure that all required dependencies for the face recognition system are installed and ready for development and testing.

These commands ensure that all required dependencies for the face recognition system are installed and ready for development and testing.

C: System Configuration

Hardware Configuration:

- **Laptop Model**: HP 15s
- **Processor**: i3-1215U
- **RAM**: 8 GB
- **Storage**: 512 GB SSD
- **Graphics Card**: Intel UHD Graphics
- **Webcam**: Integrated HD webcam

Software Configuration:

- **Operating System**: Windows 11 Pro
- **Python Version**: Python 3.9.7
- **Libraries:**
  - **OpenCV**: Version 4.5.3
  - **face_recognition**: Version 1.4.0
- **IDE**: Visual Studio Code 1.92
- **Additional Tools**:
  - dlib Compiled Binary 3.12
  - numpy 1.26.4
  - CMake