

An Ambient Assisted Living System for Alzheimer's Patients

by

Minhajul Abedin

18301224

Mohammad Abdul Ahad

18301248

A.B.M Hasan-Ul-Banna

18301143

Nibraz Khan

18201057

Ashfaq Hossain

18101658

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

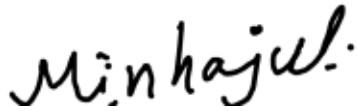
© 2022. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Minhajul Abedin
18301224



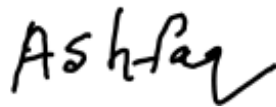
Mohammad Abdul Ahad
18301248



A.B.M Hasan-Ul-Banna
18301143



Nibras Khan
18201057



Ashfaq Hossain
18101658

Approval

The thesis/project titled “An Ambient Assisted Living System for Alzheimer’s Patients” submitted by

1. Minhajul Abedin(18301224)
2. Mohammad Abdul Ahad(18301248)
3. A.B.M Hasan-Ul-Banna(18301143)
4. Nibraz Khan(18201057)
5. Ashfaq Hossain(18101658)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 22, 2022.

Examining Committee:

Supervisor:
(Member)



Dr. Md. Golam Rabiul Alam, PhD
Professor

Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Alzheimer's is a brain disorder that gradually deteriorates the brain functions of the patients. As the disease progresses, victims start to lose their memory, thinking ability, eventually rendering them unable to perform basic tasks. They also face many difficulties namely disorientation, wandering, aggression, insomnia, hallucination, etc. What makes the situation worse is that when the caregivers try to help them most of the time they tend not to cooperate. In this paper, we have designed an AI that assists the sufferers in combating these issues by analyzing their environment, daily routine, interests, behavioral patterns, and many more factors. Using computer vision we have created a face recognition framework that identifies individuals in front of the patient & shows him/her their name, how they are related, and some photos & videos of them together. We also used an object detection system that helps prevent wandering by constantly monitoring the surroundings of the patient & notifying the caretakers about items such as keys, shoes, handbags, doors etc that could influence the patient to leave the house. The AI is instructed to alarm the attendant continuously if the patient somehow succeeds to go beyond the safe area. This feature allows the caregivers some free time as they don't need to monitor the patients 24/7 anymore. The face recognition framework achieves accuracy of 97.44% and the object detection system has mAP of 72.3% that uses YOLOv7 model. Thus, this study tries to achieve its goal to make life comparatively easier for the patients & the caregivers by making the patients self-dependent & discharging the attendants from some of their tasks.

Keywords: Alzheimer's Disease; Artificial Intelligence; Deep Learning; Object Detection; Face Recognition; Face Detection; Face Embedding; Face Classification; YOLOv4; YOLOv7; MTCNN; FaceNet; SVC; RFs

Acknowledgement

We are eternally grateful to Almighty Allah, without whom we could not have completed our thesis. Words cannot explain how thankful we are to our supervisor, Dr. Golam Rabiul Alam sir, for his time and suggestions. Also, without the kind assistance, knowledge, and skill of our honourable Lecturer Mr. Rafeed Rahman, Mr. Tanzim Reza, and Mr. Asif Shahriyar Sushmit sir our work would not have been completed. We would like to thank them for their helpful feedback and moral support in our effort. Finally, a special thanks to our amazing parents, whose prayers and support enabled us to finish our thesis work.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Nomenclature	x
1 Introduction	1
1.1 Motivation	1
1.2 Research Problem	2
1.3 Research Objective	3
2 Literature Review	4
2.1 State of the Arts in Object Detection	4
2.2 State of the Arts in Face Recognition	7
3 Methodology	9
3.1 Proposed Object Detection Framework	13
3.1.1 Data Collection	13
3.1.2 Implementation and Training	14
3.2 Proposed Face Recognition Framework	21
3.2.1 Data Collection	21
3.2.2 Implementation and Training	22
4 Result and Findings	26
4.1 Evaluation Metrics	26
4.2 Result Analysis and Comparison	27
4.2.1 Object Detection	27
4.2.2 Face Recognition	28

5 Conclusion and Future Work	30
5.1 Conclusion	30
5.2 Future Work	30
Bibliography	32

List of Figures

3.1	Top level overview of the proposed ambient assisted living system . . .	9
3.2	Proposed living system : Complete Block Diagram	12
3.3	Work plan of the proposed object detection framework	13
3.4	Basic YOLOv4 architecture [9]	15
3.5	CSPDarknet53 [10]	16
3.6	Epoch vs Precision Graph	19
3.7	Epoch vs Recall Graph	19
3.8	Epoch vs mAP@0.5 Graph	20
3.9	Epoch vs mAP@0.5:0.95 Graph	20
3.10	Work plan of the proposed face recognition framework	21
3.11	Triplet Loss	24
4.1	Model accuracy comparisn of SVC and RFs	29

List of Tables

4.1	Class Based Performance Comparison	28
4.2	Overall Performance Comparison	28
4.3	Comparison between SVC and RFs models	29

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AD Alzheimer Disease

AI Artificial intelligence

AP Average Precision

CBN Cross-Iteration Batch Normalization

CNN Convolutional Neural Network

COCO Common Objects in Context

FPS Frames Per Second

GPS Global Positioning System

LDA Linear Discriminant Analysis

LFW Labeled Faces in the Wild

LSTM Long short-term memory

mAP Mean Average Precision

MTCNN Multi-Task Cascaded Convolutional Neural Networks

PAM Partition Around Medoids

PAN Path Aggregation Network

RCNN Region-based Convolutional Neural Network

ResNet Residual Neural Network

RFs Random Forest

RNN Recurrent Neural Network

SAM Self-Attention based Deep Learning Method

SPP Spatial Pyramid Pooling

SSD Single Shot Detector

SVC Support Vector Classifier

VGG Visual Geometry Group

YOLO You Only Look Once

Chapter 1

Introduction

1.1 Motivation

Alzheimer's disease (AD) is the most prevalent kind of dementia and a degenerative brain condition. It accounts for up to 60-70 percent of dementia cases [15]. There are more than 35 million people worldwide living with Alzheimer's disease and that number is expected to double in the next 20 years [1]. It is classified into three stages: mild, moderate, and severe [16]. Additionally, survey results indicate that this condition affects adults of all ages, including seniors and middle-aged individuals [6]. The illness develops over time, and sufferers gradually lose control of their brains and bodies. The patient may be able to complete basic chores in the first two phases by themselves but they require assistance from others from time to time. Patients in the last stage, on the other hand, require intensive care.

AD is often confused with normal aging but in reality, it is much more serious than that. In [3] it was shown that the difference between a normal brain and an AD patient's brain. In the Alzheimer's brain, the cortex and hippocampus shrink and damage the areas that are responsible for thinking, planning, and remembering. Ventricles, fluid-filled spaces within the brain enlarge. For these reasons people suffering from it face many symptoms such as memory loss, decreased cognitive ability, hallucination, disorientation, aphasia, sundowning, etc.[15],[14].

There are numerous caregivers(paid and unpaid) who take care of the patient. Most of the unpaid caregivers are the family members of the patient and they face challenging situations dealing with the AD patient [16]. They struggle to maintain financial stability as the better of their day gets spent on taking care of the AD patients. Our research aims to aid AD patients as well as caregivers by building an assistive technology that helps the AD patients become more self-dependent and spare the caregivers from some of their duties in the process so that they can focus on their careers & other important aspects of life.

1.2 Research Problem

Alzheimer's disease causes a lot of hurdles in the day-to-day life of the patients in the form of memory loss, hallucination, cognitive impairment, aggression, depression, wandering, trouble navigating, aphasia, disorientation, etc[15],[14]. Among all these issues, we will address the following problems in this paper.

- **The decline of facial recognition:** They sometimes fail to remember people that are close to them. This is a common symptom of dementia. This can create an unpleasant situation for the patient and the people concerned.
- **Wandering:** Wandering can happen whenever they try to follow their old routine or they see any object which they used to bring with them when going outside such as a key, handbag, shoes, etc. or even something that means outdoor like doors, doormats, etc. can influence their brain to go outside. This can lead them to leave the house & fall into life-threatening circumstances.

Not only the patients who suffer because of AD but also their family members have a hard time looking after them. According to [16] 80% of AD patients are getting home-based care. In the US, more than 17 billion unpaid hours are given by approximately 16 million people who are friends & family members of AD patients. Informal carers (i.e. most commonly family members and friends) spent 5 hours per day on average giving assistance to people with AD in 2019[15]. Giving care to an AD patient is a 24-hour job as a result the caregivers barely have the time to work on their careers. They struggle to achieve financial stability because of that.

In our research, we have tried to build an intricate system based on visual feeds that can detect wandering causing objects and identify familiar faces. We made it a scalable system so that we have the flexibility to add more modules & features to make it even better in the future. It has both hardware and software components. Using the hardware sensors such as a camera and GPS it'll monitor the patient's activities, and based on that if it finds any undesired case it'll take necessary actions accordingly & alarm the caregiver in their app interface. Our goal is to offer a better quality of life to both the patients by making them self-dependent and the caregivers by giving them more free time than before.

The proposed system will have the following features:

- Our system detects faces from the real-time video & matches the faces with the saved face instances in the database to identify that person. If it can recognize the face instance then the system will show the name & other related information of that person in the smart glass & it will also have the option to say the name of the person through bone conduction headphones.
- The proposed system will look for objects that might influence the patient into leaving the house. It'll notify the caretakers of the position of those objects upon finding any. For instance, it'll look for the main door and if it finds any then it'll prompt the caretaker to camouflage it because if the patient notices that door he/she might open it & leave.

1.3 Research Objective

This research aims to build a state-of-the-art system that can identify known people & likelihood of causes that can influence wandering using face recognition, and object detection etc. The proposed system is scalable so that we can add more features in the future to make the life of the patients & their caregivers easier. The objectives of this research are:

- To make an AI based ambient assisted living system that will help AD patients and their caregivers.
- To create datasets for objects that are not readily available in any public datasets.
- To find suitable models for object detection and face recognition tasks for the AI based system.
- To develop a personalized system that can be integrated with wearable devices that are used in day to day life.
- To implement the system in the day-to-day life of an AD patient & evaluate its real-world efficacy.

Chapter 2

Literature Review

2.1 State of the Arts in Object Detection

Object recognition in image analysis and vision is hard in real-world scenarios. To resolve this issue, modern object detection methods like R-CNN, Mask R-CNN, SPP-net, and YOLO are discussed thoroughly in [7]. High-Quality R-CNN Object Detection using Multi-Path Detection Calibration Network is proposed in [13] to improve the data discrepancy between object proposals and refined bounding boxes in R-CNN. Two-path Convolutional LSTM is introduced in [12] that improves object detection in multi-scale images.

There are several apps and trends in Deep Learning-Based Object Detection and some of the region proposal frameworks include SPP-net, R-CNN, Mask R-CNN, etc. YOLO, SSD, YOLOv2, DSSD, and DSOD are used in generic object detection [7]. The R-CNN model includes three modules, such as region proposal, extraction of deep CNN-based features, and classification/localization. Cropping, wrapping, and multi-step training procedures distort accuracy in R-CNN. As a result, YOLO which is a grid of $S \times S$ cells came which separates images into objects and it has superior results to previous real-time object detectors and the model can process pictures at 45 frames per second. YOLOv2 came after that as discussed in [7] which increased speed and precision and it is a logistic classifier for multilabel classification for complicated datasets. However, YOLO has trouble generalizing objects with odd aspect ratios. So, SSD arrived to fix the problem and can handle objects of varying sizes by integrating predictions from numerous feature maps. Also, Deconvolutional Single Shot MultiBox Detector is an upgraded version of SSD that came out with two more modules, deconvolutional and prediction, and then Retina Net came which employs the RetinaNet focal loss to classify. PASCAL VOC 2007, PASCAL VOC 2012, ImageNet, Microsoft COCO, and OpenImages are five prominent datasets for general object recognition according to [7]. These techniques may be utilized in saliency detection, 3D object detection, aerial detection, sensing, and many more applications.

In a Multi-Path Detection Calibration Network (PDC-Net) has been proposed to acknowledge the discrepancy of data distribution between object proposals and refined bounding boxes in two-stage detectors that use object proposals [13]. Usually, the discrepancy in data distribution between object proposals and refined bounding

boxes is ignored by the two-stage detectors such as R-CNN. This could limit the detection accuracy as classification scores are given to improved bounding boxes rather than object detection. To resolve this problem, the detection result generated from object proposals by a base detector Faster R-CNN will be used as input for multiple calibration detectors for fixing incorrect outputs using refined bounding boxes. This method has successfully boosted the detection results and gives 83.1% for PASCAL VOC and 43.3% for MSCOCO benchmarks.

As Multi-scale video object detection is a challenging task, a two-path Convolutional Long Short-Term Memory(convLSTM) pyramid network is proposed to extract and convey multi-scale contextual information of different resolutions [12]. ConvLSTM resizes each input frame into four feature frames of varying resolution [12]. Multi-input convLSTM architecture uses a two-path convLSTM pyramid to input feature maps from the current time step and state information from past time steps. Top-down and bottom-up updates are utilized, with the bottom-up update output feeding the object detector. For feature extraction Res-Net-101 is used and also VGG, Inception ResNet, DenseNet, and MobileNet are used in [12]. It is one of the first approaches that introduces the connections between layers of the convolutional LSTM pyramid and makes use of contextual information for small-to-large and large-to-small changes in video object detection [12].ImageNet VID dataset is used as a dataset to compare the performance of the proposed method. It has been observed that the proposed method achieves a high improvement in multi-scale video object detection compared with the still-image baseline Faster R-CNN. The method achieves a mAP% of 82.8 while the base Faster R-CNN achieves 74.2. This approach can be computationally expensive if the number of channels in the LSTM pyramid is increased so a lightweight feature extraction network can be used and an appropriate number of channels can be used in the network to tackle this limitation [12].

The main goal is to design a manufacturing system object detector that operates quickly and is optimized for parallel computations, instead of the theoretical indicator of limited computation volume [8]. In their paper with equivalent performance, YOLOv4 runs twice as quickly as EfficientDet. AP and FPS in YOLOv3 are enhanced by 10% and 12%, respectively. One of the contributions of their research is that they actually create a robust and effective object detection model which enables everyone to train a very quick and precise object detector using a 1080 Ti or 2080 Ti GPU. Another contribution of them is to improve cutting-edge techniques like CBN, PAN, SAM, and others to make them more effective and suited for training on a single GPU. In their work, they used 1 to 8 groups in convolutional layers for GPU by using CSPDarknet53 and CSPResNeXt50 and also employ grouped-convolution for the VPU, but they avoid employing Squeeze-and-Excitement (SE) blocks in particular for the following models: GhostNet, EfficientNet-lite, MixNet, and MobileNetV3. Moreover, their extensive research demonstrates that, when it comes to object classification on the ILSVRC2012 (ImageNet) dataset, the CSPResNext50 outperforms the CSPDarknet53 by a wide margin. On the other hand, in terms of the MS COCO dataset, CSPDarknet53 outperforms CSPResNext50 in terms of object detection. There are just 16 convolutional layers ($3 * 3$), a $425 * 425$ receptive field, and 20.6M parameters in the CSPResNext50. whereas CSPDarknet53 has 27.6M parameters, $725*725$ receptive field, and 29 convolutional layers

3*3. When they run YOLOv4 on popularly used GPUs of the Maxwell, Pascal, and Volta architectures and compare it to other cutting-edge techniques, they find that YOLOv4 is faster and more accurate than the fastest and most accurate detectors and are situated on the Pareto optimality curve. Various approaches use GPUs of various architectures for inference time verification. Additionally, they provide a state-of-the-art detector that exceeds all competing options in terms of FPS and accuracy since the offered detector may be trained and used on a typical GPU with 8-16 GB of VRAM, allowing a variety of applications.

YOLOv7 exceeds all other object detectors in the range of 5 FPS to 160 FPS in terms of speed and accuracy, and on GPU V100, it has the highest accuracy of 56.8% AP of all real-time object detectors with 30 FPS or more. Both the transformer-based detector SWINL Cascade-Mask R-CNN (9.2 FPS A100, 53.9% AP) and the YOLOv7-E6 object detector exceed each other by 509% in speed and 2% in accuracy. Also, In terms of accuracy and speed, YOLOv7 surpasses YOLOR, YOLOX, Scaled-YOLOv4, YOLOv5, DETR, Deformable DETR, DINO-5scale-R50, ViT-Adapter-B, and a number of other object detectors [17]. From their paper, we learn that their focus will be on a few improved modules and optimization techniques that may increase the training cost while maintaining the same level of object detection accuracy as the suggested modules and optimization techniques are referred to as trainable bag-of-freebies. One of the contributions of their paper is that they develop a number of trainable bag-of-freebies techniques that can considerably increase real-time item identification accuracy without raising inference costs. Another contribution of this paper is that they discovered two additional problems for the evolution of object detection methods: how the re-parameterized module replaces the original module, and how the dynamic label assignment approach handles the assignment to various output layers. It also suggests ways to deal with the challenges brought on by these problems. Further from their paper, we learn that modern real-time object detectors may effectively have their parameters and computations reduced by roughly 40% and 50%, respectively, by using the approach we suggested. It also has a quicker inference speed and a greater detection accuracy. Here, In order to achieve a good trade-off between the number of network parameters, computation, inference speed, and accuracy, the model scaling method typically uses various scaling factors, such as resolution (size of the input image), depth (number of layers), width (number of channels), and stage (number of feature pyramid). Moreover, One approach to model scaling that is frequently used is network architecture search (NAS). Without establishing overly complex criteria, NAS may automatically scan the search area for appropriate scaling factors. In their paper, when compared to YOLOv5-N (r6.1), their approach is 127 frames per second quicker and 10.7% more accurate on AP. Additionally, YOLOv7 has 51.4% AP at a frame rate of 161 fps, but PPYOLOE-L only has 78 fps. YOLOv7 uses 41% fewer parameters than PPYOLOE-L overall. Moreover, Inference speed is comparable between YOLOv7-D6 and YOLOR-E6, but AP is improved by 0.8%. YOLOv7-E6E enhances AP by 0.3% while having an inference speed that is comparable to YOLOR-D6. Finally, During their research, they discovered the dynamic label assignment and replacement problems for the re-parameterized module. To resolve the issue they suggest the trainable bag-of-freebies technique to address the issue and improve object recognition precision.

2.2 State of the Arts in Face Recognition

Face recognition is a very active study area in computer vision, and it is frequently difficult to recognize faces in real time. To aid Alzheimer's patients in recognizing familiar faces, it is critical to detect faces using models with high face detection accuracy. A method for extracting face features from an attendance system was proposed in which the accuracy of the face recognition model does not degrade considerably as the number of faces increases [11]. A paper compares single and multi-modal system face detection technologies, while another proposed a deep cascaded multi-task CNN framework for joint face detection and alignment in real-time [2], [5].

A face recognition attendance system based on real-time video processing experimental data shows that the accuracy rate of the video face recognition system is up to 82% [11]. There were four different types of experiments done, including the accuracy rate of the face recognition system during actual check-in, the stability of the face recognition time and attendance attendance system, an analysis of the skip rate of the face recognition attendance system, interface settings for the face recognition attendance system, and real-time video processing [11]. Some approaches for accurately identifying faces have been suggested, including Face Feature Extraction by LDA, primary face recognition techniques, and video image recognition systems. One can see that although the identification and sign-in accuracy rate is 86% when the input faces are between 10 and 20, it drops to 80% when the faces are between 20 and 50. So, we can see that in the real-time video the accuracy does not drop greatly when the number of inputs increases.

Recent developments in single-modal and multimodal face identification are addressed in a research, which includes visual, three-dimensional, and infrared face recognition techniques. Single modal recognition is performed one at a time and has an accuracy rate of over 90%, however, there are some significant disadvantages to single modal identification, including posture, perspective, and expression changes [2]. To address this, they employ limited datasets or feature extraction to get a greater level of accuracy. As an advancement of the single-modal system, a multi-modal system is introduced, which employs multiple samples, units, and algorithms. Additionally, a fusion approach is used to recognize faces, such as a visual + infrared combination. Increased information availability is predicted to improve face recognition performance in this case since the combination of several facial modalities can overcome some recognition problems. When a combination of visual and three-dimensional recognition systems is employed the accuracy rate increases to 99 percent for numerous probes. Additionally, when Visual + 3-D+IR is employed, the recognition rate increases to 99.3 percent, outperforming baseline FRGC algorithms [2]. As seen in [11] as the number of input students increases, the accuracy rate decreases; thus, to avoid these difficulties, a multimodal recognition system may be used to get the highest possible accuracy.

Convolutional neural networks are unable to accurately recognize and align faces in real-world applications, hence a multi-task cascaded neural network architecture with primarily three phases was developed in [5]. The result is adjusted to output acceptable facial landmark locations after the first stage outputs candidate windows and the second stage rejects a significant number of non-face windows using a more complicated CNN. Online hard sample mining is used in this case to increase performance, and both face identification and alignment tests show a considerable improvement over the state-of-the-art method. The suggested system outperforms a number of difficult benchmarks, including FDDB, WIDER FACE, and AFLW for face alignment [5].

Chapter 3

Methodology

The diagram below shows the proposed Ambient assisted living system architecture. The system has two states i.e., initialization state & working state.

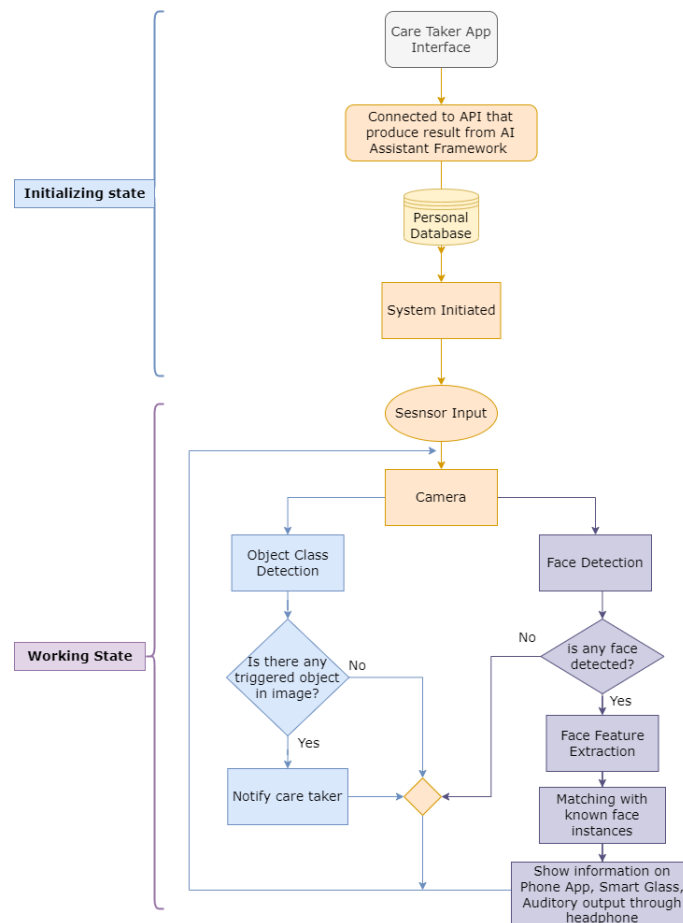


Figure 3.1: Top level overview of the proposed ambient assisted living system

In the initialization state, all the necessary data will be taken from the caretaker & it'll be stored in the database for future usage. After completing the initial state the system will be in the working state where it'll work as we proposed to assist the patients & the caretakers.

We'll set up a camera, and a smart band on the subject. A mobile app will be installed on the subject's smartphone and another mobile application will be installed on the caretaker's smartphone. Moreover, a cloud server will be deployed to process the data and notification service.

A camera will be installed on the subject which will be connected to a microcontroller with wifi accessibility such as ESP32. The microcontroller will be connected to the subject's smartphone's app using wifi. The app on the smartphone will listen to data from the microcontroller and will transfer the data to the cloud server using the POST/PUT/PATCH methods of the REST API.

The cloud server will listen to the data from the subject's smartphone application and will process the data using an appropriate processing system for a specific type of data. The server may receive the following types of data: image, GPS coordinates, and heart rate. Moreover, the caretaker will enter influential objects, known faces, and emergency contacts to initiate the system.

If the received data from the subject's application is image/video, then the data will be fed to two machine learning models for analysis. One of the models is object detection which will return the objects detected from the image or video data. If the returned objects match any of the objects which are pre-inputted to the database by the caretaker, then a push notification will be sent to the caretaker asking to take the necessary steps which were set by the caretaker. The other model, i.e., the face recognition model, will also analyze the image or video to find any known face. If it finds any known face, then it'll send the information available in the database about the person to the subject's app which the subject can use to recognize the person. If the face is recognized as unknown, then the server will store the information in the database and those face instances will be sent to the caretaker's app and the caretaker may choose to add information about them for future reference. Furthermore, whether any face is detected or not that data will be sent to the trigger detection function.

Suppose the received data from the subject's app is GPS coordinates. In that case, the information will be unicast to the caretaker application using an asynchronous WebSocket connection between the server and apps. So, the caretaker will be able to track the position of the subject. If the server finds any GPS coordinates which are stored as dangerous areas for the subject, then the server will immediately send a push notification to the caretaker's app.

A trigger detection function will weigh each input data to detect a trigger. If a trigger is detected, the function will send a push notification to the caretaker's app using the firebase push notification service. The caretaker will then acknowledge receiving the notification. If the caretaker does not acknowledge the notification in

5 minutes, then the system will generate an alarm on the caretaker's smartphone. Furthermore, if it detects no response from the caretaker then the system will generate automated calls to emergency contacts.

In addition to the above-mentioned activities, the caretaker will be able to monitor the database, add, update and delete information, and provide feedback to the system if the trigger generated was a valid trigger or not to improve the models and the system.

In our AI assistant framework, we have incorporated two features to assist Alzheimer’s patients.

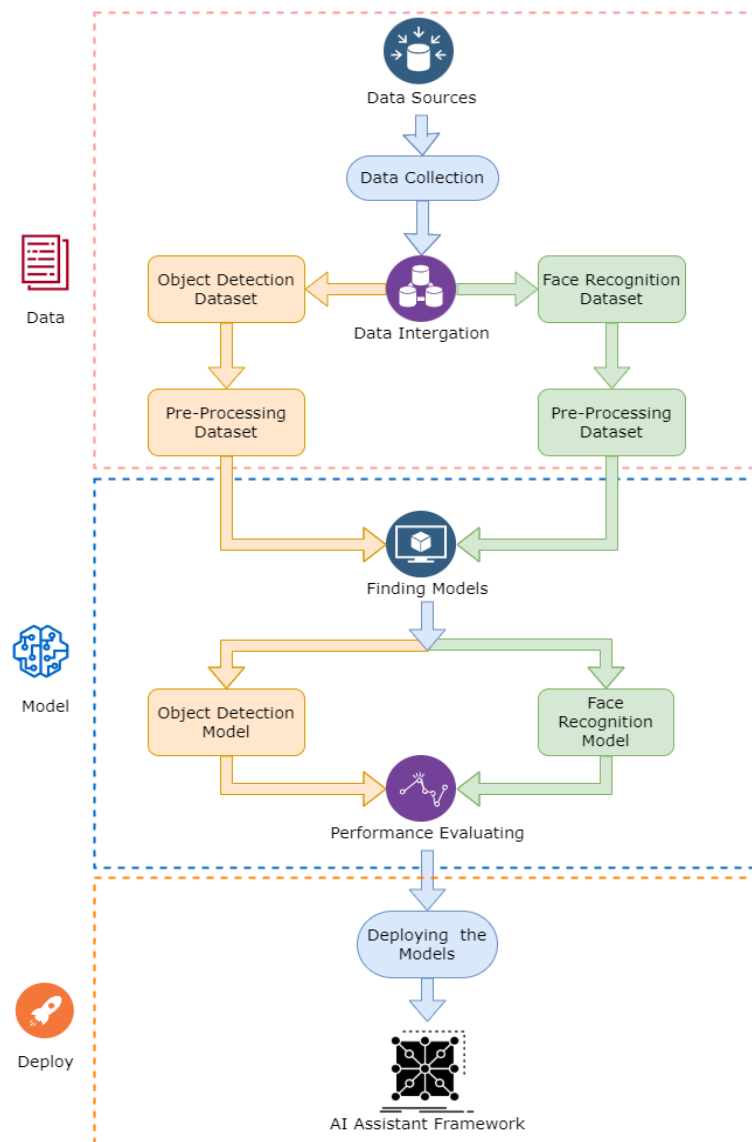


Figure 3.2: Proposed living system : Complete Block Diagram

The first feature involves detecting alarming or dangerous objects in front of Alzheimer’s patients. The latter enables the patients to recognize known people whom they have forgotten because of their miserable condition. After researching related works and papers thoroughly many older models for object detection and face recognition have been filtered out. This paper chose YOLOv4 and YOLOv7 for object detection and for face recognition pipeline MTCNN was selected for face detection, FaceNet for face embedding, and SVM and Random Forest for face classification. These models have been selected since they are highly relevant to our research.

3.1 Proposed Object Detection Framework

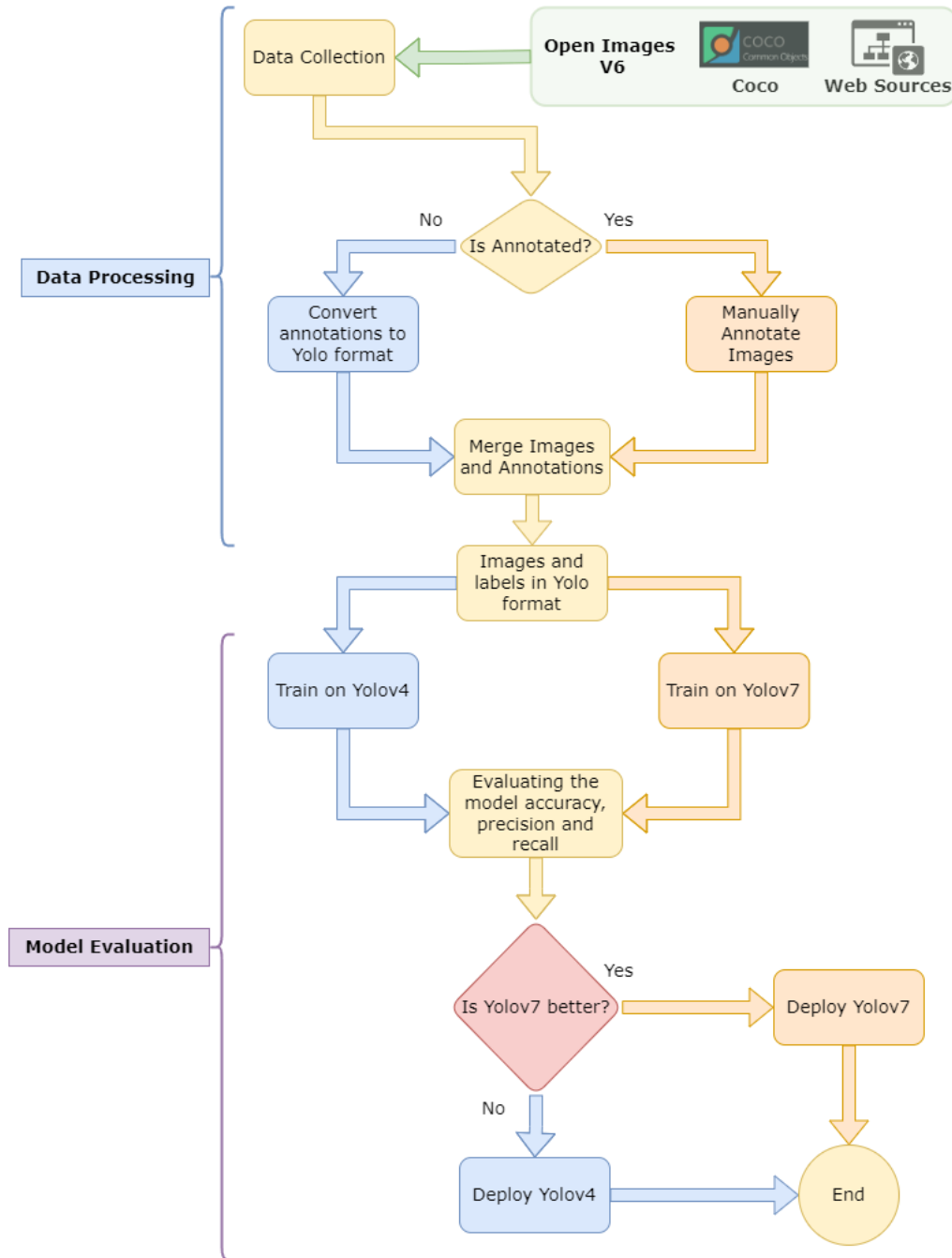


Figure 3.3: Work plan of the proposed object detection framework

3.1.1 Data Collection

During this phase of the research, we collected images of the wandering instigating objects from the internet, the MS COCO dataset, and Google Open Images.

3.1.1.1 Dataset

For this paper a custom dataset of 13 object classes has been created which consists of 70,529 annotated objects in 32,162 images. Handbag, hat, watch, backpack, knife, suitcase, scissors, fork, umbrella, key, shoe, door, and doormat are the identified wandering causing objects for Alzheimer’s patients. We downloaded key, door, doormat, and shoe images from google and hand-annotated them. Handbag, hat, watch, backpack, knife, suitcase, and scissors datasets were collected from Open Images v6. We also collected handbag, umbrella, backpack, knife, suitcase, fork, and scissors from COCO, 2017 dataset. We created our custom dataset by merging all the above-mentioned data subsets.

3.1.1.2 Data Processing

At first we downloaded annotations from COCO 2017 dataset and filtered annotations that match our class’s requirements. Then, we downloaded the annotations for Open Images v6 and dropped all the annotations that are not in our required dataset. Then we downloaded unique images for the filtered annotations. After that, we converted annotations from COCO and Open Images style annotations to YOLO style annotations. For 4 of our required classes i.e., Key, Door, Doormat, and Shoe we did not get any dataset available in any public dataset so we had to manually prepare the dataset for these 4 classes. To prepare this dataset we have downloaded images from various stock image libraries, google image search, etc. Then, we annotated the downloaded images in YOLO annotation format.

In YOLO style annotation format, annotations are saved in a text file named the same as the image name which includes annotations in each line in the following format.

class_id	x_center	y_center	width	height
0	0.45	0.46	0.69	0.45

After preparing the dataset in YOLO format, we have then created train-validation splits in a ratio of 70:30 by creating train.txt and val.txt files which contains image paths for respective splits.

3.1.2 Implementation and Training

3.1.2.1 YOLOv4

The AP and FPS improvements in YOLOv4 are well-known. In YOLOv4, real-time object detection is given priority, and training is carried out on a single CPU. On a Tesla V100, YOLOv4 achieved cutting-edge performance with 43.5 percent speed (AP) at 65 frames per second (FPS) on the COCO dataset. As, YOLO v4 is a single stage object detector so it detect objects more accurate and faster than R-CNN and fast R-CNN which are the two stage object detector. The architecture of YOLO v4 is given below-

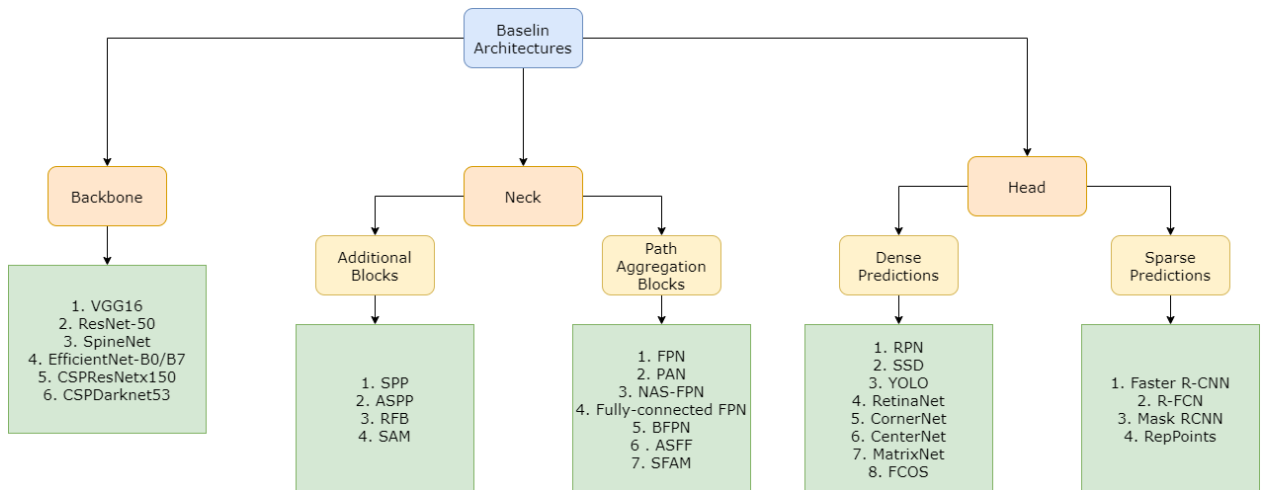


Figure 3.4: Basic YOLOv4 architecture [9]

The figure shows that YOLO v4 consists of three components, which are:

- **Backbone**

YOLOv4 essentially bases its architecture on one of the three types. the following three feature extractor models are :

- CSPDarknet53
- CSPResNext50
- EfficientNet-B3

Here, we will discuss ‘CSPDarknet53’ as It takes advantage of DarkNet-53, a Backbone that is frequently used for object identification. SPDarknet53 is specifically used by YOLOv4 as its foundation. It employs a CSPNet method that divides a feature map formed of DenseBlocks into two parts and then combines them using a cross-stage hierarchy. By avoiding the base layer, the previous section becomes the initial input for the next transition layer. Later parts of the base layer are applied employing DenseBlock. This approach reduces the computational complexity. It performs better than other ResNet models because of its improved accuracy.

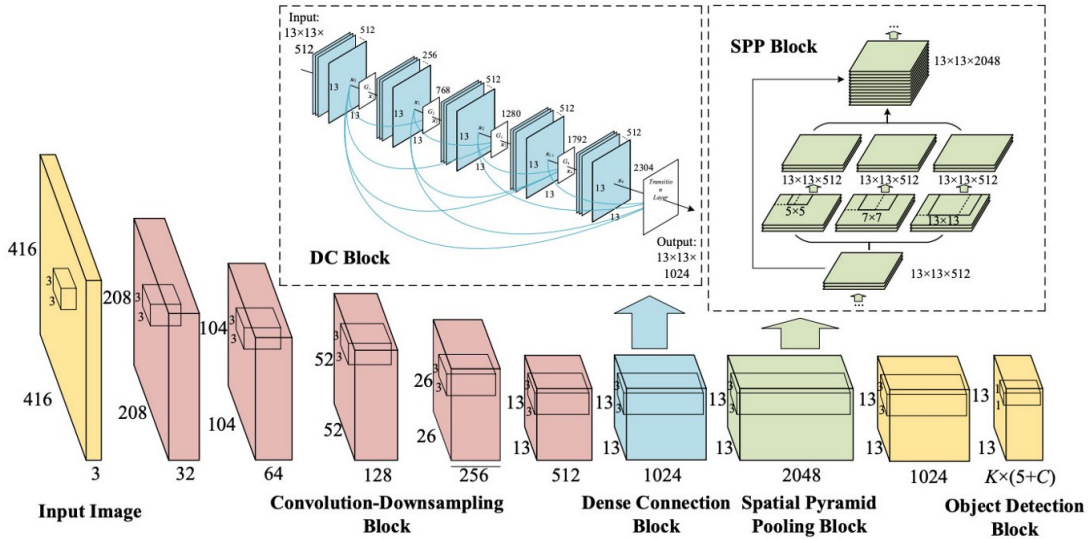


Figure 3.5: CSPDarknet53 [10]

- **Neck**

In this stage, features that were initially in the backbone are accumulated in preparation for subsequently feeding them to the head for processing. YOLOv4 offers a number of options. Here we will discuss two of the options which are “SPP” and “PAN”.

- **PAN (Path Aggregation Network):** By preserving correct semantic data or information that assists in the appropriate localization of picture sections for mask information, it functions as a neck in the YOLOv4 method and improves segmentation. There are two properties of PAN which are -

- * **Bottom-up Path Augmentation:** The spatial resolution of a picture decreases as it moves through several layers of a neural network, while the complexity of the feature rises at the same time. This makes it more difficult for high-level features to recognize pixel-level masks. When merging multi-scale characteristics in YOLOv3 FPN uses a top-down method while maintaining semantic localization data. The construction of large items becomes complex and time-consuming because more levels must be traversed by the information.

On the other hand, PANet offers more bottom-up and top-down routes than FPN. Using lateral connections from lower to top levels significantly decreases complexity. It is less difficult and lengthy because it lasts no more than 10 years.

- * **Adaptive Feature Pooling:** In the past, methods like Mask-RCNN used characteristics from just one stage to make predictions. The features from higher layers were uprooted using RIO Align Pooling. Results were occasionally anticipated incorrectly. In order to

solve this issue, PANet collected features from every layer and used Align Pooling to uproot features on each feature map.

- **SPP(Spatial Pyramid Pooling)** : SPP often referred to as Spatial Pyramid Pooling, is a methodology for gathering both fine and large-scale data. Applied on a sliding kernel with dimensions of 11, 55, 99, and 1313. To obtain the outcome, maps with different kernel sizes are merged. The receptive field improvement is the major benefit of SPP. Regardless of the size of the feature map, fixed-size features are created via spatial pyramid pooling.

The benefit of SPP is that it produces fixed results regardless of the size of the input feature map. Additionally, it uses multi-level spatial bins, making it resistant to object deformations. As it can extract pool properties at different scales, SPP is adaptable in terms of input scales.

- **Head**

In YOLOv4, the head's primary goal is to conduct prediction, which involves classifying data and regressing bounding box boundaries. The YOLOv3 head is employed. It offers details about the bounding box coordinates (x, y, h, w). It comprises the label's width, height, center, and prediction score. Each anchor box can be equipped with a YOLOv4 head.

Further, YOLO v4 uses Bag of Freebies (BoF) and Bag of Specials (BoS) for Backbone and Detector. Here, "BoF" approaches may be described as ways to reduce the cost of training or strategy in order to improve the model's accuracy. Improvements are evident in data management and data augmentation as a result of the model's improved performance without reducing inference time delay. On the other hand, "BoS" includes coherence and post-processing models, which cause a little increase in the modules' inference costs. Nevertheless, the detector's performance and accuracy are enhanced by this adjustment.

YOLOv4 is one of the most commonly used real-time object detection models. After creating the dataset a cfg file has been created where the information about batch, number of channels, max batch, and steps are stored. 3000 max batches were used for each object, so for 13 objects total of 39000 batches were used. There will be 80% to 90% steps for each batch. So, our steps are around 31200 and 35100 for 39000 batches. The image size was 416 by 416. And batch size was 64 and the subdivision was 16.

We stopped training our model after around 15200 iterations when we found out that map@0.5 was no more changing. We have got avg loss of 2.17 at 15200th iterations.

3.1.2.2 YOLOv7

An approach using a neural network called YOLOv7 makes object detection quicker and more accurate. The image is divided into a number of fixed-size grids using YOLOv7. Every grid is in charge of locating items around it. Each grid cell predicts the bounding boxes and confidence ratings for each box. These confidence ratings show the model's level of certainty that the box contains the specified object as well as how correctly it makes that assertion. If there is no item within the box, then the confidence score is given as 0.

This method considerably saves computation by maintaining both recognition and detection simultaneously in cells from the picture, but it produces a lot of duplicate predictions since several cells may forecast the same item with different bounding boxes.

YOLOv7 employs the idea of non-maximal suppression to address the above-given problem. The bounding boxes with lesser probability scores are either muted or ignored in this method. YOLOv7 chooses a bounding box with the best probability score in order to do this. The bounding boxes that have the highest Intersection over Union (IoU) with the present high probability bounding box are suppressed. Up until we have the final bounding box with precise item recognition, the processes are repeated.

The YOLOv7 algorithm combines a number of bag-of-freebies techniques to significantly improve detection accuracy while requiring fewer processing resources. The approach uses around 40% of the function parameters without sacrificing accuracy and produces better results than any existing object detection models. The YOLOv7 team has developed an extremely efficient architecture. In order to iteratively increase the network's learning capacity without changing the starting gradient, they extended the efficient layer aggregation network (E-ELAN), which only alters the compute block without affecting the transition layer.

For training YOLOv7 we have created a yaml configuration file by defining a number of classes and train-validation splits location and class names. Then, in another yaml file, we have defined our hyperparameters such as initial learning rate, weight decay, etc. We have used an initial learning rate of 0.1 and weight decay of 0.005. We have used image sizes 640 by 640 and batch size 16. For faster training, we have used transfer learning on pre-trained weights. We stopped our training at the 62nd epoch when we found out that $\text{map}@0.5$ and $\text{map}@0.5:0.95$ was not changing.

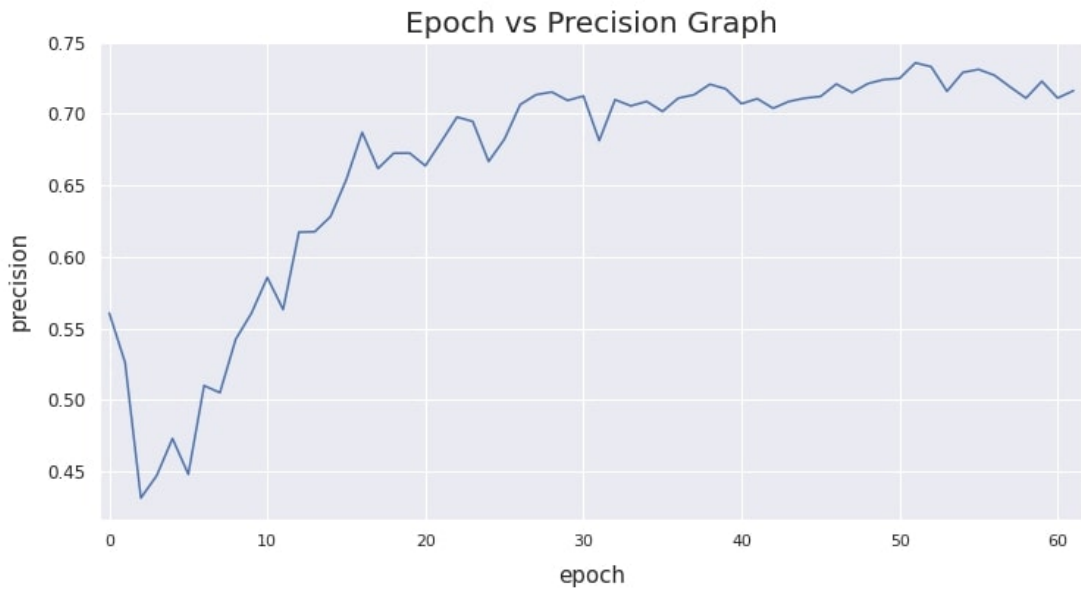


Figure 3.6: Epoch vs Precision Graph

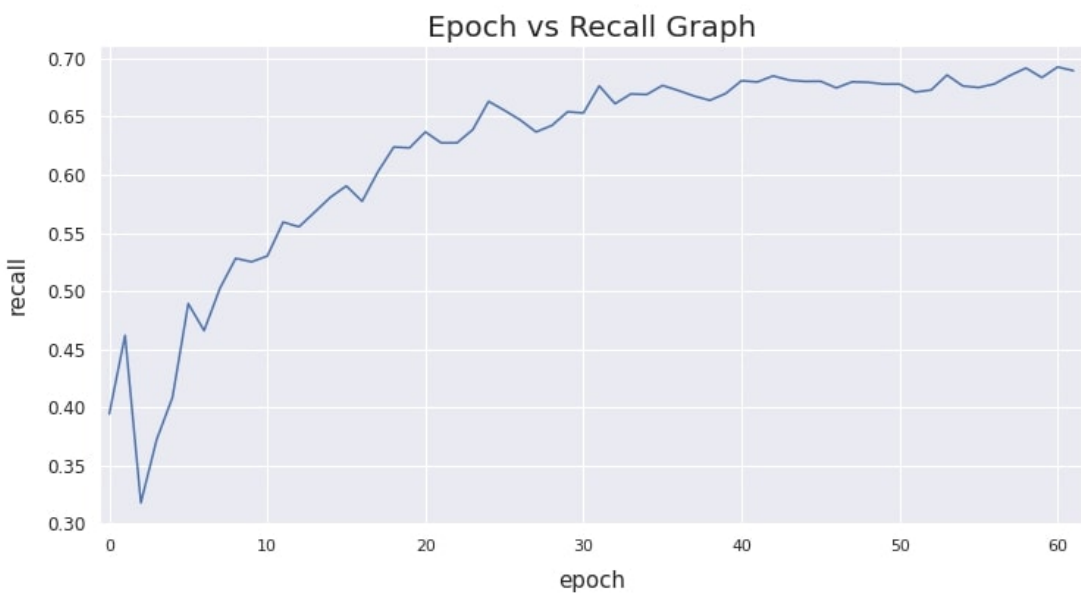


Figure 3.7: Epoch vs Recall Graph

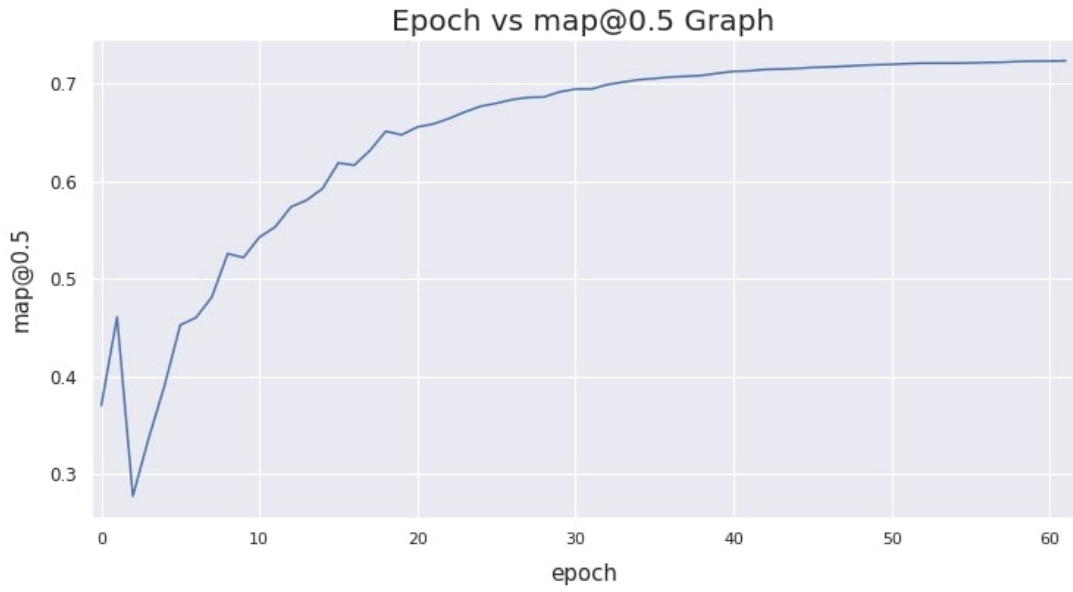


Figure 3.8: Epoch vs mAP@0.5 Graph

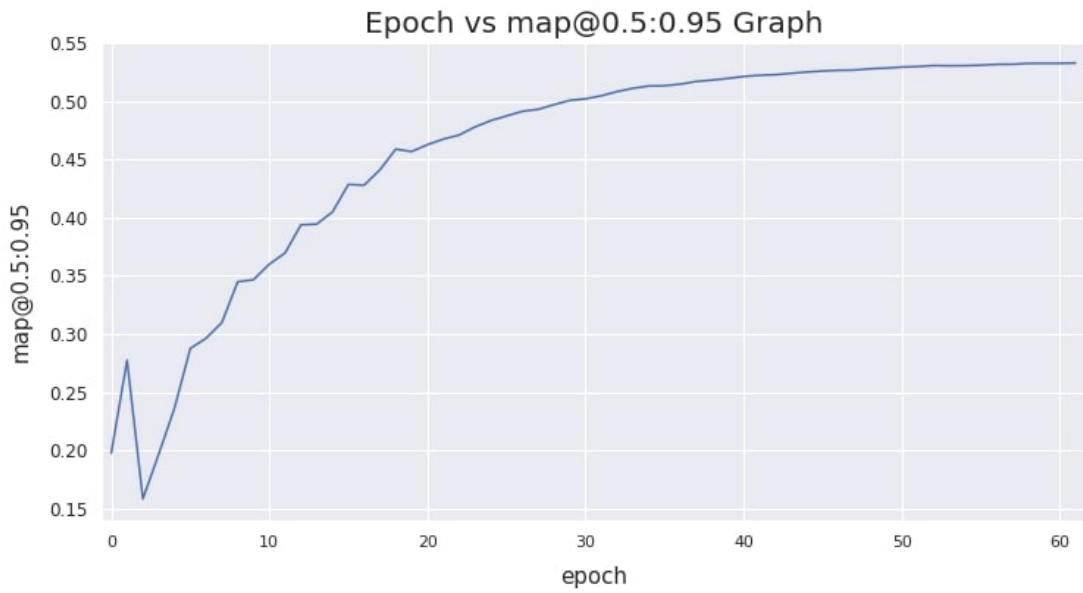


Figure 3.9: Epoch vs mAP@0.5:0.95 Graph

3.2 Proposed Face Recognition Framework

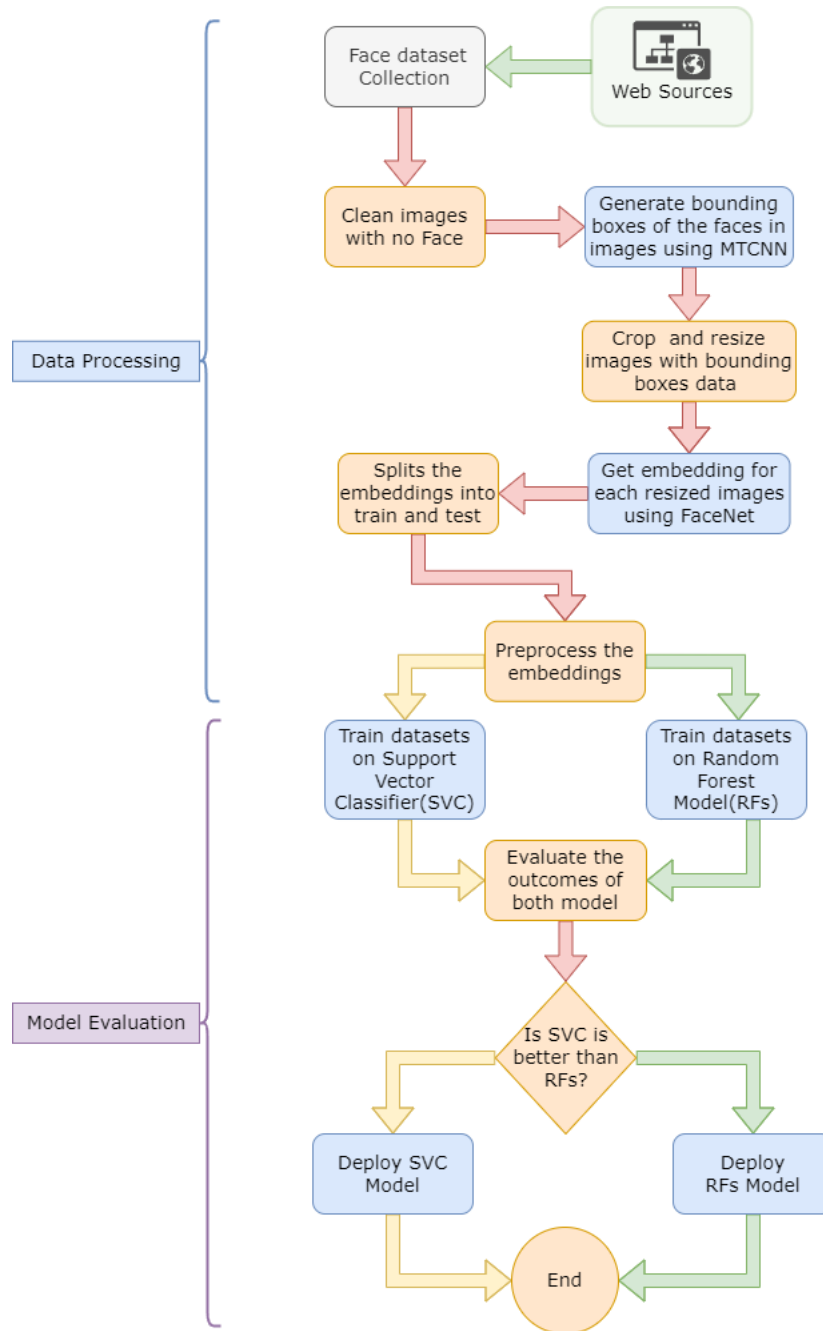


Figure 3.10: Work plan of the proposed face recognition framework

3.2.1 Data Collection

Custom dataset is prepared and collected by web scraping and high-resolution pictures from google images for classifying known faces containing 187 classes of data. A virtual Conda environment is created and the `simple_image_download` module of python is used to collect images from google.

3.2.1.1 Dataset

The customized face dataset contains a list of images accumulating famous celebrities like actors, politicians, sportsmen, ramp models, influencers, etc having 187 classes of data and almost 30 images per class adding up to 5728 image datasets. The images containing faces are in different alignments and orientations to train and test faces appropriately in real-time. It is ensured that the dataset containing images is of high resolution and the face is clearly apparent.

3.2.1.2 Data Processing

After collecting images from google, the dataset is cleaned and preprocessed by removing the duplicate, multiple people's faces, and unrelated, inconsistent persons from the subfolders renamed as the names of the individuals. The subfolders are further divided into train and test folders containing 70 percent of images(3874) in training data and 30%(1836) in the testing dataset. The preprocessed dataset is fed into the input of the face detection models for training.

The dataset is further divided newly and built for one shot learning specially, with 1 picture in each subfolder in the training set and the rest of the photos in the testing dataset in each subdirectory of the entities, to test for faster training and testing of the suggested models.

3.2.2 Implementation and Training

Face recognition module consists of 3 stages, that are face detection, face embedding, and face classification stage. Here, we will briefly discuss the different stages of the face recognition module and how we have detected, embedded and classified faces using the models described in the modules.

3.2.2.1 Face detection using MTCNN

The face detection module will detect the faces at first and create the bounding boxes. Those bounding boxes will be then fed as input to the face recognition stage. For this purpose, we have used MTCNN(MultiTask Cascaded Convolutional Neural Network).[5] It is a framework developed to detect the face and the face alignment. It has three convolutional networks which are, P-network (Proposal network), R-network (Refine network), and O-network (Output network). At first, when it detects an image it resizes that into various sizes so that it can detect the different sizes of faces. After resizing it, P-network does the first detection by scanning the images but it gives max false results as the threshold is low. Also, P-network is mainly used for obtaining the candidate windows and their bounding box regression vectors. After that, the candidates from P-network go to the R-network to get more refinement, it filters the given result to get a precise bounding box. Finally, it goes to O-network to perform the last clarifying of the bounding box to get the final output. By this, we can detect the face and also find the precise bounding box. Moreover, it detects facial landmarks such as the eyes, nose, and two sides of the mouth. It can achieve 13 fps on FHD videos and can go up to

45 fps on the rescaled video. So, we can say that the three primary outputs of this network are facial landmark localization, bounding box regression, and face/non-face classification.

- **Face/non-face classification:** It is a binary classification problem that uses cross-entropy loss

$$L_i^{det} = -(y_i^{det} \log(p_i) + (1 - y_i^{det})(1 - \log(p_i))) \quad (3.1)$$

Here, p_i is probability produced by network whereas y_i^{det} is ground-truth level whose value is between 0 to 1.

- **Bounding box window:** As, there can be too many candidate windows so, for each of them the offset can be between it and the nearest ground truth. For each candidate, we use Euclidean loss.

$$L_i^{box} = \|\hat{y}_i^{box} - y_i^{box}\|_2^2 \quad (3.2)$$

Here, \hat{y}_i^{box} is the target obtained from the network, and y_i^{box} is the ground truth coordinate. (3.3)

- **Facial Landmark Localization:** It is calculated by using the same formula which is Euclidean loss.

$$L_i^{landmark} = \|\hat{y}_i^{landmark} - y_i^{landmark}\|_2^2 \quad (3.4)$$

Here, $\hat{y}_i^{landmark}$ is target obtained from network and $y_i^{landmark}$ is ground truth coordinate. (3.5)

We have used MTCNN to detect our faces from the images. When it detect any faces we collect the co ordinates of bounding boxes given by MTCNN. By using. the bounding boxes we have extracted the images than resize it into 160 by 160 matrix.

3.2.2.2 Face Embedding using FaceNet

In this stage, our Face Recognition module will identify faces with help of the FaceNet model [4]. It is a one-shot model proposed by google researchers. It has achieved state-of-the-art in the datasets “Labeled Faces in the Wild (LFW)” and “Youtube Face Database”. This model learns high-quality face mapping from the images using deep learning models such as The Zeiler & Fergus style networks and the recent Inception type networks. To train the model it uses triplets of aligned matching faces and non-matching faces. A triplet basically consists of an anchor image, a matching image to the anchor image, and a non-matching image of the anchor image. The Triplet Loss process in the training minimizes the distance between the anchor and the positive/matching image while maximizing the distance between the anchor and the negative/non-matching image as shown in the demonstration below.

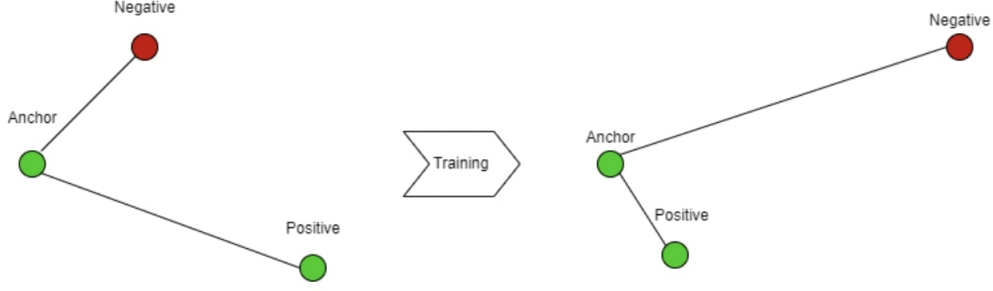


Figure 3.11: Triplet Loss

An image is embedded into d -dimensional Euclidean spaces that is represented by $f(x) \in \mathcal{R}$. Also, A constraint is added that the embedding has to live on the d -dimensional hypersphere, representing by $\|f(x)\|_2 = 1$. Motivated by k nearest neighbor classifier, here it ensures that the distance between an anchor image, x_i^a of a specific person and all images of the same person, x_i^p must be less than the distance between all images of other persons, x_i^n .

This constraint can be shown as,

$$\|x_i^a - x_i^p\|_2^2, \forall (x_i^a, x_i^p, x_i^n) \in \tau \quad (3.6)$$

Here, α is a distance between positive and negative pairs, τ is the set of all possible triplets, and N is the cardinality of the training set.

Then, the normalized loss,

$$\sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+ \quad (3.7)$$

When generating all the triplets, a good portion of the triplets satisfies the given constraint. As a result, they don't contribute to the training, rather they make the convergence slower as they still have to be passed through the network. So, it is very important to use hard triplets that don't satisfy the constraints, therefore, have the potential to improve the model.

In our proposed face detection framework, after creating bounding box co-ordinates using pixel values of the dataset by MTCNN, the pixel values of the training and testing dataset is given as input into the keras FaceNet model. The face embeddings of 128×128 co-ordinate vector matrices of the images which contains the essential facial features of the corresponding images. Separate embeddings for training and testing dataset images are extracted to be used to classify the known faces.

3.2.2.3 Face classification

Once we get the face embedding vector or feature extraction, we can use those vectors to find the similarity between two faces. Basically, the distance between two face embedding vectors expresses the similarity between two faces. If the distance

between two faces is too low then they are similar to each other and vice versa. After the feature is extracted via face embeddings, we can use classifications like SVC(Support Vector Classifier) and Random Forest to classify the faces.

Face Classification Using Random Forest

Random Forest, a supervised learning algorithm, is also used for classifying faces that utilize decision trees based on varieties of samples and takes a majority vote for classifying the final face recognition as output. It bags different training samples known as the Bootstrap Aggregation technique in which random samples of data subsets are created and the final output is based on majority voting known as the aggregation process. Each subset of data is trained independently and individual decision trees are constructed for each sample to generate outputs lastly via majority voting or averaging the final class is determined. Because of its diversity, immunity, parallelization, and stability, it is considered one of the vigorous classification algorithms.

Using Random Forest Classifier, the model is fit using parameters as `n_estimators` of 100, `criterion` as Gini, and `min_samples_split` of 2. The faces are classified as having training accuracy of 99.77% and testing accuracy of 95.153%.

Face Classification using SVC(Support Vector Classifier)

Due to high accuracy and less computational power, Support Vector Classifier(SVC) is utilized for classifying faces which determines a hyperplane in N-dimensional space(N denoting the number of features) that uniquely identifies the data points. The data points are split and the optimal hyperplane is distinguished by having the maximum margin between classes' data points. The data points on either side of the plane attribute to different classes. Support vectors are primarily data points closer to the hyperplane. The classes are uniquely identified by the output of the linear function of either 1 or -1, therefore the reinforcement range lies between the margin range of values [1,-1]. The loss function that helps maximize the margin between the data points is the hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else } c(x, y, f(x)) = (1 - y * f(x))_+ \end{cases} \quad (3.8)$$

The regularization parameter is added to the cost function to create an equilibrium between margin and maximization, the weights are updated by taking partial derivatives with respect to weights and the regularization parameter is hyper-tuned for better classification.

Support Vector Classifier(SVC) is utilized to detect the faces by first normalizing the training and testing data using l2 normalization techniques. The training and testing classes are label encoded so that the model identifies the classes efficiently. The Linear SVC model is trained and tested for face detection obtaining a training accuracy of 99.568% and tested for 97.440%.

Chapter 4

Result and Findings

4.1 Evaluation Metrics

For calculating precision, recall, F1 score and mAP etc. we need the following metrics,

- **True Positives (TP):** The model predicted the positive class correctly.
- **True Negatives (TN):** The model predicted the negative class correctly.
- **False Positives (FP):** The model predicted the positive class incorrectly.
- **False Negatives (FN):** The model predicted the negative class incorrectly.

Precision is a popular metric for assessing the performance of a machine learning model. It reflects the accuracy of the model's positive prediction. We use the following formula to determine precision:

$$precision = \frac{TP}{TP + FP} \quad (4.1)$$

Recall is the number of correct positive class predictions produced out of all correct positive cases in the dataset. We use the following formula to calculate recall:

$$recall = \frac{TP}{TP + FN} \quad (4.2)$$

The harmonic mean of recall and precision is known as the F1-score. The following formula combines recall and precision into a single number,

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (4.3)$$

Moreover, The accuracy of object detection is determined by IoU. We utilised an IoU threshold of 0.5 for this study, which implies that anything below it counts as a FN and anything beyond it counts as a TP. In essence, its calculation is an overlap ratio between the ground truth table and the anticipated bounding box. Further, The degree of overlap between two bounding boxes that forecast the same class for various grid cells is determined using the IoU. The two independent bounding box predictions are not ignored by the non-max suppression algorithm if there is not enough overlap, say IoU 0.5, as there is a good chance that they are two different

objects belonging to the same class. Pick the bounding box set with the highest class probability if the IoU is high, which indicates that overlap is high (so one set of bounding boxes is discarded by the non-max suppression algorithm).

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} \quad (4.4)$$

Finally, A metric called Mean Average Precision (mAP) is used to assess object detection algorithms like Fast R-CNN, YOLO, Mask R-CNN, etc. Recall values between 0 and 1 are used to calculate the average precision (AP) values. Confusion Matrix, Recall, Precision and IoU are the criteria upon which mAP is based on. The formula below is used to determine mAP-

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4.5)$$

Moreover, The mAP encompasses both FP and FN and incorporates the trade-off between recall and precision. Due to this characteristic, the majority of detecting applications may use mAP as a measure. In addition, other criteria are employed for assessing the accuracy of models at various object sizes.

The sensitivity is the proportion of TP that is correctly predicted by the model. It is calculated using the following formula

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (4.6)$$

The specificity is the proportion of TN that is correctly predicted by the model. It is calculated using the following formula

$$\text{specificity} = \frac{TN}{TN + FP} \quad (4.7)$$

Accuracy denotes the proportion of predictions that were made by the model. It is calculated using the following formula

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.8)$$

4.2 Result Analysis and Comparison

4.2.1 Object Detection

From the above tables, we can see that average precision for most of the object classes in YOLOv7 are higher than YOLOv4. YOLOv4 performed better in manually annotated objects. For example, YOLOv7 showed 37.5 % better average precision in detecting Handbags, YOLOv4 showed 17% better average precision in detecting Doors. However, In YOLOv4 minimum average precision is 40.30% where as in YOLOv7 it is 55.4%. Moreover, In YOLOv7 variance of average precision is lower

class	Average Precision (YOLOv4)	Average Precision (YOLOv7)
handbag	40.30%	55.4%
hat	59.83%	64.8%
watch	73.77%	82.5%
backpack	53.13%	65.6%
knife	52.06%	64.2%
suitcase	79.93%	68.0%
scissors	60.15%	74.1%
fork	61.87%	76.0%
umbrella	60.45%	68.5%
key	82.11%	80.4%
shoe	87.96%	77.1%
doormat	89.60%	72.2%
door	93.09%	79.0%

Table 4.1: Class Based Performance Comparison

	YOLOv4	YOLOv7
precision(%)	62	71.4
recall(%)	60	69
f1-score(%)	61	70.2
mAP@.5(%)	68.79	72.3
mAP@.5:.95(%)	-	53.31

Table 4.2: Overall Performance Comparison

than that of YOLOv4.

From the overall test result it is found that, YOLOv7 shows around 15% higher average precision than YOLOv4 in average precision. Moreover for both recall and F1-score, YOLOv7 shows approximately 15% higher scores than YOLOv4. mAP@0.5 in YOLOv7 is 5.1% better than YOLOv4 and YOLOv7 has achieved 53.31% score in mAP@.5:.95.

From the analysis above, we found that YOLOv7 outperformed YOLOv4 in almost all metrics. As a result we are proposing to use YOLOv7 for our Trigger Object Detection module.

4.2.2 Face Recognition

After gathering the data, we used MTCNN for face recognition and FaceNet for embeddings to prepare it for training the Support Vector Classifier (SVC) and Random Forest (RF) models. On the training data, the Random Forest model produced an accuracy of 99.77%, precision of 99.52%, and recall of 98.76%, greatly exceeding the SVC model’s accuracy of 99.57%, precision of 99.56%, and recall of 99.59%. On the test dataset, however, Support Vector Classifier surpassed the other techniques, with an accuracy of 97.44%, precision of 97.05%, and recall of 97.74%, whereas RFs

	Support Vector Classifier(SVC)		Random Forest(RFs)	
	Train	Test	Train	Test
Accuracy (%)	99.57	97.44	99.77	95.15
Precision(%)	99.56	98.05	99.52	95.97
Recall(%)	99.59	97.22	98.76	94.35
Specitivity(%)	99.59	97.22	98.76	94.35
F1-score(%)	99.57	97.47	99.14	94.71

Table 4.3: Comparison between SVC and RFs models

models had an accuracy of 95.15%, precision of 95.97%, and recall of 94.34%.

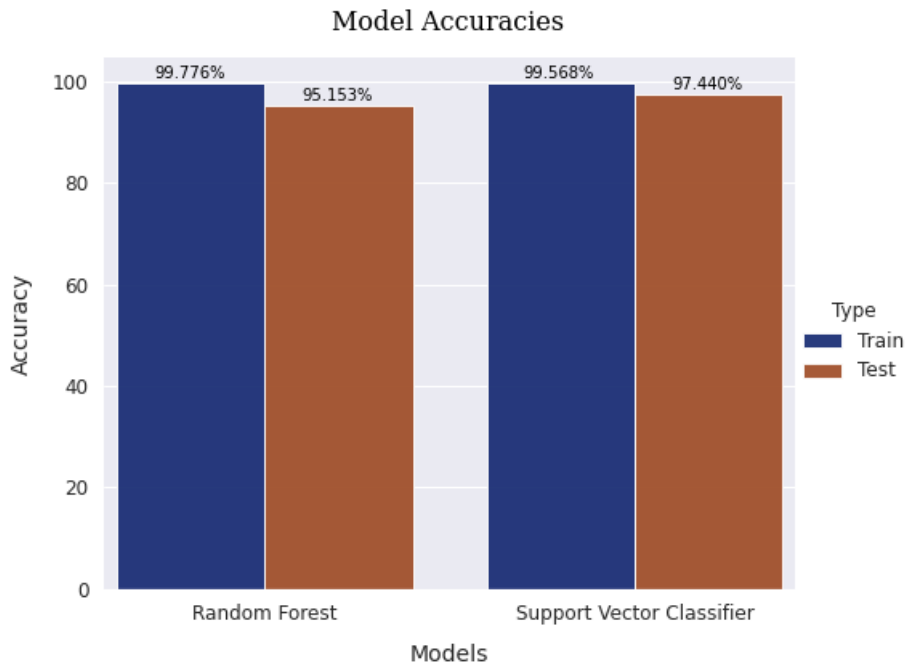


Figure 4.1: Model accuracy comparison of SVC and RFs

We selected Support Vector Classifier to incorporate with our face recognition framework for face classification since it beat the SVC model on the test dataset and achieved an accuracy of 97.55%. As a result, our face recognition system now includes MTCNN for face detection and producing bounding box for face cropping, FaceNet for face embeddings, and Support Vector Classifier for face classification.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

People often confuse AD with normal aging when it is not the same. Many studies have been conducted to detect AD in its early stages but very little research has been done to build an all-inclusive system to serve AD patients & their caregivers. The aim of this study was to fill this void by creating a bespoke system for every user that will cater to their needs. The wandering issue was addressed by implementing a state-of-the-art object detection model, YOLOv7 for detecting wandering inducing objects. We found that YOLOv7 has shown 15% better performance in almost all evaluation metrics than YOLOv4. To mitigate the decline in facial recognition problems a novel face recognition framework was proposed. In this framework, a face was detected by using MTCNN, then face embeddings were extracted using FaceNet, and lastly, the face was classified or identified using SVC and we achieved 97.44% accuracy in test set. Hopefully, through this research, it will be able possible to offer the patients & their caregivers a better quality of life than before.

5.2 Future Work

In this study, a novel AI Assistant framework was proposed for AD patients but just like any other work there are scopes for further improvements. An emotion and hallucination detection feature can be added that can monitor the patient's emotional state at all times. The system will try to detect the emotional state of the patients using their voice & speech patterns and heart rate reading. If it detects that they are experiencing any negative emotional change then it'll notify the caretaker about it and also play calming music through the patient's headphones if they are wearing any. It'll also look for patients' frequently used words when they are hallucinating. These words will be acquired from the caretakers at the initial setup of the system. When the system detects that the patient has said any of the words it will then cross-check with the current emotional state of the patient & also with the heart rate sensor reading. If all these readings indicate that the patient is hallucinating then the system will notify the caretaker about the possible hallucination of the AD patient. Lastly, we will update our models in the future based on the advancement of technology in the related fields to serve our target audience better

Bibliography

- [1] J. Tung, H. Snyder, J. Hoey, A. Mihailidis, M. Carrillo, and J. Favela, “Everyday patient-care technologies for alzheimer’s disease,” *IEEE Pervasive Computing*, vol. 12, no. 4, pp. 80–83, 2013. DOI: 10.1109/MPRV.2013.63.
- [2] H. Zhou, A. Mian, L. Wei, D. Creighton, M. Hossny, and S. Nahavandi, “Recent advances on singlemodal and multimodal face recognition: A survey,” *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 6, pp. 701–716, 2014. DOI: 10.1109/THMS.2014.2340578.
- [3] R. Alattas and B. D. Barkana, “A comparative study of brain volume changes in alzheimer’s disease using mri scans,” in *2015 Long Island Systems, Applications and Technology*, 2015, pp. 1–6. DOI: 10.1109/LISAT.2015.7160197.
- [4] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.
- [5] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. DOI: 10.1109/LSP.2016.2603342.
- [6] G. Gupta, A. Gupta, V. Jaiswal, and M. D. Ansari, “A review and analysis of mobile health applications for alzheimer patients and caregivers,” in *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 2018, pp. 171–175. DOI: 10.1109/PDGC.2018.8745995.
- [7] L. Aziz, M. S. B. Haji Salam, U. U. Sheikh, and S. Ayub, “Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review,” *IEEE Access*, vol. 8, pp. 170 461–170 495, 2020. DOI: 10.1109/ACCESS.2020.3021508.
- [8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, *Yolov4: Optimal speed and accuracy of object detection*, 2020. DOI: 10.48550/ARXIV.2004.10934. [Online]. Available: <https://arxiv.org/abs/2004.10934>.
- [9] Y. V. 0. Introduction, *Yolov4*, <https://medium.com/visionwizard/yolov4-version-0-introduction-90514b413ccf>, Last accessed on 2022-09-20, May 2020.
- [10] Y. V. 3. P. Workflow, *Yolov4 network pathway for input 416x416*, <https://medium.com/visionwizard/yolov4-version-3-proposed-workflow-e4fa175b902>, Last accessed on 2022-09-20, May 2020.
- [11] H. Yang and X. Han, “Face recognition attendance system based on real-time video processing,” *IEEE Access*, vol. 8, pp. 159 143–159 150, 2020. DOI: 10.1109/ACCESS.2020.3007205.

- [12] C. Zhang and J. Kim, “Video object detection with two-path convolutional lstm pyramid,” *IEEE Access*, vol. 8, pp. 151 681–151 691, 2020. DOI: 10.1109/ACCESS.2020.3017411.
- [13] X. Chen, H. Li, Q. Wu, K. N. Ngan, and L. Xu, “High-quality r-cnn object detection using multi-path detection calibration network,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 2, pp. 715–727, 2021. DOI: 10.1109/TCSVT.2020.2987465.
- [14] National Health Service, *Alzheimer’s disease*, <https://www.nhs.uk/conditions/alzheimers-disease/>, Last accessed on 2022-09-20, 2021.
- [15] World Health Organisation, *Dementia*, <https://www.who.int/news-room/fact-sheets/detail/dementia>, Last accessed on 2022-09-20, 2021.
- [16] J. Yu, *Personal Cognitive Assistive Robot for Alzheimer’s Disease Patients’ Practicing of COVID-19 Hygiene Measures*. Rochester Institute of Technology, 2021.
- [17] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, 2022. DOI: 10.48550/ARXIV.2207.02696. [Online]. Available: <https://arxiv.org/abs/2207.02696>.