Assisting Blind People with AI and Audio by Using Smart Glass and Smartphone

Priyanka Shilevant, Supasun Khumpraphan, Vishap Chauhan, Cynthia Elijah, Kudirat Sofola, Yashang Dubey, Anju Thomas

Abstract--The daily activities of those who are blind or have their vision impaired are limited particularly due to the need to navigate dependently. Several artificial intelligence approaches have been developed for the purpose of supporting this set of people with the ability to live an independent life. Despite this, there are still limiting factors that impede the seamless independence of blind people. Hence, this project has employed the use of AI, ML and deep learning models to carry out object identification and image processing, to be trained by users' data and then transmitted to the server using encrypted communication, to ensure users' privacy and provide them with a notification of any object identified. This research will provide blind people with the experience of living a more independent life.

I. Introduction

Traditionally, individuals who are visually impaired have relied on other people, their canes and guide dogs for navigation and mobility; however, there are huge safety concerns and discrimination that they struggle with in their daily activities. For instance, the Guardian newspaper reported as recently as the 22nd of May 2022 that senior executive at the charity Guide Dogs said he felt "publicly humiliated" when he and his guide dog, Faldo, were illegally told to leave a Marks & Spencer shop in west London. He said the incident felt like a "kick in the nuts" and that it highlighted a continuing problem of blind and partially sighted people being refused entry or ejected from shops because of their guide dogs. Dave not only felt humiliated, but his freedom and independence were taken away from him because he relies on his guide dog for mobility [1]. Another recently reported case was by BBC news, about the discrimination against a blind student, Kelsey Trevett, who was left shaken and excluded when he and his friend were turned away from a restaurant because Kelsey had his trusted guide dog with him [2]. Another recently reported case was by BBC news, about the discrimination against a blind student, Kelsey Trevett, who was left shaken and excluded when he and his friend were turned away from a restaurant because Kelsey had his trusted guide dog with him [2].

Independent mobility is required for those with vision impairments to live a fulfilling life. However, mobility might be difficult for blind people due to visual loss. As a result, the quality of life for blind and visually impaired people suffers. According to research, a Guide Dogs survey found that three-quarters of guide dog owners have experienced being illegally turned away from public buildings. Also, accidents are particularly common in mobility-related occupations. In 2010, a survey was done on accidents that blind or legally blind people had while performing their mobile tasks. According to the findings of 300 interviews, approximately 34% of legally blind and more than 45% of blind participants had head-level accidents once a month or less, while approximately 18% of legally blind and 9% of blind interviewees had the same type of accident more frequently, monthly. Medical repercussions were reported in 23% of the accidents. In all, 7% of people fell while walking once or more every month. Due to their limits in object recognition above knee-level, navigation, and orientation, supportive tools like the long white cane and guide dogs were unable to avert these incidents. Moreover, these tools suffer from certain drawbacks; for example, a cane is ineffectual over long distances, crowded places, and cannot provide information regarding dangerous objects or car traffic when crossing the street, whereas training of guide dogs is cumbersome and expensive, and dogs require special attention when caring for them. According to the World Health Organization (WHO), there are 253 million individuals with vision impairment, 36 million of whom are blind, and 217 million who are partially blind or suffer from mild to severe visual problems. The population of people who have irreversible blindness has necessitated the use of assistive technology for vision rehabilitation to optimize functioning and reduce disability. The identification of landmarks is now used by visually impaired and blind people to discover their locations on a route to determine orientation and navigation. Further, although smartphone applications such as voice assistance and navigation maps for BVI people are evolving rapidly, proper and complete use is still low [3].

Our research was directed at the use of artificial intelligence and audio to support blind people using smartphones. We carried out extensive background research on what has been implemented in the past and the shortcomings of their outcome. With lessons learnt from the various studies, and a knowledge of possibilities, we implemented a federated learning system to recognize objects from visual inputs from our mobile phone camera while also instructing users with audio and using image detection model (MobileNet SSD). Our research was targeted at how accurately and efficiently our system could carry out object recognition. Federated learning allows for smarter models, lower latency, and less power consumption, all while ensuring privacy. And this approach has another immediate benefit: in addition to providing an update to the shared model, the improved model downloaded on the phone can also be used immediately, powering experiences personalized by the way you use your phone. [4]Hence, our project aimed to achieve reduced latency and improved accuracy. By implementing a federated learning system, the distributed system of the edge computing device could reduce the dependency on heavy computing systems required as well reducing the latency as it is close to the speed of the network. Our study used a federated learning approach, wherein the devices used a shared prediction model. It allows multiple users to train the same model. This helps keep users' personal data on the end users' devices by pre-processing the data on the smartphone. It also saves the need to store the data on the cloud, thus saving storage space. Pushing the computation and prediction on the user's device reduces the latency caused while constantly communicating to the cloud and getting the required input. Federated learning is a collaborative way of learning, where each user's unique data will train the machine learning model. The model's learning will further be sent to the cloud using encrypted communication and then averaged with other users' updates. This was achieved with the utilization of some hard/software, libraries and IDEs to implement and test our project. The table below recounts them. Hence, our project aimed to achieve reduced latency and improved accuracy. By implementing a federated learning system, the distributed system of the edge computing device could reduce the dependency on heavy computing systems required as well reducing the latency as it is close to the speed of the network. Our study used a federated learning approach, wherein the devices used a shared prediction model. It allows multiple users to train the same model. This helps keep users' personal data on the end users' devices by pre-processing the data on the smartphone. It also saves the need to store the data on the cloud, thus saving storage space. Pushing the computation and prediction on the user's device reduces the latency caused while constantly communicating to the cloud and getting the required input. Federated learning is a collaborative way of learning, where each user's unique data will train the machine learning model. The model's learning will further be sent to the cloud using encrypted communication and then averaged with other users' updates. This was achieved with the utilization of some hard/software, libraries and IDEs to implement and test our project. The table below recounts them.

A. *Software Requirements*

Table 1 below introduces the software, libraries and IDE that will be used while implementing our project. Details of some of the software are indicated below the table:

Integrated development environment (IDE)	**Android Studio, Intellij IDEA, Visual Studio**
Front end Development	**XML**
Backend development	**Java, Kotlin**
Deep Learning Model	**MobileNet SSD (CNN based), OpenCV**
IOT	**Smart Phone**
Software-as-a-service	**RabbitMQ**
Development operating system	**Windows 10, Ubuntu, MAC OS**
Framework	**TensorFlow, Keras, Tensorflow Lite**

B. *Integrated development Environment (IDE)*

- **Visual Studio Code:** A simplified code editor that supports development tasks such as debugging, task execution, and version management. It seeks to provide developers only the tools they need for a rapid code-build-debug cycle, leaving more complicated processes to full-featured IDEs like Visual Studio IDE.

- **IntelliJ IDEA:** IntelliJ IDEA is an intelligent, context-aware IDE for working with Java and other JVM languages like Kotlin, Scala, and Groovy on all sorts of applications.

- **Android Studio:** Android Studio is an integrated development environment that allows you to create apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to break down your project into functional parts that you can create, test, and debug separately.

C. Front End Development

XML: eXtensible Markup Language, or XML is a markup language created as a standard way to encode data in internet-based applications. Android applications use XML to create layout files. Unlike HTML, XML is case-sensitive, requires each tag to be closed, and preserves whitespace.

D. Backend Development

- **Java:** Java is the platform of choice for creating programs that use managed code and run-on mobile devices.

- **Kotlin:** Kotlin is a general purpose, free, open source, statically typed “pragmatic” programming language initially designed for the JVM (Java Virtual Machine) and Android that combines object-oriented and functional programming features.

E. Deep Learning Model

- **MobileNet SSD:** *MobileNet SSD* is an object detection model that computes the bounding box and category of an object from an input image. This Single Shot Detector (SSD) object detection model uses MobileNet as backbone and can achieve fast object detection optimized for mobile devices.

- **OpenCV:** OpenCV is the huge open-source library for computer vision, machine learning, and image processing. By using it, one can process images and videos to identify objects, faces, or handwriting in real-time.

F. IOT

IoT: The Internet of Things (IoT) describes the network of physical objects— “things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. We will be using our mobile Motorola moto g50 - camera 48MP sensor

G. Software-as-a-service

RabbitMQ: RabbitMQ is a message broker that allows clients to connect over different open and standardized protocols such as AMQP, HTTP, STOMP, MQTT, MQTT over WebSocket and WebSocket/Web-Stomp.

H. Development operating system

Development work was carried out using these operating systems- Windows 10, Ubuntu, MAC OS

I. Framework

- **TensorFlow:** TensorFlow is an open-source library by Google primarily for deep learning applications. It provides a high-level API, and complex coding isn’t needed to prepare a neural network, configure a neuron, or program a neuron.

- **Keras:** Google created the high-level Keras deep learning API to implement neural networks. It is used to make the implementation of neural networks simple and is developed in Python. Additionally, different backend neural network computations are supported.
- **TensorFlow Lite:** TensorFlow Lite is a set of tools that enables on-device machine learning by helping developers run their models on mobile, embedded, and edge devices. [5]

II. RELATED WORK

In this section, we review similar work that have been carried out using a system of smart glasses and other wearable devices, smart phone apps and integrated AI to help the Blind or Visually Impaired (BVI) to detect objects or obstacles and provide audio navigation to help them live independent lives. We also look at some practical applications of federated learning.

In Jyun-You Lin, Chi-Lin Chiang et al's studies, they developed a system that could potentially help BVI people to understand their environment to improve their quality of life. The visually impaired people use smart glasses to take pictures of the scene, and the photos are immediately stored in SD card, and uploaded to the back-end server. After performing object identification, the YOLO v3 (You Only Look Once) deep learning model will output the results in text format that TTS can understand. The detection result is then output using TTS in a voice-based way so that visually impaired individuals may clearly understand the items in front of them and their locations. The average recognition rate they obtained was 96.3% achieving higher recognition rates with bigger objects (such as transportation vehicles) in comparison with smaller ones (such as animals and food).
They also recorded lower recognition rates with similar objects such as black dogs and black cats. In addition, their system took an average of 3.788 seconds from taking a picture with their smart glasses to uploading the photo and finally to making voice results. The model treats object detection as a single regression problem putting the entire image directly into the neural network to predict the position of Bounding Box and its corresponding category probability. Their studies made use of the COCO dataset which can recognize up to 80 objects (such as cats, dogs, horses, birds, bicycles, cars, trucks, traffic lights, chairs, and beds). This study did not make use of Federated learning, however Wi-Fi was used for wireless Internet access and uploading information. Once a picture was taken, the captured photo was read into the photo buffer of the Client and transferred to the Server. The server then receives the uploaded photo by the Client and stores it in the photo buffer of the Server. The YOLO object detection was called, and the text file generated by YOLO was read into the buffer side of the server and then the data in the text buffer of the server was downloaded to the client [6].

In their paper titled "Real Time Object Detection with Speech Recognition Using TensorFlow Lite", Ganesh Khekare1 and Kalpeshkumar Solanki's research focused on using a lightweight object detection model for real-time detection in this case, SSD MobileNet. Thier system was an android-based system designed for visually impaired persons with a goal to identify the type and location of objects in front of them so that they may be able to shop in a supermarket. The system builds an object detection system based on an SSD network as well as an object classification system based on a MobileNet network and builds two sets of image databases containing 9 categories in total for training a neural network. Once a picture is taken, it is sent to the network through the Android-TensorFlow interface, and the android app will display the detected category and the confidence of the category in real-time. If the accuracy is greater than 80%, then the TTS announces what the detected object is. In the object detection module, the RGB camera collects colour images to obtain a three-dimensional image matrix. The image is then passed through the SSD network, and the parameters and hyperparameters of the network have been trained and achieved over 90% accuracy [7].

Crossing the road successfully is a major challenge that BVI people face daily because errors could lead to collisions with vehicles. In 2020, Wan-Jung Chang, Jian-Ping Su and others published their research on a system that used AI edge computing-based technology to help BVI people safely cross zebra-crossings. The scenario of their experiment involved a BVI person crossing the road and the smart glasses making a recording of the beginning of zebra crossing. To achieve the goal of real-time high accurate zebra-crossing recognition, they adopted a single shot multi-box detector (SSD) which was installed in an AI edge computing core as their deep learning training module. The SSD uses a single convolutional neural network to detect the object in an image. They used zebra-crossing photo data for deep learning training. The photos consisted of 600 images with a size of 640x480. They had 200 scenes during the day, 200 views at night, and 200 pictures of slippery ground, respectively. The assistive system composed of smart glasses, an intelligent walking cane, and an intelligent waist-mounted box which received the current traffic light signal information and gave permission via audio on when to cross to a BVI person during the day or at night, or on a slippery road. The system consisted of a built-in GPU based embedded system development kit as an AI edge computing core and implementing deep learning image recognition on the intelligent waist-mounted box. An image sensor (camera), an IR sensor, and a G-Sensor were mounted on the proposed smart glasses for recording real-time video and fall/suspended obstacles detection. The recorded video is transmitted to the waist-mounted box in real time. The intelligent waist-mounted box receives the current pedestrian traffic signs information, which provides the visually impaired voice guidance service, and assists them to recognize whether there is a deviation from the zebra-crossing when walking at an intersection to reduce the chance of collisions between them and vehicles. The zebra-crossing recognition experiments were successfully done in three different situations (during the day, night, and slippery ground). The recognition rate was as high as 90%. The architecture of their system consisted of an intelligent walking cane, which integrates a Bluetooth (BT) module, a low power wide areas network (LPWAN) communication module, a GPS module, G-Sensor, a vibration motor module, and an MCU module. BT module was responsible for communication with the proposed smart glasses while the LPWAN module was used to upload the GPS position and the fall message to the online information platform. Their experiment was successful with a 90% recognition rate [8].

"Spatial orientation and navigation are the cornerstones of mobility," according to Passini et al. (Passini, Dupré, & Langlois, 1986). Electronic solutions have been developed to help or replace traditional instruments for people's movement, navigation, and object identification to serve persons with visual impairments. Many research groups and organizations have investigated and developed helpful solutions to help visually impaired and blind people with direction and navigation so they can move independently and freely. An inventory of 146 different devices was divided into two groups (object detection & orientation and navigation); depending on their purpose in a literature analysis on then-existing electronic support aids for mobility undertaken by Roentgen et al. from October 2007 to March 2008 [9]. On object detection with distance calculation, past studies have shown that determining the distance between the objects in a scene and the camera sensor from 2D images is feasible by estimating depth images using stereo cameras or 3D cameras. The outcome of depth estimation is relative distances that can be used to calculate absolute distances to be applicable. However, distance estimation is very challenging using 2D monocular cameras. A further study presented a deep learning framework that consists of two deep networks for depth estimation and object detection using a single image. Firstly, objects in the scene are detected and localized using the You Only Look Once (YOLOv5) network. In parallel, the estimated depth image is computed using a deep autoencoder network to detect the relative distances. The proposed object detection-based YOLO was trained using a supervised learning technique, in turn, the network of depth estimation was self-supervised training. The presented distance estimation framework was evaluated on real images of outdoor scenes. The results achieved in this study showed that the proposed framework is promising and yields an accuracy of 96% with RMSE of 0.203 of the correct absolute distance [10].On object detection with distance calculation, past studies have shown that determining the distance between the objects in a scene and the camera sensor from 2D images is feasible by estimating depth images using stereo cameras or 3D cameras. The outcome of depth estimation is relative distances that can be used to calculate absolute distances to be applicable. However, distance estimation is very challenging using 2D monocular cameras. A further study presented a deep learning framework that consists of two deep networks for depth estimation and object detection using a single image. Firstly, objects in the scene are detected and localized using the You Only Look Once (YOLOv5) network. In parallel, the estimated depth image is computed using a deep autoencoder network to detect the relative distances. The proposed object detection-based YOLO was trained using a supervised learning technique, in turn, the network of depth estimation was self-supervised training. The presented distance estimation framework was evaluated on real images of outdoor scenes. The results achieved in this study showed that the proposed framework is promising and yields an accuracy of 96% with RMSE of 0.203 of the correct absolute distance [10].

In order to achieve the best security, machine learning must take data privacy into account. People are becoming less willing to share their own personal data as security awareness rises, which substantially impedes the advancement of deep learning. A lot of data is needed for AI services. Most businesses do not have enough data to support data-hungry AI services. In most cases, data is not shared between firms or even between departments within the same company. Most of an individual user's data also includes personal information about them, such as their biographical information, health status, etc. It is dangerous to send the raw, unprotected data to the deep learning server in this circumstance. Malicious nodes can more easily exploit the federated learning system concept. There are also several other issues, such as numerous heterogeneous users and the data themselves, which make the algorithm more complicated to process. Some of them are brand-new issues, while others are the same problems as regular deep learning. Furthermore, federated learning frameworks may even be more severely impacted by the difficulties that other classical algorithms face. The federated learning system can be easily broken, for instance, if the dataset only contains positive labels [11]. Federated learning can be used for data privacy and security. Federated learning works on the end edge computing devices and data get aggerated to the server. All the data remains on the end edge device. In Fahad Ahmed KhoKhar, Jamal Hussain Shah, Muhammad Attique Khan, Muhammad Sharif,Usman Tariq, Seifedine Kadry, “A review on federated learning towards image processing” they found that federated learning is the smartest way to achieve data security and privacy. Edge devices send batches of data by encryption technique and the server computes and sends results in batches. Federated learning is essentially a distributed framework of the deep learning. It can improve the effectiveness of the model through the model aggregation of multiple clients based on ensuring the security of data privacy. The growing significance of data privacy is what gave rise to federated learning [12]. In Nishat I Mowla's work, they studied UNMANNED aerial vehicles (UAVs) with adaptive federated reinforcement learning to prevent interference in unmanned aerial vehicles. Then they used Q-learning to detect interference and learn the noise automatically without the need for an instructor and avoiding interference. This was a very good way to prevent spatial retreat. If there was no federated reinforcement learning, more noise would be found in unmanned aerial systems which would increase the chances of plane crashes even more. Moreover, the unmanned aerial vehicle system is a very challenging problem because it must be able to learn in new environments with no manoeuvres and needful high manoeuvrability because the need to always inspect the area and natural disasters, which makes it suitable for using federated reinforcement learning that can learn by itself rather than centralized knowledge. As a result of research and use of dataset ns-3, it was seen that the federated jamming detection mechanism, which is a federated reinforcement protection method learning was 39.9% higher than the distributed mechanism. In addition, the flying ad-hoc network (FANET) is a network made up of node UAVs to distribute communications over one another using a wireless medium and transfer them to Multi-Access Edge Computing (MEC) servers, but this communication is poor, and the signal will be disturbed. For this reason, we need to find an anti-interference method, which is an adaptive federated reinforcement learning method. [13]

III. METHODOLOGY

Our study used a federated learning approach, wherein the devices use a shared prediction model. It allows multiple users to train the same model. This helps keep users' personal data on the end users' devices by pre-processing the data on the smartphone. It also saves the need to store the data on the cloud, thus saving storage space. Pushing the computation and prediction on the user's device reduces the latency caused while constantly communicating to the cloud and getting the required input. [14]. Federated learning is a collaborative way of learning, where each user's unique data will train the machine learning model. The model's learning will further be sent to the cloud using encrypted communication and then averaged with other users' updates.

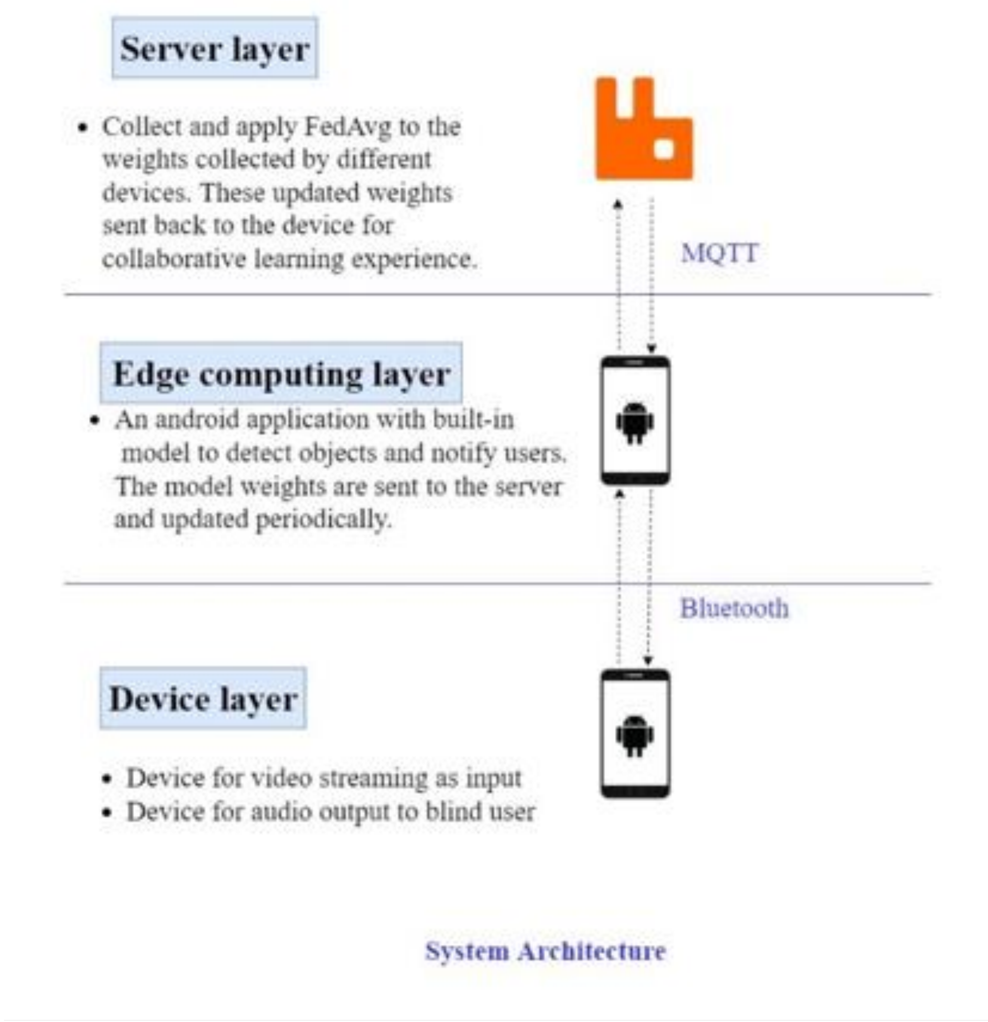


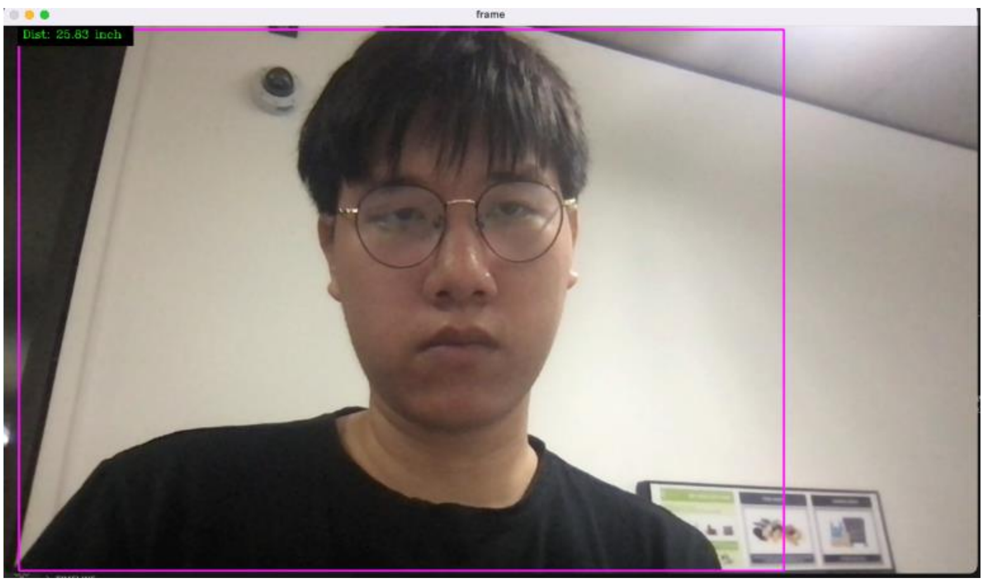
Figure 1: System architecture for assisting blind people with AI and smart glasses

a. Device layer

Internet of Things networks like wearable devices have sensors that allow them to collect huge data in real time. We used our smartphone (Motorola moto g50 - camera 48MP sensor) on the device layer which provides us with image frames of the surroundings. The images are then inputted into the android app. A video stream is generated and further sent back to the smartphone, which processes (using OpenCV) and sends the details via audio to the smart glasses. The image is continuously transferred to the android app to process and detect objects.

b. Edge Computing layer

Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services. The "edge" as any computing and network resources along the path between data sources and cloud data centers. Data is increasingly produced at the edge of the network; therefore, it would be more efficient to also process the data at the edge of the network. The data produced by IoT will never be transmitted to the cloud, instead it will be consumed at the edge of the network. A lot of delays are caused when there is constant communication between the cloud layer and the smartphone, which makes the technology unreliable in our project for dangerous situations. All the processing and prediction of the obstacles when done within mobile device saves time and avoids issues like network drop, extra storage and computation required to separately process the image/video frames. Hence, our android app is designed to process images and communicate with the user, by initiating a video stream. Object Detection is also carried out here, using MobileNet SSD because a small model that works fast with low latency does not consume much processing power. It is designed to work with limited resources, which is suitable for use with the phone.



For object distance measurement, the method we have utilized is to take a picture of the object and measure the object's size and distance from it with a tape measure relative to what the computer calculates and equalizes and as we move the camera further or closer, the variable D (distance between the camera and the object) is no longer a known variable; but we know the value F (image focus distance), this value does not change. If we do not adjust the camera, we can calculate the value D automatically [15].

Also on the edge computing layer, we make use of federated learning to periodically download the current model and compute an updated model in the android app using local data. These locally trained models (using TensorFlow and Keras) are then sent from the device back to the central server where they are aggregated, i.e., averaging weights, and then a single consolidated and improved global model is sent back to the devices.

We also made use of Transfer Learning. Transfer Learning is a technique that effectively uses knowledge of an already learned model to solve another new task with minimal re-training or fine-tuning. The requirement for a large amount of labeled data is a major problem in solving some critical domain-specific tasks, particularly the applications for the medical domain, where the creation of large-scale, high-quality annotated medical datasets is very complex and expensive.

Deep learning requires a massive amount of training data compared to traditional machine learning methods. Although academics are working hard to optimize it, the typical deep learning model requires a lot of computational power, such as a GPU-enabled server. To solve this issue, Deep Transfer Learning (DTL), a DL-based Transfer Learning method, was developed. By selecting a pre-trained model (trained on another big dataset of the same target domain) for a fixed feature extractor or for further fine-tuning, DTL greatly minimizes the requirement for training data and training time for a target domain-specific task [16] [17].

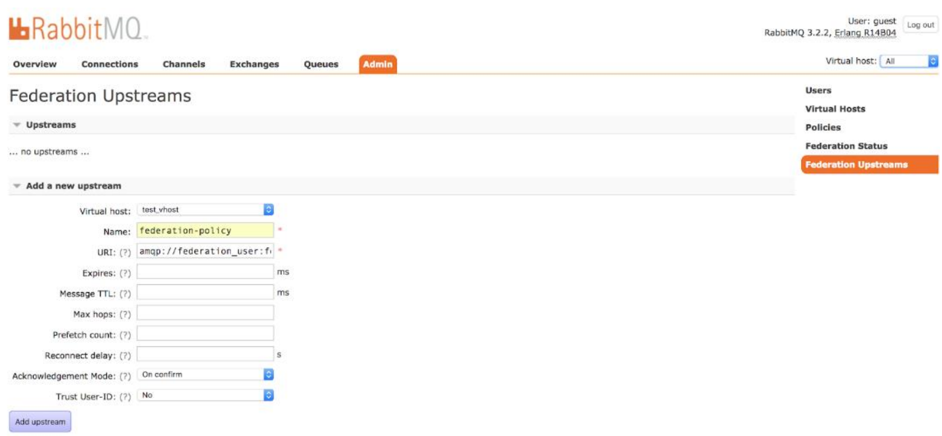
- **Server layer**

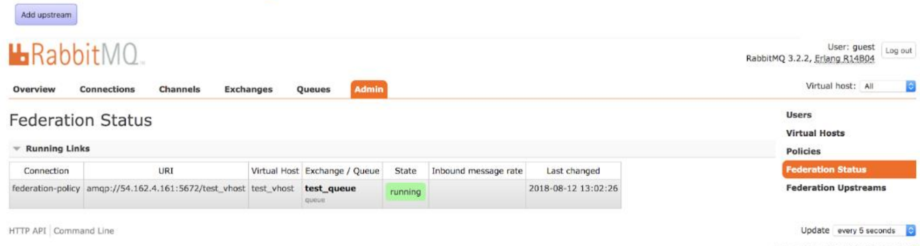
The server layer is where we deploy and run our machine learning model which is then processed by learnings of all the edge devices to implement the federated learning. Federated learning is introduced in this layer, wherein the data is decentralized. A model which is deployed on the server layer is shared between different mobile devices, thus keeping the data on the devices themselves. This is a collaborative way of learning, where each user's unique data will train the machine learning model. The model's learning will further be sent to the cloud using encrypted communication and then averaged with other users' updates. We intend to use a pre-trained deep learning model by comparing accuracies. By introducing the edge layer, we prevent sharing the user's data on the cloud. Heavy processing is done in the edge layer and the server layer will only process the data where our model fails to meet the requirement and accuracy. This layer further adds more leverage to store any unprocessed data which cannot be done on the user's device.

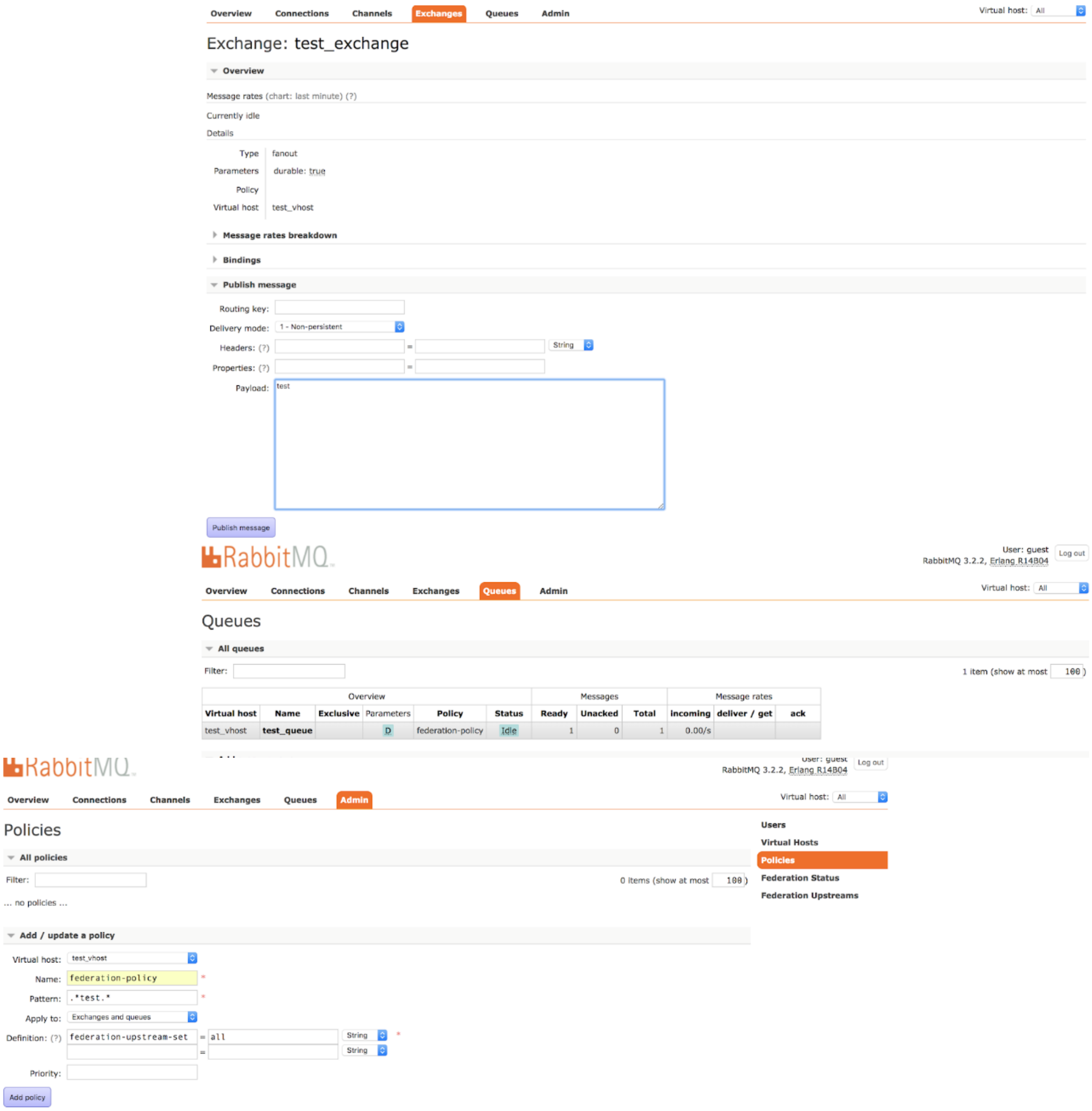
Federated learning supports this layer as it will enable the smart phone to learn a prediction model while keeping all the training data on device, decoupling the ability to do machine learning from the need to store the data in the cloud. This way, the smart phone downloads the current model, improves it by learning from data on it, and then summarizes the changes as a small, focused update. Only this update to the model is sent to the server, using encrypted communication, where it is immediately averaged with other user updates to improve the shared model. All the training data remains on the device, and no individual updates are stored on the server. [18]

RabbitMQ was utilized for communicating between this layer and the edge computing layer. It is a message broker that allows clients to connect over different open and standardized protocols such as AMQP, HTTP, STOMP, MQTT, MQTT over WebSocket and WebSocket/Web-Stomp. MQ Telemetry Transport protocol is often used in the IoT (Internet of Things) world of connected devices. It is designed for built-in systems, mobile phones, and other memory and bandwidth sensitive applications. MQTT provides an asynchronous communications protocol; the sender and receiver of the message do not need to interact with the message queue, the topic, at the same time. Messages placed onto the topic are stored until the recipient retrieves them or the messages time out. MQTT is for effective use by bandwidth-sensitive applications. [19] [20]

Rabbit MQ Implementation Steps





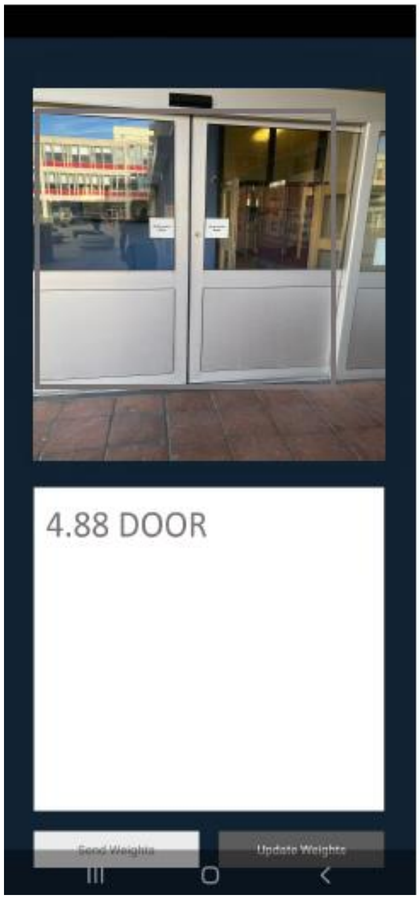


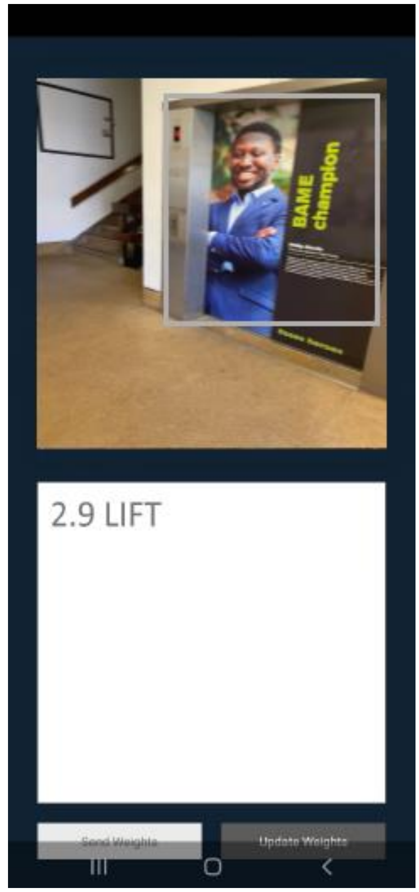
System Security

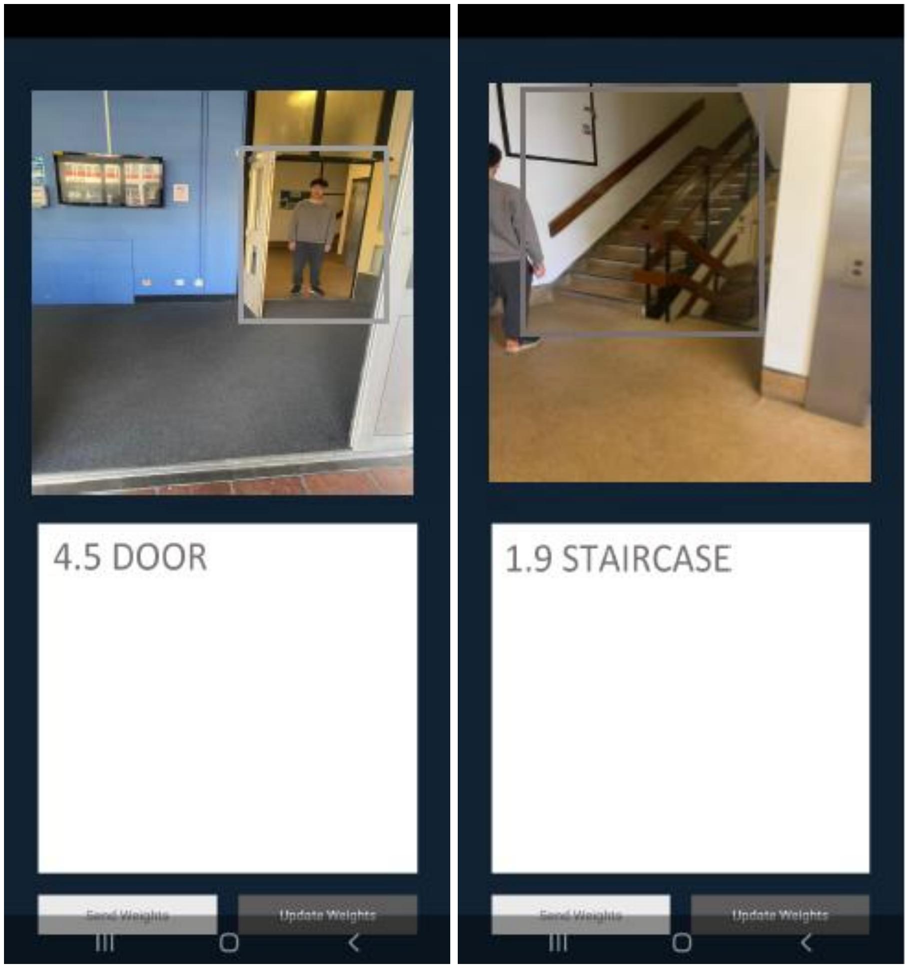
The federated learning system we have employed avoids sending actual users' data instead, we utilize the weights. However, our android app currently stores the weights sent and received from the server temporarily, on the device’s memory which might make it accessible.

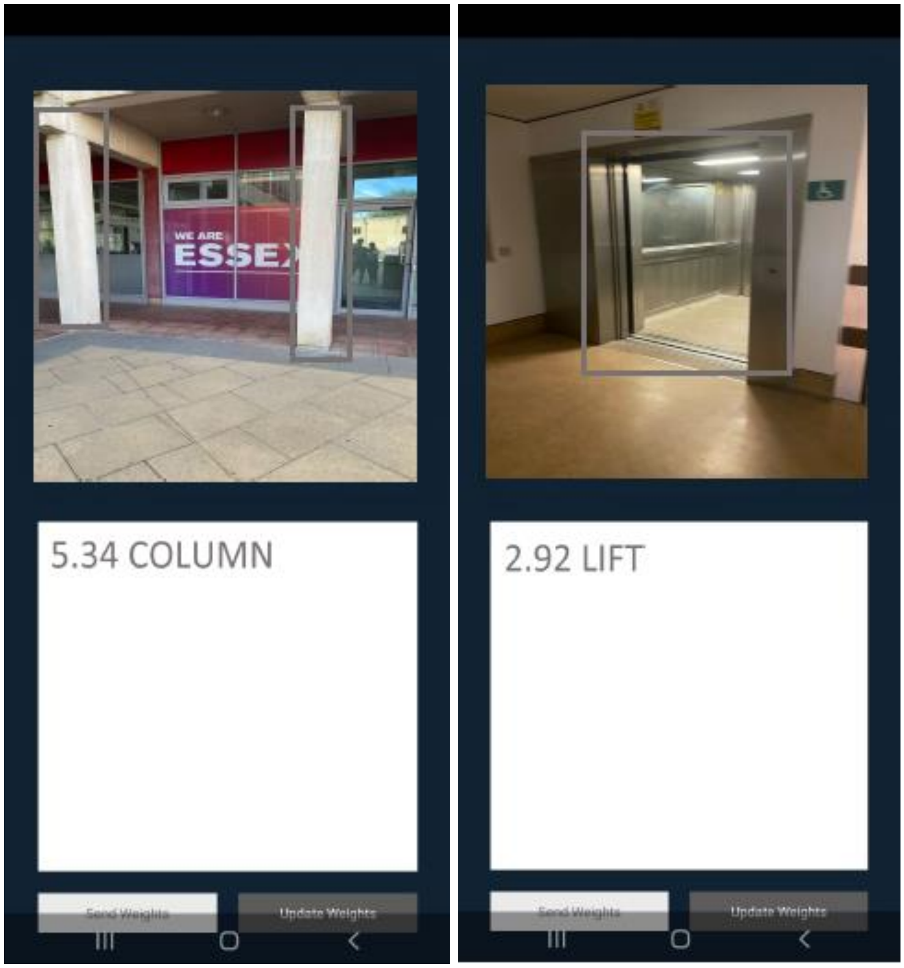
Scenario

We intend to perform our experiment in one of the labs and navigate successfully from the CSEE department's main entrance while avoiding any obstacles like pillars, staircases, chairs, tables, devices, doors, and other human beings. We also aim to identify objects such as computers, books and human beings. The images below show some of our scenario from the starting point to the ending point of our experiment:


4.88 DOOR
Send Weights
Update Weights


BAME
champion
2.9 LIFT
Send Weights
Update Weights


4.5 DOOR
Send Weights
Update Weights
1.9 STAIRCASE
Send Weights
Update Weights



Implementation

The flow chart below shows how the system is deployed:

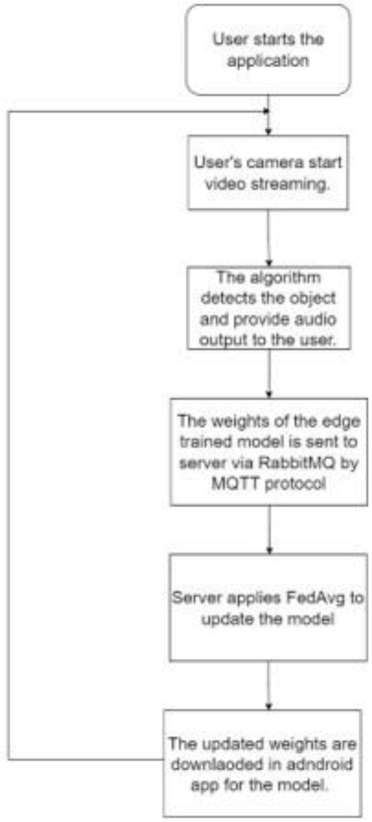


Figure 2- showing how the system is deployed

IV. EVALUATION

We developed an image classifier for training object detection which is an example of supervised learning. For distance calculations which makes it a regressor example, we use precision, recall, and F1 score as methods of evaluation.

a. Evaluation in terms of Accuracy

Our Image classifier will be based on the CNN model, for choosing the correct model we must evaluate in terms of accuracy, memory usage, CPU time, and model size. There are pre-existing models which are available like Dense Net, ResNet50, Inception, and so on.

Accuracy means considering the prediction of how much we achieve. Here we must process the Chair table and door, we are following the Supervised learning where we will train our model with existing images and other data sources. There are pre-existing models available in terms of their accuracy. By considering the top 7 pre-existing models. We have found the models below for Top 1 and Top 5 Accuracy.

Model	Top-1 Accuracy	Top-5 Accuracy
Xception	79.00%	94.50%
VGG19	71.30%	90.00%
ResNet50	74.90%	92.10%
ResNet152V2	78.00%	94.20%
InceptionV3	77.90%	93.70%
InceptionResNetV2	80.30%	95.30%
MobileNetV2	71.30%	90.10%
DenseNet121	75.00%	92.30%
DenseNet169	76.20%	93.20%
DenseNet201	77.30%	93.60%
NASNetMobile	74.40%	91.90%

Figure 3 showing the top-1 and top-5 accuracy refer to the model's performance on the ImageNet validation dataset. [25]

- Top-1 accuracy is the conventional accuracy, which means that the model answer must be exactly the expected answer.
- Top-5 accuracy means that any of your models that gives the 5 highest probability answers must match the expected answer.

We have found that the best accuracy was for the InceptionResNetV2 model. i.e., 80.3% and 95.3% when we checked for Top-1 Accuracy and Top-5 Accuracy respectively. In our model, we will aim to gain maximum accuracy by training our model.

b. The Efficiency of the Proposed System

(i) **CPU time:** is the exact amount of time that the CPU has spent processing data for a specific program or process. Here in our model, we have followed Edge computing where we are doing computation in the mobile app, which will save us time. By considering the above pre-existing model available when we observed we found the best CPU time was for MobileNetV2, which was 25.9ms per inference step. Table E.2. In our approach, we will reduce the computation time by developing a lightweight Android application and doing the computation in the smartphone itself. [26]

(ii) **Model Size:** The Model size is essential when we deploy our model, it saves time on computation. However, in our model, we aimed to make ultra-light applications so we can instruct blind people without any delay. By observing pre-existing models, we found that the Mobile Net model memory size is 14 MB in Size. In Our approach, we aim to reduce the file size to make our model lightweight.

Model	Size (MB)	CPU Time	GPU Time
Xception	88	109.4	8.1
VGG19	549	84.8	4.4
ResNet50	98	58.2	4.6
ResNet152V2	232	107.5	6.6
InceptionV3	92	42.2	6.9
InceptionResNetV2	215	130.2	10
MobileNetV2	14	25.9	3.8
DenseNet121	33	77.1	5.4
DenseNet169	57	96.4	6.3
DenseNet201	80	127.2	6.7
NASNetMobile	23	27	6.7

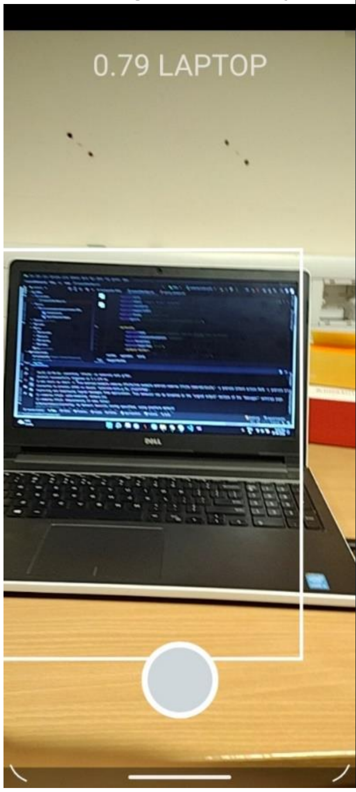
The table above shows top models in respect of Model, Size, CPU and GPU Performance on the ImageNet validation dataset.

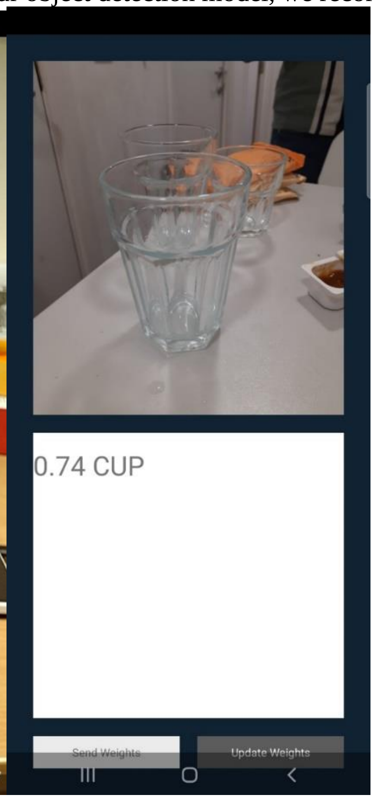
Time per inference step is the average of 30 batches and 10 repetitions.

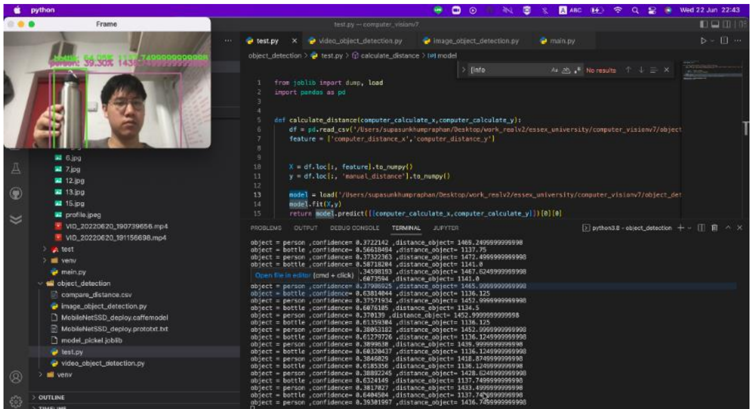
- CPU: AMD EPYC Processor (with IBPB) (92 core)
- RAM: 1.7T
- GPU: Tesla A100
- Batch size: 32

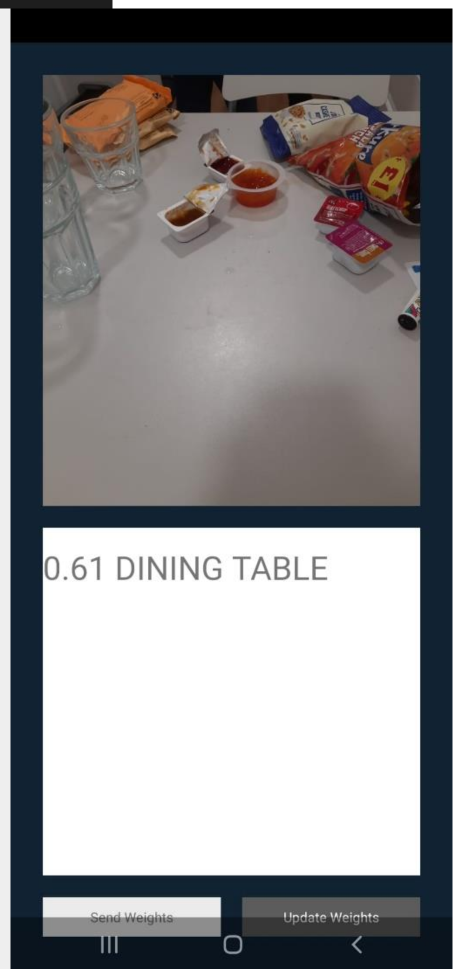
V. TESTING/RESULTS

In examining the efficiency of our object detection model, we recorded some results which are represented in the images below:







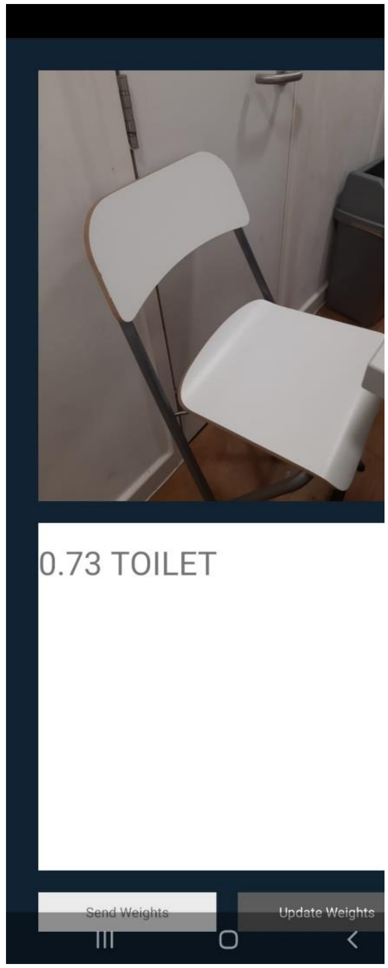


VI. FUTURE WORK

Looking forward, further attempts could prove quite beneficial to similar research, to intensify the efficiency of object detection models and that of text recognition. For instance, our object detection model sometimes detects multiple things some of which are irrelevant or inaccurately detected especially because of similar features like a wall and an elevator door. Also, our model did not factor in text recognition, however, during our data collection stage, we realized that there are instances where doors or buildings are labelled, and the label is needed for users to properly navigate and arrive at the right destination. Moreover, there are instances where detected objects might have texts on them, it will be necessary for models to be able to read texts detected on objects, generally.

Additionally on object detection, we recommend the identification of different types of doors, especially automatic doors which do not need to be manually opened or closed, so the user knows to wait for the door to open or close, however the case may be. Also, we noticed during our data collection process that the user must be mindful of the gap between elevator doors and the landing floor; an audio prompt like "please be mindful of the gap between the elevator's door and the landing floor as you exit" will be very helpful.

Similarly, an audio prompt that mentions the current floor number as users make use of an elevator will be helpful, as well as mentioning that the elevator's door(s) is opening or closing; the buttons in the elevator are on the left or right, in order to indicate which floor, the user is going to. In the same vein while using the stairs, the object identification model should be able to identify the staircases and send an audio prompt that states which side of the user the staircase is. For instance, an audio prompt mentions that "the stairs leading upstairs are on your left-hand side and the stairs leading downstairs are on your right-hand side".



The above image, for example, is a chair but is being detected as a toilet.

VII. CONCLUSION

This application will help the visually impaired to carry out their daily activities without relying so much on others, enabling a more independent life for them. By implementing a federated learning approach, the distributed system of the edge computing devices will reduce the dependency on heavy computing systems required. Our research also aimed to reduce the latency as it is close to the speed of the network. This can be further enhanced by deploying the model to a larger scale which will help to collect more training data and improve the accuracy of the model which is on the cloud.

VIII. PROJECT MANAGEMENT

The Gantt chart was employed to help us plan and execute the project. It helped us assess how long the project would take, determine the resources needed, and plan the order in which tasks will be completed.

Name and student ID	Work Duty
Vishap Chauhan (2110837)	T1,T2, T3, R1, R2, T4, T5,T9, T10, T11, T12, T13
Supasun Khumpraphan (2110366)	T1,T2, T3, T4, T5, T9, T10, T11, T12, T13
Kudirat Sofola (2111423)	T1,T2, T3 ,T4 ,T5, T9, T11, T12, T13
Cynthia Elijah (2112166)	T1,T2, T3,T6, T7, T10, T11, T12, T13
Priyanka Vilas Shilevant (2111310)	T1,T2, T3,R1, R2 , T6, T7, T10, T11, T12, T13
Yashang Dubey (2111712)	T1,T2, T3, T8, T10, T11, T12, T13
Anju Thomas (2111481)	T1,T2, T3,T8, T10, T11, T12, T13

A. Gantt Chart

Task/ activity	W30	W31	W32	W33	W34	W35	W36	W37	W38	W39
Planning (T1)										
Requirement (T2)										
Functional Requirement (R1)										
Non-Functional Requirement (R2)										
Design (T3)										
Create android application (T4)										
Text to Speech (T5)										
Object Detection (T6)										
Federated learning (T7)										
Cloud (T8)										
IoT (T9)										
Edge Computing (T10)										
Testing (T11)										
Document (T12)										
Presentation (T13)										

IX. References

[1] M. Weaver, "The Guardian," 2022. [Online]. Available: https://www.theguardian.com/society/2022/may/22/anger-as-man-with-guide-dog-told-to-leave-marks-spencer-shop.

[2] "BBC News," [Online]. Available: https://www.bbc.co.uk/news/uk-england-oxfordshire-61618015.amp .

[3] Q. Lu, "Feasibility Study of a "Smart" Aid for the Visually Impaired and Blind's Independent Mobility in Outdoor Environments," 2018.

[4] B. M. a. D. Ramage, "Federated Learning: Collaborative Machine Learning without Centralized Training Data," 2017.

[5] T. S. A. K. Z. M. S. M. T. A. &. S. V. Li, "Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems," 2020.

[6] C. -L. C. M. -J. W. C. -C. Y. a. M. -C. C. J. -Y. Lin, "Smart Glasses Application System for Visually Impaired People Based on Deep Learning," 2020.

[7] K. S. Ganesh Khekare, "REAL TIME OBJECT DETECTION WITH SPEECH RECOGNITION USING TENSORFLOW LITE," 2022.

[8] W. -J. Chang, "An AI Edge Computing Based W earable Assistive Device for Visually Impaired People Zebra-Crossing Walking," 2020.

[9] U. R. G. G. J. S. M. &. D. W. L. P. Roentgen, "Inventory of electronic mobility aids for persons with visual impairments: A literature review. Journal of Visual Impairment & Blindness, 102(11), 702-724.," 2008.

[10] A. M. D. G. A. S. C. J. P. D. &. R. H. A. Masoumian, "Absolute distance prediction based on deep learning object detection and monocular depth estimation models. In Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence, Artificial Intelligence Research and D," 2021.

[11] K. S. X. Z. C. e. a. Zhang, "Challenges and future directions of secure federated learning: a survey. Front. Comput. Sci. 16, 165817 (2022). https://doi.org/10.1007/s11704-021-0598-z," 2021.

[12] N. I. T. N. H. D. I. &. C. K. Mowla, "AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET. Journal of Communications and Networks, 22(3), 244-258.," 2020.

[13] A. a. M. D. G. a. A. S. a. C. J. a. P. D. a. R. H. A. Masoumian, "Absolute distance prediction based on deep learning object detection and monocular depth estimation models. In Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence, Artificial Intelligence Rese," 2021.

[14] Q. L. Y. C. Y. K. Y. C. T. &. Y. H. Yang, "Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 13(3), 1-207," 2019.

[15] "Sony," [Online]. Available: https://www.sony.co.th/th/electronics/support/articles/00267921.

[16] 2020. [Online]. Available: https://odsc.medium.com/what-is-federated-learning-99c7fc9bc4f5.

[17] A. G. A. S. S. F. S. Abu Sufian, "A Survey on Deep Transfer Learning to Edge Computing for Mitigating the COVID-19 Pandemic,," 2020.

[18] [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[19] B. M. a. D. Ramage, 2017. [Online]. Available: Federated Learning: Collaborative Machine Learning without Centralized Training Data.

[20] [Online]. Available: https://www.cloudamqp.com/docs/mqtt.html.

[21] "wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Focal_length .

[22] [Online]. Available: https://www.sony.co.th/th/electronics/support/articles/00267921 .

[23] 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[24] A. K. S. A. T. a. V. S. T. Li, "Federated Learning: Challenges, Methods, and Future Directions," in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60".

[25] [Online]. Available: https://keras.io/api/applications.

[26] [Online]. Available: https://keras.io/api/applications.

[27] N. I. T. N. H. D. I. &. C. K. Mowla, *AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET. Journal of Communications and Networks, 22(3), 244-258.,* 2020.

[28] B. M. a. D. Ramage, 6 April 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. [Accessed 06 2022].

[29] [Online]. Available: https://www.sony.co.th/th/electronics/support/articles/00267921 .

[30] A. K. S. A. T. a. V. S. T. Li, *Federated Learning: Challenges, Methods, and Future Directions," in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60,* 2020.