OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

AI-Based Real-Time Face Tracking and Following Camera Using Raspberry Pi

Priyanka K, Ashwin Gowda L , Naveen Kumar S, Ravi Kumar B R , Lavanya Vaishnavi ,

[1]*UG scholar, Dept. of Electronics and Communication Engineering, R L Jalappa institute of Technology, Doddaballapur, Affiliated to VTU Belagavi. India*

[a]*priyankapriyanka71169@gmail.com* , [b]*ashwingowdal40@gmail.com*

[c]*naveenkumars27736@gmail.com* , [d] *ravikumarbr082004@gmail.com*

[2]*Assistant Professor, Department of Electronics and Communication Engineering*

R L Jalappa Institute of Technology, Doddaballapur, Affiliated to VTU Belagavi, India

lavanyavaishnavi@gmail.com

Abstract: The AI-Based Real-Time Face Tracking and Following Camera using Raspberry Pi is an intelligent embedded vision system designed to automatically detect, track, and follow a human face in real time. Traditional camera systems require manual positioning and constant adjustment, which limits their effectiveness in dynamic environments such as surveillance, video conferencing, robotics, and human–machine interaction. This project addresses these limitations by integrating artificial intelligence with compact, low-cost hardware to achieve autonomous camera control.

The system is built around a Raspberry Pi, which serves as the main processing and control unit. A camera module continuously captures live video frames and forwards them to the Raspberry Pi for processing. Face detection and tracking are performed using AI and machine learning techniques implemented through TensorFlow and OpenCV. By analyzing each frame, the system identifies facial features and calculates the position of the face relative to the camera's field of view. Based on this information, control signals are generated to drive servo motors, enabling the camera to pan and tilt smoothly so that the detected face remains centered.

The proposed system demonstrates real-time performance while maintaining low power consumption and cost efficiency, making it suitable for both academic and practical applications. The modular design allows for easy scalability and integration with additional features such as multiple face tracking, face recognition, or IoT-based remote monitoring. Experimental results show that the system can accurately track moving faces under varying lighting conditions with minimal delay.

Overall, the AI-based face tracking and following camera using Raspberry Pi highlights the potential of combining embedded systems with artificial intelligence to create smart, autonomous vision solutions. This project provides a strong foundation for future advancements in intelligent surveillance, robotics, and interactive camera systems.

Keywords: Artificial Intelligence, Computer Vision, Face Detection, Face Tracking, Haar Cascade, OpenCV, Raspberry Pi, Real-Time Processing, Servo Motors, TensorFlows

---

I. INTRODUCTION

Artificial Intelligence (AI) is rapidly transforming modern technology by enabling machines to perform tasks that typically require human intelligence. One of the most significant and emerging applications of AI is real-time visual tracking, which allows machines to detect, analyze, and follow objects or individuals using image processing techniques. Face detection and tracking have become essential components of many advanced systems, including automated surveillance, human–computer interaction, assistive robotics, biometric authentication, smart cameras, automobile safety systems, and social robotics. Cameras capable of automatically adjusting their viewing direction to follow a person are widely used in security monitoring, video conferencing, smart classrooms, and interactive robotic platforms.

This project, titled "AI-Based Real-Time Face Tracking System**,"** focuses on designing and implementing an intelligent camera system that can locate a human face within a live video stream and physically adjust the camera's orientation to keep the face centered in the frame. The system is built using a Raspberry Pi as the main controller, OpenCV as the computer vision and face detection engine, and servo motors to provide mechanical movement of the camera.

A webcam continuously captures video frames, which are processed in real time using OpenCV to detect facial features through machine-trained algorithms. Once a face is detected, the Raspberry Pi calculates the position of the face relative to the video frame and generates appropriate control signals to the servo motors. Two servo motors enable pan (left–right) and tilt (up–down) movements, allowing the camera to dynamically follow the face as it moves.

The system primarily uses Haar Cascade face detection, a classical yet efficient machine learning technique developed by Viola and Jones. This method employs Haar-like features and the AdaBoost algorithm to form a cascade of classifiers that quickly reject non-face regions and focus on potential face areas. Due to its low computational requirements and fast execution, Haar Cascade detection is well suited for real-time applications on embedded platforms like the Raspberry Pi.

Overall, this project demonstrates how embedded AI and computer vision can be effectively combined to create an autonomous, low-cost, and practical face-tracking system.

The camera is a vital component of an AI-based real-time face tracking and following system, as it provides the visual input required for face detection and tracking. In a Raspberry Pi–based setup, the camera continuously captures live video frames that are processed by artificial intelligence and computer vision algorithms to identify and track human faces in real time. The performance of the entire system largely depends on the camera's resolution, frame rate, and compatibility with the Raspberry Pi.

Commonly, a Raspberry Pi Camera Module or a USB webcam is used in this application. The Raspberry Pi Camera Module is widely preferred due to its compact size, low power consumption, and direct connection to the Raspberry Pi via the CSI interface. It supports high-quality image capture and smooth video streaming, which are essential for accurate face detection. The camera supplies real-time image data to the Raspberry Pi, where libraries such as OpenCV and AI frameworks like TensorFlow analyze each frame to locate facial features.

Once a face is detected, its position within the frame is calculated, and this information is used to control pan and tilt servo motors. The camera then moves automatically to keep the face centered, enabling continuous and smooth tracking of a moving subject. Overall, the camera acts as the "vision sensor" of the system, making intelligent tracking, automation, and real-time interaction possible in AI-based embedded applications.

1.1. Video Processing using Tenserflow.

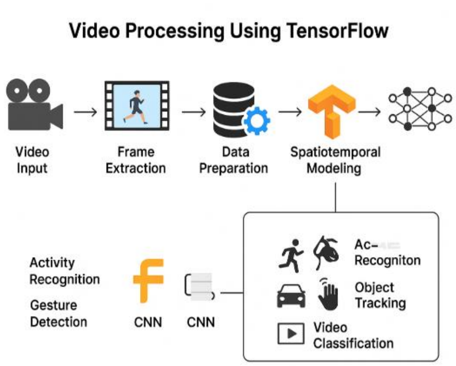

Figure.1 overview of Video Processing using Tenserflow.

In Figure.1 we are showing that Video processing using TensorFlow is widely adopted for developing intelligent video analysis systems due to its flexibility, scalability, and strong support for deep learning models. Video data consists of sequential frames with temporal information, making analysis complex. TensorFlow simplifies this process by enabling models to learn both spatial and temporal features, supporting applications such as object tracking, activity recognition, gesture detection, surveillance automation, and video classification.

The video processing pipeline begins with frame extraction using tools like OpenCV. Video frames are resized, normalized, and batched before being processed. TensorFlow's tf.data API enables efficient data pipelines for loading, preprocessing, shuffling, and augmenting large datasets in parallel. Augmentation techniques such as flipping, rotation, cropping, and brightness adjustment improve model generalization.

Convolutional Neural Networks (CNNs) are used to extract spatial features from individual frames, forming the basis for object detection and scene understanding. TensorFlow provides pre-trained CNN models such as MobileNet, Inception, and EfficientNet through TensorFlow Hub, allowing transfer learning for faster training and improved accuracy.

To capture temporal dynamics, TensorFlow supports Recurrent Neural Networks (RNNs) including LSTM and GRU, which model frame-to-frame motion and sequence patterns. Additionally, 3D CNNs learn spatial and temporal features simultaneously and are effective in applications like surveillance and autonomous driving.

TensorFlow also supports video object detection and tracking using models such as YOLO, SSD, and EfficientDet. Deployment is simplified through TensorFlow Lite and TensorFlow.js for real-time processing on edge devices and web platforms.

1.2. Face Recognisation using TensorFlow.

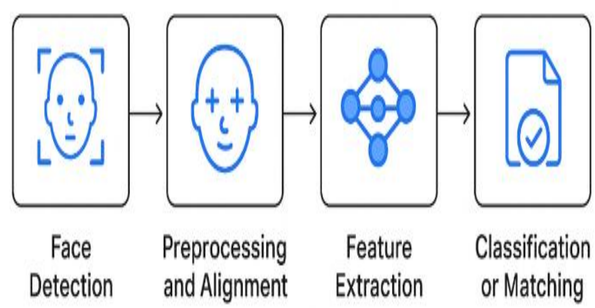

Figure.2 abstract view of Face Recognisation using TensorFlow.

In Figure.2 shows the Face recognition using TensorFlow is a powerful and widely used approach for applications such as identity verification, security systems, attendance monitoring, and human–computer interaction. TensorFlow provides a comprehensive ecosystem of deep learning tools, pre-trained models, and flexible APIs that enable the development of accurate and real-time face recognition systems.

The process begins with face detection, which identifies and localizes human faces within an image or video frame. TensorFlow supports detection models such as MTCNN, Haar Cascades through OpenCV integration, and SSD-based detectors, which also extract facial landmarks to support proper face alignment. After detection, preprocessing is performed, including resizing, normalization, and alignment of faces to a standard orientation, which significantly improves recognition accuracy.

Feature extraction is the core stage of face recognition and is typically handled by deep Convolutional Neural Networks (CNNs). TensorFlow supports popular architectures such as FaceNet, Inception-ResNet, and MobileFaceNets. These models generate facial embeddings high-dimensional vectors that represent unique facial characteristics. FaceNet uses triplet loss to minimize the distance between embeddings of the same individual while maximizing separation between different individuals.

Recognition is achieved by matching embeddings using distance metrics or classifiers such as SVM, KNN, or Softmax. TensorFlow also supports real-time deployment through TensorFlow Lite and TensorFlow.js, enabling execution on mobile, embedded, and web platforms. Despite challenges such as lighting variations, occlusion, and privacy concerns, TensorFlow remains a leading framework for robust and scalable face recognition systems.

II.LITERATURE SURVEY

Kim et al. proposed a real-time criminal face recognition system for surveillance using deep learning and face tracking. The system integrates RetinaFace with Feature Pyramid Networks and Deformable Convolutions for accurate face detection under varying scales and poses. FaceNet is used to extract discriminative facial embeddings, while SORT with Kalman filtering ensures consistent identity tracking across frames. A cumulative score-based identification strategy improves reliability by reducing frame-level misclassification. Experimental results showed over 30 FPS performance with high accuracy, demonstrating the system's suitability for real-world intelligent surveillance and crime prevention applications[1].

This paper presents Trackez, an IoT-based real-time object tracking system using Mez and the Follow-Satisfy-Loop (FSL) algorithm. The system achieves 3D tracking using only 2D pixel data, eliminating the need for depth sensors. Mez dynamically adapts video quality based on network conditions, significantly reducing latency, while FSL maintains object centering through servo control. Implemented on Raspberry Pi with cloud-based CNN detection, the system demonstrated improved tracking accuracy and latency reduction. Trackez is well suited for robotics, smart surveillance, and edge-based IoT vision systems[2].

Narayanan et al. introduced a YOLOv9-based framework for human face detection and counting in complex environments. The model incorporates Programmable Gradient Information (PGI) to enhance gradient flow and prevent overfitting. Using multi-scale feature fusion and noise-aware preprocessing, YOLOv9 achieved improved detection accuracy compared to previous YOLO versions. The system performed well under occlusion, varying illumination, and mixed human-animal faces. Experimental results showed high FPS and reduced error rates, making it suitable for real-time surveillance, crowd monitoring, and edge-device deployment [3].

This extended study further validates the Trackez framework by analyzing its performance under varying motion speeds and network conditions. The combination of Mez and FSL enables latency-tolerant tracking and stable servo control without depth estimation. The system demonstrated over 80% accuracy improvement and significant latency reduction. Experiments conducted in indoor environments confirmed robustness for slow and fast-moving objects. Despite limitations related to servo vibration and computational cost, the study highlights Trackez as an effective solution for real-time IoT-based object tracking applications[4].

This review paper surveys stereo-camera-based 3D face tracking techniques. It discusses depth estimation using disparity maps and tracking methods such as KLT, particle filtering, Mean Shift, and CamShift. The paper highlights how depth information improves robustness against pose variation and occlusion compared to 2D approaches. It also examines probabilistic filters like PHD for multi-face tracking. While stereo systems enhance accuracy, challenges such as illumination sensitivity and computational complexity remain. The review provides valuable insights into traditional and hybrid 3D face tracking methodologies[5].

Ren et al. proposed a cross-camera multi-face tracking system based on a Double Triplet Network (DTN). The DTN improves feature discrimination by simultaneously minimizing intra-class and maximizing inter-class distances. YOLOv3 is used for face detection, while Chinese Whisper clustering reduces redundancy within each camera view. Facial embeddings are shared across cameras for re-identification. Experimental results showed high accuracy and scalability across large datasets. The system is suitable for smart city surveillance, public security, and multi-camera monitoring environments[6].

This paper presents a real-time human tracking system combining CNN-LSTM for spatiotemporal feature extraction with Q-learning for adaptive tracking decisions. The CNN extracts visual features, while LSTM captures motion patterns across frames. Reinforcement learning optimizes tracking actions under dynamic conditions. Experimental results demonstrate improved tracking stability and accuracy compared to traditional methods. The approach is effective in handling occlusion and abrupt motion changes, making it suitable for intelligent surveillance and autonomous monitoring systems[7].

Kang and Ma developed a real-time eye tracking system capable of handling both bare and sunglasses-wearing faces. The system uses machine learning-based eye region detection and temporal filtering to ensure robustness under varying lighting conditions. Designed for augmented reality head-up displays, the method achieves high accuracy and low latency. Experimental evaluations confirm reliable eye tracking even with partial occlusion. The study demonstrates the feasibility of real-time eye tracking for AR applications, driver monitoring, and human computer interaction systems[8].

Problem Statement

Conventional cameras require manual adjustment to keep a person in view, which is impractical in dynamic environments such as surveillance, presentations, and robotic systems. A Raspberry Pi–based AI tracking camera offers a low-cost, low-power, and intelligent solution capable of autonomous face tracking.

However, the Raspberry Pi has limited CPU/GPU resources, power constraints, and strict real-time requirements. Traditional computer vision techniques like Haar cascades are computationally efficient but lack robustness under varying lighting, occlusions, and pose changes. In contrast, modern deep-learning models provide higher accuracy but require optimization techniques such as pruning and quantization to achieve real-time performance on the Pi.

In addition, accurate face tracking demands smooth and stable pan-tilt control using servo motors, which requires well-tuned control algorithms, real-time feedback, and reliable GPIO interfacing. The system must also remain cost-effective, modular, and easy to replicate, allowing future enhancements such as multi-face tracking, cloud integration, or advanced control strategies.

This project aims to develop a real-time AI-based face tracking and following camera using Raspberry Pi that dynamically detects and tracks a human face, automatically controlling pan-tilt movements within a 180° range to maintain continuous focus.

III.OBJECTIVE

1. To design and develop a low-cost, intelligent face tracking camera system using Raspberry Pi suitable for real-time applications.
2. To implement AI-based face detection and tracking algorithms capable of accurately identifying and following a human face under varying lighting and movement conditions.
3. To achieve real-time performance on resource-constrained hardware by optimizing computer vision and deep learning models for efficient execution on Raspberry Pi.
4. To control pan–tilt camera movement automatically using servo motors based on the detected face position to keep the face centered in the frame.
5. To ensure smooth and stable camera motion through effective control algorithms and precise GPIO-based servo interfacing.
6. To maintain continuous face tracking within a wide field of view, enabling up to 180° pan and tilt coverage.
7. To develop a modular and scalable system architecture that allows future enhancements such as multi-face tracking, face recognition, or cloud connectivity.
8. To reduce the need for manual camera operation in applications such as surveillance, smart monitoring, presentations, and human–robot interaction.
9. To create an energy-efficient solution suitable for long-duration operation in embedded and portable environments.
10. To validate system performance in terms of accuracy, responsiveness, and reliability under real-world conditions.

Methodology of implementation.

Block diagram

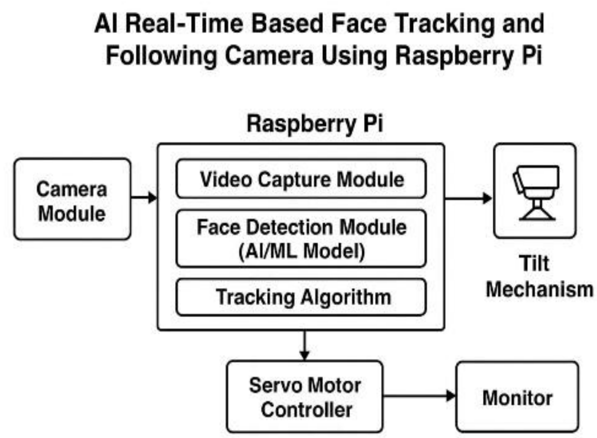

Figure.3 Block diagram

In Figure.3 shows that the block diagram illustrates the functioning of an AI real-time face-tracking and face-following camera system using a Raspberry Pi. This system integrates computer vision, machine learning, and hardware control to automatically detect a person's face and physically follow their movement using mechanical rotation and tilt mechanisms.

The process begins with the Camera Module, which continuously captures live video frames from the environment. These frames are sent directly to the Video Capture Module inside the Raspberry Pi. This module is responsible for converting raw camera data into a format suitable for analysis. It ensures stable frame rates, manages buffer handling, and prepares each video frame for further processing.

Once a frame is captured, it is forwarded to the Face Detection Module, which uses AI or machine learning techniques to detect human faces within the image. This module typically relies on models such as Haar cascades, HOG-based detectors, or deep learning frameworks like OpenCV DNN, TensorFlow Lite, or MediaPipe. The main role of this module is to analyze each frame and determine whether a face is present, and if so, identify its exact coordinates usually via a bounding box around the detected face. These coordinates form the essential input for the next stage, the tracking algorithm.

The Tracking Algorithm takes the face coordinates and calculates how far the face is from the center of the camera's field of view. If the detected face shifts to the left, right, up, or down, the algorithm determines the required correction by computing the direction and degree of rotation necessary for the camera to realign itself. Common tracking techniques include centroid tracking, Kalman filtering, or simple proportional control (PID-like adjustments). This algorithm ensures smooth and continuous tracking rather than abrupt movements.

The calculated movement instructions are then sent to the Servo Motor Controller, which acts as the interface between the Raspberry Pi and the physical motors. The controller uses PWM (Pulse Width Modulation) signals to precisely control the angle of the servo motors. These motors actuate the Tilt Mechanism, enabling the camera to rotate horizontally (pan) or tilt vertically. As the servo motors adjust the camera's orientation, the system physically follows the detected face, keeping it centered in the frame.

The processed video feed or status output may be displayed on a connected Monitor. This allows the user to visually observe the real-time tracking performance, debugging details, or system feedback. It also helps in verifying the accuracy of face detection and camera positioning.

The system workflow begins with the Camera Module, which captures continuous video frames and sends them to the Raspberry Pi's Video Capture Module for formatting, buffering, and frame-rate stabilization. These processed frames enter the Face Detection Module, where AI-based techniques such as Haar Cascades, HOG detectors, OpenCV DNN, TensorFlow Lite, or MediaPipe identify human faces and extract their coordinates through bounding boxes. These coordinates are then passed to the Tracking Algorithm, which determines how far the detected face is from the center of the camera's field of view. Using methods like centroid tracking, Kalman filtering, or proportional control, the algorithm calculates the required movement direction and angle needed to re-center the face, ensuring smooth, uninterrupted tracking.

The movement instructions from the tracking stage are sent to the Servo Motor Controller, which generates precise PWM signals to adjust the servo motors. These motors operate the Tilt Mechanism, enabling both panning and tilting so the camera can physically follow the face in real time. As the servos reposition the camera, the system maintains the subject at the center of the frame. The processed output or live video feed can be viewed on a connected monitor, allowing users to observe tracking behavior, verify accuracy, and check system performance. This complete workflow ensures efficient, real-time face tracking with smooth camera movement and reliable detection.

Flowchart of implementation

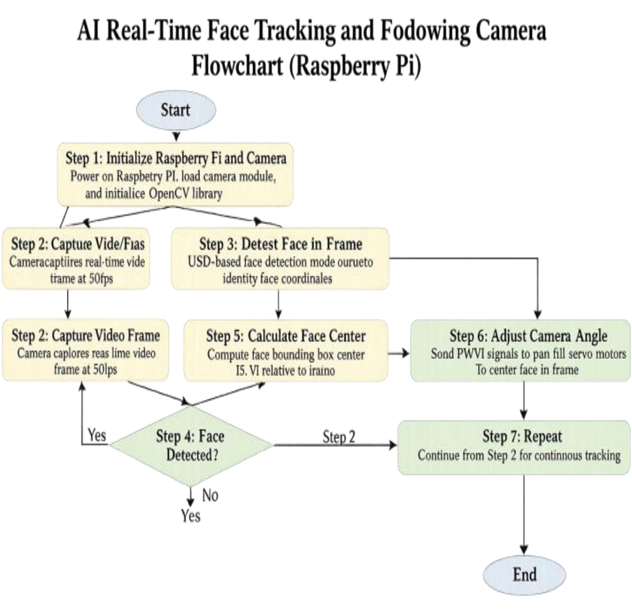

Figure.4 Flowchart of implementation.

In Figure.4 we are show that the flowchart illustrates the working process of an AI real-time face tracking and following camera system implemented using a Raspberry Pi. The system begins with the initialization stage, where the Raspberry Pi is powered on, the camera module is activated, and essential computer vision libraries such as OpenCV are loaded into memory. This initialization ensures that the hardware and software environment is properly configured for capturing and analyzing video data. Once initialized, the camera immediately starts capturing continuous real-time video frames, typically at a rate of around 30 frames per second. These live frames form the input for the face detection module. As each video frame is captured, the system sends it to the AI/ML-based face detection model, which may use deep learning methods, Haar cascades, or more advanced pretrained neural networks to identify human faces within the frame. The detection model scans the frame and determines whether a face is present by drawing a bounding box around detected facial regions.

After detecting a face, the system proceeds to compute the center coordinates (x, y) of the face within the captured frame. This step is crucial because accurate tracking depends on understanding how far the face is from the center of the camera's field of view. The algorithm retrieves the bounding box values typically represented by width, height, and top-left corner coordinates and calculates the center point of the face. This allows the tracking module to determine the directional offset between the face's actual position and the ideal center of the frame. A decision block then checks whether a face has been successfully detected. If no face is present, the system simply returns to the video capture step and continues scanning each incoming frame until a face appears. This ensures uninterrupted scanning and eliminates unnecessary motor movement.

When a face is detected, the system moves to the camera adjustment stage. At this point, the algorithm compares the face's center with the predefined frame center and calculates how much the camera must rotate horizontally (pan) or vertically (tilt) to bring the face back to the center. Based on this calculation, the Raspberry Pi sends PWM (Pulse Width Modulation) signals to the attached servo motors. These motors control the pan-tilt mechanism, allowing the camera to physically follow the subject's movement. If the face moves left, right, up, or down, the algorithm generates correction signals to rotate the camera accordingly, keeping the face perfectly centered. This dynamic adjustment creates the effect of a camera that automatically follows a person in real time.

The system loops back to the frame capture step, enabling continuous and uninterrupted face tracking. This cycle capture, detect, calculate, adjust runs multiple times per second, ensuring smooth real-time performance. The flowchart thus represents a complete intelligent tracking pipeline where video acquisition, AI-based face detection, mathematical coordinate processing, and servo motor control work together seamlessly. This integrated approach allows the Raspberry Pi to function as an autonomous smart camera capable of actively monitoring, tracking, and following a human face without manual intervention.

Haar cascade face detection

In an AI-based face tracking system using TensorFlow, Haar Cascade face detection serves as a lightweight and efficient method for real-time face localization. While TensorFlow typically uses deep-learning models for face detection, Haar Cascades provide a classical machine-learning approach that runs extremely fast on CPUs and embedded platforms such as the Raspberry Pi. This makes them ideal as an initial detection stage before more advanced TensorFlow models refine the results. Haar Cascades operate by analyzing simple rectangular Haar-like features—edges, intensity differences, and contrast patterns commonly found in facial regions. These features are calculated using an integral image technique, which enables rapid computation even with limited hardware resources.

The effectiveness of Haar Cascades comes from the AdaBoost algorithm, which plays a key role in selecting the most important features from thousands of possibilities. AdaBoost (Adaptive Boosting) trains the classifier by combining many weak learners into a strong classifier capable of distinguishing faces from non-face regions with high accuracy. The classifier is structured into multiple stages, forming a cascade. Early stages quickly eliminate background areas with minimal computation, while later stages perform more detailed analysis on regions likely to contain a face. This step-by-step rejection process significantly reduces computational load.

In TensorFlow-based face tracking systems, integrating Haar Cascades offers several advantages. They provide fast and consistent face localization, reduce dependency on high-power deep-learning models, and serve as a reliable fallback when neural network inference becomes slow or resource-heavy. Combining Haar Cascades with TensorFlow enhances responsiveness and overall tracking stability. The hybrid approach ensures real-time performance, making it well-suited for embedded applications such as surveillance, robotics, and smart camera systems where both speed and accuracy are required.

Hardware and Software Used

The hardware components used in this AI-based face tracking system form the foundation for real-time image capture, processing, and camera movement. The core processing unit is the Raspberry Pi 4 Model B, a compact single-board computer equipped with a 40-pin GPIO header that supports I2C, SPI, UART, and PWM interfaces for controlling external devices. The Raspberry Pi Camera Module V2, featuring an 8MP Sony IMX219 sensor, connects through the CSI interface to capture high-resolution images and 1080p video for computer vision tasks. Two SG90 micro servo motors provide the physical movement required for the camera's pan and tilt functions, offering a 0°–180° rotation range controlled through precise PWM signals. These motors are supported by a servo mount or bracket that ensures smooth mechanical motion and stable camera alignment. A 5V, 2A (or higher) power supply is essential to deliver consistent power to both the Raspberry Pi and the servo motors, preventing voltage drops. Jumper wires and a breadboard are used to establish flexible, solder-free electrical connections during circuit setup and prototyping.

The software environment integrates multiple tools and libraries to enable efficient image processing, detection, and hardware control. Raspberry Pi OS (Buster/Bullseye) acts as the operating system, providing the platform for running Python scripts, OpenCV, and supporting libraries. Python 3.9 serves as the primary programming language for implementing the face tracking algorithm, integrating TensorFlow for detection, OpenCV for video processing, and GPIO libraries for servo control. The OpenCV library captures frames from the camera, preprocesses them, and detects face positions, acting as the interface between the camera and AI models. NumPy is used for handling image arrays and performing fast mathematical operations required during preprocessing and coordinate calculations. The imutils library simplifies common OpenCV tasks such as resizing and rotating frames, helping maintain smooth and consistent real-time tracking performance. PuTTY, a lightweight terminal emulator, is used for remote access to the Raspberry Pi via SSH, enabling configuration, debugging, and system management from a separate computer. Together, these hardware and software components create a robust, scalable, and efficient face tracking system.

IV. RESULT AND DISCUSSION

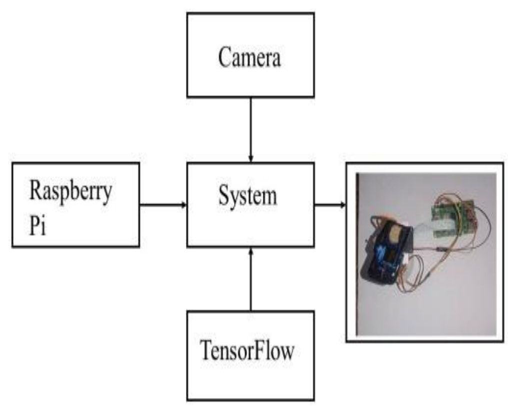

Figure 5: Block diagram

The diagram illustrates the architecture of an AI-based real-time face tracking and following camera system using a Raspberry Pi. The camera captures live video and sends it to the central system for processing. The Raspberry Pi acts as the main controller, managing data flow and hardware control. TensorFlow is integrated into the system to perform face detection and tracking using AI algorithms. Based on the processed results, control signals are generated to drive the servo motors, enabling the camera to move and continuously follow the detected face automatically and accurately.

AI-Based Real-Time Face Tracking and Following Camera Using Raspberry Pi in the dim light.

Trail output 1: Face focused to centre

In Figure 6 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labeled “Chandan H Y, 7th Semester ECE.” and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is center, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames,detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

Figure 6: face focused to centre

Trail output 2:- Face focused to left side.

In Figure 7 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labeled “Chandan H Y, 7th Semester ECE.” and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is left side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

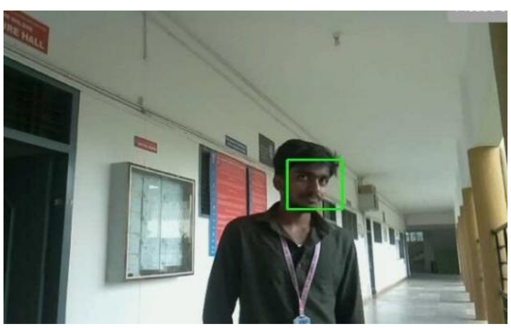

Figure 7: Face focused to left side.

Trail output 3:- face focused to right side.

In Figure 8 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labeled “Chandan H Y, 7th Semester ECE.” and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is right side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

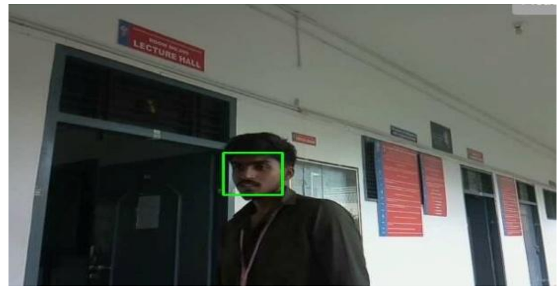

Figure 8: face focused to right side.

Trail output 4:- face focused to centre.

In Figure 9 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labeled “Anusha B M, 7th Semester ECE.” and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is center, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

Figure 9: face focused to centre.

Trail output 5:- face focused to left side

In Figure 10 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labeled “Anusha B M, 7th Semester ECE.” and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is left side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

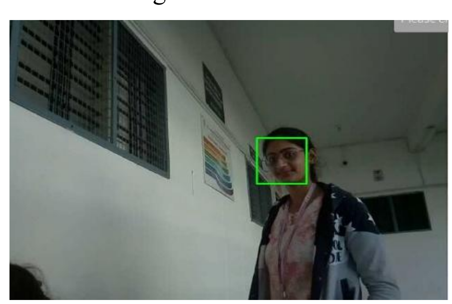

Figure 10: face focused to left side.

Trail output 6:- face focused to right.

In Figure 11 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labeled "Anusha B M, 7th Semester ECE." and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is right side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

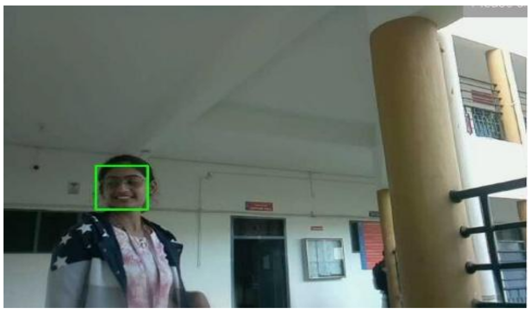

Figure 11: face focused to right side.

AI-Based Real-Time Face Tracking and Following Camera Using Raspberry Pi in the Bright light.

Trail output 7:- Face focused to centre.

In Figure 12 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labelled " Naveen Kumar S, 7th Semester ECE." and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is centre, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

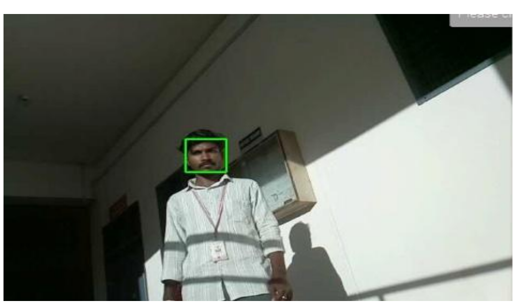

Figure 12: Face focused to centre.

Trail output 8:- Face focused to left side.

In Figure 13 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labelled " Naveen Kumar S, 7th Semester ECE." and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is left side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

Figure 13: Face focused to left side left side.

Trail output 9:- Face focused to right side.

In Figure 14 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labelled " Naveen Kumar S, 7th Semester ECE." and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is right side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detect facial features, and adjust the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

Figure 14: Face focused to right side.

Trail output 10:- Face focused to centre.

In Figure 15 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labelled " Priyanka K, 7th Semester ECE." and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is centre, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

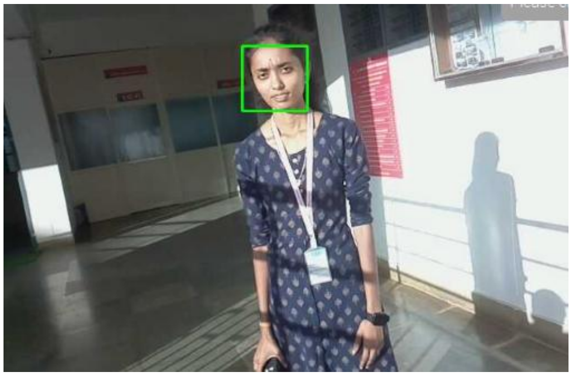

Figure 15: Face focused to centre.

Trail output 11:- Face focused to left side.

In Figure 16 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labelled " Priyanka K, 7th Semester ECE." and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is left side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

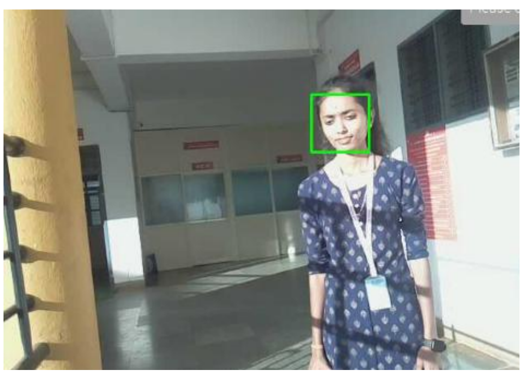

Figure 16: Face focused to left side.

Trail output 12:- Face focused to right side.

In Figure 17 we shows the functioning of an AI-Based Real-Time Face Tracking and Following Camera System using Raspberry Pi under dim-light conditions. The system accurately detects and tracks the face of the subject labelled " Priyanka K, 7th Semester ECE." and highlights the face with a green bounding box, confirming accurate identification by the OpenCV-based model. The face is right side, showing the servo-driven pan–tilt mechanism actively tracking the subject. The Raspberry Pi processes live video frames, detects facial features, and adjusts the camera position automatically. This demonstration proves effective integration of computer vision and embedded hardware for intelligent face tracking in an indoor environment.

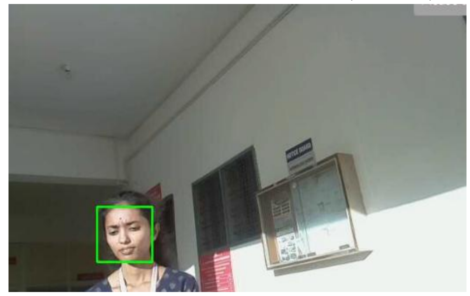

Figure 17: Face focused to right side.

V.CONCLUSION

The AI-Based Real-Time Face Tracking and Following Camera using Raspberry Pi successfully demonstrates the practical integration of artificial intelligence with embedded hardware to achieve an autonomous, intelligent visual tracking system. This project proves that compact, low-cost platforms such as the Raspberry Pi are capable of performing real-time face detection and tracking with satisfactory accuracy, responsiveness, and reliability. By continuously capturing live video, processing each frame using AI-based algorithms, and dynamically controlling camera orientation through a pan–tilt servo mechanism, the system achieves smooth and consistent tracking of a moving subject without the need for manual intervention.

The implementation highlights the Raspberry Pi as a powerful and versatile platform for machine learning, computer vision, and automation applications. The effective use of Python programming, OpenCV image processing, TensorFlow-based face detection, and GPIO-controlled motor actuation demonstrates strong coordination between software intelligence and hardware control. This seamless interaction ensures precise identification of face positions and accurate generation of corrective movements, validating the feasibility of embedded AI systems in real-world environments.

The developed system offers wide-ranging applications, including smart surveillance, automated lecture recording, robotics, interactive monitoring systems, and personal content creation. By eliminating the requirement for human-operated camera adjustments, the system enables hands-free and intelligent camera operation, improving efficiency and usability in dynamic scenarios such as presentations, conferences, and human–robot interaction environments. Its low-cost, compact design and energy-efficient operation make it suitable for continuous deployment in both personal and professional settings.

Despite its strengths, certain limitations are observed. Performance may degrade under low-light conditions, and the processing capabilities of the Raspberry Pi impose constraints on handling complex deep-learning models. Additionally, the current implementation primarily supports single-face tracking, and rapid subject movements can affect tracking accuracy. These limitations open avenues for future enhancements, such as integrating infrared cameras for low-light operation, adopting advanced deep-learning models like YOLO or MediaPipe for improved detection accuracy, and enabling multi-face tracking.

Future developments may also include cloud-based processing, face recognition, gimbal-based stabilization, battery-powered operation, gesture and voice control, and AI-driven analytics. Such enhancements would significantly increase the system's intelligence, adaptability, and functionality. Overall, the project successfully meets its objectives and establishes a strong foundation for the development of advanced, intelligent vision-based systems that integrate AI, IoT, and embedded technologies.

VI.REFERENCES

1. G.R. Chhimpa, A. Kumar, S. Garhwal, Dhiraj, F. Khan and Y. -K. Moon, "Revolutionizing Gaze-Based Human–Computer Interaction Using Iris Tracking: A Webcam-Based Low-Cost Approach With Calibration, Regression and Real-Time Re-Calibration," in *IEEE Access*, vol. 12, pp. 168256-168269,2024,doi:10.1109/ACCESS.2024.3498441
2. H. -B. Kim, N. Choi, H. -J. Kwon and H. Kim, "Surveillance System for Real-Time High-Precision Recognition of Criminal Faces From Wild Videos," in *IEEE Access*, vol. 11, pp. 56066-56082,2023,doi: 10.1109/ACCESS.2023.3282451.
3. N. Faruqui, M. A. Kabir, M. A. Yousuf, M. Whaiduzzaman, A. Barros and I. Mahmud, "Trackez: An IoT-Based 3D-Object Tracking From 2D Pixel Matrix Using Mez and FSL Algorithm," in *IEEE Access*, vol. 11, pp. 61453-61467,2023,doi: 10.1109/ACCESS.2023.3287496.
4. S. Perikamana Narayanan, M. Sabarimalai Manikandan and L. R. Cenkeramaddi, "YOLOv9-Based Human Face Detection and Counting Under Human-Animal Faces ,Complex Imaging Environments,and Image Qualities," in *IEEEAccess*, vol.13,pp.129600-129637, 2025, doi: 10.1109/ACCESS.2025.3591247.
5. N. Faruqui, M. A. Kabir, M. A. Yousuf, M. Whaiduzzaman, A. Barros and I. Mahmud, "Trackez: An IoT-Based 3D-Object Tracking From 2D Pixel Matrix Using Mez and FSL Algorithm," in *IEEE Access*, vol. 11, pp. 61453-61467,2023,doi: 10.1109/ACCESS.2023.3287496.
6. F. Alqahtani, J. Banks, V. Chandran and J. Zhang, "3D Face Tracking Using Stereo Cameras: A Review," in *IEEE Access*, vol. 8, pp. 94373-94393,2020,doi: 10.1109/ACCESS.2020.2994283.
7. G. Ren, X. Lu and Y. Li, "A Cross-Camera Multi-Face Tracking System Based on Double Triplet Networks," in *IEEE Access*, vol. 9, pp. 43759-43774, 2021, doi: 10.1109/ACCESS.2021.3061572.
8. D. A. Maharani, C. Machbub, P. H. Rusmin and L. Yulianti, "Real-Time Human Tracking Using Multi-Features Visual With CNN-LSTM and Q-Learning," in *IEEE Access*, vol. 12, pp. 13233-13247, 2024, doi: 10.1109/ACCESS.2024.3355785.
9. D. Kang and L. Ma, "Real-Time Eye Tracking for Bare and Sunglasses-Wearing Faces for Augmented Reality 3D Head-Up Displays," in *IEEE Access*, vol. 9, pp. 125508-125522, 2021, doi: 10.1109/ACCESS.2021.3110644.
10. D. A. H. Ollachica, B. K. A. Asante and H. Imamura, "Autonomous UAV Implementation for Facial Recognition and Tracking in GPS-Denied Environments," in *IEEE Access*, vol. 12, pp. 119464-119487, 2024, doi: 10.1109/ACCESS.2024.3447899.
11. X. Yang, X. Jia, M. Yuan and D. -M. Yan, "Real-time facial pose estimation and tracking by coarse-to-fine iterative optimization," in *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 690-700, Oct. 2020, doi: 10.26599/TST.2020.9010001.
12. B. Gao, Z. Li, D. Zhang, Y. Liu, J. Chen and Z. Lv, "Roadside Cross-Camera Vehicle Tracking Combining Visual and Spatial-Temporal Information for a Cloud Control System," in *Journal of Intelligent and Connected Vehicles*, vol. 7, no. 2, pp. 129-137, June 2024, doi: 10.26599/JICV.2023.9210034.
13. J. Panda, P. Kumar Nanda and T. Pradhan, "Particle Filter-Based Video Object Tracking Scheme With Target Remodeling and Reinitialization and Its Hardware Implementation Using Raspberry Pi," in *IEEE Access*, vol. 12, pp. 98285-98305, 2024, doi: 10.1109/ACCESS.2024.3428321.
14. N. Faruqui, M. A. Kabir, M. A. Yousuf, M. Whaiduzzaman, A. Barros and I. Mahmud, "Trackez: An IoT-Based 3D-Object Tracking From 2D Pixel Matrix Using Mez and FSL Algorithm," in *IEEE Access*, vol. 11, pp. 61453-61467, 2023, doi: 10.1109/ACCESS.2023.3287496.
15. Lavanya Vaishnavi D A and Anil Kumar C. "Evaluating Supervised Learning Classifier Performance for OFDM Communication in AWGN-Impacted Systems." *Results in* Engineering 26 (2025): 105178. https://doi.org/10.1016/j.rineng.2025.105178.