EXPERT SYSTEM FOR VISUALLY CHALLENGED USING AI

by

Adarsh Verma (2147102)
Amber Ujjwal Linda (2147104)
Ankur Sharma(2147105)

Under the guidance of

Dr. THIRUNAVUKKARASU V

A Project report submitted in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications of CHRIST (Deemed to be University)

Dec - 2022

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CERTIFICATE

This is to certify that the report titled ***EXPERT SYSTEM FOR VISUALLY CHALLENGED STUDENTS USING ARTIFICIAL INTELEGENCE*** *is a bonafide record of work done by* ***Adarsh Verma (2147102), Amber Ujjwal Linda (2147104) , Ankur Sharma (2147105)*** *of CHRIST (Deemed to be University), Bengaluru, in partial fulfillment of the requirements of V Trimester MCA during the year 2022.*

Head of the Department **Faculty In charge**

Valued-by:	Faculty In-charge:	Dr.THIRUNAVUKKAR ASU
	Register Number(s) :	2147105 2147104 2147102
	Examination Centre :	CHRIST (Deemed to be University)
	Date :	17 December 2022

ACKNOWLEDGEMENTS

We would like to express our gratitude to our Vice-Chancellor, Dr. Fr. Abraham Vettiyankal Mani, Pro-Vice Chancellor Dr.Fr. Joseph CC, our Head of Department(HOD), Dr. Ashok Immanuel V, Coordinator Dr. Shoney Sebastian and the faculty for providing all the required facilities to accomplish the project.

We would like to convey our gratitude to our project guide, Dr. Thirunavukkarasu V, for giving us a constant source of inspiration and help in preparing of the project, correcting our work and providing encouragement throughout the project.

Finally, We would like to thank our parents and friends, without them this project would not have been completed.

Ankur Sharma - 2147105

Adarsh Verma - 2147102

Amber Ujjwal Linda - 2147104

ABSTRACT

Image processing is the process of converting an image into a digital format and carrying out specific procedures to extract some usable information from it is known as image processing. When implementing certain specified signal processing techniques, the image processing system typically interprets all pictures as 2D signals. Using specialized software, optical character recognition (OCR) transforms scanned text images into electronic text so that digitized material may be searched, indexed, and retrieved. image recognition refers to a computer program's capacity to recognize items, locations, persons, writing, and activities in pictures. To recognize images, computers can employ machine vision technology in conjunction with a camera and artificial intelligence software. Image recognition refers to a computer program's capacity to recognize items, locations, persons, writing, and activities in pictures. To recognize images, computers can employ machine vision technology in conjunction with a camera and artificial intelligence software.

This project helps people that are visually challenged to carry on the tasks for which they are dependent on someone else to be done. This project covers the processes such as text-to-speech conversion for office documents, library books, and any other printed text document and campus tours for the students' using audio output. This project can also be used for identifying any teacher based on the Face recognition concept and voicing out their details to the student. This complete project will majorly focus on the visually impaired students and help the students who are visiting the college for the first time. The development of this project aims to replace manual involvement with audio, and students do not need to rely on another person to complete the process.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This chapter gives a brief description, objective and scope the system proposed. Thechapter also talks about the existing systems and the limitations of that along with the proposed system and it's advantages.

1.1 PROJECT DESCRIPTION

Cell phones are an essential part of cutting-edge lifestyles and in today's era there is there is no task which cannot be completed without the cell phone. Proposed system is an android application that helps using voice commands. The utility is evolved for visually impaired human beings. After unlocking the mobile, the application can be launched with voice commands. The structure accept voice commands and volume button actions to perform different operations to them. To act for different tasks, it first translates the voice into textual content, which gives output in the form of voice. It performs simple features such as a campus tour of the college. We will use the concept of image processing to train our model to detect the different places inside the campus. This will help the visually impaired students to know about the campus, and by using this module, the visually impaired students will be independent enough to explore the campus on its own. In the next module, we will perform Teacher recognition task which uses the concept of face detection, which uses image recognition and computer vision. The next module is Text to Speech recognition using which the student can scan any printed document and convert it to speech, and this module will support the students for documentation in the office and the library. The last module is about how the user will navigate through the landing page.

1.2 EXISTING SYSTEM

The existing apps designed for visually impaired persons are as follows: Voice Over, Talk back, Be my eyes, Seeing AI, Aira, Lookout. These are the applications which use the concept of artificial intelligence and machine learning, they have most of the functions which are dependent on the mobile camera and the speaker.

These components collect most of the information for processing the data. Some of the examples are text to speech, face recognition, reading the text out loud, detecting the object in front of them, scene descriptor and also scans the barcode of an item from a supermarket and gives the detailed information about the same. These functionalities use the concept of Natural language processing, sentiment analysis, computer vision and many more concepts. The limitation of the existing projects are that they require internet connection, projects are are too general and they are not user specific

1.3 PROPOSED SYSTEM

The proposed system is going to be implemented as an android mobile application. It consists of four functionalities that will be useful for the users who are visiting CHRIST UNIVERSITY. These four mentioned modules are: a landing page, this page will direct the users to the other three modules as the project is for users who are visually impaired it will ask for some input from the users side to select one of the functionality using a button or a gesture. The second functionality will consist of text to speech recognition for this the application will ask for permissions from the user which will direct to use the camera and microphone. This functionality will read any printed document and output the data in a audio format which was scanned from camera. The next functionality is Teacher recognition, this is similar to face recognition in this the module few teachers images will be trained from different angles and will be stored in the database so that if a teacher from the collected dataset is in front of the camera then the application will respond with the teacher's name. The last functionality is the campus tour which will tell the user about the current location place he/she is present.

1.4 OBJECTIVE

The objective of the project is to tackle the problem statement mentioned with helpof using AI/ML , by performing upon Image processing on the input image and predict the output by using Machine. The objective is also to train a proper model from the dataset that can successfully perform

translation. By fulfilling the above requirements, the user can input an image and can get output as speech/audio to help visually disabled students. The project should be able to achieve the following transforming upon not only a single input but multiple different inputs over the three modules.

1.5. PURPOSE, SCOPE, AND APPLICABILITY

1.5.1. Purpose

(i). The project aims to help visually disabled student of CHRIST university to carry on the tasks for which they are dependent on someone else to be done.

(ii). This project can be implemented in many places where the user needs to get an output as speech. Example - To read any printed document and output the data in a audio format, to recognize a teacher and to know about the CHRIST Bangalore central campus.

(iii). Since there is no current system for this kind of use in CHRIST university, this project will improve the current system records by giving an approximate imageaccording to the current time or future.

1.5.2. Scope

To get an accurate result from the trained model we assume that the Input image is clear, text is printed only and the face of the individual is not tampered i.e.,he has not undergone any plastic surgery, his face is not damaged.

1.5.3. Applicability

We define the applicability of this project more in-line with the CHRIST university students and teachers, speacially designed for visually disabled students. The model can help the visually impaired students to know about the campus and teachers. By using this mobile application, the visually impaired students will be independent enough to explore the CHRIST

UNIVERSITY Bangalore campus on their own.

1.6. OVERVIEW OF THE REPORT

The report sticks to a linear path and contents are aligned accordingly. All the components and sub-components are clear and well defined, and modularized inthe proper way.

The **First** chapter is the **Project Introduction**, it contains sub modules as Project Description, Project Description, Existing System, Objectives, Scope and Applicability which introduces the project aim and what it tends to achieve.

The **Second** chapter is **System Analysis and Requirements,** it contains **Problem Definition**, **Requirements Specification**, **Block Diagram**, **System Requirements, User Characteristics**, **Software and Hardware Requirements, Constraints** and **Conceptual Models.**

The **Third** chapter is **System Design,** it encompasses all the important criteria related to making the model such as **System Architecture, Module Design** and **Interface and Procedural Design.**

The **Fourth** chapter is the **Implementation,** which consists of **Implementation Approaches**, **Coding Standard** and **Coding Details.** This part is the development part of the project which requires technical and programming knowledge. Here the training and testing of the model is done after coding. The logical base of the code will impact every aspect of the project, hence making it a very important module.

The **Fifth** chapter is the Testing, this part comes after the implementation or coding. It consists of **Test Cases** and **Testing Approaches**.

The **Sixth** and the last is **Conclusion**. It comprises of **Design and Implementation Issues, Advantages, Limitation** and **Future Scope of the Project.**

2. SYSTEM ANALYSIS AND REQUIREMENTS

System Analysis and Requirements is the second chapter, and it includes the following: Problem Definition, Requirements Specification, Block Diagram, System Requirements, User Characteristics, Software and Hardware Requirements, Constraints, and Conceptual Models. This section outlines the issue statement the project will attempt to solve as well as any further choices and plans of action made in response to it. It also makes note of a crucial element: the system requirements, which are crucial for each project.

2.1. PROBLEM DEFINITION

Visually disabled people have a tough time finding good reading materials in accessible formats. Also the office forms are not easy to fill without the help of others. Other challenge that they face is recognizing their teachers and they had to ask most of the time when they encounter them. Another challenge for the visually disabled person, especially the one with the complete loss of vision, is to navigate around places. Obviously, they can roam easily around their houses without any help because they know the position of everything in house. But, unfortunately, this is not done in most of the places. This creates a big problem for visually disabled people who might want to visit the place.

The project aims to tackle this problem by creating a model using the latest AI/ML frameworks, which can convert a given image or set of images into age-transformed data. Different individuals can use this different modules of the proposed application to successfully transform several types of visual data into speech. CHRIST university can use this application to help their visually disabled students, other students, teachers, visitors, etc.

2.2. REQUIREMENTS SPECIFICATION

2.2.1 Functional Requirements

Dataset: The .py and .jason files contains important pre-processing parameters. This is the first segment of the entire operation. It is also responsible for the segregation of data. It accepts only particular data formats (JPG or PNG). The separated data is used to train the model.

YOLOV5: The cutting-edge, real-time object identification programme YOLO was trained on the COCO dataset. An entire image is processed by a single neural network. The programme divides the image into areas and forecasts probability and bounding boxes for each region.
Health care, security surveillance, and self-driving cars are just a few of the industries that have adopted YOLO because of its speed and accuracy. Many versions of this model have been released since the Ultralytics team began working on its improvement in 2015. The fifth iteration of this algorithm, known as YOLOv5, will be examined in this article.

CNN Model: The CNN model of neural networks enables us to derive more accurate representations of the image material. In contrast to traditional image recognition, which requires you to define the image characteristics directly, CNN starts with the raw pixel data from the image, trains the model, and then automatically extracts the features for improved categorization.

Real Data: instances of real data, photographs of real individuals. These events are used as encouraging examples by the discriminator during training.
Fake Data: examples of fake data that the generator has produced. These situations are used as negative examples by the discriminator during training.

2.2.2 Non-Functional Requirements

Constraints

The Only Constraints in the working system are regarding image type. The image should be in proper format (JPG or PNG), and the image should necessarily consist of a human face of 215 x 215 minimum. However, the Image resolution can be higher.

Response time

The response time is limited to 1-2 sec. Depending on the inputs from the user, this may increase by a few mili-seconds but does not hamper the overall performance.

Re-usability

The trained model is robust; no matter the number of uses, the Output remains consistent throughout.

Capacity:

No limitations are present currently in the working system. The user can input n- number of inputs and receive proper Output.

2.3 ARCHITECTURE DIAGRAM

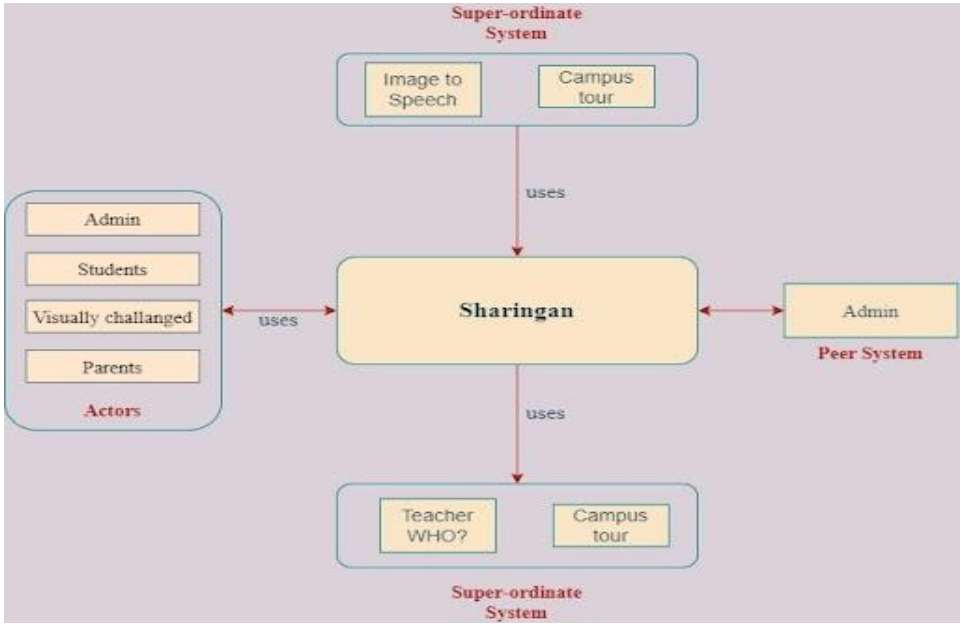


Fig 2.1 Architecture Diagram

2.4. SYSTEM REQUIREMENTS

2.4.1. User Characteristics

Our model is supposed to cater to a range of users, but the underlying characteristics are the same for all the users, i.e the character should have a minimum configured system and an image data to work with the model. Barring this criteria the system will reject the input from the user.

2.4.2. Software and Hardware Requirements

2.4.2.1 Software and Hardware Requirements for training the model

The Dataset used is highly demanding in terms of system requirements. Dataset: We are using the .json file (with images) to train our CNN model for facial recognition and for campus tour modules. The CNN has multiple convolution and pooling layers and we are using RELU as activation function. We are using base64 format to share and process the text images.

Software Requirements

- OS: Windows 10
- Visual Studio: 1.74
- Python: Python 3.10
- Compiler: Node v16.17.0
- Mobile platform: Android and IOS
- Android version: Android 5.0 and above
- Mobile app - Expo: 2.26.6

- Dependencies:

“body-parser”: “^1.20.1”
"cors": "^2.8.5”
"express": "^4.18.2"
"@react-navigation/native": "^6.0.14"
"@react-navigation/native-stack": "^6.9.2"
"axios": "^1.2.0" "expo": "~47.0.8"
"expo-camera": "~13.1.0"
"expo-dev-client": "~2.0.1"
"expo-font": "~11.0.1"
"expo-speech": "~11.0.0"
"expo-status-bar": "~1.4.2"
"react": "18.1.0"
"react-native": "0.70.5"
"react-native-gesture-handler": "~2.8.0"
"react-native-reanimated": "~2.12.0"
"react-native-safe-area-context": "4.4.1"
"react-native-screens": "~3.18.0"
"react-native-volume-manager": "^1.2.1"
"react-native-web": "~0.18.9"
"react-dom": "18.1.0"
"@expo/webpack-config": "^0.17.2"
Table no. 1.1

Here are the **Minimum System Requirements**:

- CPU: Intel Core i5 8th Gen
- CPU SPEED: 2.5GHz (or Above)
- RAM: 8 GB
- OS: Any
- Hard Disk: 500 GB

Recommended System Requirements:

- CPU: Intel Core i5 8th Gen (or above)
- CPU SPEED: 3.7GHz (or Above)
- RAM: 16 GB
- OS: Any
- Hard Disk: 1TB

2.4.2.2. Software and Hardware Requirements for the trained model

Hardware Requirements

Here are the **Minimum System Requirements**:

- CPU: Dual Core
- CPU SPEED: 1 GHz
- RAM: 2 GB
- OS: Any
- STORAGE: 8 GB

Software Requirements

- Operating System - Certified Distribution of WINDOWS
- Visual Studio
- Web Browser - Google Chrome
- Database(Backend) - SQL
- Python 3.6
- Tesseract
- Pytorch

2.4.2.3. Android Requirements

- Mobile platform: Android and IOS
- Android version: Android 5.0 and above
- Mobile app - Expo: 2.26.6
- Project's .apk file INSTALLED
- RAM : 3 GB (and above)
- CPU processor: Octa-core Max 2.02GHz
- Storage: 10GB(and above)

2.4.3. Constraints

Hardware Limitations: Requires high-end computing infrastructure, including both high-configuration CPU & GPU with the latest updated drivers. Not fulfilling, which will lead to high training time. Language Requirements: Requires support of python and all AI/ML libraries and packages. Not fulfilling, which will lead to the non-functioning of the model.

Reliability Requirements: Since the project deals with images of people, the privacy issue must be considered while creating a reliable environment.

Now that you have understood the problem domain and the requirements, produce a model of the system, reflecting the operations/functions that can be performed on/by the system and the allowable sequence of these operations. Conceptual Models can include Context Diagram, Data Flow Diagrams (At least 0 and 1 Level), Entity Relationship (ER) Diagrams, Class Diagrams, System Flow Charts, etc.

2.5. CONCEPTUAL MODELS

2.5.1. Data Flow Diagram

- The base model can be conceptualized by using subsystems of YOLOV5.
- In the first step image as an input is provided.
- The next step contains requirement validation, which checks If the image satisfies the required criteria or not
- Next comes the pre-processing stage, where the image undergoes certain pre-processing parameter.
- Then the image undergoes the transformation phase, under which it gets transformed into speech using a trained Machine learning model.
- In the last step, an output is to produce the audio message using mobile.

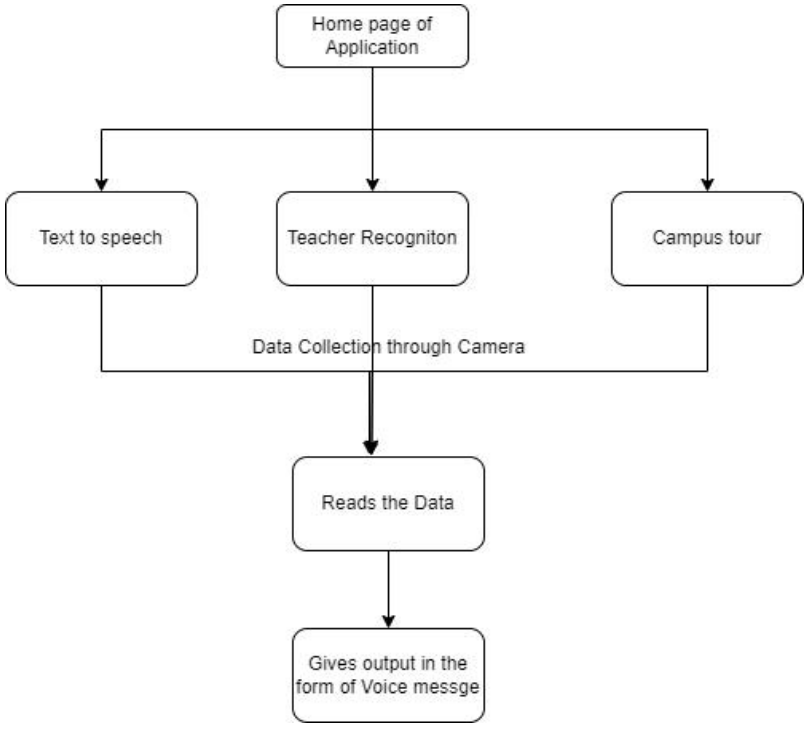


Fig 2.2 DFD for the proposed system

3. SYSTEM DESIGN

This chapter contains the various designs related to the system that encompass all the sub-modules and subsystems of the main system. It also covers the system architecture,data flow, and other design aspects.

3.1 SYSTEM ARCHITECTURE

The system architecture can be divided into two parts: the training and the trained.The upper half of the system architecture is the system's training component, which contains several modules.

Image Dataset : The Face dataset is generated using teachable learning.

Name: Face

Type: Image Dataset

Format: JPG

Number of Images: 50

Dimensions: 200x200

Color space: RGB

Pre-processing is an important module with several pre-processing parameters. It further ensures the images follow the proper parameters before reaching the main algorithm module.

YOLOV5: The images are trained using yolov5 and COCO128 and then the test images are feeded to check the models accuracy. We will also be using the functionalities in Android to access the camera to recognize the image. This includes the facial recognition and campus building recognition in our project.

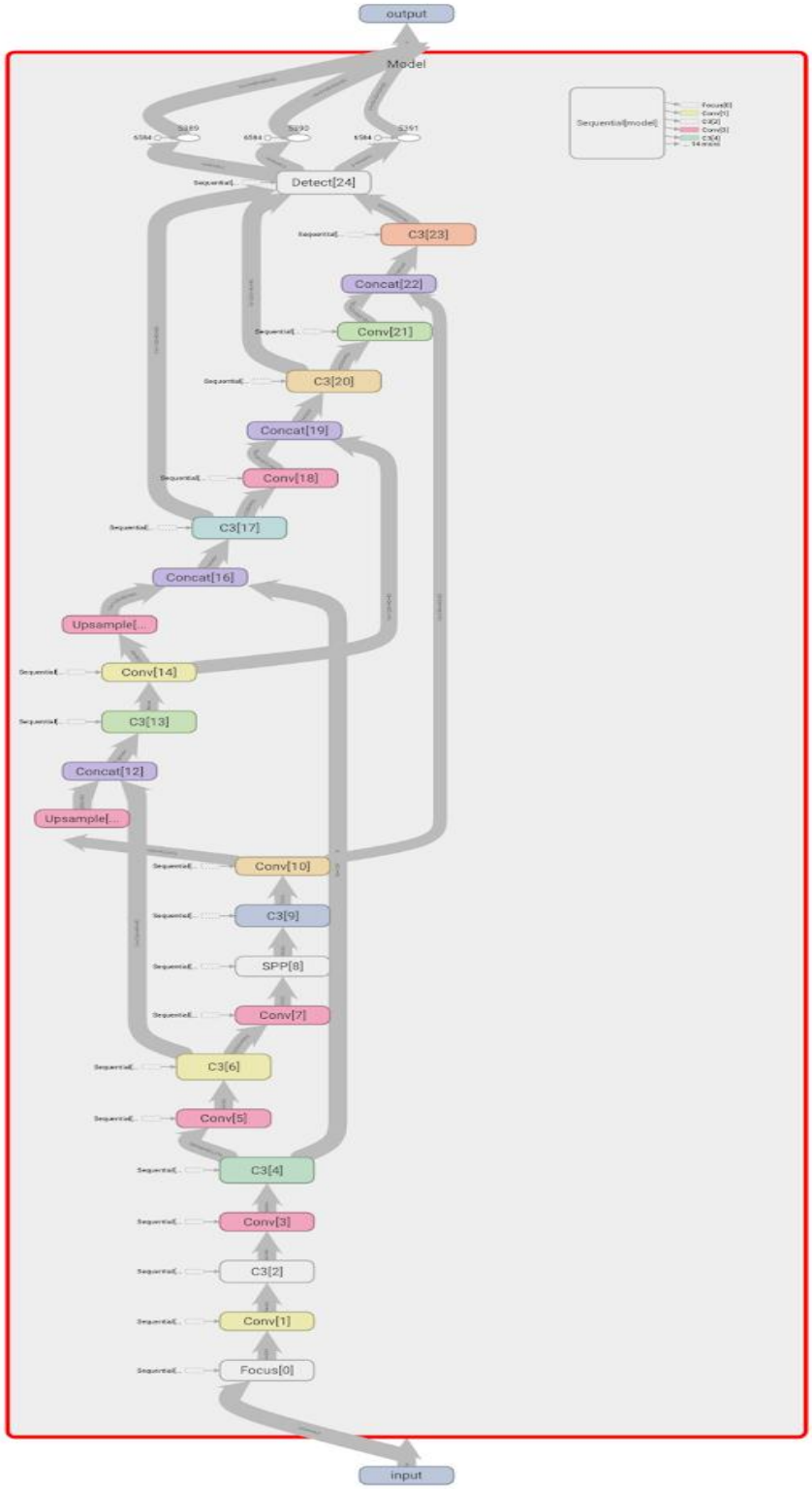


Fig 3.1 YOLOV5 Model Architecture

3.2 ER DIAGRAM

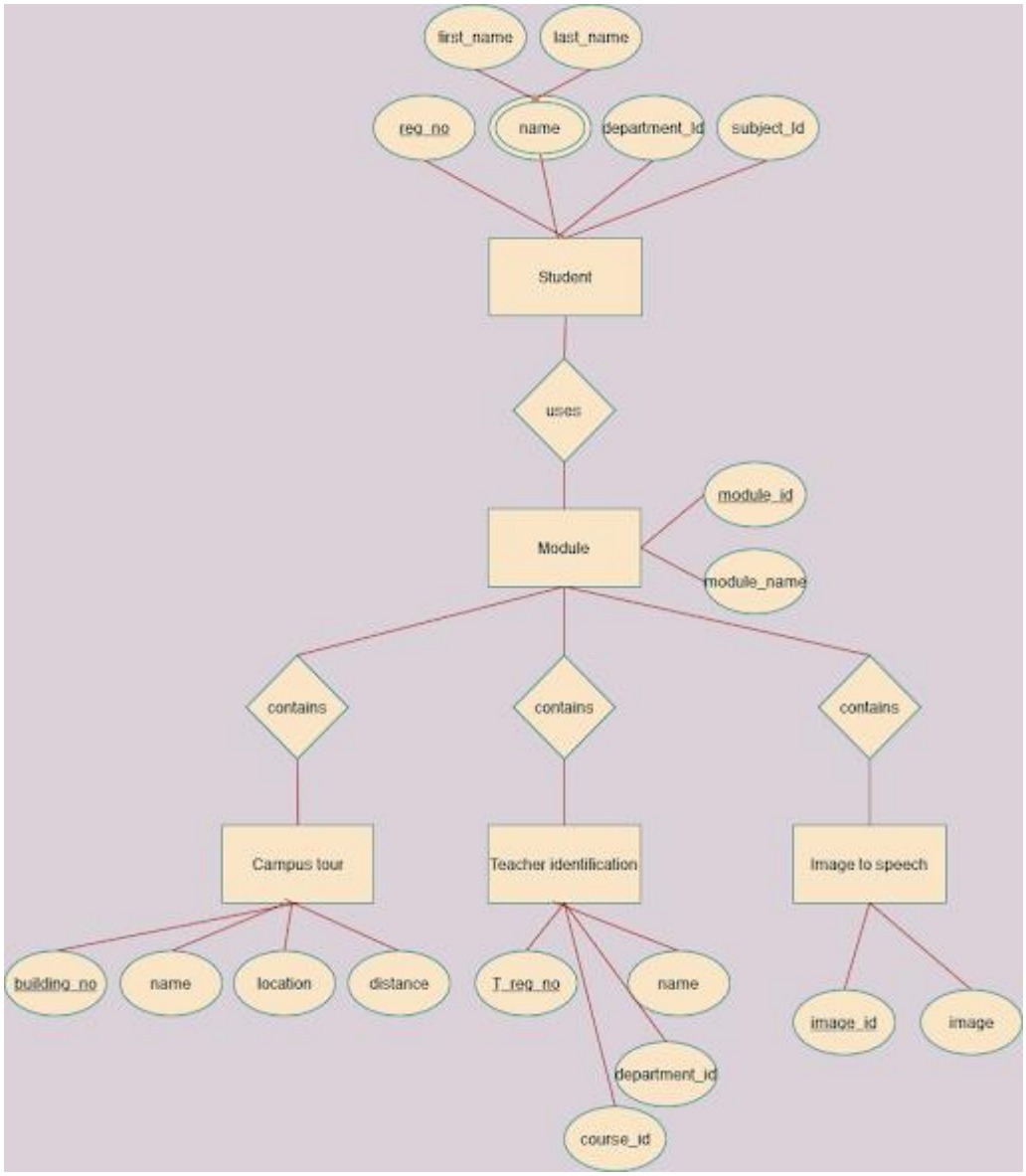


Fig 3.2 Entity Relationship Diagram

3.3 CLASS DIAGRAM

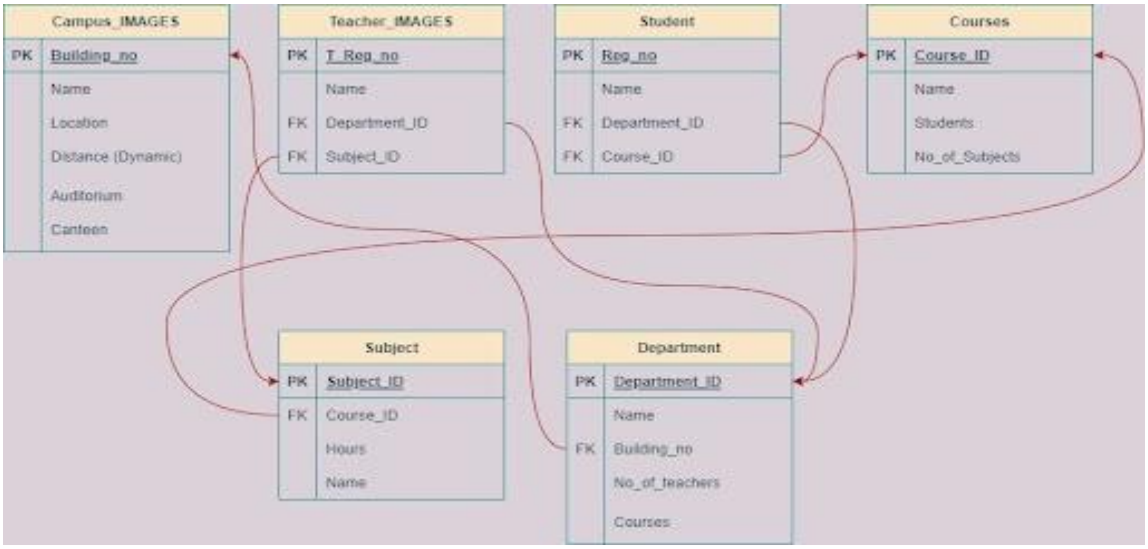


Fig 3.3 Class Diagram

3.4 INTERFACE DESIGN AND PROCEDURAL DESIGN

The UI incorporated here was made by keeping in mind the user perspective; the Interface's theme, design, and components have a minimalist tone. While creating the UI, the latest interfaces in similar domains were reviewed for reference.

3.4.1 USER INTERFACE DESIGN:

The Interface follows very simple and user-friendly operations. After unlocking the mobile, the application can be launched with voice assistants like Google, Siri, Alexa, etc. It have a navigation page to navigate through the modules. The application accept volume button actions to perform operations according to them. Volume down button will navigate to the next module and will speak out the module name. Volume up button will select the module and opens the camera for user.A voice instructing you the procees to proceed will be heard. Follow it. Example: If you enter “Teacher Recognition” module, it will say “Align your camera to the text and double click to capture the image”.

3.4.2. MODULE DESCRIPTION

The user interface consists of the three modules

3.4.2.1. IMAGE TO SPEECH

When the user selects this module, then the camera will automatically be turned on and the camera will start detecting the images using the functionality of Tesseract OCR and the images with the text will automatically get recognized with the help of bounding boxes or annotated image box . The text in the image will be converted into text with the help of google text to speech and finally the recognized image's text will be voiced out with the help of python play sound.

3.4.2.2. TEACHER RECOGNITION

When the user selects this module, then the camera will automatically be turned on and the camera will start detecting the images. We have trained the model on YOLOv5 and COCO128 using these two deep learning algorithms. Now once the camera is pointed at any teacher, this functionality will voice out the information about them.

3.4.2.2. TEACHER RECOGNITION

When the user selects this module, then the camera will automatically be turned on and the camera will start detecting the images. We have trained the model on YOLOv5 and COCO128 using these two deep learning algorithms. Now once the camera is pointed at any building in the campus, this functionality will voice out the information about them.

3.4.3. USER INTERFACE

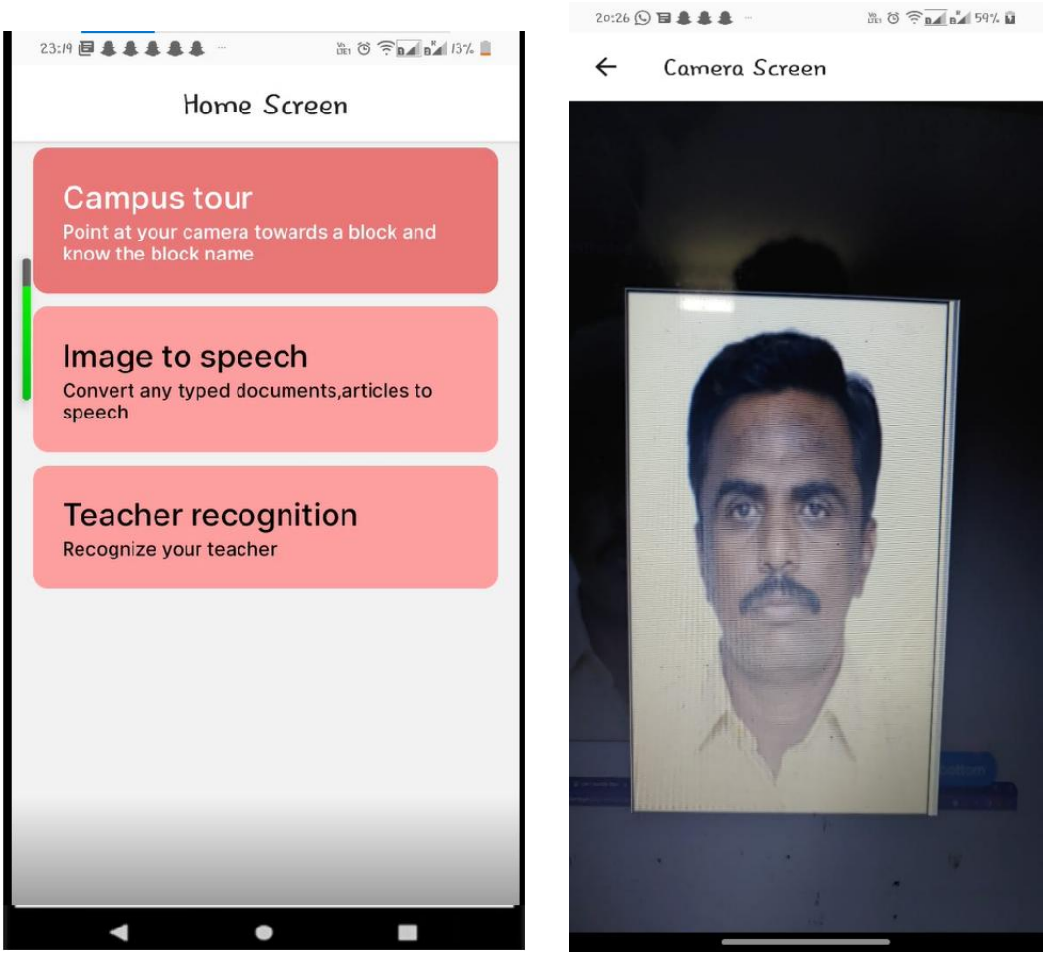


Fig 3.4. Mobile application UI Output

4. IMPLEMENTATION

Implementation, which is divided into Implementation Approaches, Coding Standards, and Coding Details, is covered in this chapter. This part is the development part of the project which requires technical and programming knowledge. After coding, the model is trained and tested in this case. The logical foundation of the code will have an impact on all areas of the project, making it a crucial module.

4.1 IMPLEMENTATIONS APPROACHES

The implementation approach opted towards converting the objectives into an actual working model is the Agile approach. An agile approach is a value-centered approach to project management; the agile project management process allows for processing projects in discrete stages or cycles. The approach is adaptable, and projects with dynamic characteristics would benefit from it. Project managers working in this environment would treat milestones as "sprints" to continuously adapt to sudden changes brought on by client feedback. Small software projects with highly collaborative teams or requiring frequent iterations are best suited for it.

4.2 CODING STANDARD

- Proper coding standards were followed during the coding and implementation stage.
- Since the project is AI/ML oriented, the entirety of the code is written using python and nodeJS.
- Comments were added at every juncture within the code.
- The code was properly modularized by grouping the tasks into proper well-defined functions.
- Naming conventions were implemented to identify each component easily. Error and Exception Handling at every critical section of code for error handling.

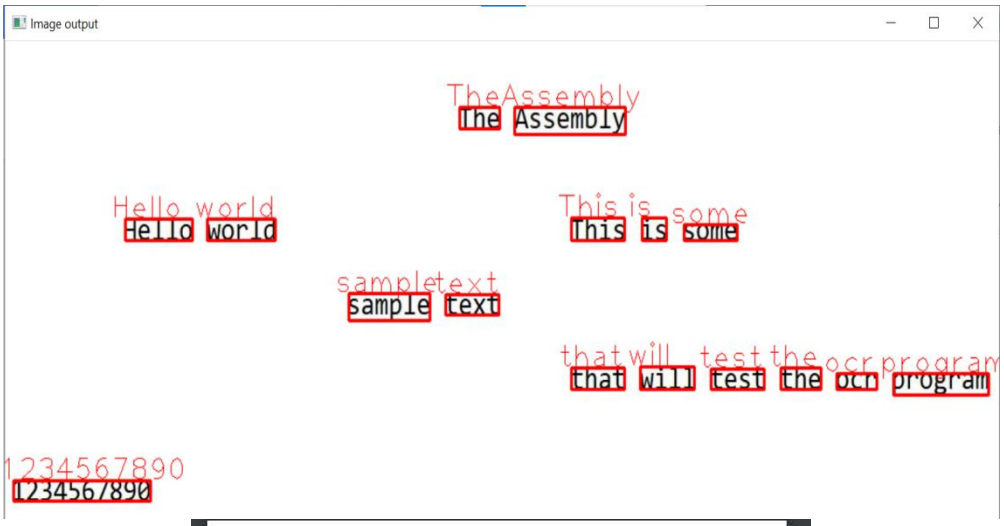
4.3 MODULE CODE AND OUTPUT SCREENSHOT

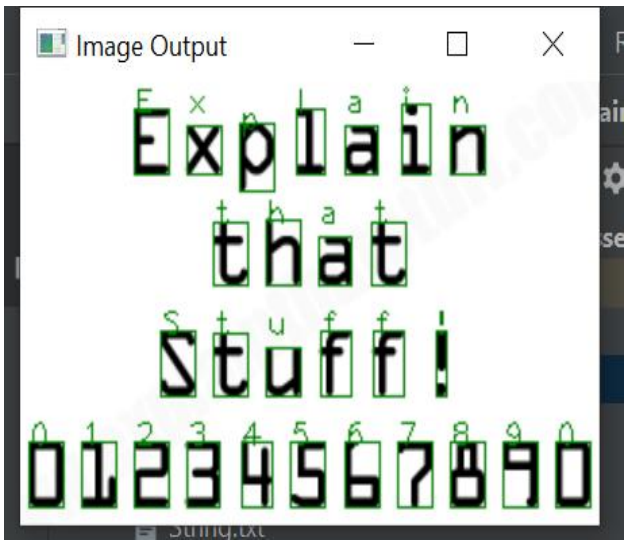
4.3.1. Module 1

```
        #python code for OCR, text to speech
        import cv2
import pytesseract
import base64


def img_to_speech(img_in_base64):
    pytesseract.pytesseract.tesseract_cmd = "C:/Program Files (x86)/
        Tesseract-OCR/Tesseract.exe"
    image_64_decode = base64.decodebytes(img_in_base64)
    image_result = open('target.jpg', 'wb')  # create a writable image
         and write the decoding result
    image_result.write(image_64_decode)

    final = cv2.imread('target.jpg')
    data4 = pytesseract.image_to_data(final)
    #print(data4.splitlines())
    str = ""
    for z, a in enumerate(data4.splitlines()):
        # Counter
        if z != 0:
            # Converts 'data1' string into a list stored in 'a'
            a = a.split()
            # Checking if array contains a word
            if len(a) == 12:
                # Storing values in the right variables
                x, y = int(a[6]), int(a[7])
                w, h = int(a[8]), int(a[9])
                # Display bounding box of each word
                cv2.rectangle(final, (x, y), (x + w, y + h), (0, 0, 255), 2)
                # Display detected word under each bounding box
                cv2.putText(final, a[11], (x - 15, y),
          cv2.FONT_HERSHEY_PLAIN, 2, (0, 0, 255), 1)
                str = str + a[11] + " "
```

```
        # Output the bounding box with the image
    cv2.imshow('Image output', final)
    cv2.waitKey(0)
    return str

image = open('C:/Users/ankur/Downloads/specialised_project/
        Real-time-OCR-Text-To-Speech-with-Tesseract-main/
        Capture_1.JPG', 'rb')
image_read = image.read()
image_64_encode = base64.b64encode(image_read)
speak = img_to_speech(image_64_encode)
print(speak)
```

Output:





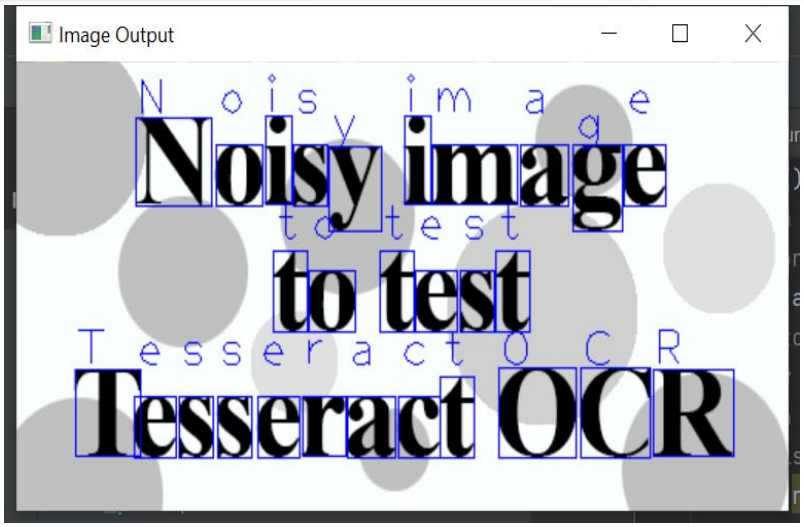


Fig 4.1. Outputs for OCR text recognition

4.3.2. Module 2

```
        *Python code for teacher recognition
        import os
import boto3
import cv2
import pytesseract
from gtts import gTTS
from playsound import playsound
import matplotlib.pyplot as plt

# Connects pytesseract(wrapper) to the trained tesseract module
pytesseract.pytesseract.tesseract_cmd = "C:\Program Files (x86)\
        Tesseract-OCR\Tesseract.exe"

dict_images = { "C:/Users/ankur/Downloads/specialised_project/
        senthil.jpg" : "Professor Senthilnathan",
"C:/Users/ankur/Downloads/specialised_project/thiru.jpg" : "Professor
Thirunavukkarasu", "C:/Users/ankur/Downloads/specialised_project/
        Adarsh.jpg" : "Adarsh", "C:/Users/ankur/Downloads/Ankur/
        Ankur_Sharma.jpg" : "Ankur",
"C:/Users/ankur/Downloads/specialised_project/Amber.jpg" : "Amber"}

def compare_faces(sourceFile, targetFile):
    client = boto3.client('rekognition')
```

```
    imageSource = open(sourceFile, 'rb')
    imageTarget = open(targetFile, 'rb')

    response = client.compare_faces(SimilarityThreshold=80,
                                    SourceImage={'Bytes': imageSource.read()},
                                    TargetImage={'Bytes': imageTarget.read()})

    for faceMatch in response['FaceMatches']:
        position = faceMatch['Face']['BoundingBox']
        similarity = str(faceMatch['Similarity'])

        print('The face at ' +
              str(position['Left']) + ' ' +
              str(position['Top']) +
              ' matches with ' + similarity + '% confidence')

    imageSource.close()
    imageTarget.close()
    return len(response['FaceMatches'])


def main():
    # Video feed
    video = cv2.VideoCapture("https://192.168.137.230:8080/video")
    # Setting width and height for video feed
    video.set(3, 640)
    video.set(4, 480)

    # Capture one frame from the video feed
    extra, frames = video.read()
    data4 = pytesseract.image_to_data(frames)

    print("OK")
    cv2.waitKey(0)

    img = frames
    cv2.imwrite("filename.png", img)
```

```
target_file = "C:/Users/ankur/Downloads/specialised_project/
    Real-time-OCR-Text-To-Speech-with-Tesseract-main/

    filename.png"
name = ""

for i in dict_images:
    source_file = i

    face_matches = compare_faces(source_file, target_file)
    #print("Face matches: " + str(face_matches))

    if face_matches == 1:
        name = dict_images[i]
        break

if name == "":
    line = "Not a teacher at Christ"
    language = 'en'

    speech = gTTS(text=line, lang=language, slow=False)
    speech.save("test.mp3")
    playsound("test.mp3")
    return 0

from PIL import Image

input_image = Image.open(target_file)
target_image = Image.open(i)

#target_image.show()
#input_image.show()

plt.figure(figsize=(10, 10), constrained_layout=False)

plt.subplot(121), plt.imshow(input_image)
plt.title("Input Image", fontsize=10), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(target_image)
```

```
    plt.title("Target image", fontsize=10), plt.xticks([]), plt.yticks([])

    plt.show()

          cv2.waitKey(0)

    line = name + "from Department of Computer Science,
         Christ university"

    language = 'en'
    speech = gTTS(text=line, lang=language, slow=False)
    speech.save("test.mp3")

    playsound("test.mp3")


if __name__ == "__main__":
    main()
```

Output:

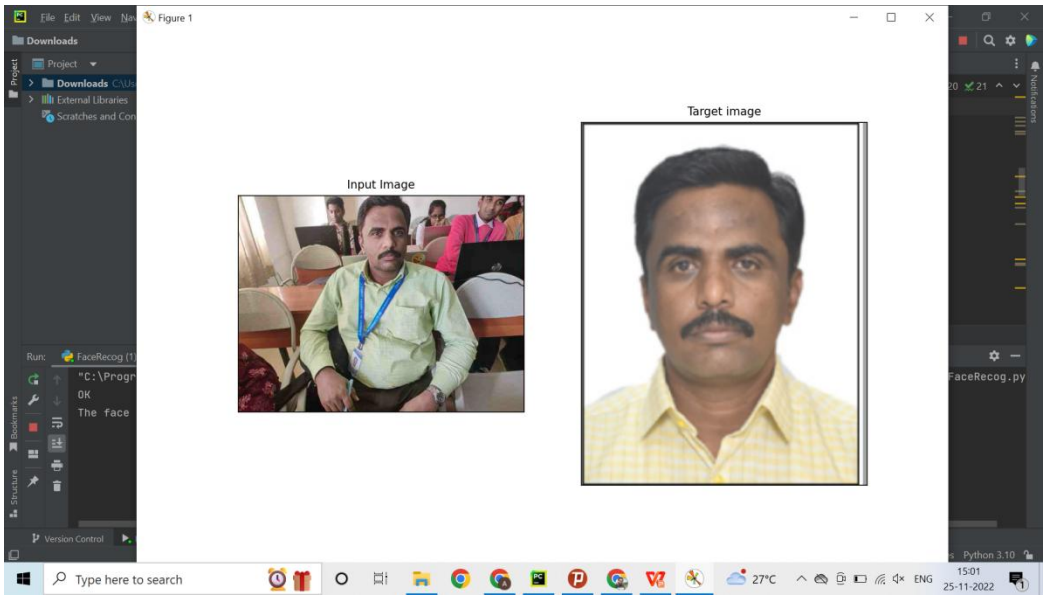


Fig 4.2. Output for teacher recognition

4.3.4. Navigation Page in Native React

```
// listen to volume changes (example)
useEffect(() => {

    // VolumeManager.showNativeVolumeUI({ enabled: true});
    setVolume(0.8)

    let sum = 0
    const volumeListener = VolumeManager.addVolumeListener((result) => {

        sum += 1
        if(result.volume > 0.8 || result.volume === 1){
            // increaseId()
            if(sum === 4){
                console.log("volume up")
                setVolume(0.8)
                sum = 0
                navigation.navigate("camera-screen",{
                    selectedId:selectedId
                })
            }
        }else if(result.volume < 0.8 || result.volume === 0){
            // decreaseId()
            if(sum === 4){
                increaseId()
                setVolume(0.8)
                sum = 0
            }
        }
    });

const tapGesture = Gesture.Tap()
.numberOfTaps(2)
.maxDuration(500)
.maxDelay(250)
.maxDistance(10)
.onStart(captureImage);


return (
    // outputImg ? <View style={{flex:1}}><img style={{flex:1}} src={`data:image/p
    !loading ? <Camera style={styles.camera} type={type} ref={r => cam = r}>
        <GestureDetector gesture={tapGesture}>
            <View style={styles.buttonContainer}/>
        </GestureDetector>
    </Camera>
    :<Loader/>
)
}
```

5. TESTING

After the project's implementation or coding phase, the fifth chapter is the testing section. Writing test cases and doing extensive testing during this step ensures that the designed and implemented components are all functioning as intended. Test Cases andTesting Approaches make up the model.

5.1 TEST CASES

5.1.1 Test case table

Test Id	Test Name	Input Specification	Expected output	Actual Input	Actual Output
1.1	Input	1. Navigate through the modules using volume down button 2. Press the volume up button to select a option and enter into the function 3. User could capture the image by pointing at a object	1.Accept the input 2. Camera should open 3. Image was captured	1.Pressed the volume down button 2. Entered into the function 3. Camera was pointed at a object	1. Accepted 2. Camera was turned on without any issues 3. Image was captured successfully
2.1	Requirement and validation	1. Point the camera at a printed document 2. Point the camera at a Teacher 3. Point the camera to a campus building	1. Accept 2. Accept 3. Accept	1. Document was read 2 Teacher was Identified 3. Teacher was Identified	1. Accept 2. Accepted 3. Accepted

Table no. 2.1 Test Cases

5.2 TESTING APPROACH:

Unit Testing was chosen as the testing approach and was performed on the model. The model was divided into smaller units to affiliate with the testing standards. Testcases were designed by keeping the units in consideration.

Unit testing: A bit of code is checked to see if it functions as intended as part of this software testing. The name comes from the fact that tests are carried out one unit at a time. By identifying bugs or errors in the code level early in the development lifecycle, unit testing aims to lower the cost of bug fixes. Unit testing is a crucial component and a money saver for the company because the cost of fixing bugs increases exponentially as the Software Development Lifecycle moves into its later phases.

It is a component of Test-Driven Development (TDD), a pragmatic process that uses continuous testing and modification to produce a product. This is also the first stage of software testing, undertaken before additional approaches such as integration testing. Unit tests are generally isolated to ensure that no external code or functions are used by the unit. Testing can be performed manually, although it is more commonly automated.

5.3 Test Report

Test Id	Test Name	Input Specification	Expected output	Actual Input	Actual Output	Output
1.1	Input	1. Navigate through the modules using volume down button 2. Press the volume up button to select a option and enter into the function 3. User could capture the image by pointing at a object	1.Accept the input 2. Camera should open 3. Image was captured	1.Pressed the volume down button 2 Entered into the function 3. Camera was pointed at a object	1 Accepted 2 Camera was turned on without any issues 3. Image was captured successfully	PASS
2.1	Requirement and validation	1. Point the camera at a printed document 2. Point the camera at a Teacher 3. Point the camera to a campus building	1. Accept 2. Accept 3. Accept	1. Document was read 2 Teacher was Identified 3. Teacher was Identified	1 Accepted 2 Accepted 3 Accepted	PASS

Table no. 2.2 Test Report

6. CONCLUSION

6.1. DESIGN AND IMPLEMENTATION ISSUES

The training phase of the model is the most demanding aspect of the entire project. This can easily hamper the design and implementation and create issues such as delays and time mismanagement.

Since the training time is high, the coding part has to be faultless, as changes in the code will lead to further training and more time delays.

Attributes such as scaling and performance will directly depend on the type of system used to train the model. Without a high-end configuration system, the project's scalability aspect will severely hamper other related attributes.

For implementation purposes, the user side should have a minimum system requirement fulfilled system to successfully and easily run the pre-trained model.

6.2. ADVANTAGES AND LIMITATIONS

Advantages:

- The project stays grounded with its approach and provides a realistic output type. It doesnot over-saturate the product in terms of getting an output.
- The system also accepts inputs from the user side and works with them. It performs transformation successfully over the number of inputs and produces proper Output.
- The model successfully fulfils its objective of successfully creating a mobile application for Visually disabled students while following all standards and parameters.

Limitations:

- Does not work well in situations where the entity in the input image is not clear. In such cases, the results are not accurate.
- Sometimes accepts inputs with no proper facial attribute, thus affecting the Output for teacher recognition.
- Does not work accurately on dysfunctional faces or faceswith any anomaly.

6.3 Future Scope of the Project

i. Other universities can use this model extensively in collaboration with new technology to improve their service towards their students.
ii. Hospitals can use this model extensively in collaboration with other medical technology to improve their service towards their patients.
iii. Libraries can use this model extensively in collaboration with new technology to improve their service towards their customers/visitors.
iv. Offices can use this model extensively in collaboration with new technology to improve their service towards their employees.

REFERENCES

[1] Face Aging With GAN, Siyao Sun, Alex Wang {siyaosun, cwang16}@stanford.edu https://github.com/siyaofd/GANSTER

[2] GANs for Face Aging problems — What does your face look like in the next few years? Neurons AI Feb 9, 2021 https://neurondai.medium.com/

[3] A Gentle Introduction to CycleGAN for Image Translation Jason Brownlee, August 5, 2019, https://machinelearningmastery.com/what-is-cyclegan/

[4] A. Lanitis, C. J. Taylor, and C. J. Taylor, T. F. Cootes. 2002. Toward Automatic Simulation of Aging Effects on Face Images. IEEE Trans. Pattern Anal. Mach. Intell. 24, 4 (April 2002), 442-455. Jinli Suo, Song-Chun Zhu, Shiguang Shan, and Xilin Chen. 2010. A Compositional and Dynamic Model for Face Aging. IEEE Trans. Pattern Anal. Mach. Intell. 32, 3 (March 2010), 385-401.

[5] Burt, D. M. and David I. Perrett. "Perception of age in adult Caucasian male faces: computer graphic manipulation of shape and colour information." Proceedings. Biological sciences 259 1355 (1995): 137-43 .

[6] Bernard Tiddeman, Michael Burt, and David Perrett. 2001. Prototyping and Transforming Facial Textures for Perception Research. IEEE Comput. Graph. Appl. 21, 5 (September 2001), 42-50.

[7] Bhowmick, Alexy & Hazarika, Shyamanta. (2017). An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends. Journal on Multimodal User Interfaces. 11. 1-24. 10.1007/s12193-016-0235-6.

[8] Al-Razgan M, Almoaiqel S, Alrajhi N, Alhumegani A, Alshehri A, Alnefaie B, AlKhamiss R, Rushdi S. A systematic literature review on the usability of mobile applications for visually impaired users. PeerJ Comput Sci. 2021 Nov 22;7:e771. doi: 10.7717/peerj-cs.771. PMID: 34901428; PMCID: PMC8627222.

[9] Dobosz, Krzysztof. (2017). Designing Mobile Applications For Visually Impaired People.

[10] Gannady G. Kalach, Navigation System Based on the Fuzzy Logic Expert System, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, issue 6, November-December, 2019. https://doi.org/10.30534/ijatcse/2019/02862019.