REAL-TIME FACE DETECTION AND RECOGNITION

A PROJECT REPORT

Submitted by

NANDHINI K (2303811724322075)

In partial fulfillment of requirements for the award of the course

AGI1242 – MACHINE LEARNING TECHNIQUES

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNACOLLEGEOFTECHNOLOGY
(An Autonomous Institution, affiliated to Anna University Chennai and approved by AICTE, New Delhi)

SAMAYAPURAM–621112

DECEMBER, 2024

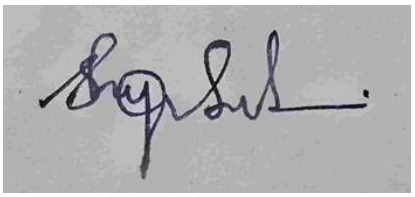
K. RAMAKRISHNACOLLEGEOFTECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM–621112

BONAFIDECERTIFICATE

Certified that this project report titled "**REAL - TIME FACE DETECTION AND RECOGNITION**" is the bonafide work of **NANDHINI K (2303811724322075)**, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.



SIGNATURE

Dr.T.AVUDAIAPPAN M.E.,Ph.D.,

HEADOFTHEDEPARTMENT

ASSOCIATE PROFESSOR

Department of Artificial Intelligence

K. Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

SIGNATURE

Mr.SHYAMSUNDAR T, M.E.,

SUPERVISOR

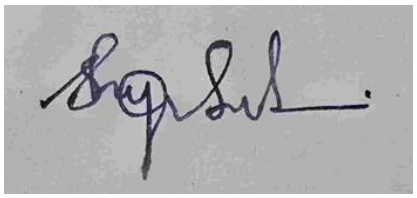
ASSISTANT PROFESSOR

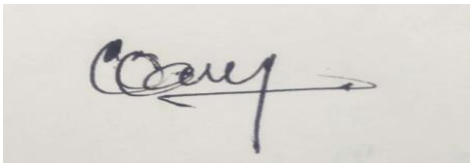
Department of Artificial Intelligence

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voice examination held on **7/12/2024**





INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on "**REAL - TIME FACE DETECTION AND RECOGNITION**"is the result of original work done by us and to the best of our knowledge, similar work has not been submitted to "**ANNA UNIVERSITY CHENNAI**" for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted in the partial fulfillment of the requirement of the award of the course **AGI1242 –MACHINE LEARNING TECHNIQUES.**

Signature

NANDHINI K.

STUDENT NAME

Place: Samayapuram

Date: 7/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, "**K. Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman,**Dr.K.RAMAKRISHNAN,B.E.**,for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr.N.VASUDEVAN,M.TECH.,Ph.D.**,Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T. AVUDAIAPPAN, M.E., Ph.D.,** Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**,for providing his en couragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mr**.**SHYAMSUNDAR T, M.E.,** Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for his incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and LabTechnicians of our departments who rendered their help during the period of the work progress.

VISIONOFTHEINSTITUTION

To emerge as a leader among the top institutions in the field of technical education.

MISSION OF THE INSTITUTION

- Produce smart technocrats with empirical knowledge who can surmount the global challenges.
- Create adiverse,fully-engaged,learner-centriccampus environment to provide quality education to the students.
- Maintain mutually beneficial partnerships without alumni, industry, and Professional associations.

VISIONOFDEPARTMENT

To excel in education, innovation, and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

MISSION OF DEPARTMENT

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conducive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAMEDUCATIONALOBJECTIVES

Graduates will be able to:

1. PEO1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

2. PEO2: Provide industry-specific solutions for the society with effective communication and ethics.

3. PEO3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAMSPECIFICOUTCOMES (PSOs)

PSO1: DomainKnowledge

Toanalyze,design,and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO2:QualitySoftware

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO3:Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems.

PROGRAMOUTCOMES(POs)

Engineering students will be able to:

Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

The engineerand society: Apply reasoning informed by contextual knowledge to assess societal,health,safety,legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

Environment and sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Life-longlearning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

Real-time face detection and recognition are crucial components of computer vision, enabling systems to detect, track, and identify individuals in various environments. Face detection involves identifying the presence of human faces within digital images or video streams, typically achieved using methods like Haar Cascades, HOG (Histogram of Oriented Gradients), or deep learning techniques such as CNN (Convolutional Neural Networks). In contrast, face recognition goes a step further by distinguishing and identifying specific individuals based on facial features.The main challenges in real-time face detection and recognition include variations in lighting, face orientations, and occlusions, all of which must be handled for robust and accurate performance. However, with the development of more advanced models and efficient algorithms, these systems are becoming increasingly accurate and fast, making them feasible for use in diverse real-world applications.

TABLEOFCONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS :

CV – Computer Vision

AI – Artificial Intelligence

ML – Machine Learning

CNN – Convolutional Neural Network

FPS – Frames Per Second

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT

Real-time face detection and recognition systems play a crucial role in applications such as security, authentication, and personalized services. Using computer vision techniques, the system detects faces in live video feeds captured from a webcam, employing Haar Cascade Classifiers for face detection. The process involves converting each frame to grayscale, which enhances the detection efficiency. Once faces are identified, bounding boxes are drawn, and labels are applied. In the future, the system can be extended to include face recognition, enabling the identification and authentication of individuals based on their facial features for advanced applications like access control and personalized user experiences.

1.2 PURPOSE AND IMPORTANCE OF PROJECT

The purpose of this project is to develop a real-time face detection and recognition system that can identify and track human faces in video streams. By leveraging computer vision techniques such as Haar Cascade Classifiers, the system aims to detect faces efficiently and accurately from a live webcam feed. The primary objective is to provide a robust, real-time solution that can be extended to include face recognition, enabling the system to identify specific individuals based on their facial features.

The importance of this project lies in its wide range of applications across multiple industries. In security, face detection and recognition are used for surveillance and access control, offering a reliable and non-intrusive method of identifying individuals. In the realm of personalized services, it can be used for user authentication, creating customized experiences based on the individual's identity. Additionally, this technology is also integral to modern smart devices

and systems, where it enhances user interaction and convenience. By combining simplicity with functionality, this project can contribute to the growing demand for efficient, real-time identification and security systems in both public and private sectors.

1.3 OBJECTIVE

The objective of this project is to develop a real-time face detection and recognition system that uses computer vision techniques to detect and identify faces in live video streams. The system utilizes Haar Cascade Classifiers to efficiently detect faces from a webcam feed, with future plans to extend it to face recognition for identifying individuals based on their facial features. By converting video frames to grayscale, the system optimizes detection performance for real-time applications. This project aims to be applicable in areas such as security, access control, and personalized services, providing a non-intrusive and reliable method for identification and authentication. Ultimately, it strives to contribute to the growing demand for intelligent, real-time systems in both public and private sectors, offering enhanced safety and user interaction.

1.4 PROJECT SUMMARIZATION

This project focuses on building a real-time face detection system that processes video streams in real time, detecting faces and marking them with bounding boxes. The system leverages the Haar Cascade Classifier, an effective machine learning-based method for detecting objects, which in this case, is used for face detection. The system operates by capturing frames from a live video feed, converting them to grayscale, and applying the classifier to detect human faces. Once faces are detected, the system draws rectangles around the faces and displays them in the output feed. The project's future goal is to expand this system to recognize and authenticate individuals, creating opportunities for applications in security, access control, and personalized user

experiences. The importance of this project lies in its potential to provide a fast, efficient, and scalable solution for real-time face detection and recognition.

CHAPTER 2

PROJECT METHODOLOGY

2.1 INTRODUCTION TO SYSTEM ARCHITECTURE

The system architecture diagram provided outlines the flow of operations for a real-time face detection and recognition system. It highlights the key components involved and how they interact to perform face detection tasks.

1. **VideoCapture**: This component is responsible for starting the video capture from the user's camera. If the camera is inaccessible, an error message is displayed, prompting the user to check their connection or camera settings.
2. **HaarCascade**: Once the video feed is captured, the frames are processed using Haar Cascade Classifiers. This is where the face detection algorithm runs. The frame is first converted to grayscale and then analyzed for faces, with the coordinates of detected faces being returned.
3. **Window**: The processed frame (with face detection and coordinates) is then displayed in a window. This window allows the user to see the output of the face detection in real time.
4. **User Interaction**: The user can press 'q' to exit the loop, stopping the video feed and closing the windows. When the loop is exited, the system releases the camera and closes the output window to clean up resources.

2.2 DETAILED SYSTEM ARCHITECTURAL DIAGRAM

The detailed system architecture of the real-time face detection and recognition system consists of several key components. The User Interface (UI**)** displays the live video feed and feedback to the user while allowing user input, such as exiting the system. The Video Capture Module captures frames from the webcam and sends them

to the Face Detection Module, which detects faces in each frame and returns their coordinates. Optionally, the Face Recognition module compares the detected faces to a known database to identify individuals. The Processing and Display Module draws bounding boxes around detected faces and displays the processed video feed in real time. The Exit Mechanism listens for user commands, such as pressing 'q', to terminate the system, release camera resources, and close windows. Additionally, the Error Handling component ensures that any issues, such as camera access errors, are properly managed and displayed to the user. This architecture facilitates real-time face detection and recognition while ensuring smooth user interaction and error management.

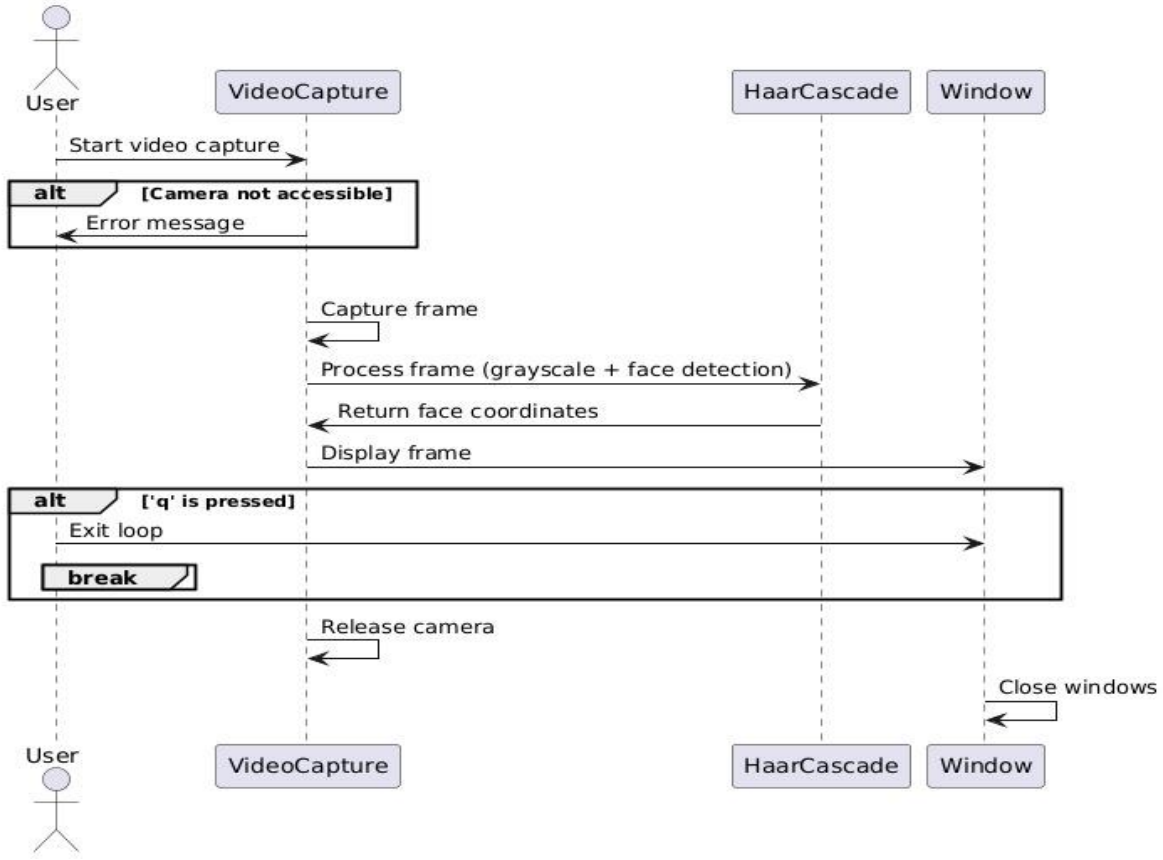


Fig 2.1: Architecture Diagram

CHAPTER 3

MACHNE LEARNING ALGORITHM USED

3.1 HAAR Cascade Classifier

The Haar Cascade Classifier is a machine learning-based object detection technique widely used for tasks such as face detection, eye detection, and other object recognitions in images and videos. It is an efficient and well-established method that leverages the concept of Haar features

3.1.1 Simplicity:

- **Easy to Understand**: The basic concept of Haar Cascade involves using simple rectangular features to detect patterns in images, making it easy to implement and understand.
- **Lightweight**: The classifier is computationally less intensive compared to more complex methods like deep learning-based models. It is well-suited for systems with limited computational resources.

3.1.2 Real-Time Detection:

- **Fast Processing**: The Haar Cascade Classifier is capable of detecting objects (like faces) in real-time due to its efficient use of Haar-like features and the integral image technique.
- **Cascade Structure**: The multi-stage cascade structure allows for quick rejection of non-object regions, improving processing speed by focusing only on promising regions.

3.1.3 Robust to Small Variations:

- **Tolerance to Variations**: Although not perfect, the Haar Cascade Classifier can detect objects under moderate variations in size, orientation, and lighting

conditions. This makes it more adaptable to real-world scenarios compared to simpler detection methods.

3.1.4 Scalable:

- Besides face detection, the Haar Cascade Classifier can be trained to detect other objects like pedestrians, cars, or animals, making it versatile for various applications.

3.1.5 Low Memory Usage:

- Efficient Memory Consumption: The algorithm is optimized to consume less memory compared to more complex machine learning models, such as deep neural networks, which is beneficial for mobile devices or systems with limited memory.

3.2 ADVANTAGES AND DISADVANTAGES

3.2.1 Advantages of Haar Cascade Classifier:

- The Haar Cascade Classifier is fast and efficient, allowing real-time object detection, which makes it ideal for applications like video surveillance and live face detection. It comes with pre-trained models, reducing the need for extensive training. The algorithm is computationally lightweight, making it suitable for devices with limited resources. Its simplicity makes it easy to implement, and it can be trained for different objects, enhancing its versatility in various tasks.

3.2.2 Disadvantages of Haar Cascade Classifier:

- The Haar Cascade Classifier has lower accuracy compared to modern deep learning methods, particularly in complex scenarios with occlusion or poor lighting. It can produce false positives and false negatives, affecting reliability. The classifier struggles with scale variations, poor image quality, and complex backgrounds. Additionally, training a custom classifier requires large amounts of labelled data, which can be time-consuming and resource-intensive.

CHAPTER 4

MODULES

4.1 OpenCV (cv2) Module

Description:

A library used for computer vision tasks like image processing and video capturing. In the program, it facilitates webcam video capture, frame processing, and displaying results.

4.2 cv2.rectangle Module

Description:

Draws geometric shapes, such as rectangles, on images. It is used to highlight detected faces by drawing rectangles around them in each video frame.

4.3 cv2.putText Module

Description:

Adds text to images or video frames. In the program, it labels detected faces with the word "Face" near the rectangles for easy identification.

4.4 cv2.imshow Module

Description:

Displays images or video frames in a window. It shows the processed video frames with detected faces highlighted by rectangles and labeled with "Face."

CHAPTER 5

CONCLUSION & FUTURE SCOPE

5.1 CONCLUSION:

Real-Time Face Detection and Recognition using Haar Cascade Classifiers highlights an efficient approach for detecting faces in real-time. Its simplicity, speed, and ease of implementation make it ideal for basic applications. While effective in controlled environments, it faces limitations in handling complex scenarios like varying lighting, occlusions, or orientations. Overall, it demonstrates the practical utility of Haar Cascades in real-time face detection tasks.

5.2 FUTURE ENHANCEMENT

The future scope of this project lies in enhancing the robustness and accuracy of the face detection system. One avenue for improvement is integrating deep learning models (such as Convolutional Neural Networks) to handle more challenging real-world conditions, including variations in lighting, pose, and occlusion. Additionally, the system can be expanded to include face recognition capabilities, enabling it to identify and verify individuals, which would be beneficial in applications such as security systems, attendance tracking, or personalized user experiences. Exploring the integration of multi-object detection and expanding the use of face detection in dynamic environments (e.g., in crowded places or outdoor settings) are other potential areas for development. Furthermore, with advancements in hardware and processing power, real-time detection and recognition systems could be scaled up for use in more sophisticated applications like autonomous vehicles and surveillance systems.

APPENDICES

APPENDIX A-SOURCE CODE

```
import cv2

# Load the Haar Cascade for face detection
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

# Start the video capture (0 for default webcam)
video_capture = cv2.VideoCapture(0)

if not video_capture.isOpened():
    print("Error: Unable to access the camera")
    exit()

print("Press 'q' to exit the application.")

while True:
    # Capture frame-by-frame
    ret, frame = video_capture.read()

    if not ret:
        print("Error: Failed to capture video frame.")
        break

    # Convert the frame to grayscale (Haar Cascade works on grayscale images)
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detect faces
    faces = face_cascade.detectMultiScale(
        gray_frame,
        scaleFactor=1.1,  # How much the image size is reduced at each scale
        minNeighbors=5,   # Minimum number of neighbors a rectangle should have to be
retained
        minSize=(30, 30)  # Minimum size of the detected face
    )

    # Draw rectangles around detected faces and label with 'Face'
```

```
    for (x, y, w, h) in faces:
        # Draw rectangle around the face
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

        # Add label 'Face' near the top of the rectangle
        cv2.putText(frame, "Face", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0,
255, 0), 2)

    # Display the resulting frame
    cv2.imshow("Face Detection", frame)

    # Break the loop when 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the capture and close OpenCV windows
video_capture.release()
cv2.destroyAllWindows()
```

APPENDIX B–SCREENSHOTS

RESULT AND DISCUSSION

OUTPUT: