SMART PEOPLE COUNT SYSTEM BASED ON FACE RECOGNITION USING DEEP LEARNING APPROACH

A PROJECT REPORT

Submitted by

MAHESHSRIRAM T

19ITR045

NIRESH N

19ITR054

ARAVINTH S

19ITL109

in partial fulfilment of the requirements

for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMMUNICATION AND COMPUTER SCIENCES

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638060

DECEMBER 2022

DEPARTMENT OF INFORMATION TECHNOLOGY

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638 060

BONAFIDE CERTIFICATE

This is to certify that the Project Report entitled "**SMART PEOPLE COUNT SYSTEM BASED ON FACE RECOGNITION USING DEEP LEARNING APROACH"** is the bonafide record of project work done by **MAHESHSRIRAM T(19ITR045), NIRESH N (19ITR054)** and **ARAVINTH S (19ITL109)** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in **INFORMATION TECHNOLOGY** of Anna University Chennai during the year 2022-2023.

SUPERVISOR **HEAD OF THE DEPARTMENT**

(Signature with seal)

Date:

Submitted for the end semester viva voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF INFORMATION TECHNOLOGY

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638 060

DECLARATION

We affirm that the Project Report titled "**SMART PEOPLE COUNT SYSTEM BASED ON FACE RECOGNITION USING DEEP LEARNING APPROACH"** being submitted in partial fulfilment of the requirements for the award of Bachelor of Technologyis the original work carried out by us. It has not formed the 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.

MAHESHSRIRAM T
(19ITR045)

NIRESH N
(19ITR054)

ARAVINTH S
(19ITL109)

Date:

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Date: Name and Signature of the Supervisor with seal

ABSTRACT

The Systems that count people using vision have a wide range of possible uses, such as managing public resources and conducting video surveillance. The majority of works in the literature rely on the identification and tracking of moving objects while presuming that all moving objects are people. Algorithmic and computational challenges must be overcome in order to recognize many faces simultaneously. The YOLOv5 method is used in this research of image processing to identify objects in an more efficient and accurate manner. A unique small-sized smart-counting system comprised of less expensive hardware and a cloud-based object-counting software server to execute an accurate counting function is introduced. The cloud-based people-counting program adapts a model to the goal of counting authorized and unauthorized people using the deep learning-based YOLOv5 algorithm. These findings show that the suggested smart people counting technique correctly identifies the people. The findings of the suggested technique show that it effectively matches the people image using deep learning's YOLOv5 algorithm to improve security, achieving an improved accuracy of 98.53% with cloud-based storage that allows users to quickly check the smart-counting system's results.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

First and foremost we acknowledge the abundant grace and presence of Almighty throughout different phases of the project and its successful completion.

We wish to express our extreme gratefulness to our beloved Correspondent **Thiru.A.K.ILANGO B.Com.,M.B.A.,LLB.,** and all the trust members of Kongu Vellalar Institute of Technology Trust for providing all the necessary facilities to complete the project successfully.

We would like to convey our sincere gratitude to our beloved Principal **Dr. V. Balusamy B.E. (Hons.), M. TECH., Ph.D.,** for giving us the chance to complete the project.

We express our gratitude to **Dr.R.THANGARAJAN M.E., Ph.D.,** Head of the Department, Department of Information Technology, for his valuable suggestions.

We are thankful to our project **Dr. R. SHANTHAKUMARI ME, Ph.D.,** Associate Professor and **Ms.V.P.GAYATHRI ME.,** Assistant Professor, Department of InformationTechnology for their valuable supervision, which have been very helpful to complete the project.

We are highly indebted to **Dr. T. ABIRAMI M.E., Ph.D.,** Associate Professor, Department of Information Technology, for her valuable supervision, which have been very helpful to complete the project.

We are appreciative to the department of **Information technology** faculty members for their invaluable advice and assistance.

CHAPTER 1

INTRODUCTION

1.1 SMART PEOPLE COUNT SYSTEM

The necessity of safe guards in various colleges and hostels has been highlighted for the security purpose to prevent the unauthorized person. Owing to the fact that most of the safe guards cannot be able to check each and every person manually. Popular cameras often use sensors. The majority of sensor-based cameras are created for a specifically in structured-environment in order to achieve the high-accuracy because sensors are sensitive to environmental factors. As a result, these techniques are not applicable everywhere and cannot be compared to other techniques.

Figure 1.1.1 Smart People Count System

Image sensor safety measures have been created recently as a result of improvements in image processing methods. The items were initially numbered by removing their edges. The requirement that the objects should be easily distinguishable from the background was crucial in this situation.

Recent deep learning-based counters, advantage is to detect items reliably in any environment as long as they have had enough training data. These object- detection models do, however, still have limitations in that they do not function well enough whenthe object density in the image is high. To suggest a novel approach to detect objects initially andthen recognize faces using the YOLOv5 method, which is based on deep learning, to get around this constraint.

To enable the entire process of identifying the objects and to run a deep-learning-based detector on a low-cost embedded platform, the matching hardware device is just as crucial as the software. The following is a summary of significant contributions:

1. By using the YOLO (You Only Look Once) algorithm, which is based on deep learning for counting objects.
2. By creating a pipeline of all the operations to the hardware system that can work in tandem with the cloud-based deep learning software server, to show the potential of the smart counting system as a general-purpose counting device and to determine whether an person is permitted or unauthorized.
3. To regardless of hardware device specs, the cloud-based deep learning software server displays consistent performance. It follows that the more customers who uses the server, the more users can accommodate at low cost.

1.2 DEEP LEARNING METHODS

Deep Learning based methods for classifying and segmenting images currently delivers state-of-the-art performance. Many DL-based CNN models, such as AlexNet, VGGNet, GoogLeNet, Residual Net, and DenseNet, have been created and evaluated to perform better at classification for visual recognition tasks. In that deep learning different algorithms, Yolov5 method to count the permit and non-permit persons.

1.2.1 YOLO Algorithm

The acronym YOLO stands for "You Only Look Once." This majority of primary and detects various elements in a picture (in real-time). The item recognition in YOLO, which is conducted out as a prediction model, provides the probability for each class of the found photos. YOLO is a technique that employs neural network models to detect objects in real time. This algorithm's prominence emanates from its accuracy and speed. It has been used to recognize living creatures, living beings, ticket machines, and traffic signals, among many other things.

Figure 1.2.1 Yolo Grid Producing Diagram

1.2.2 YOLOv5 Algorithm

Each of the four YOLOv5 models—s, m, l, and x—offers a distinct level of detection accuracy and performance, as demonstrated below. YOLOv5's mAP (accuracy) is 55.6 with 17GFlops (computational power). YOLOv3-416, in contrast, had a mAP of 55.3 for 65.86 GFlops. YOLOv5 and other object detectors are trained to find objects. To train the model and teach it how to recognize things, a large number of photos and their corresponding annotations are used. This train produces a model file as its output. The MS COCO dataset was used to train the collection of models for YOLOv5.

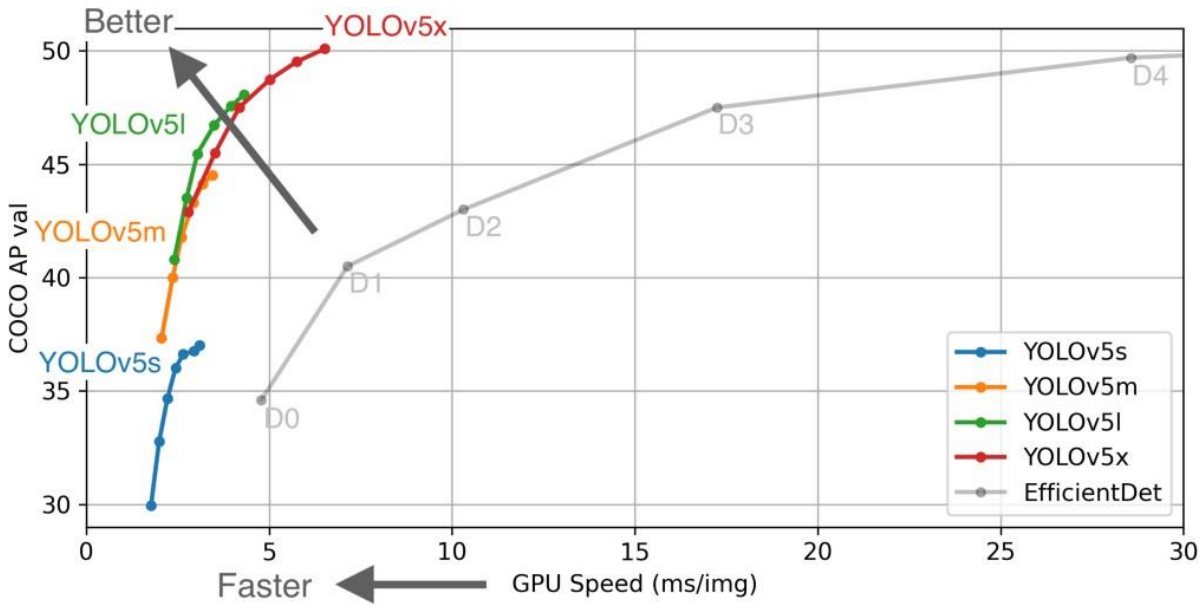

Figure 1.2.2 YOLOv5 Chart Diagram

CHAPTER 2

LITERATURE REVIEW

In this section, we are going to see about the related works regarding the smart counting system. According to **[1],** Colleges have historically been very concerned about student attendance, and manually recording attendance requires a lot of work from instructors. Because the current smart biometric counting system is not automatic, it costs time, is difficult to maintain, and necessitates a line of people waiting to have their fingerprints scanned to mark their people attendance. Everyone in the modern culture, the smartphone and is constantly online. In this study, attendance will be tracked via smartphones, which practically all faculty members have access to. Back propagation neural networks, quicker region-based convolutional networks, and single shot detector are a few examples of well-liked object identification techniques. In unified attendance structure is built on the face-based YOLO V3 algorithm.

To utilize the YOLO V3 algorithm, which was originally developed to detect objects—roughly 9000 objects—but in study, tweaked it to detect faces. The YOLO V3 system countsas well as notifies us of the number of the pupils presence in a class. Following that, it becomes required to recognize faces, which is accomplished using the Azure face API, which uses facial- features to detect and be confidently recognize a face for identification. Following a match betweenall student data in the face database and the creation of a spreadsheet for the particular time and date, each student's attendance will be recorded. By following this procedure, a report and system are generated at the conclusion of each month.

According to **[2],** Infrared or pressure sensors are typically the foundation of traditional counting systems. Despite being inexpensive, they are difficult to connect with video surveillance systems. Vision-based people of the counting systems have been grown in popularity in recent years for a variety of settings, including hot spots, roadways, and buildings. Two primary methodologies were used by

authors to develop approaches in the literature. Mehta et colleagues used classifiers to make it easier to locate and count the objects in the scene.

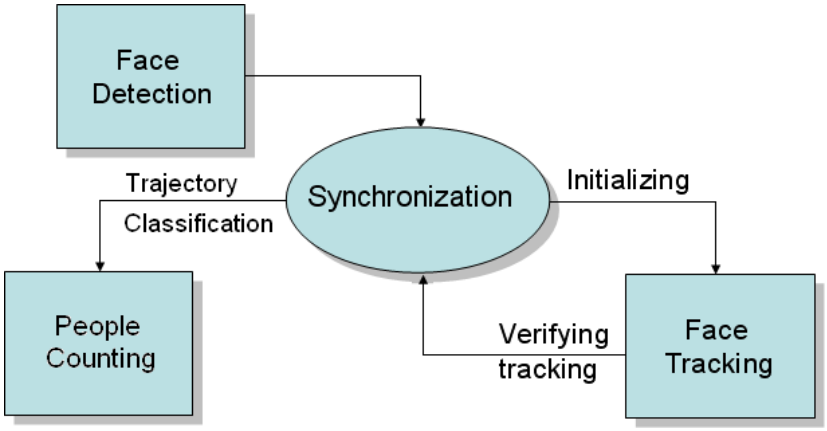

Figure 2.1. System framework

This study addresses the problem of the counting individuals going toward the best camera in a confined area where the lighting conditions are reasonably steady and people are typically facing the camera, such as the entry to a supermarket, bank, or bus. Based on these scenes, to suggest a method that offers a number of enhancements over the existing literature. The use of the face detector to confirm that the items being tallied are individuals is the first advancement. Second, a scale-invariant Kalman filter is suggested to deal with the abrupt changes in face scales in the scene.

[3] Automation refers to the machine's replacement of people. Without human intervention, the automation system will operate automatically. There is an urgent need to save energy in every aspect due to the rise in population and energy usage. In both business and non-commercial contexts, people counting and observation are of interest. To control any or all of the

electrical equipment in the office thanks to the proposed solution. This project explains how to implement and control the office equipment effectively. In addition, the system will count the number of employees present and track the number of visitors to the office. As a result, it regulates the electrical appliances to reduce electricity use, creating a no polluting and good atmosphere.

Most individuals spend their time in workplaces these days. Because the working environment has a direct impact on how effectively individuals perform their jobs, the office setting should be enjoyable for all parties involved. Without human intervention, the automation system will operate automatically. There is an urgent need to save energy in every aspect due to the rise in population and energy usage. The project switches from a sensor-based technology to a vision-based one. The suggested system automates utilizing face recognition. The light and fan will automatically turn on whenever a face is identified and turn off when no faces are found. To control any or all of the electrical equipment in office thanks to the proposed solution. This undertaking

According to **[4],** Mass production has brought attention to how important counters are across the board in the manufacturing sector [1-3]. It was important to build a different counter for each product while dealing with multiple sorts of products because the majority of the prior counters used for production often count only a single type of product. As a result, numerous experiments on all- purpose counters have been carried out over time. Popular counters [4] use sensors. Since sensors are susceptible to environmental factors, the majority of sensor-based counters are created for environments that are precisely constructed in order to attain high accuracy. As a result, these techniques are not applicable everywhere and cannot be compared to other techniques. Counters with image sensors have been developed recently as a result of improvements in image-processingmethods.

[5] Due to lower installation and video storage costs, closed-circuit television (CCTV) is now used more frequently to secure buildings. Selective access to places is a key worry for many organizations because to the excess of crime and terrorism. Conventional techniques, such as a password and smart card, are unreliable and insecure. Comparatively, facial recognition is a

trustworthy and clever biometric identification technique. The scientific community has struggled with the difficult task of automatic facial recognition. Applications including biometrics, surveillance, security, identification, and authentication have all widely used it. Face recognition typically makes use of high-dimensional data, which increases its computational complexity. Additionally, incorrectly recognized features can slow down the recognition process even more. Consequently, the interest in powerful facial recognition methods to ascertain whether two facial images belong to same person is increasing rapidly.

[6] Using Bluetooth signals from their phones, Bluetooth-based systems collect information about the students in the classroom and record their attendance. Because nearly 95% of college students have their phones with them, this system appears to be very practical. To complete the system, proxy removal methods can be implemented. The system's main flaw, however, is its lack of usability. ABluetooth-enabled device can only connect to up to eight other devices at once. This is because theMaster and Slave concept restricts a device's connection to only eight other devices at once. As a result, when the number of students in a classroom is in the single digits, this system cannot be used [18]. It was discovered after further consideration. Face detection and tracking were implemented using a Convolutional Neural Network (CNN). A model of an automated attendance system was proposed by authors Akbar et al. [7]. Their system detects and counts students as they enter and exit the classroom using a combination of Radio-Frequency Identification (RFID) and FaceRecognition. It keeps track of each student's attendance records and provides pertinent informationas needed. Their strategy was to collect data in real time.

According to **[7],** Smart pixels are often used to describe pixel sensors that carry out considerable amounts of data processing, such as feature extraction or intricate mathematical processes. Smart pixels have the ability to deliver a high level of fine-grained parallelism, enabling the focal plane array's (FPA) smart pixels to process multiple pieces of data simultaneously. Fine-grained parallelism can speedup processing, reduce latency, use less memory to store temporary results, and increase throughput. Additionally, the smart pixel can reduce die area and power consumption when it functions in the analog domain.

In findings demonstrate that analogue pixel-level processing with a slight fill factor penalty can produce outcomes that are on par with a completely digital version of the method. Analogue and digital processing stages make up the two stages of the heterogeneous SIS architecture. The analogue stage is a bidimensional smart-pixel array that may be set up to either extract image features during integration time or capture a picture as a traditional CIS. The digital step comprises of typical readout circuitry and a digital coprocessor that uses the features that were retrieved to perform facial recognition. After parasitic extraction of the complete array the simulated the analog smart-pixel array to validate the overall SIS architecture.

[8] Based on these scenes, suggest a method that differs from the literature in a number of ways. Utilizing a face detector to confirm that the objects being counted are individuals is the first improvement. Second, a scale-invariant Kalman filter is suggested to handle the significant variations in face scales in the scene. To manage facial occlusions, it is then integrated with a kernel-based object tracking system. Last but not least, it provides a method for counting individuals by automatically categorizing face trajectories, which are represented by an angle histogram.

The pros are in dual. On one of the hand, the problems are to be filtered to from consideration. improving counting accuracy by reducing false trajectories brought on by erroneous face detection in the system architecture. In contrast, automatic trajectory categorization dispenses with the need for manual and empirically developed methods for calculating the number of persons in each picture.

[9] To found that there are many time-consuming and unsuccessful alternatives, such as biometric, RFID-based and others. Hence, from each perspective of time, the superior and more trustworthy arrangement is the one that will transcend the earlier framework. To design an image processing algorithm for detecting faces of the students in the classroom as a result of developing a strong and effective participation structure. The technology can be improved in the future to be more accurate and able to distinguish faces in all lighting conditions. The system can be enhanced to distinguish between identical twins by identifying students through their eyes.

[10] The suggested approach outperforms the current methods with a success rate of 99% for face identification and 93% to 95% for face recognition. One of the most crucial things in any institution is the attendance record of the students. The face of a person can be recognized by computers froma digital photograph or video. By contrasting the real-time image with the database image, it can be done. It is necessary to compare the face characteristics extracted from the real-time images within the facialist characters of the database image saved. The faculties have the ability to lessen the load of taking attendance thanks to the automated attendance management system. Face recognition is used by this method to automatically take attendance. However, it could be challenging.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 PROJECT DESCRIPTION

The Smart Count Attendance System has a wide scope to automate the complete security and surveillance system. Here, using building an IoT-based Surveillance System using the ESP32 Camera are connected with the main admin device. It receives an input from the camera and communicates with the cloud-based object detecting server to exchange the data. And develop a cloud based low-cost hardware device for object counting and detection using deep learning algorithm to detect the authorized and unauthorized persons viewed through web application.

3.2 SYSTEM REQUIREMENT ANALYSIS

3.2.1 HARDWARE REQUIREMENTS

Processor : Intel I3 or above.

RAM : 8GB

Components : ESP32, High Definition Camera, UART Cable.

3.2.2 SOFTWARE REQUIREMENTS

Operating System : Android 4.2 or above, IOS 9 or above,

Windows 7 or higher.

AWS Services : AWS Recognition, Dynamo DB.

Languages : Python 3.7.

3.3 SOFTWARE DESCRIPTION

3.3.1 Python

Python is dynamically semantic, interpreted, and object-oriented. Its built-in highlevel data buildings, shared with dynamic typing, build it excellent for Faster Development and as a scripting or glue language for integrating existing systems. Python's simple syntax encourages readability, which reduces software maintenance costs. Python's support for modules and packages allows programmers to be more flexible and reuse code. The Python interpreter and its extensive standard library are available for free download and distribution in binary or source form for all major platforms.

3.3.2 AWS S3 Bucket

Amazon S3 is the object-storage service that provides business scalability, availability of information, protection, and effectiveness. Any amount of information isstored and kept safe by clients and sectors for virtually any use case, including data storage, cloud
- based services, and mobile apps. You can maximize costs, organize the data, and customize perfectly alright user access to meet the necessary business, organizational, and compliance issues with cost-effective storage classifications and simple device management.

3.3.3 Dynamo DB

The Amazon DynamoDB is a services that was managed by the Amazon. It is not necessary to hires professionals to manage the NoSQL installation Developers do not have to worry about settings up and configuring in distributed da Amazon DynamoDB offers high throughput with very low latency. Due to distributed natures of DynamoDB's data placements of requesting routings the algorithm, latencies stay constant as datasets grow.

It handles all of the complexities in the scaling, partitioning, and the re-partitioning of the data across more resources to meet the I/O performance in requirements. Amazon DynamoDB is built-in scalability. There is no need to be the concerned about predefined data-storage for the limitsof each table. Data of any size can be kept and retrieved. As the table expands, DynamoDB will expand automatically as more data is stored.

3.4 HARDWARE DESCRIPTION

To design and develop a cloud based low-cost hardware device for object counting and detection using deep learning algorithm to predict the authorized and unauthorized persons using YOLOv5 Algorithm by using ESP32 Camera with UART Cable to implement the automated counting system using AWS Cloud and proposes the web-based Application for Automated Counting Process. The cloud-based object detecting server connects to the ESP32 over a socket-based communication for quick transmission of data and receipt. This allows the server to quickly receive images from the camera and exchange information with the ESP32.

3.4.1 BLOCK DIAGRAM

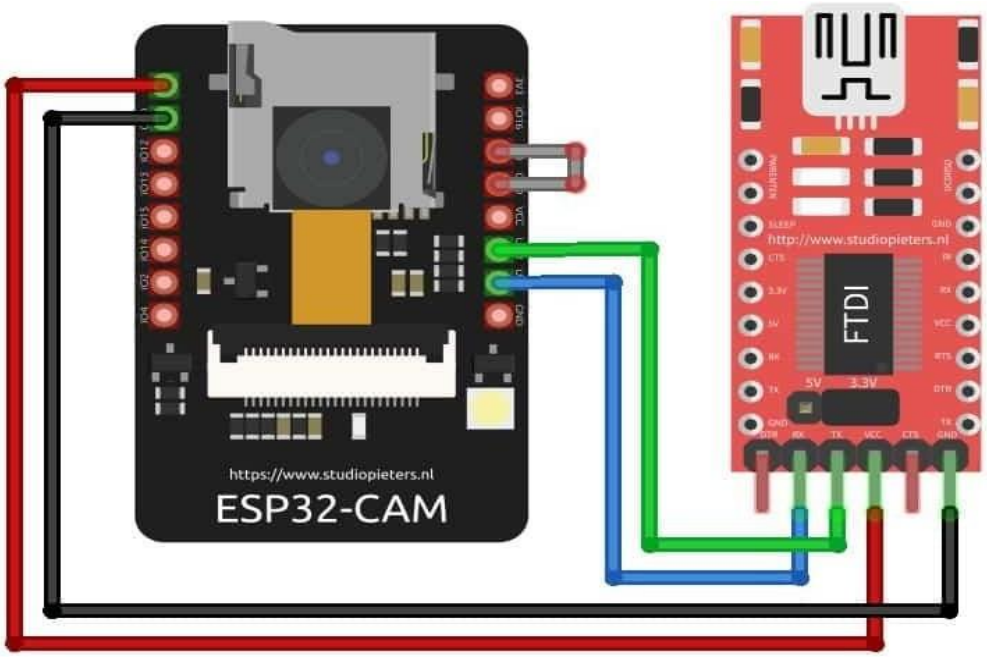

Figure 3.4.1 Existing System Block Diagram

3.4.2 ESP 32 CAMERA MODULE

A compact, inexpensive development board built on the ESP32 platform is called ESP32-CAM. It is the best option for the Internet of Things based applications, building the prototypes and the make-it-yourself projects conditions. The board includes two powerful 32-bit LX6 CPUs, WIFI, classic Bluetooth, and low power BLE. The main frequency system adjustment runs from the range of 80MHz to 240MHz, and it uses the 7-stages of pipeline architecture system, and the on-chip sensor, the hall sensor, the temperature sensor and different sensors. It is fully compliant with Bluetooth 4.2 and WIFI address 802.11b/g/n/e/i standards and can be utilized either in master mode to create the standard network controller system or in slave mode to add the networking functionality to other host the MCUs. The ESP32-CAM is applicable in numerous IoT applications. It is appropriate for wireless industrial controls, the smart home devices, and the wireless system.

Figure 3.4.2 ESP32 Camera diagram

3.4.3 UART

A UART is a microchip that converts data from the CPU to peripheral devices for serial-to-parallel conversion and from parallel to serial for transmission to peripheral devices. In order to reduce the amount of software required to handle the communication link between a computer and a peripheral device, the UART chip contains control capabilities and the ability to send an interrupt request to the CPU. The asynchronous serial connection between a computer and a peripheral device attached to the computer's serial port is handled by the UART controller, which also translates data from serial to parallel and vice versa. By doing so, the computer may communicate with modems and other serial devices. A UART is essentially a microchip that processes data entering and leaving serial ports, such the RS232 serial port interface on a computer. UART hardware was developed to support this kind of communication. However, these obsolete cables and connectors that use UART communication have almost all been superseded by USB.

Figure 3.4.3 Module of UART

CHAPTER-4

PROPOSED METHOD

4.1 OVERVIEW OF PROPOSED SYSTEM

Most of the security guards cannot be able to identify the authorized and unauthorized persons. Theproposed solution is to monitor each person through face recognition by cloud based hardware withdeep learning algorithms to detect the authorized and unauthorized persons viewed through web app. Detecting objects in densely packed settings is challenging for object-detecting techniques to switch to an object-detecting strategy that work ed well for counting tasks. For quick data transmission and reception, the cloud-biased object detection server connects to the ESP32 througha socket connection. The main computer device is coupled to an ESP32 camera. It receives an pictures from the cameras and communicates with a cloud-oriented object detection server to exchange data. a web-based user interface (UI) that makes it simple for users to view the results ofthe smart counting system's counting.

4.2 PROPOSED BLOCK DIAGRAM

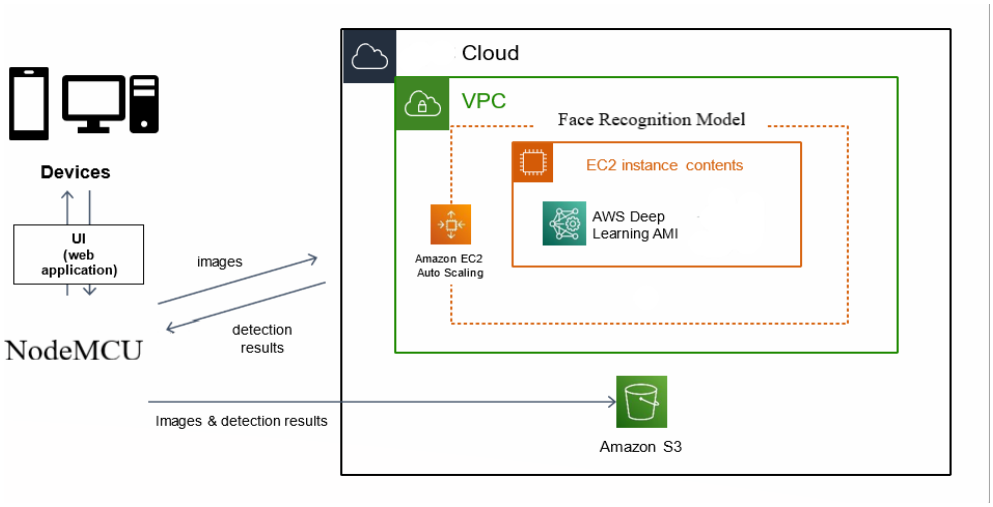

Figure 4.2 Proposed Method of Block Diagram

4.2.1 Working principle of AWS with ESP32 Camera

A managed cloud service called AWS IoT Core enables connected devices to quickly and securely communicate with other gadgets and cloud-based software. You can send and receive MQTT messages to and from AWS IoT Core using the MQTT messaging service. By obtaining an AWS IoT certificate, learning how to attach the things, and connecting to AWS IoT using MQTT, to construct an AWS IoT thing that represents the ESP32 by using the Publish/Subscribe capability and the AWS IoT Core dashboard.

The connection between an ESP32-CAM and an AWS IoT Core is shown in the following diagram. Publishing and subscribing to MQTT topics are both supported. As a result, the device is able to send any information it chooses to AWS IoT Core and get orders in return.

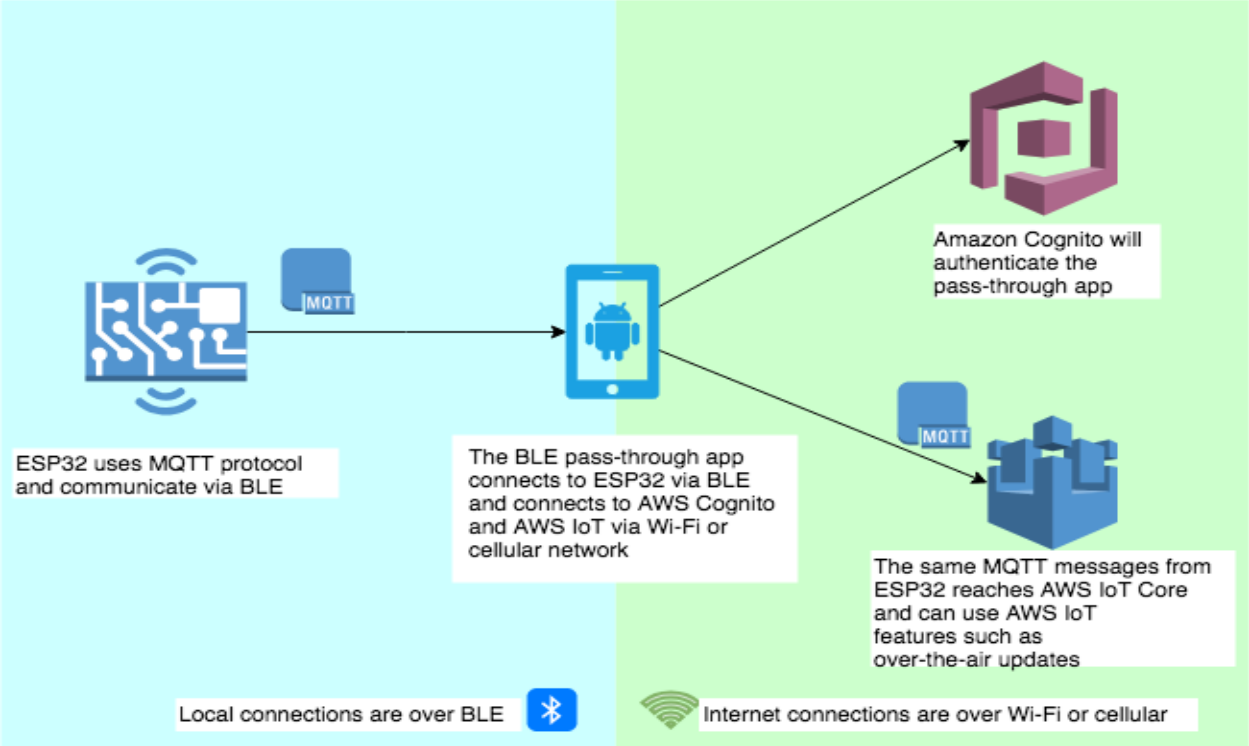

Figure 4.2.1 Principle of AWS with ESP32 Camera Diagram

In this project, to capture the image of the authorized person, that was uploaded into AWS S3 bucket by using LAMDA function method that will triggered all the images in Amazon S3 bucket.

By using Amazon S3 bucket are used to convert the faceprint of the corresponding images. Next it sends the corresponding images into DynamoDB with recognition unique ID of the image.

For capturing the image, using live camera (CCTV camera) or product (ESP32 with camera module). In that image, face of the image using CV2 Cascade to create faceprint with unique id. And then, it will move into the AWS Rekognition Service.

By using the AWS Rekognition Services, compare the faceprints with Dynamo Database faceprints return the accuracy of the comparisons and face print of the both images. It checks the corresponding face person in the database, if it is found in that database, it assume that it is a authorized person or unauthorized person in real time data (time, live face image, name and etc.,) stored in database.

A web-based user interface (UI) that makes it simple for users to view the results of the smart counting system's counting. By using web-based application will show the live footage, authorized and unauthorized count and total number of person crossed in that footage. In that application, it provides an option to add new authorized person into the database. To make use as a surveillance CCTV Camera to monitor and record the footage of the live video to make as a good security purpose system.

4.2.2 AWS Cloud Services

The world's most complete full cloud-platform, Amazon Web Service (AWS)which offers the more than 200 fully-functional services get from data centers spread out throughout the globe. A scalable and affordable cloud computing option is offered by Amazon Web Service, a platform online. AWS is the widely used big-cloud computing platforms that provides a different of on-demanding services to businesses to help them to scale and growing, including compute power, the database storage, the content of distribution, etc. According to the needs of the user, AWS often operates in several distinct configurations. The server map and configuration type must be visible to the user in relation to the AWS service. The most trustworthy pricing structure is offered by AWS because it is so economical. Suppose, for example, that a user wants to use a cloud server for an hour.

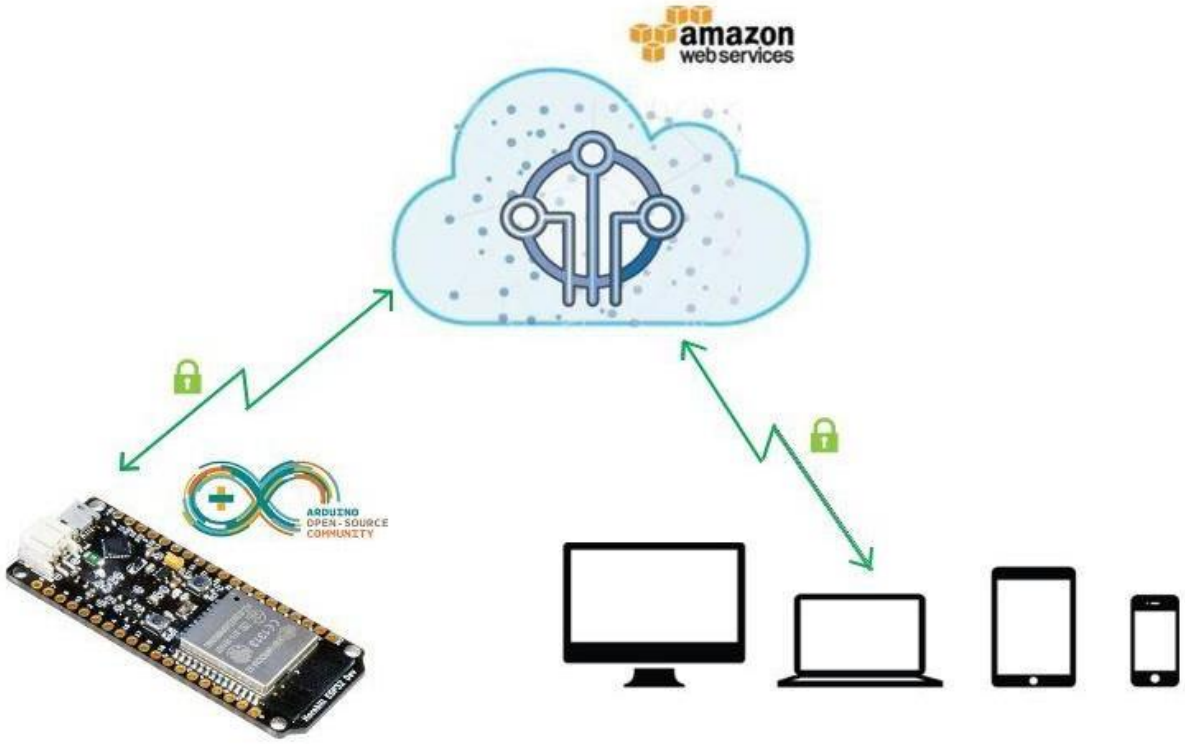

Figure 4.2.2 AWS Cloud Service Diagram

4.2.3 Features of AWS Cloud Services

- AWS offers web services in a variety of fields, including compute, storage, databases, and networking.
- The simplicity of use of AWS is its first strength. Customers can safely host both new and old apps with the help of AWS.
- AWS offers well-documented APIs in addition to a GUI, the AWS management console.
- There are a ton of examples and code snippets available in addition to this guide to help you embrace AWS.
- The flexibility of AWS is the second strength.
- AWS gives you the freedom to select elements of your virtual environment, such the operating system.
- If there isn't a lot of traffic coming in, AWS enables you to dynamically reallocate resources, or boost these resources if there is.
- A secure or security phrase While working in the AWS Cloud environment, you will hear a lot of noise.
- AWS offers complete security, from the operational and software safeguards against unauthorized use of your resources to the physical security of the underlying data centers.
- Like in operating system, the operating programming languages, the web-based application platform, database and variety of different services you require are all selectable through AWS.
- A wide range of international cloud-based services are provided by Amazon web services by includes of compute, the storages, databases, analytics and the networking mobile development, the developing tools, the management tools, Internet of Things, the security, etc.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 PERFORMANCE METRICS

5.1.1 YOLO v5 Model Architecture

YOLO v5 is a single target object detector with three important aspects, just like any other single target object detector.

- Model Neck
- Model Backbone
- Model Head

Model-Backbone is primarily used to retrieve core aspects from an input image by employing CSP (Cross-Stage-Partial-Networks), which are employed as a foundation in YOLO v5 to retrieve rich data from an input image by employing informative features.

5.1.2 YOLO v5 Model Architecture

It is recommended to train with over 1500 images per class and over 10,000 instances per class to achieve a robust YOLOv5 model. It is also advised to include up to 10% background images in order to reduce false-positive errors. Because my dataset is so small, I'll use transfer learning techniques to narrow the training process.

Most annotation platforms support YOLO labeling format export, which provides one annotations text file per image. Each text file contains one annotation for each of the image's objects. The annotations range from 0 to 1 YOLOv5 is straightforward and dependable. It requires much less computational power than other architectures while producing comparable results and performing much faster.

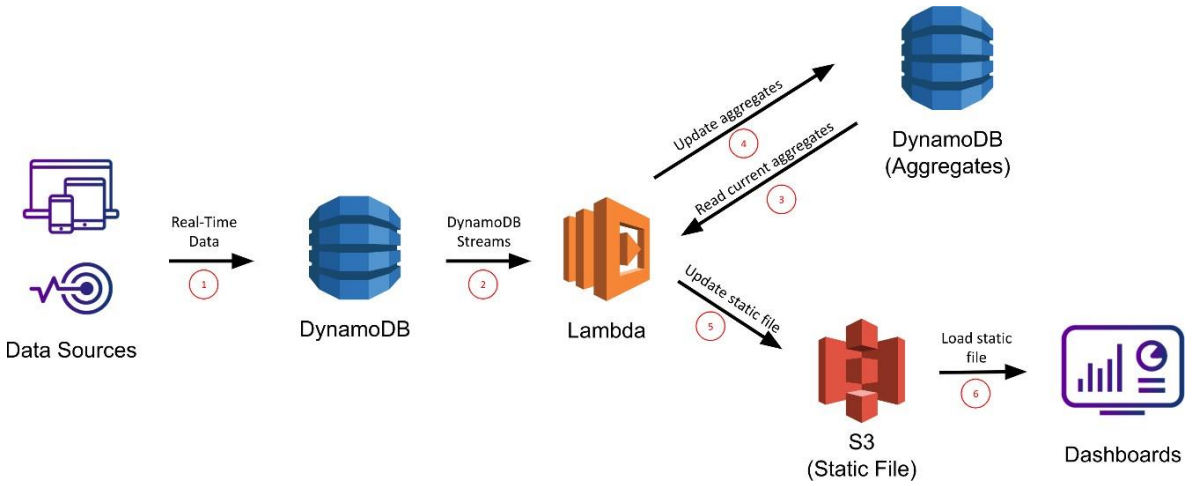

Figure 5.1.2 Block Diagram of AWS S3

In this article, the performed basic CRUD operations in Dynamo DB, such as creating, inserting, deleting, and updating records. To discuss data querying and aggregation. This, however, is insufficient. Here are some external resources you should look into if you want to learnmore about Dynamo DB. AWS is an Amazon web services database that is highly scalable, low latency, and highly secure. It debuted in 2012, following MongoDB. This is used in a variety of industries, including retail and media.

5.1.3 Accuracy of YOLO v5 Model

At the moment, to achieve an favorable balance of 98.56% accuracy and speed using the You-Only-Look-Once (YOLOv5s) model for image detection systems. As a result, the study employs the YOLOv5s version. The model is trained on 600 remote sensing data from the UCAS-AOD datasets and evaluated on 400 additional images taken from the same datasets. YOLOv5 achieves the 99.3% in 74 seconds, based on the test findings. It demonstrates that this prototype can detect object faces with greater precision in live time.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Thus the "**SMART PEOPLE COUNT SYSTEM BASED ON FACE RECOGNITION USING DEEP LEARNING APROACH**" has been designed and tested successfully. It was created by integrating functionality from every piece of hardware used. The system's ability to run autonomously has been evaluated. The ESP32 Camera records a person's image in the AWS Cloud, compares it, counts the authorized and unauthorized people, and keeps the results in the cloud. A web-based user interface (UI) that makes it simple for users to view the results of the smart counting system's counting.

Deep learning object detection methods will also be used in future studies to improve forecast accuracy. By expanding this project, it is possible to have a significant impact on the workplace, colleges, and other institutions, etc. For future scope to update the web- based application into mobile application for good responsibility and security purposes to easy access of users. And also, system integrates more than one camera footages to recognize the count of authorized and unauthorized person to reduce the cost of multiple computer system and to make the control of single computer system. To improve an accuracy of identifying and matching the computer prints by using YOLOv7 version to make improved fast and security for future purposes.

APPENDIX – 1 (SOURCE CODE)

```
import io
import os
import cv2
import boto3
import numpy as np
import pandas as pd
from PIL import Image
import boto3 as boto3
import mysql.connector
from datetime import date
from base64 import b64encode
from datetime import datetime
from IPython.display import display
from werkzeug.utils import secure_filename
from flask import Flask, request, render_template, jsonify, redirect, Response
# Defining Flask App
app = Flask(_name_)
# To get invoke the home page...
@app.route('/')
def home():
    # Get date and Time...
    today_dat = date.today().strftime("%d/%m/%Y")
    today_time = datetime.now().strftime("%H:%M")
    #Get connection to MySql...
    connection = mysql.connector.connect(host='localhost', database='face', user='root', password='')
    cursor = connection.cursor()
    #Get How many data from authorized table...
    sql_fetch_blob_query = """SELECT COUNT(*) from face_data_auth"""
```

```
    cursor.execute(sql_fetch_blob_query)
    auth_record = cursor.fetchall()
    for row in auth_record:
        auth_count = row[0]
    #Get How many data in Unauthorized table...
    sql_fetch_blob_query = """SELECT COUNT(*) from face_data_unauth"""
    cursor.execute(sql_fetch_blob_query)
    unauth_record = cursor.fetchall()
    for row in unauth_record:
        unauth_count = row[0]
    print("Date = ", today_date)
    print("Time = ", today_time)
    print("Authorized person count = ", auth_count)
    print("Unauthorized person count = ", unauth_count)
    return render_template('index.html', today_dat = today_dat, today_time = today_time, auth_count =
auth_count, unauth_count = unauth_count)
#TO get a live video stream...
def generate_frames():
    #capture the live video footage...
    # capture = cv2.VideoCapture('aaa.mp4')
    capture = cv2.VideoCapture(0)
    while True:
        boolean, frame = capture.read()
        # Convert into grayscale...
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        # Load the cascade...
        # face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        face_cascade = cv2.CascadeClassifier('haarcascade_profileface.xml')
        # Detect faces...
        faces = face_cascade.detectMultiScale(gray, 1.1, 4)
        # Draw rectangle around the faces and crop the faces...
        for (x, y, w, h) in faces:
```

```
            cv2.rectangle(frame, (x-50, y-50), (x+w+30, y+h+30), (0, 0, 255), 1)
            # crop a gace image...
            finimg = frame[y-50:y+h+30, x-50:x+w+30]
            # compare this image with AWS Rekognition face images...
            cmpface(finimg)
        # load to the buffer and send it to HTML...
        ret, buffer = cv2.imencode('.jpg', frame)
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```

To compare the real time image to AWS images...

```
def cmpface(finimg):
    try:
        # save in local storage...
        cv2.imwrite("test.jpg", finimg)
        # Rekongnition service config...
        rekognition = boto3.client('rekognition', region_name='us-east-1')
        dynamodb = boto3.client('dynamodb', region_name='us-east-1')
        # Convert into binary image formate...
        image = Image.open('test.jpg')
        stream = io.BytesIO()
        image.save(stream, format="JPEG")
        image_binary = stream.getvalue()
        # Get responce from Rekognition service...
        response = rekognition.search_faces_by_image(CollectionId='famouspersons', Image={'Bytes':
image_binary})
        # Check Authorized person or NOT by compare all faceprints...
        found = False
        for match in response['FaceMatches']:
            print(match['Face']['FaceId'], match['Face']['Confidence'])
            face = dynamodb.get_item(TableName='face_recognition', Key={'RekognitionId': {'S':
match['Face']['FaceId']}})
```

```
        # If Found person or matches faceprint...
        if 'Item' in face:
            # get deatails of the faceprints...
            print("Found Person: ", face['Item']['FullName']['S'])
            name=str(face['Item']['FullName']['S'])
                        found = True
            try:
                # SQL database config...
                connection = mysql.connector.connect(host='localhost', database='face', user='root',
password='')
                cursor = connection.cursor()
                sql_insert_query = """INSERT INTO face_data_auth(username, photo) VALUES
(%s,%s)"""
               # Convert data into tuple format and push it to cloud...
                insert_tuple = (name, image_binary)
                result = cursor.execute(sql_insert_query, insert_tuple)
                connection.commit()
                print("Image inserted successfully...", result)
               # Get wether any issue arrise...
            except mysql.connector.Error as error:
                print("Failed inserting Image... {}".format(error))
            # close the Connection from SQL database...
            finally:
                if connection.is_connected():
                    cursor.close()
                    connection.close()
                    print("MySQL connection is closed...")
                home()
      # If cannot found a person or not maches faceprints...
       if not found:
          print("Person cannot be recognized")
```

```
            name = "unauthorized"
            try:
                # SQL database config...
                connection = mysql.connector.connect(host='localhost', database='face', user='root',
password='')
                cursor = connection.cursor()
                sql_insert_query = """INSERT INTO face_data_unauth(username, photo) VALUES
(%s,%s)"""
                # Convert data into tuple format and push it to cloud...
                insert_tuple = (name, image_binary)
                result = cursor.execute(sql_insert_query, insert_tuple)
                connection.commit()
                print("Image and file inserted successfully...", result)
                # Get wether any issue arrise...
                except mysql.connector.Error as error:
                print("Failed inserting BLOB data into MySQL table {}".format(error))
                # close the Connection from SQL database...
            finally:
                if connection.is_connected():
                    cursor.close()
                    connection.close()
                    print("MySQL connection is closed...")
                    home()
    except:
        print("No faces found...")
#to get a new name for upload image to S3 bucket...
def newname():
    bucket = "ambi-persons-images"
    folder = "index"
    s3 = boto3.resource("s3")
    s3_bucket = s3.Bucket(bucket)
    files_in_s3 = [f.key.split(folder + "/")[1] for f in s3_bucket.objects.filter(Prefix=folder).all()]
```

```
    count = len(files_in_s3)+1
    name="image_"+str(count)+".jpg"
    return name
# Upload to the AWS S3 Bucket...
@app.route('/sent', methods=['GET', 'POST'])
def sent():
    # Get user name...
    newusername = request.form['newusername']
    if request.method == "POST":
        # Get user Image...
        image = request.files['newuserimages']
        if image.filename == '':
            print("Image must have a file name")
            return redirect(request.url)
        # Save the image...
        newimgname=newname()
        image.save(newimgname)
        # Get list of objects for indexing
        images = [(newimgname, newusername)]
        # Iterate through list to upload objects to S3
        s3 = boto3.resource('s3')
        for img in images:
            file = open(img[0], 'rb')
            object = s3.Object('ambi-persons-images', 'index/' + img[0])
            ret = object.put(Body=file, Metadata={'FullName': img[1]})
        return render_template('index.html')
    return render_template('addperson.html')
# Authorized persons management...
@app.route('/auth')
def auth():
    try:
```

```
    # Get a connection from SQL....
    connection = mysql.connector.connect(host='localhost', database='face', user='root', password='')
    cursor = connection.cursor()
    sql_fetch_blob_query = """SELECT * from face_data_auth"""
    cursor.execute(sql_fetch_blob_query)
    record = cursor.fetchall()
    # Get all the data from database...
    for row in record:
        # print("Id = ", row[0], )
        # print("Name = ", row[1])
        # print("image = ", row[2])
        # print("date and time = ", row[3])
        Name = row[1]
        with open("db_img.jpg", 'wb') as file:
            file.write(row[2])
        # Get new file name...
        imgname="auth_face_"+Name+".jpg"
        # Get new dir file or create...
        if not os.path.isdir('static'):
            os.makedirs('static')
        # store in local...
        img = cv2.imread("db_img.jpg")
        cv2.imwrite('static/'+imgname, img)
    # Send to html file...
    return render_template('auth.html',result=record)
# check any issue will arrise...
except mysql.connector.Error as error:
    print("Failed to read data from MySQL… {}".format(error))
    return render_template('auth.html',result=record)
# Close the connection...
finally:
```

```
        if connection.is_connected():
            cursor.close()
            connection.close()
            print("MySQL connection is closed")
            return render_template('auth.html',result=record)
# Unauthorized persons management...
@app.route('/unauth')
def unauth():
    try:
        # Get a connection from SQL....
        connection = mysql.connector.connect(host='localhost', database='face', user='root', password='')
        cursor = connection.cursor()
        sql_fetch_blob_query = """SELECT * from face_data_unauth"""
        cursor.execute(sql_fetch_blob_query)
        record = cursor.fetchall()
        Number = 1
        # Get all the data from database...
        for row in record:
            with open("db_img.jpg", 'wb') as file:
                file.write(row[2])
            # Get new file name...
            imgname="unauth_face_"+str(Number)+".jpg"
            Number=Number+1
            # Get new dir file or create...
            if not os.path.isdir('static'):
                os.makedirs('static')
            # store in local...
            img = cv2.imread("db_img.jpg")
            cv2.imwrite('static/'+imgname, img)
    except mysql.connector.Error as error:
        print("Failed to read BLOB data from MySQL table {}".format(error))
```

```
    finally:
        if connection.is_connected():
            cursor.close()
            connection.close()
            print("MySQL connection is closed")
    return render_template('unauth.html', result=record)
@app.route('/addperson')
def addperson():
    return render_template('addperson.html')
@app.route('/video_feed')
def video_feed():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')
# Our main function which runs the Flask App
if _name_ == '_main_':
    app.run(debug=True)
```

APPENDIX – 2(SCREENSHOTS)

Figure 6.1 User Interface of WebApp

Figure 6.2 Authorized Person List

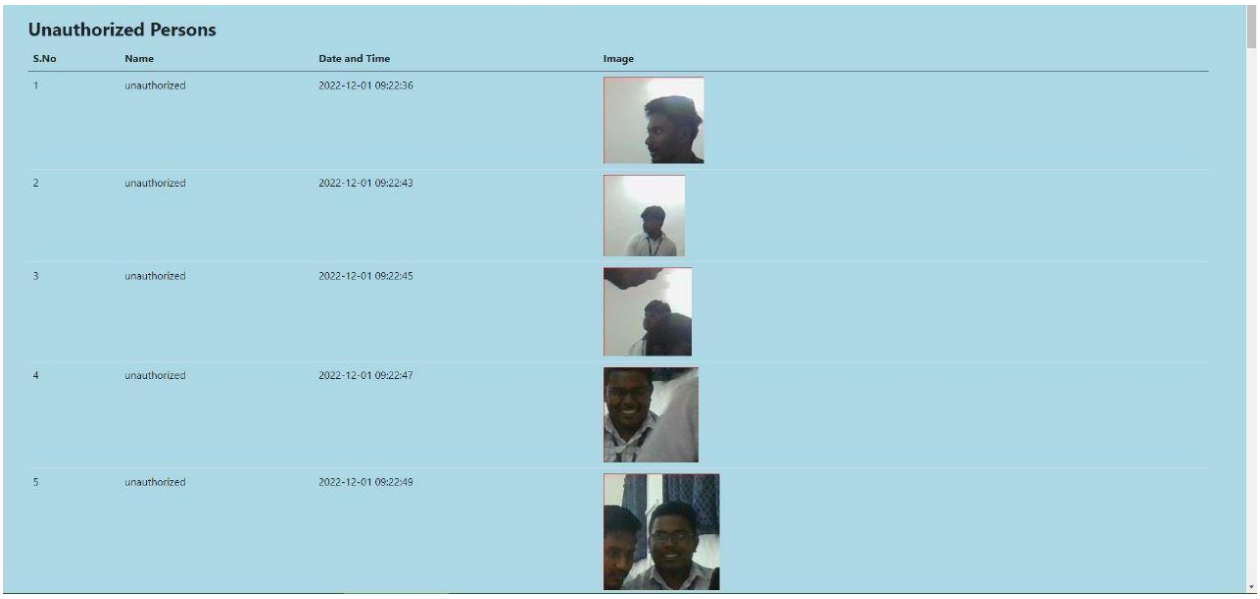

Figure 6.3 Unauthorized Person List

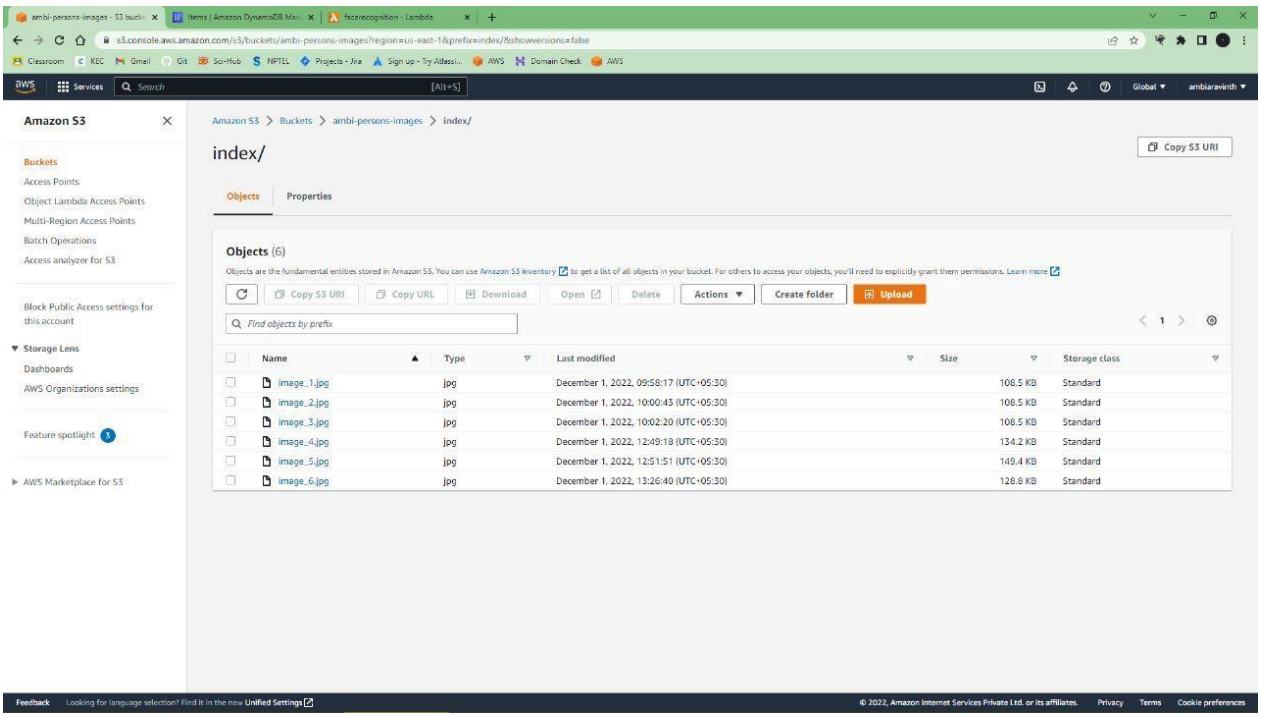

Figure 6.4 Data Storage in AWS S3

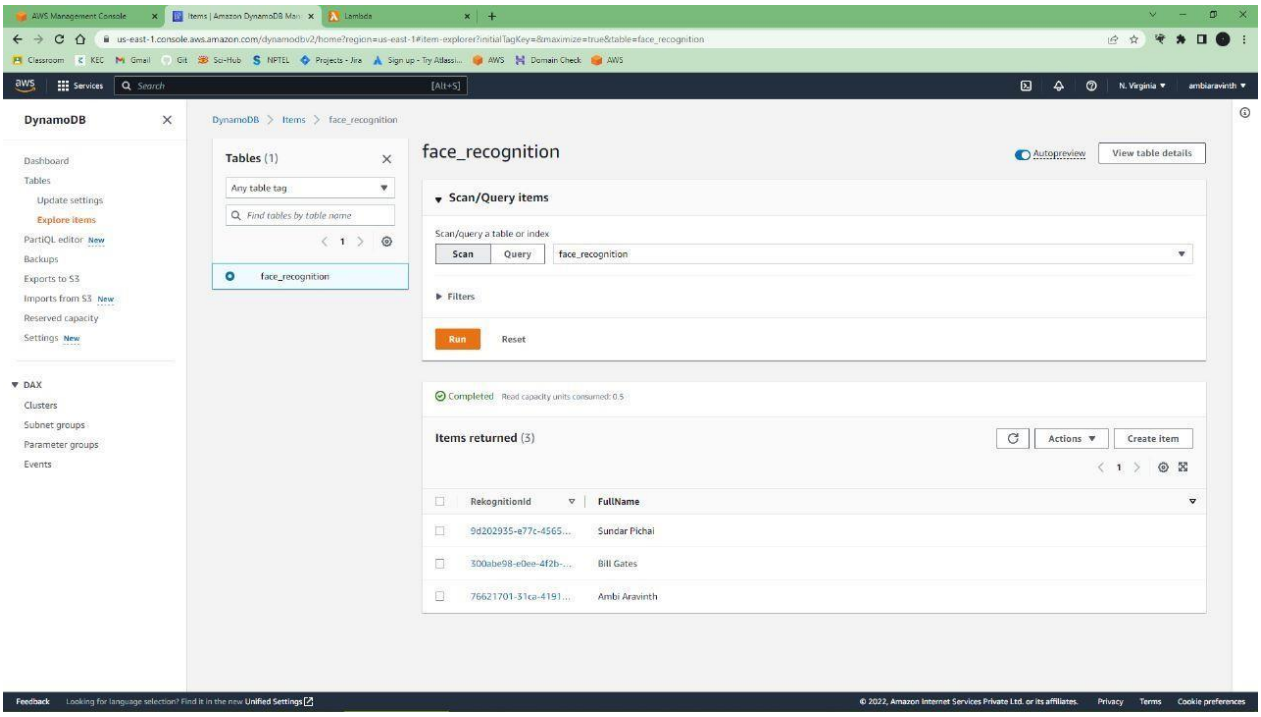

Figure 6.5 Data Storage in DynamoDB

REFERENCES

[1] Lee, H., Moon, J., and S. Lim Smart Counting System Based on Object Detection Using Deep Learning, Yu, S.; Lee, K.-B. Remote Sensing, 2202, 14, 3761.

[2] Y. Xue, N. Ray, J. Hugh, and G. Bigras Regression cell counting with a convolutional neural network. In the 2016 Springer Berlin/Heidelberg publication, Proceedings of the European Conference on Computer Vision, pp. 274–290.

[3] The smart image recognition mechanism for crop harvesting system in intelligent agriculture by Horng, G.J., Liu, M.X., and Chen, C.C. IEEE Sensors J, 20, 2766–2781, 2019.

[4] Mobilenetv2: Inverted residuals and linear bottlenecks. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. 4510-4520 in Salt Lake City, UT, USA, 18-23 June 2018, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[5] W. Liu; D. Anguelov; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.

[6] Sikandar Khan, Adeel Akram, Nighat Usman, “Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV,” 31 March 2020.

[7] Xi Zhao, Emmanuel Dellandrea and Liming Chen, " A People Counting System based on Face Detection and Tracking in a Video" 2019.

[8] Y. PAVAN KUMAR, S. NARENDRA REDDY, G. AKHIL, CH.SAI, " Office Automation with Face Detection and Counting System using Haar Cascade" 2019.

[9] Jiwon Moon, Sangkyu Lim, Hakjun Lee, Seungbum Yu and Ki-Baek Lee , "Smart Count System Based on Object Detection Using Deep Learning" 2019.

[10] Danish Ali Chowdhry, Aqeel Hussain, Muhammad Zaka Ur Rehman, Farhan Ahmad, Arslan Ahmad, Mahmood Pervaiz, "Smart Security System for Sensitive Area Using Face Recognition"2020.

[11] Mr. Rajvardhan Shendge, Mr. Aditya Patil, Mrs. Tejashree Shendge," Smart Attendance System using Face-Recognition" 2022.

[12] M. Rivai, Suwito, M. Ashari and M. A. Mustaghfirin, "Face Recognition on a Smart Image Sensor Using Local Gradients" 2021.

[13] Kittipong Tapyou, Pannawich Chaisil, Jirapond Muangprathub "Smart School Attendance System using Face Recognition with Near Optimal Imaging", 2021.

[14] Smitha Shekar B, Harish G, Aishwarya B R, Badri Narayan S, Chinmayeshree K B, "Smart Classroom Attendance System Using Face Recognition and Raspberry Pi" ,2017.

[15] M.Kasiselvanathan, Dr.A.Kalaiselvi, Dr.S.P.Vimal, V.Sangeetha ," Smart Attendance Management System Based On Face Recognition Algorithm’’, 2021.