Dissertation on

"Efficient AI-Driven Edge Surveillance using Edge Computing"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE22CS320B – Capstone Project Phase - 2

Submitted by:

Govind Subramanian	**PES1UG22CS222**
Herman Singh Umrao	**PES1UG22AM067**
Akshaj B Seerpu	**PES1UG22AM018**
Uday V	**PES1UG22CS662**

Under the guidance of

Prof. Dinesh Singh
Associate Professor
PES University

January - May 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Efficient AI-Driven Edge Surveillance using Edge Computing'

is a bonafide work carried out by

Govind Subramanian	**PES1UG22CS222**
Herman Singh Umrao	**PES1UG22AM067**
Akshaj B Seerpu	**PES1UG22AM018**
Uday V	**PES1UG22CS662**

in partial fulfilment for the completion of sixth semester Capstone Project Phase - 2 (UE22CS320B) in the Program of Study - **Bachelor of Technology in Computer Science and Engineering** under rules and regulations of PES University, Bengaluru during the period Jan. 2025 – May. 2025. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 6th semester academic requirements in respect of project work.

Signature Prof. Dinesh Singh Associate Professor	Signature Dr. Mamatha H R Chairperson	Signature Dr. K S Sridhar Registrar

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **"Efficient AI-Driven Edge Surveillance using Edge Computing"** has been carried out by us under the guidance of **Prof. Dinesh Singh, Associate professor,** and submitted in partial fulfillment of the course requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester Jan – May 2025. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

Govind Subramanian	**PES1UG22CS222**
Herman Singh Umrao	**PES1UG22AM067**
Akshaj B Seerpu	**PES1UG22AM018**
Uday V	**PES1UG22CS662**

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Prof. Dinesh Singh, Department of Computer Science and Engineering, PES University, for his/her continuous guidance, assistance, and encouragement throughout the development of this UE22CS320B - Capstone Project Phase – 2.

I am grateful to the project coordinator, Dr Priyanka H, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Mamatha H R, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. K.S. Sridhar, Registrar, PES University, for his help.

I am deeply grateful to Prof. Jawahar Doreswamy, Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University, for providing me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

This project presents an efficient edge-AI based CCTV surveillance system designed for medium-footfall institutional facilities. The system leverages existing CCTV infrastructure combined with edge computing devices to perform real-time video analytics without requiring significant cloud resources.

Processing data at edge helps reduce latency for AI based detection, at the same time helping with security and also lowering required bandwidth. Our proposal will use very lightweight computer vision algorithms optimized for deployment on the edge to perform some of the monitoring tasks such as people detection, occupancy tracking, and anomaly detection.

Our proposed architecture allows facilities to upgrade their conventional CCTV systems with the capability for AI, while minimizing infrastructure cost while maintaining data privacy. Key features in the proposal include real time monitoring, local data processing, customizable alerts, and basic analytics reporting.

This system is designed for facilities that have 50 to 500 people coming daily, making it perfect for smaller and medium-sized institutions that want a less expensive, uncomplicated security system without the complexity of cloud-based or server based solutions.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Recent rapid development in artificial intelligence and edge computing systems have provided a gateway to upgrade modern surveillance systems. With such high-end AI-powered security solutions, there is still a large market gap for cost-effective, efficient monitoring solutions for medium-sized facilities.

Traditional CCTV systems, although ubiquitous, include much human monitoring with intelligence lacking in regard to automatically detecting potential threats or even analysis of the patterns. In contrast, high-tech cloud-based AI surveillance systems are very expensive and sometimes too complex for most smaller institutions and thus require infrastructure upgrades and bandwidth capabilities.

This project idea addresses the given challenges by developing an edge-AI based CCTV surveillance system specifically designed for facilities with moderate daily visitor traffic (50-500 people). Our solution balances functionality, cost-effectiveness, and privacy by upgrading the existing CCTV infrastructure with edge computing capabilities. Local video feed processing using lightweight AI algorithms will make it possible to monitor in real-time and provide simple analytics without requiring continuous connectivity to the cloud or the installation of expensive hardware upgrades.

Our solution centers on providing practical, applicable solutions that strengthen security operations but are mindful of resource constraints and institutional privacy concerns more typically associated with medium-sized institutions, such as small educational facilities, healthcare clinics, and office spaces. This system embodies a major step forward in democratizing the application of intelligent surveillance capabilities at the edge.

CHAPTER II

PROBLEM DEFINITION

Facilities with a medium footfall of 20-200 visitors per day suffer greatly to implement proper surveillance solutions that balance between the security needs and practical constraints. Traditional CCTV systems are very common, but suffer from a few key limitations:

- Resource Constraints:
  - Small to medium-sized businesses cannot afford costly server infrastructure or pricey, cloud-based analytics platforms
  - Monthly fees on cloud subscription and bandwidth requirements drive high-end surveillance features out of reach
  - Fewer IT personnel and scarce skillset to manage complex systems

- Operational Inefficiencies:
  - Manual monitoring of multiple video feeds tends to be labor-intensive and error-prone
  - Monitoring an individual movement requires reviewing multiple, distinct video footages coming from different cameras
  - Real-time threat detection is dependent on human vigilance and reaction
  - Does not have automated alerts and incident reporting features

- Technical Limitations:
    - Basic CCTV systems do not offer intelligent monitoring features with just a simple recording function
    - Cannot detect anomalies or patterns without automating the system
    - Constant surveillance cannot be maintained at peak hours
    - Restricted ability to derive insights from collected video

- Privacy and Compliance
    - Security mandates needs to be balanced against privacy regulation
    - Data needs to be stored and processed in accordance with local privacy laws
    - Risk of transmitting sensitive video to cloud servers

CHAPTER III

DATA

3.1 Overview

The data source comes from publicly recorded videos in which an individual person traverses the range of capture for a single camera. These videos are used as source material for training AI systems designed for real-time surveillance applications. The objective of the dataset is to offer a varied collection of frames with the subject moving, allowing for the creation of AI systems that can identify, track, and monitor the subject's presence in different environments. The dataset is particularly intended to aid AI-based edge surveillance systems that run effectively without the need for centralized cloud servers.

3.2 Dataset

The data is composed of frames removed from videos involving a single individual moving in the field of view of the camera. A script tailored to process the videos searches for full-screen frames where the individual is fully within view. The frames are stored in a folder for later use. The frames record different points in the motion of the subject, giving a wide variety of images that can be used for object detection, tracking, and classification. The dataset is also curated to capture only the individual, making it possible to use it to train models to detect, track, and analyze human subjects in video. The generated collection of frames is essential in training AI models that can run in real-time on edge devices with limited computation resources.

CHAPTER IV

DESIGN DETAILS

4.1 Novelty

This system presents a new combination of low-cost edge computing and AI-driven surveillance aimed at medium-footfall institutions which are usually ignored by conventional and cloud-based security solutions.

4.2 Innovativeness

The method exploits lightweight computer vision models such as YOLO and FaceNet, implemented within Raspberry Pi-based systems, providing real-time AI analytics without incurring costly infrastructure or constant internet connectivity.

4.3 Interoperability

Designed to retro-fit existing CCTV installations, the system facilitates support for multiple camera sources and transparent communication between devices in local networks through standardized protocols and modular design.

4.4 Performance

Edge device-based local video processing guarantees low-latency AI inference for real-time detection and alerting. Hardware-accelerated operations and frame-level optimization methods improve throughput on even limited devices.

4.5 Security

Through local processing of sensitive video data and minimization of cloud reliance, the system improves data protection and adheres to local privacy policies. Local, encrypted communication adds further protection to device interaction.

4.6 Reliability

Edge architecture provides reliable operation even in the event of internet unavailability. Local processing prevents single-point-of-failure conditions that are common in centralized systems, improving system dependability.

4.7 Legacy to Modernization

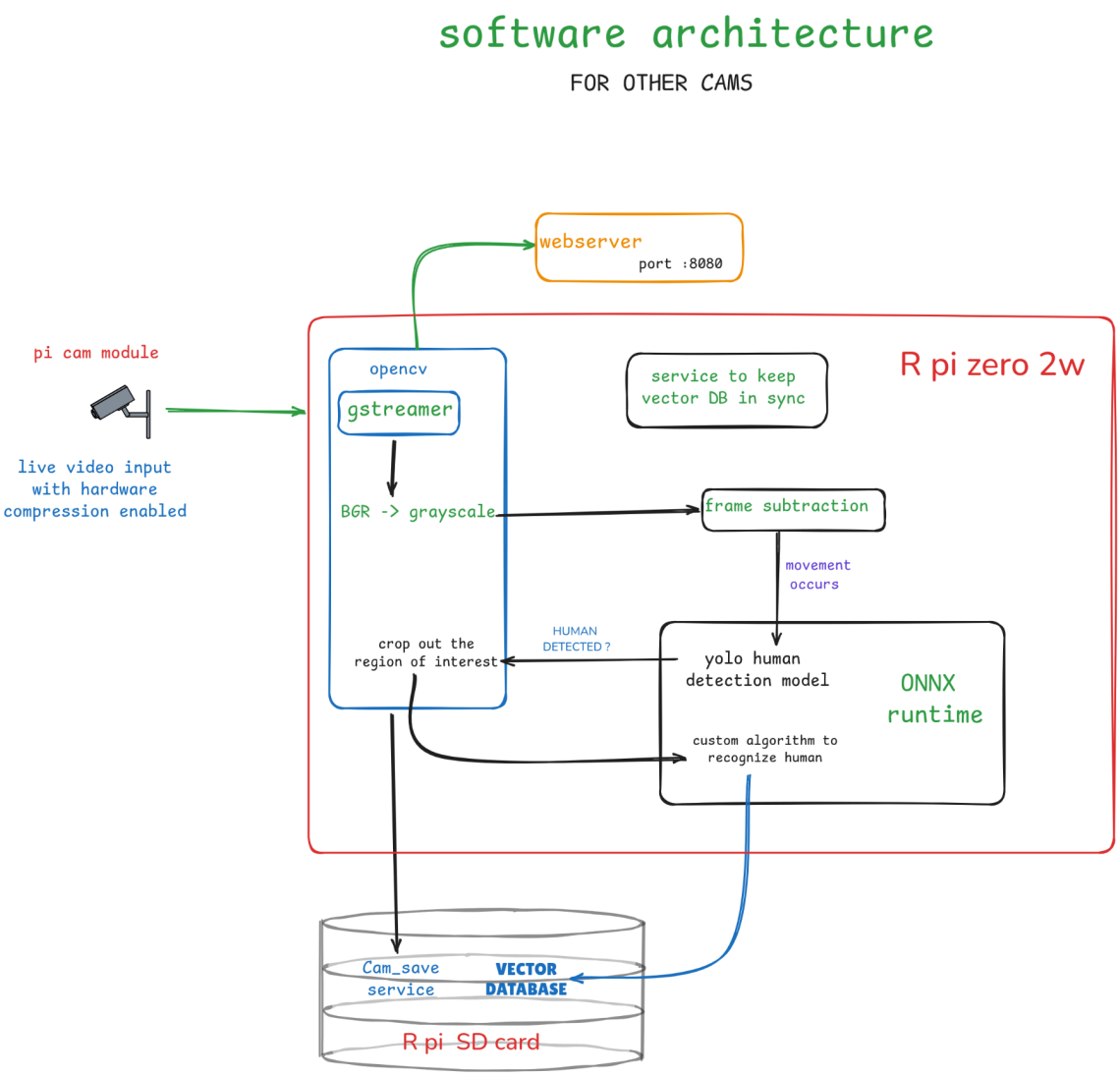
By using AI-based functionality in place of upgrade, the project fills the gap between legacy monitoring systems and cutting-edge smart surveillance, making it possible to make legacy equipment useful again.

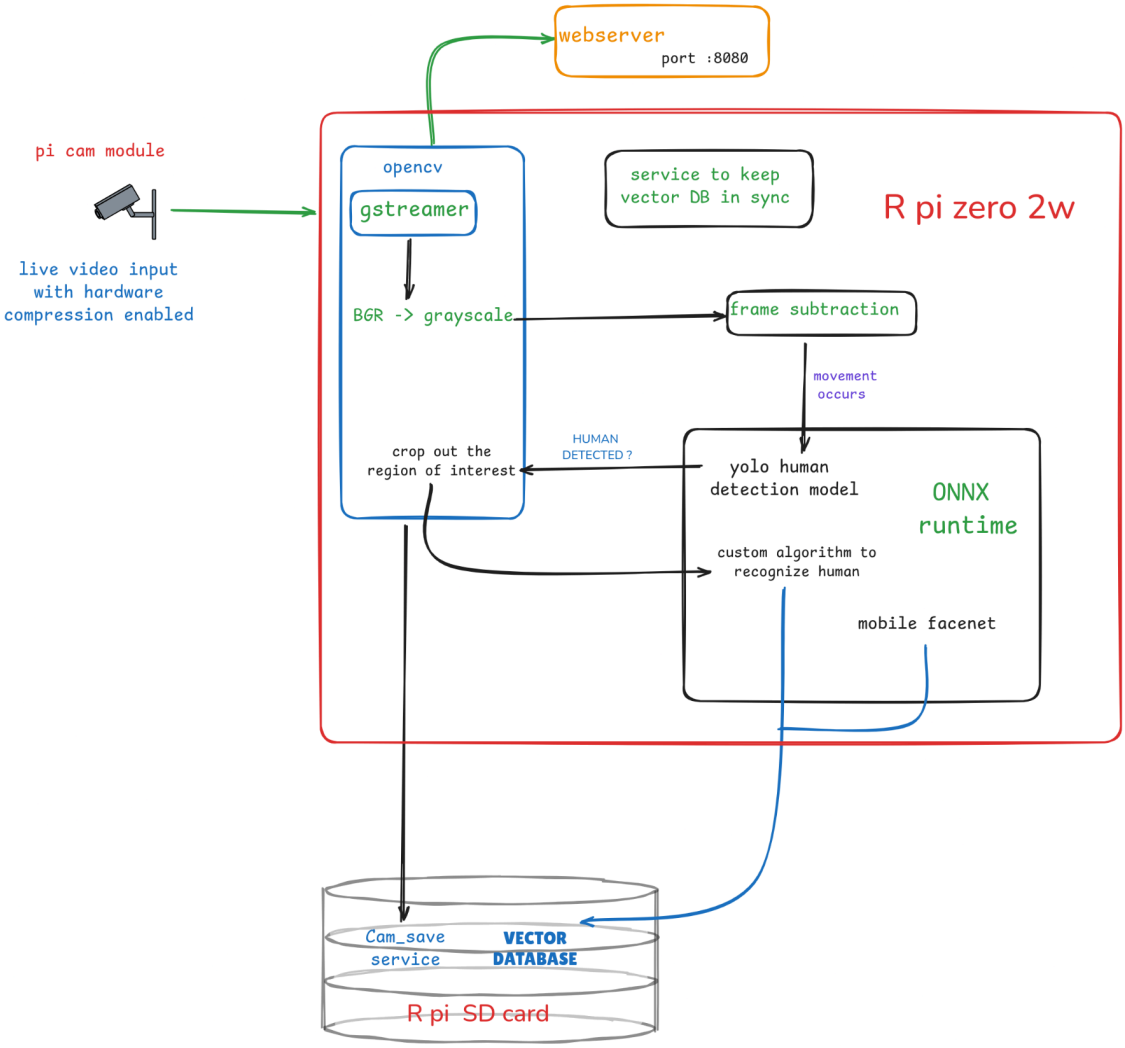
4.8 Reusability

The architecture has modular design and abstract AI components to enable reuse in other edge applications like smart retailing, access management, and crowd management systems with minor adjustments.

CHAPTER V

HIGH LEVEL SYSTEM DESIGN /SYSTEM ARCHITECTURE

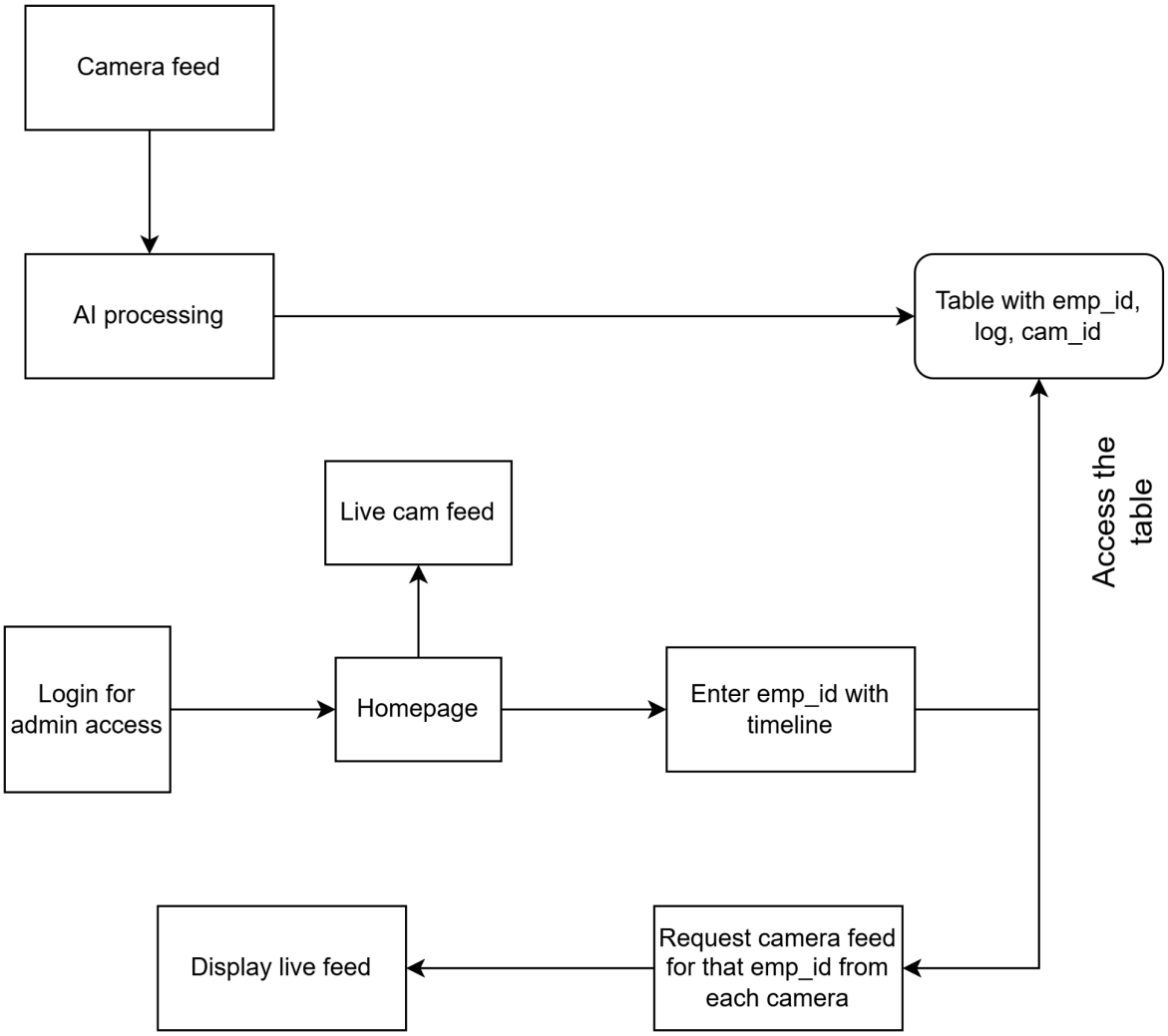



software architecture
FOR MAIN CAMS
webserver
port :8080
pi cam module
live video input
with hardware
compression enabled
opencv
gstreamer
BGR -> grayscale
service to keep
vector DB in sync
R pi zero 2w
frame subtraction
movement
occurs
HUMAN
DETECTED ?
crop out the
region of interest
yolo human
detection model
ONNX
runtime
custom algorithm to
recognize human
mobile facenet
Cam_save
service
VECTOR
DATABASE
R pi SD card

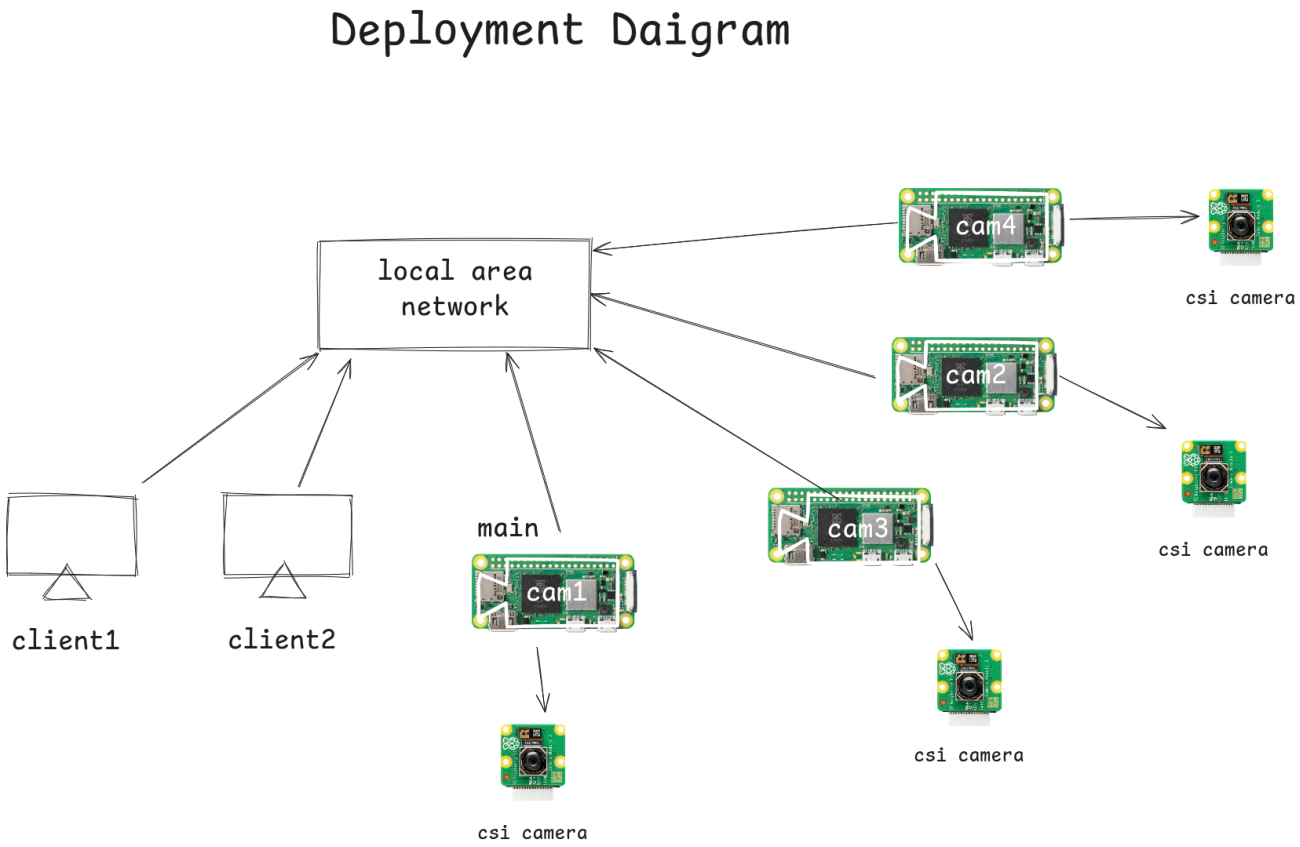
CHAPTER VI

DESIGN DESCRIPTION

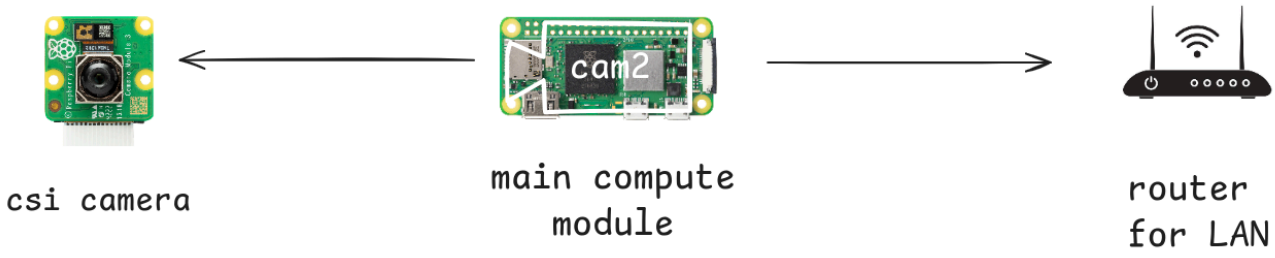
6.1 User Interface Diagram



6.2 Packaging and Deployment Diagram



6.3 External Interfaces Diagram



CHAPTER VII

TECHNOLOGIES USED

- yolo
- onnx
- shutil
- g++
- python3
- opencv
- cross-compilation
- r pi OS
- gstreamer
- cpp based webserver
- cmake
- makefile
- freeCAD
- cura
- ssh
- git

CHAPTER VIII

DATA PREPROCESSING AND IMPLEMENTATION

split_data.sh | convert.py 3

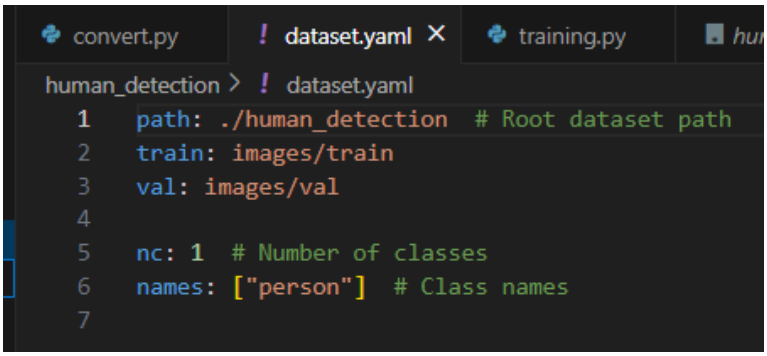
C: > Users > Govind Subramanian > Downloads > FLIC-full > FLIC-full > convert.py > ...

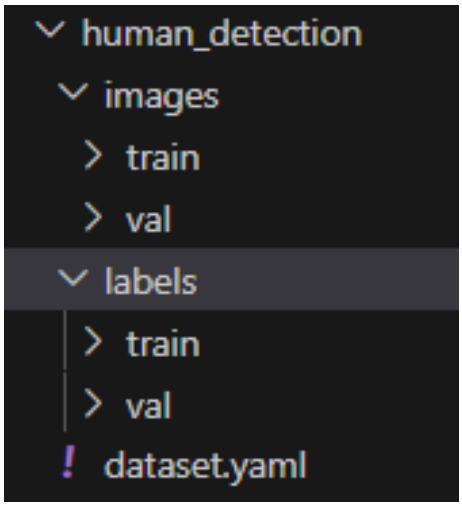
```python
from ultralytics import YOLO
import cv2
import os
import numpy as np

# Load YOLO model (pre-trained on COCO)
model = YOLO("yolov8n.pt")

# Define paths
image_dir = "C:/Users/Govind Subramanian/Downloads/FLIC-full/FLIC-full/images"  # Input images
label_dir = "C:/Users/Govind Subramanian/Downloads/FLIC-full/FLIC-full/labels"  # Output labels
os.makedirs(label_dir, exist_ok=True)

# Get image files
image_files = [f for f in os.listdir(image_dir) if f.endswith(('.jpg', '.png'))]

# Process each image
for image_file in image_files:
    image_path = os.path.join(image_dir, image_file)
    img = cv2.imread(image_path)
    h, w, _ = img.shape  # Image dimensions

    # Run YOLO detection
    results = model(image_path, imgsz=640, conf=0.5, max_det=50)

    # Prepare label file
    label_file = os.path.join(label_dir, os.path.splitext(image_file)[0] + ".txt")
    with open(label_file, "w") as f:
        for result in results:
            for box in result.boxes:
                cls = int(box.cls.item())  # Class index
                if cls == 0:  # Only keep 'person' detections
                    x_center, y_center, box_w, box_h = box.xywhn.tolist()[0]  # Normalized bbox
                    f.write(f"0 {x_center} {y_center} {box_w} {box_h}\n")

    print(f"Processed: {image_file}")

print("✅ All images processed! Labels saved in:", label_dir)
```

$ split_data.sh

C: > Users > Govind Subramanian > Downloads > FLIC-full > FLIC-full > $ split_data.sh

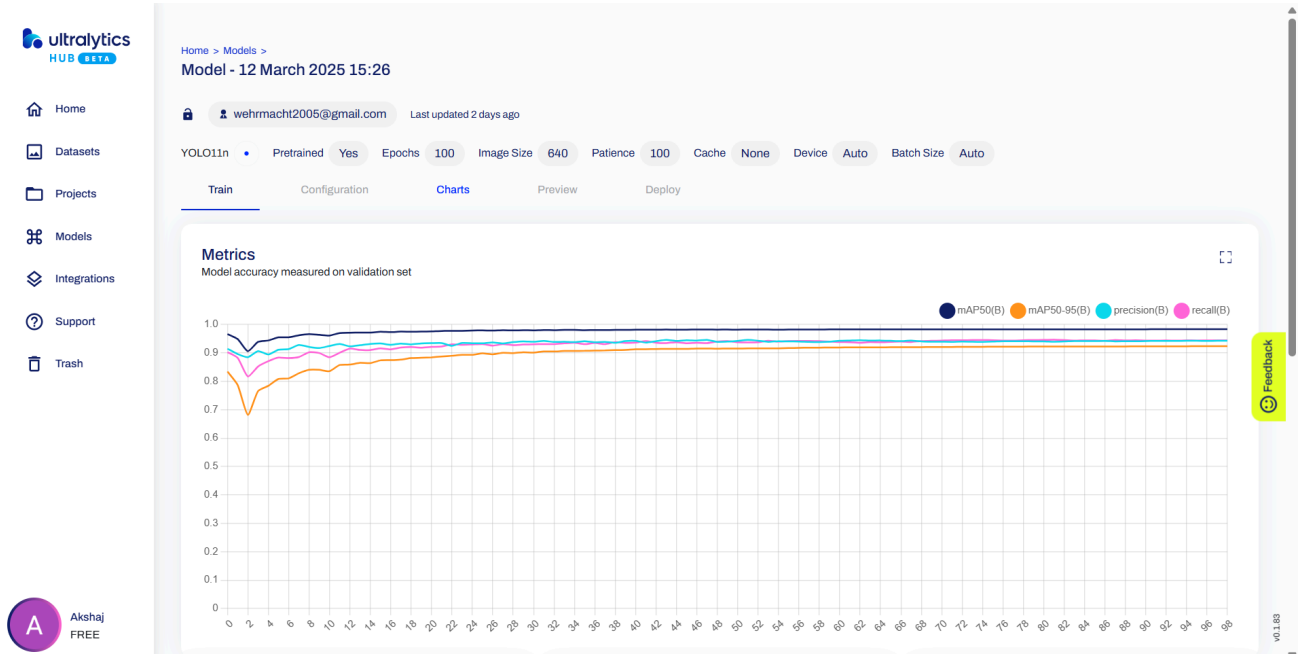
```bash
#!/bin/bash

# Set dataset directories
DATASET_DIR="$(pwd)"  # Get the current directory dynamically
IMAGES_DIR="$DATASET_DIR/images"
LABELS_DIR="$DATASET_DIR/labels"

# Create train/val directories
mkdir -p "$IMAGES_DIR/train" "$IMAGES_DIR/val"
mkdir -p "$LABELS_DIR/train" "$LABELS_DIR/val"

# Get a list of image files
mapfile -t IMAGES < <(find "$IMAGES_DIR" -maxdepth 1 -type f \( -name "*.jpg" -o -name "*.png" \))

TOTAL_IMAGES=${#IMAGES[@]}
TRAIN_COUNT=$((TOTAL_IMAGES * 80 / 100))

# Shuffle images
shuf -e "${IMAGES[@]}" > shuffled_images.txt

# Move files to train and val folders
COUNT=0
while IFS= read -r IMAGE; do
    FILENAME=$(basename "$IMAGE")
    LABEL_FILE="$LABELS_DIR/${FILENAME%.*}.txt"

    if [ "$COUNT" -lt "$TRAIN_COUNT" ]; then
        mv "$IMAGE" "$IMAGES_DIR/train/"
        [ -f "$LABEL_FILE" ] && mv "$LABEL_FILE" "$LABELS_DIR/train/"
    else
        mv "$IMAGE" "$IMAGES_DIR/val/"
        [ -f "$LABEL_FILE" ] && mv "$LABEL_FILE" "$LABELS_DIR/val/"
    fi
    COUNT=$((COUNT + 1))
done < shuffled_images.txt

# Cleanup
rm shuffled_images.txt

echo "✅ Dataset split completed! Train: $TRAIN_COUNT, Val: $((TOTAL_IMAGES - TRAIN_COUNT))"
```






```
(yolo_env) C:\Users\Govind Subramanian\Downloads\FLIC-full\FLIC-full>python convert.py
✅ All images processed! Labels saved in: C:/Users/Govind Subramanian/Downloads/FLIC-full/FLIC-full/human_detection/labels
```

Base model implementation:



CHAPTER IX

CONCLUSION OF CAPSTONE PROJECT PHASE - 2

During this stage of the project, we have been able to obtain and preprocess both of the necessary datasets that are critical for developing and training our AI models. The preprocessing process included cleaning, normalization, and formatting the data to be compatible with the light-weight computer vision algorithms we intend to use on edge devices. This step has provided a solid foundation for precise detection and analytics in subsequent stages.

We have finished one of the two fundamental AI model developments. That model is working on real-time human detection employing optimized YOLO-based architecture, designed for efficient execution in resource-limited devices like Raspberry Pi. That second model which will be implemented for facial identification and identity comparison employing FaceNet is still being developed and integrated in the follow-up phase.

In addition, we have also deployed a simple surveillance system based on Raspberry Pi boards. The system consists of a live video stream server from the camera, providing real-time access to video using a locally served web interface. The current setup also incorporates initial processing like frame differencing and motion detection, which will be used as an initiator for launching the AI detection pipeline.

This pilot work shows the possibility of installing AI-fortified surveillance features on low-cost, edge-based devices and paves the way for integrating more sophisticated analytics and logging capabilities in future stages.

CHAPTER X

PLAN OF WORK FOR CAPSTONE PROJECT PHASE - 3

In Phase 3 of our capstone, we will move closer to the ultimate integration and testing of our intelligent edge-based CCTV surveillance system. The main focus will be on finalizing the system architecture through the creation of the second AI model and the implementation of end-to-end connectivity between components.

1. Creation of the Second AI Model

- Start developing and optimizing the second mandatory AI model
- Train and fine-tune the model on the preprocessed dataset for precise identity recognition on real-time camera streams.
- Keep the model lightweight and efficient on Raspberry Pi-class hardware.

2. Testing and Validation Camera-wise

- Perform testing on single Raspberry Pi camera configurations.
- Test performance of the current human detection pipeline in real-time environments.
- Integrate the face recognition model with live feeds to support tracking of known identities.

3. Integration of Vector Database for Logging

- Use a vector database (such as FAISS or Milvus) to store facial embeddings and detection metadata.
- Allow rapid querying of identity data and log entries for analysis after the event.
- Apply logic to correlate people to camera locations and timestamps.

4. System Integration and Web Dashboard

- Start integrating all the system pieces into a cohesive solution.
- Improve the web interface to enable visualization of logged events and individual tracking.
- Provide synchronized functionality between video feed, detection events, and logged metadata.

5. Testing and Iterative Improvement

- Conduct thorough testing across various scenarios (e.g., changing lighting, crowd density).
- Tune detection thresholds, alert triggers, and system responsiveness.
- Collect performance benchmarks and debug bottlenecks prior to final deployment.

REFERENCES/BIBLIOGRAPHY

[1] Á. Carro-Lagoa, V. Barral, M. González-López, C. J. Escudero, and L. Castedo, "CITIC Research Center & Department of Computer Engineering, University of A Coruña, 15071, A Coruña, Spain," received Mar. 8, 2023, revised Jul. 26, 2023, accepted Sept. 8, 2023, available online Sept. 12, 2023, version of record Sept. 18, 2023.

[2] Y. Martínez-Díaz, H. Méndez-Vázquez, L. S. Luevano, L. Chang and M. GonzalezMendoza, "Lightweight Low-Resolution Face Recognition for Surveillance Applications," 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021

[3] P. Ramya, R. Rajeshwari, "A Modified frame difference method using correlation coefficient for background subtraction", 6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India.

[4] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, Charles Xie. 2021. Milvus: A PurposeBuilt Vector Data Management System. In Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), June 20–25, 2021, Virtual Event, China. ACM, New York, NY, USA, 14 pages. https: //doi.org/10.1145/3448016.3457550

[5] Chen, Sheng & Liu, Yang & Gao, Xiang & Han, Zhen. (2018). MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices. 10.48550/arXiv.1804.07573.

1. CCTV https://en.wikipedia.org/wiki/Closed-circuit_television
2. Ultralytics YOLO https://docs.ultralytics.com/#yolo-a-brief-history
3. Mobile Facenet https://github.com/pedroprates/mobile-face-net
4. Milvus https://milvus.io/
5. Raspberry Pi Zero https://www.raspberrypi.com/products/raspberry-pi-zero/
6. Ingenic https://en.ingenic.com.cn/products-detail/id-21.html

AKSHAJ_B-PW25_DS_01.pdf

2,027 Words

0% Matches