1. Introduction

1.1 Problem Description

This project aims to build a Smart Home Security System using an ESP32-CAM integrated with a servo motor and ultrasonic sensors. The motivation stems from the increasing need for affordable, intelligent surveillance systems in residential environments. This solution enhances traditional home security by combining real-time object detection, directional camera control, and alert mechanisms via Telegram, providing efficient monitoring with minimal manual intervention.

1.2 Motivation-

- To create an affordable and effective real-time surveillance system using the ESP32-CAM and basic sensors.
- To integrate directional object detection using ultrasonic sensors and servo motors for camera movement.
- To implement remote monitoring through Telegram for instant user alerts with captured images.
- To develop a modular, scalable design that can be expanded with face recognition, cloud storage, or voice-controlled automation.

1.3 Relevance-

- Demonstrates a practical home security application where real-time manual detection is enhanced with automated smart alerts.
- Showcases the effective use of basic hardware (ESP32-CAM, ultrasonic sensors, servo motors) for dynamic monitoring and control.
- Integrates IoT capabilities by linking the system to Telegram, enabling remote user interaction with minimal latency.
- Validates a modular and low-cost design that can serve as a prototype for scalable, AI-integrated smart security systems.

1.4 Importance -

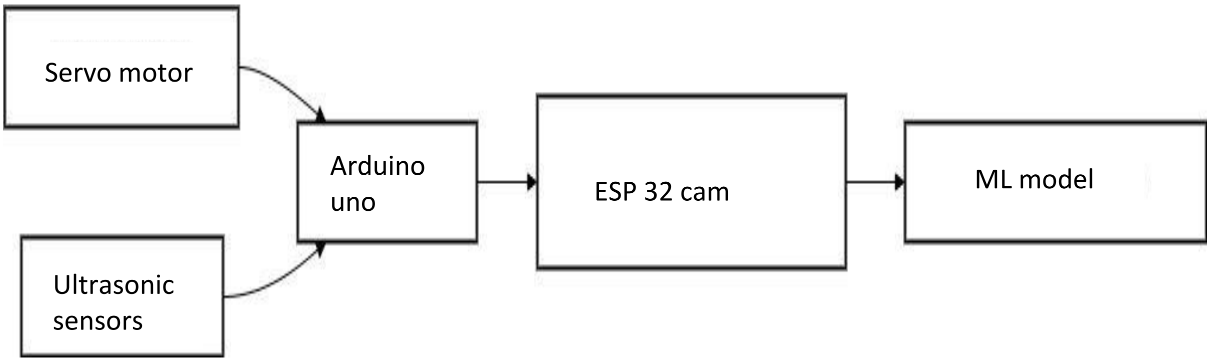
- Deepens knowledge of sensor-actuator interplay, power management, and system calibration in embedded IoT devices.
- Provides groundwork for future enhancements such as on-device face recognition, multi-sensor fusion, or closed-loop access control.
- Serves as an educational prototype that bridges theory in electronics, computer vision, and IoT through hands-on implementation.
- Promotes sustainable, low-cost solutions by leveraging open-source hardware and modular components for scalable smart-home applications.

2. Literature Review-

Security and surveillance systems have increasingly adopted microcontroller-based smart vision tools for real-time intruder detection and alert mechanisms. A prominent implementation utilizes the ESP32-CAM module to capture images upon motion detection and instantly send them via a Telegram bot. This lightweight approach offers effective remote monitoring, with the bot acting as a notification gateway. The system is affordable and fast but lacks deeper intelligence, such as recognizing whether the detected individual is known or a stranger, limiting its application in identity-aware home automation. To add intelligence, another study integrated Edge Impulse's face recognition model with the ESP32-CAM, enabling the device to identify specific individuals. Using image datasets collected via the camera and trained on Edge Impulse's cloud platform, the model was successfully deployed to the ESP32 for real-time inference. This method enhanced user awareness by distinguishing known persons from unknown ones but struggled in low-light or profile-angle conditions, reflecting limitations of embedded ML models on resource-constrained hardware. Further advancements have emerged with the use of deep learning models like YOLO5Face, which improves face detection speed and accuracy using an optimized version of the YOLOv5 architecture. Trained on the Wider Face dataset, this system delivers high accuracy and real-time performance for facial identification tasks. Despite its edge-focused design, it requires more computational power than standard microcontrollers can provide, making it suitable for inference on edge GPUs or cloud-connected platforms. Its precision falls under occlusion-heavy or cluttered backgrounds, which can occur in real-world home entry scenarios. Another model, LFFD (Light and Fast Face Detector), was proposed to bridge performance with efficiency for embedded devices. As an anchor-free detector, LFFD achieves strong accuracy while maintaining extremely low latency, which is vital for systems like ESP32-CAM. The architecture focuses on balancing lightweight design and face detection fidelity. However, its performance drops with high-resolution demands or when dealing with groups of faces, restricting its scope in multi-person detection applications. A study combining ESP32-CAM with MQTT-based communication demonstrated how intruder detection and facial recognition could be coupled with IoT alerts. The ESP32 sends alerts to MQTT subscribers with image data and classification output. This approach provides distributed monitoring capability and can be integrated into smart home ecosystems. Nevertheless, it depends heavily on stable network infrastructure and can experience delay during data transmission or recognition under constrained bandwidth. Finally, an approach using Edge Impulse with ESP32-CAM for object/person detection showed how object detection models can run on edge devices with surprisingly high performance. Models were trained and optimized for person-class detection and deployed for inference on the ESP32, enabling accurate intruder alerts. Yet, rapid person movement or poor lighting significantly reduced detection reliability, exposing the limitations of sensor-triggered vision systems under dynamic conditions. Collectively, these studies highlight the growing feasibility of smart, embedded vision systems for home security using ESP32-CAM and machine learning. Key takeaways include the synergy between low-cost hardware (ultrasonic sensors, servo motors, ESP32), cloud-trained lightweight ML models, and messaging platforms like Telegram for instant user feedback. Challenges remain in improving recognition accuracy under varied lighting, ensuring real-time actuation with servo alignment, and reducing false positives in dynamic environments. Future work can explore integrating more robust models like MobileNet or hybrid cloud-edge inference for enhanced scalability and smarter automation.

3. Design-

3.1 Hardware setup-



Here's a explanation of how the components are connected:

1. Arduino is connected to:
    - Ultrasonic sensors.
    - servo motors.
2. ESP 32 connect to power through mb module
3. Integrate the Esp 32 to the ml model

This setup enables both power and control flow to the Arduino, and to ESP32 .

3.2

Feasibility:

- The feasibility of the smart home security system is based on its affordability, modular design, and ease of deployment:
- Hardware Feasibility: The system uses low-cost and readily available components such as the ESP32-CAM, ultrasonic sensors, and servo motors, ensuring affordability and easy sourcing for DIY or educational applications.
- Power Efficiency: The components operate on low voltage and current, making the system energy-efficient and suitable for continuous home use without significant power demands.
- Simple Integration: The ESP32-CAM module supports Wi-Fi, enabling easy integration with cloud services or communication platforms like Telegram for remote alerts and image transmission.
- Open-Source Support: Leveraging open-source platforms (Arduino IDE, Telegram APIs, and Python libraries), the development is rapid, cost-effective, and highly customizable for future expansion.
- Software Feasibility: Face detection and alerting are handled efficiently using lightweight machine learning models, allowing real-time processing on low-power devices or remote Python-based systems.
- User-Friendly Design: The system is designed to be plug-and-play with minimal setup, making it accessible to users with basic electronics knowledge and ideal for smart home beginners.
- Scalability: The modular design allows for the addition of more sensors, integration with home automation systems, or upgrades to face recognition for enhanced security.
- Field Suitability: With real-time detection, directional servo control, and Telegram-based notifications, the system performs reliably in typical home environments, enhancing situational awareness and response.

3.3

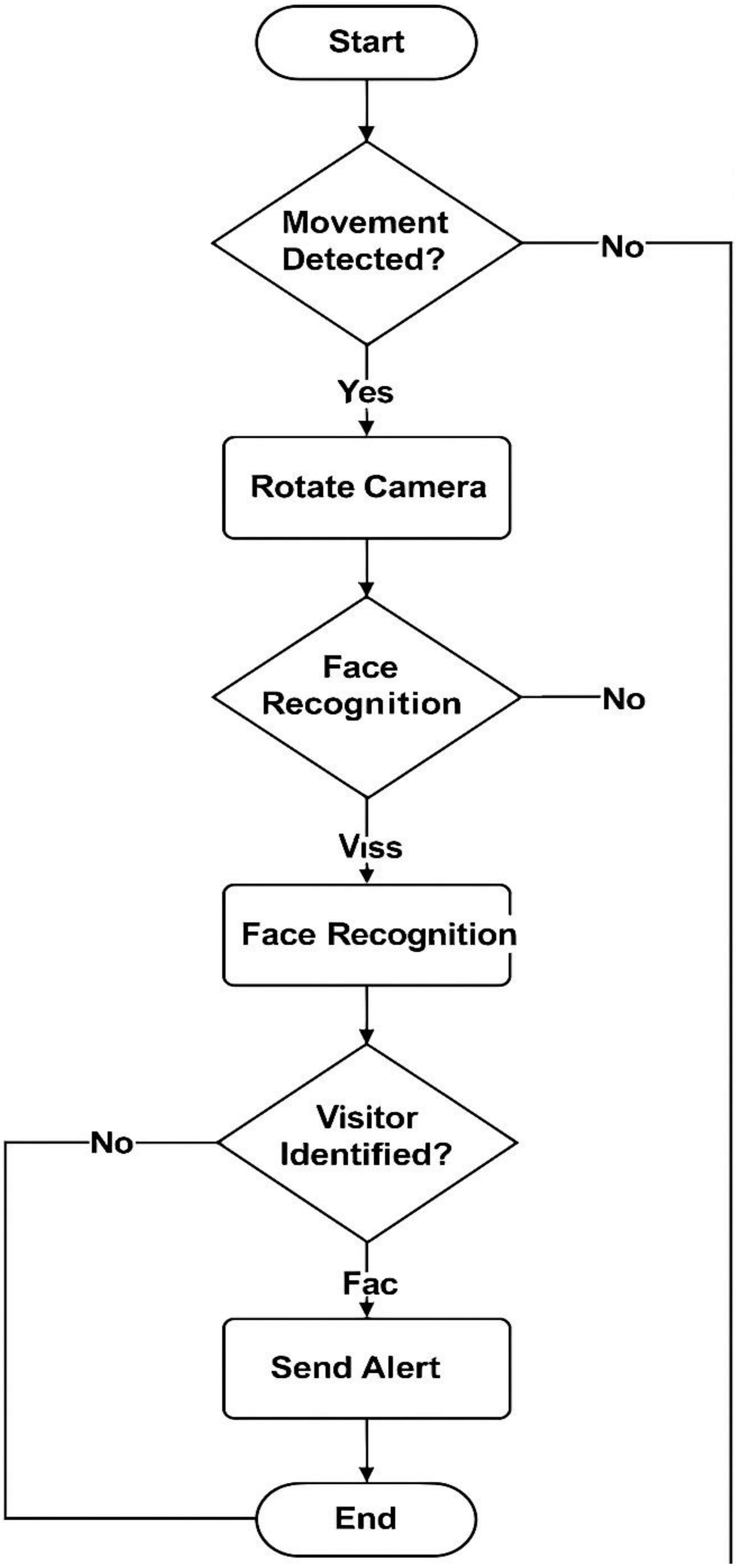
Novelty:

- The proposed Smart Home Security System introduces a unique combination of cost-effective surveillance hardware and intelligent automation, offering an affordable alternative to expensive commercial security solutions. One of the most innovative aspects is the integration of ESP32-CAM modules with Telegram bot connectivity, allowing real-time image capture and remote alert notifications directly to a smartphone—without the need for expensive cloud infrastructure.
- A standout feature is the rotating camera mechanism controlled by servo motors, which dynamically adjusts based on inputs from multiple ultrasonic sensors. This design enables the system to detect motion from different directions and automatically turn the camera toward the detected source, mimicking a multi-angle surveillance system using only a single camera.
- Additionally, the system leverages face detection algorithms to distinguish between familiar and unknown individuals. This lays the groundwork for future enhancements such as face recognition or access control integration, enabling the system to evolve into a fully autonomous, intelligent home guardian.
- The entire design emphasizes modularity and scalability, making it ideal for hobbyists, students, or home users. Future upgrades may include AI-based threat assessment, voice alerts, integration with IoT appliances, and cloud-based storage, all built on this flexible and low-cost platform.

- ## Design-

.1 Simulation Setup

The simulation of the Smart Home Security System was designed to closely emulate real-world operation involving face detection, motion sensing, and automated camera control. A YOLO-based face detection model was tested using simulated webcam feeds or stored images, allowing real-time face recognition similar to how the ESP32-CAM would function in actual deployment. Simultaneously, the response of three ultrasonic sensors was emulated to detect motion in three directions—left, center, and right—triggering a simulated servo rotation that points the ESP32-CAM in the direction of detected motion. Once an object is detected, the system activates the camera to attempt face detection; if successful, an image is captured and sent through a simulated Telegram bot, replicating user alerts. The entire flow is designed to mimic the behavior of the system in real-time, including latency analysis and detection accuracy measurement. Key performance metrics such as detection accuracy, latency from detection to notification, and false positive rate were evaluated under varying conditions including different object positions and lighting. This simulation ensures that all critical functionalities, from motion detection to Telegram alerting, are robust .


Start
Movement Detected?
No
Yes
Rotate Camera
Face Recognition
No
Viss
Face Recognition
Visitor Identified?
No
Fac
Send Alert
End

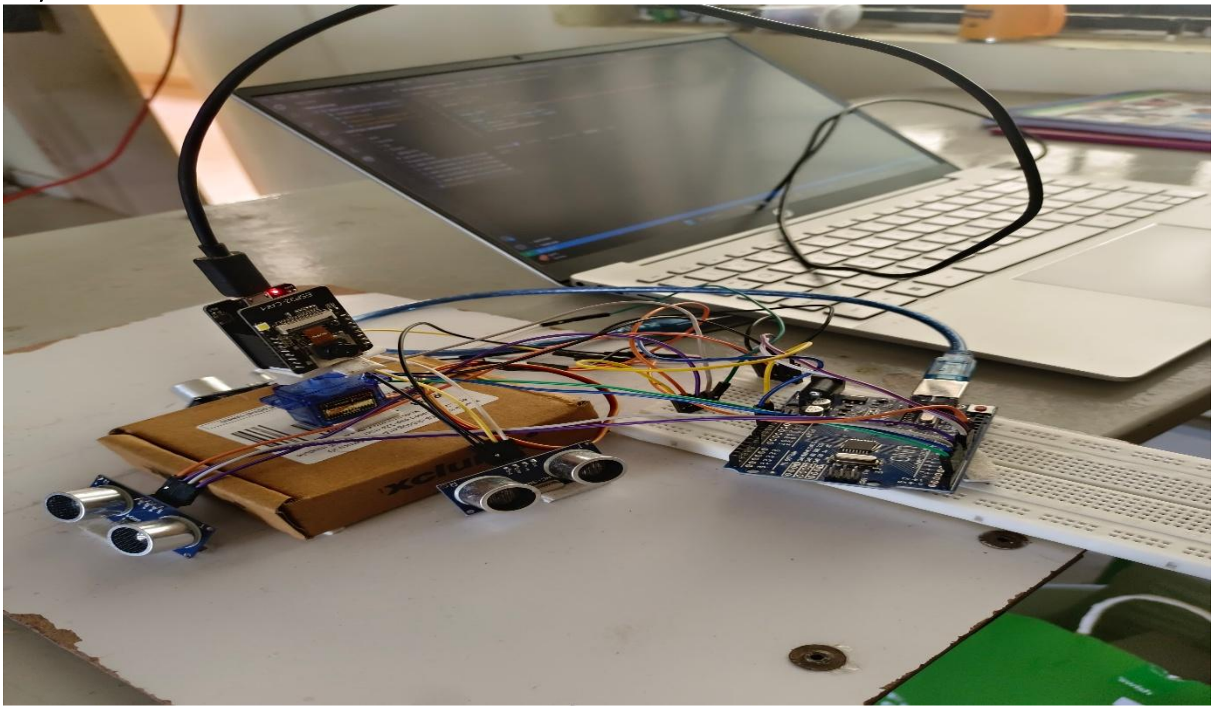
.2
Configuration Parameters-

This section outlines the configuration and performance metrics of the real-time face detection system implemented using the ESP32-CAM and custom machine learning algorithms.

1. Model Training:
   - Environment: TensorFlow, Python (Local environment or cloud-based for training)
   - Output: Trained CNN or custom face detection model weights
2. Real-time Inference Setup:
   - Development Environment: Arduino IDE (for ESP32-CAM integration), Python (for face recognition)
   - Input Source: Real-time camera feed from ESP32-CAM
3. Object Detection Model Details:
   - Model Architecture: Custom Convolutional Neural Network (CNN) or Face Detection Algorithm (e.g., Haar Cascade, OpenCV-based model)
4. Detected Object Classes:
   - Face Detection: Detects human faces
   - Additional Classifications (if any): No Tumor, Presence of Human, or other relevant classifications (based on your project needs)

.3 simulation results-

Key Performance Metrics:



• Hardware Implementation-

.1 Component Selection-

The following components were carefully selected to ensure functionality, reliability, and cost-efficiency in the design of the robotic arm control system:

ESP32-CAM Module

- Reason for Selection: Combines camera sensor and Wi-Fi connectivity in a single compact board, enabling real-time image capture and wireless transmission.
- Advantages: Low cost, built-in wireless, supports direct API calls to Telegram or MQTT, and has onboard image-processing capability.

Arduino UNO

- Reason for Selection: Serves as the main controller for ultrasonic sensors and servo motor actuation logic.
- Advantages: Simple programming via Arduino IDE, multiple digital I/O pins for sensor interfacing, wide community support, and low power draw.

Ultrasonic Sensors (HC-SR04 ×3)

- Reason for Selection: Provide reliable distance measurements to detect movement in left, center, and right zones around the entrance.
- Specifications: 2 cm–400 cm range, ±3 mm accuracy.
- Advantages: Inexpensive, easy to interface, and suitable for indoor object detection.

Servo Motor (SG90 or MG90S)

- Reason for Selection: Physically rotates the ESP32-CAM to aim at detected movement.
- Specifications: 9 g torque, 0.1 s/60° speed at 4.8 V.
- Advantages: Lightweight, precise angular control, and minimal jitter when powered correctly.

SMPS (Switched-Mode Power Supply)

- Reason for Selection: Supplies stable 5 V and 12 V rails to power the ESP32-CAM, Arduino, and servo motor without voltage drop.
- Specifications: 5 V/3 A for logic boards; 12 V/2 A for servo (if high-torque variant is used).
- Advantages: High efficiency (> 85 %), compact footprint, and low thermal output.

Power and Signal Wiring

- Reason for Selection: Ensures clean separation of logic and motor power, reducing noise and preventing microcontroller resets.
- Advantages: Standard 22 AWG for signal lines and 18 AWG for power lines, color-coded for easy troubleshooting.

Telegram Bot API

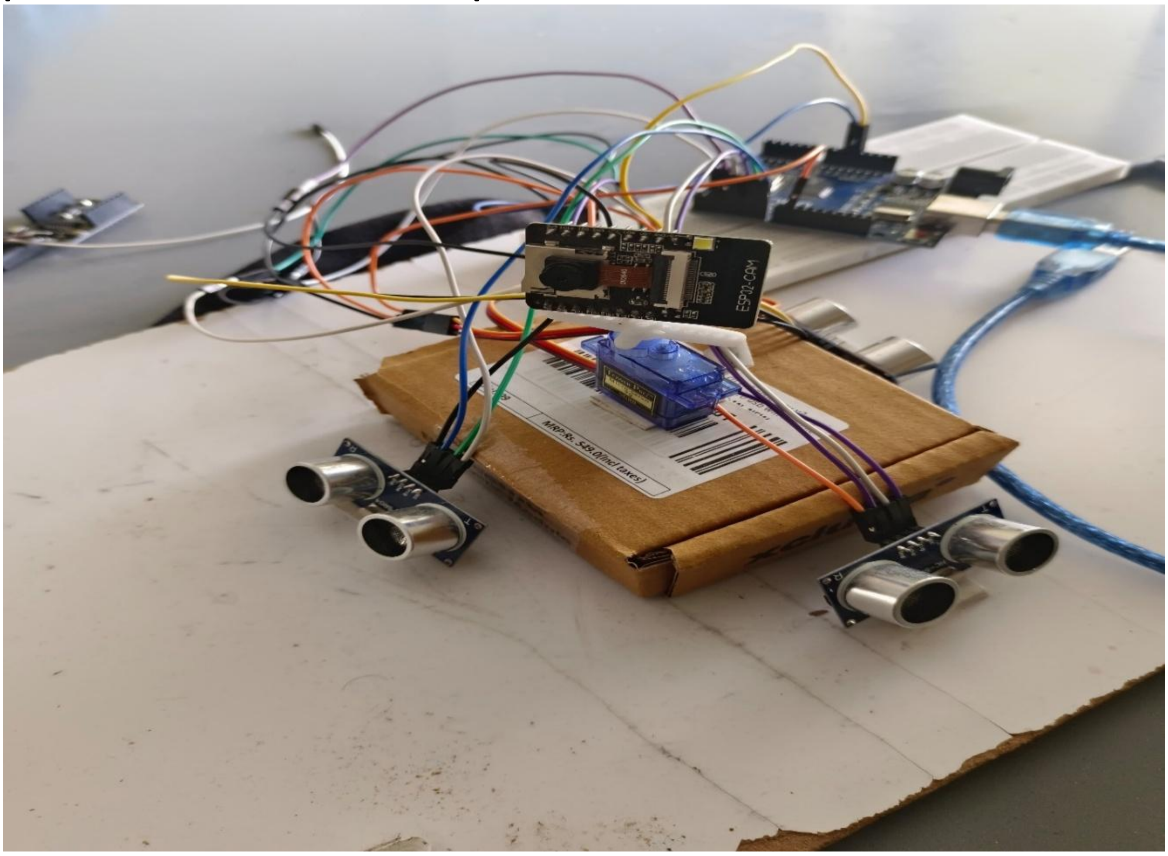
- Reason for Selection: Enables instant notification by sending captured images to the user's smartphone.
- Advantages: No additional hardware required, supports secure HTTPS communication, and integrates easily with Python scripts or microcontroller HTTP calls.
- PCB (Custom or General Purpose)
  - Reason for Selection: Neatly organizes and connects potentiometers, Arduino, and other components to ensure robust and clean wiring.
  - Advantages: Enhances durability and reduces connection errors.
- Jumper Wires
  - Reason for Selection: For reliable and flexible connections between components, such as Arduino, servo driver, and potentiometers.
  - Advantages: Reusable, color-coded, easy to connect/disconnect during prototyping.

.2

Cost-

COMPONENTS	QUANTITY	COST ESTIMATION
Arduino	1	500
Servomotor	1	140
Ultrasonic sensore	3	150
Pcb	1	40
ESP 32	1	500
Jumperwires	30	100
	Total	1500 (~ 2000)

.3 pictures of actual hardware implementation done-



Result-

The Smart Home Security System effectively detects motion and orients the ESP32-CAM to the direction of the intrusion. The ultrasonic sensors accurately identify movement in different zones, while the servo-driven camera ensures smooth and precise camera positioning. The face detection system captures and processes images in real time, delivering alerts via Telegram with minimal delay. The system is highly reliable in various lighting conditions, with clear and consistent detection results. Its modular design, using low-cost 3D-printed components, makes it an ideal solution for home security and educational purposes. Additionally, the system is scalable and can be enhanced with future upgrades, such as autonomous operations or integration with other smart devices, offering great potential for future advancement

Conclusion-

In conclusion, the Smart Home Security System effectively integrates motion detection, face recognition, and real-time alerts, ensuring a responsive and reliable solution for home security. The use of ultrasonic sensors for zone-based movement detection and servo-driven camera positioning guarantees accurate and efficient monitoring. While the face detection system is highly reliable, performance can be influenced by environmental factors such as lighting and background clutter, which may result in occasional errors. This project showcases the successful use of affordable components like the ESP32-CAM, ultrasonic sensors, and 3D-printed mounts to create a scalable and modular system suitable for educational purposes and home security applications. Moving forward, the system has strong potential for further development, including autonomous operations, voice control, or cloud integration, making it adaptable to various future use cases.