AI BASED REMOTE EXPERT SYSTEM FOR GUIDING THE USERS ON MACHINE ACTIVITIES

Vivek Kumar Varma Nadimpalli, Gopichand Agnihotram, Pandurang Naik

ABSTRACT

Creating a remote expert system that will assist the user in repair or maintenance works of any manufacturer machine. The paper proposes a method where the user will able to collect the device images data or the video stream of device data in real-time and remote expert who will able to guide the user to perform a repair or maintenance task on the device. The object detection intelli-gence will be added on both the user side and expert side, which will be leveraged by the expert to guide the user. As an applica-tion, the paper describes different models trained on annotated data of dialysis device and based on accuracy metric the model will be chosen and ported into the mobile device and AR devices. With the help of machine data sent by the user the expert will guide the repair steps of the machine remotely and user will per-form each step at machine side using object detection of different parts of machine.

Index Terms*— Object Detection, Augmentation, Real time Streaming, MobileNet, TensorFlow Object Detection, TensorFlow Lite, ARCore, ARKit, Single Shot Detection, MobileNet, Res-Net50, Expert system. YOLO Model, MobileNet FPN.*

Reference to this paper should be made as follows:

Vivek Kumar Varma Nadimpalli, Gopichand Agnihotram, Pandurang Naik. *'AI BASED REMOTE EXPERT SYSTEM FOR GUIDING THE USERS ON MACHINE ACTIVITIES, Int. J. of Electronics Engineering and Applications, 8, Issue II, July-Dec 2020, pp 49-62, DOI-10.30696/IJEEA.VIII.II.2020. 49-62.*

Biographical notes:
Vivek Kumar Varma Nadimpalli, Gopichand Agnihotram, *Wipro Technology Limited, Wipro CTO Office, Elec-tronic City, Bangalore – 560100, India..*

Pandurang Naik. *Wipro Technology Limited, Wipro CTO Office, Elec-tronic City, Bangalore – 560100, India.*

I. INTRODUCTION

Remote assistance systems will play a key role in the mar-ket for the ongoing pandemic and the relevance of these sys-tems has become necessary now. It is one of the crucial tasks that need to be addressed in the current situation, and a lot of research and development is taking place to improve existing systems and bring innovations into the smart remote assis-tance. One of the critical tasks that remote assistance are used is for repair or maintenance work. The pandemic has created a situation that needs remote expert systems in which the ex-pert of the device cannot reach the location of the equipment and needs to access and operate the device remotely to per-form the task. So, the expert will need help from the user. The expert will guide the user with steps to complete the necessary tasks. One of the significant problems of remote expert sys-tems is the knowledge gap between the expert and user of the device. The user may not be knowing entirely about the de-vice, i.e. every part of the equipment and how to re-pair/periodic maintenance of the device. The knowledge gap between user and expert can be filled by the implementation of object detection intelligence on both the sides of the streaming and augmenting the description of the detected object on top of that. For the expert, the object detection will help in finding the various parts of the devices and the faults parts in the device. The augmented description of the parts will help the user to identify the solution for repairing or maintenance task on the device. For the user, the expert's in-structions may be confusing due to the lack of knowledge on the equipment and lead to errors. The object detection and augmentation will help the user to identify the correct equip-ment to perform the instruction given by the expert.

For computing object detection, our approach consists of a deep learning powered method which robustly locates and identifies various equipment in an image/video stream and robust enough to handle any orientation, position, back-ground and other variants. Our primary aim is to port the sys-tem into smartphones. For this, we have developed two ap-proaches in the application of dialysis machine. For the first approaches, we used the region-based convolution neural networks for training the data. The model was trained on a GPU in a remote server and we have developed flask API to access the trained model. The client application which resides in the smartphone will send the frames from stream through the flask API to the remote GPU where the training model deployed. The training model will receive the image/ frames from the video stream and it will process the image to com-pute the location of object and object class. This information will have sent back from remote GPU server to the smartphones (mobile). The other approach was to train acces-sible mobile models on device data for object detection on smartphones (mobiles). We have trained various Single Shot Detection (SSD) models with the device data of the dialysis machine. The trained models are further processed to convert them into mobile accessible models and are ported into the mobile app. We compared these models based on different metrics such as inference time, precision and recall values.

As part of streaming we used WebRTC for real-time com-munication to mobile applications and web browsers. With the help of peer to peer communication, it will allow audio and video communication inside mobile apps. The WebRTC ap-plication has a wide range of support from various entities, like Android, iOS. We can integrate streaming and object de-tection applications and port into smartphones. We have built the mobile applications for both Android and iOS mobile de-vices using android studio and Xcode, respectively.

This paper is organized as follows, section II explains the related work on different models available for object detec-tion and the applications of the artificial intelligence on re-mote collaborations and solving the remote tasks with expert help. The section III and its sub-sections, discusses about remote assistance approach, setting the streaming application and object detection along with the Augmentation. This sec-tion also discusses about the various approaches for object detection for the system and training the models based on the data. Section IV discusses the results of various models and approaches used for object detection on dialysis machine. This section also discusses about the inference obtained from the results and trade-off based on application use. Section V describes the conclusions and future enhancements followed by References section VI.

2. RELATED WORK

Tsung-Yi Lin et al., [1] proposed object detection using deep learning approaches which will take input and passed to a neural network model that was trained on a dataset of imag-es. The model will process the image and detect the objects and the locations. The paper used Convolutional neural net-work architecture to train on the dataset, which will be used in many applications based on images and videos. The authors described that the object detection will be done by initial train-ing with the proposed regions, which will have sent to neural network. The neural network will classify the proposed region.

Joseph Redmon et al., [2] discusses about object detection approach. You Only Look Once (YOLO) framework object detection is a regression problem to spatially separated bound-ing boxes and associated class probabilities. This paper dis-cusses about a single neural network predicts bounding boxes and class probabilities directly from full images in one evalua-tion.

Ren, Shaoqing, et al., [3] discusses about a method in which, object detection networks depend on region proposal algorithms to hypothesize object locations. The paper intro-duced a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network. Thus, enabling nearly cost-free region proposals. RPN is a ful-ly convolutional network that simultaneously predicts object bounds and object scores at each position. The method also merged with Fast RCNN to produce better performance.

Shifeng Zhang et al., [4] described filter out negative an-chors to reduce search space for the classifier, and coarsely adjust the locations and sizes of anchors to provide better initialization for the subsequent regressor. Authors designed a transfer connection block to transfer the features in the an-chor refinement module to predict locations, sizes and class labels of objects in the object detection module.

Howard, Andrew et al. [5] proposed a method based on a streamlined architecture that uses depth-wise separable con-volutions to build light weight deep neural networks. Authors discussed two simple global hyper-parameters that efficiently trade-off between latency and accuracy. The hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. To improve the accuracy on SSD MobileNet, Mo-bileNet FPNs can be trained.

Kirillov, Alexander et al. [6] exploited the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. It pro-poses a top-down architecture with lateral connections which is developed for building high-level semantic feature maps at all scales. To further improve the performance ResNet50 has been used [7]. He, Kaiming et al. [7] presented a residual learning framework to ease the training of networks that are substantially deeper. The method explicitly reformulates the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions.

Lapointe et al. [8] discusses about the remote collabora-tions such as remote guidance of field tasks using Augmented Reality (AR) technologies. Authors provided reviews of multi-ple literature describing AR-based remote guidance tasks and discusses the task setting, technical requirements and discuss-es the multiple applications in this technology. He describes how the artificial intelligence algorithms helps the efficiency in the applied tasks of remote assistance.

Huang et al. [9] discusses a collaboration scenario involving a remote helper guiding in real time while a local worker per-forming a task. The authors proposed a novel approach to improve the process of collaboration but also improve the special awareness of the remote participants. The authors developed a 3D system based 3D shared space and 3D hand gestures called HandsIn3D. This system uses a head tracked stereoscopic HMD that allows the helper to be immersed in the virtual 3D space of the workers workspace. The system captures the 3D hands of the helper and fuses the hands into shared workspace.

Chen Alvin et al. [10] describes a portable robotic device capable of introducing needles and catheters into deformable tissues such as blood vessels to draw blood or deliver fluids autonomously. These Robotic cannulation predictions are driven by a series of deep convolutional neural networks that encode spatiotemporal information from multimodal image sequences to guide real-time serving. In this paper, the authors demonstrated image based robotic tracking studies which has the ability of device to segment, classify, localize, and track peripheral vessels in the presence of anatomical variability and motion.

Zillner et al. [11] describes an augmented reality remote col-laboration system leveraging high fidelity and dense scene reconstruction for intuitive and precise remote guidance. A local worker can use this system to automatically generate a 3D mesh of the surrounding and stream it to a remote expert. The remote expert can navigate and explore the reconstructed environment independently of the local worker in six degrees of freedom. The text and image based annotations can be placed in the scene the expert can draw the strokes on surfac-es intelligently which can help the local worker. The recon-struction allows the remote expert to segment colored objects from the mesh and use the 3D model to create simple anima-tions to convey precise instructions to the local worker.

Chen et al. [12] demonstrated the utility of augmented real-ity microscope in the detection of metastatic breast cancer and the identification of prostate cancer with latency compat-ible in real time use. The augmented reality microscope will remove barriers towards the use of AI designed to improve the accuracy and efficiency of cancer diagnosis. Thus, the artifi-cial intelligence and augmented reality provides the cost effec-tive and quality healthcare to the users.

3. SOLUTION APPROACH

This section describes the deep learning based approaches for objects detection and object detection models are de-ployed in mobile devices to detect the objects. This infor-mation will help the user to interact with the expert in guiding the machine activities such as repair or periodic maintenance activities.

The solution architecture diagram and the detailed explana-tion of each module is given below.

The proposed solution approach uses peer to peer commu-nication based streaming WebRTC API which builds the communication between user and the expert. The Deep Learning based approaches are used for training the models for object detection module and the solution is ported in the android and iOS mobile devices. The augmentation libraries like ARKit, ARCore are used for tracking the objects and augmenting device information on top the detected objects. The solution architecture diagram has shown in the Figure 1 and the details of solution approach has explained below.

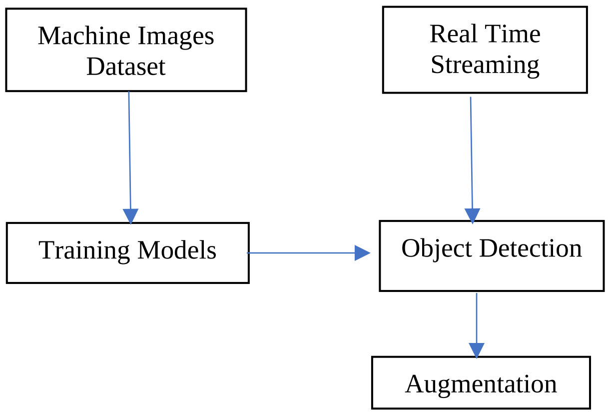

Figure 1: Solution Architecture

In the first step, the model is trained using deep learning ap-proaches on huge machine image datasets for detecting the various parts/objects in the machine. Once the object detec-tion has been trained the model will be ported into the android and iOS devices. The object detection will be ported in two different approaches. One of the approach is to train the deep learning model and deploy the model on the GPU. The model will be accessed using Flask APIs. The other approach is to train deep learning models and convert the model into mobile accessible models and port into mobile application.

We have trained our solution with dialysis machine to test our approach. Collected the images of the dialysis device by taking images of devices at different orientations. As part of this method we have Collected 1079 images which includes details about the various parts/objects in the device. We have Annotated all the images using OpenLabelIMG annotation tool and created the XML annotation file in Pascal VOC for-mat. The object detection is trained with deep learning models on annotated data. The object detection is built on the streaming module that will stream audio and video data from user side and expert side. The object tracking and augmenta-tion module is built over object detection module that will help the expert to guide the user in performing a repair and maintenance work. The details of streaming module, object detection module, augmentation module will be explained in further sub sections.

A. *Object Detection*

In this subsection, we have trained the model with two ap-proaches as part of object detection. As one approach de-pends on GPU based training and detection will happen at client level (mobile device) remotely. Another approach is based on MobileNet where training will happen with lesser layers at GPU. This solution is optimized and ported into mo-bile device for object detection. In this approach, we don't have to connect to the GPU server remotely as detection will happen locally in mobile device.

Approach 1: Training and detection Remotely

Training: The data collected is divided into 80% training, 10% validation and 10% test data. Training the object detec-tion using deep learning model will take a lot of computation power and time. So, we are performing transfer learning on our device data. Here we are using Yolo model and Tensor-Flow object detection models on dialysis machine data and compared the results on both the methods. In TensorFlow object detection, the model was finalized with the Region based Convolutional neural networks (RCNN) which was trained on COCO dataset since the model has good inference time and mAP values on COCO dataset for transfer learning. For Yolo model the Darknet model has been used.

Detection Remotely:

As part of object detection, we have collected 1079 images of the dialysis machine at different orientation. As the deep learning models need higher data, so used image augmenta-tion techniques like random flip, rotation, adding blurriness on images

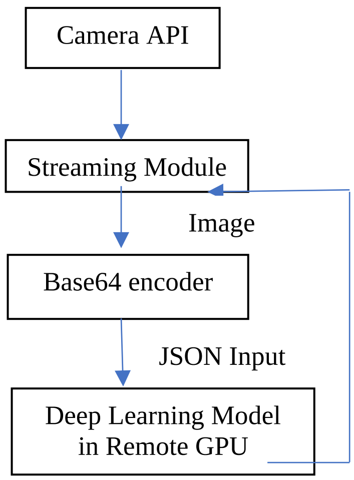

Figure 2: Input data processing for detection in Deep Learning Model

to increase the size of the dataset. The data set was annotated with 59 labels using OpenLabelIMG library in pascal VOC format.

The input data processing steps for detection will be explained in Figure 2. For this approach, the input data will be a JSON file that consists of the image data which is converted to the base64 format. The smartphone will have the stream of the video of the device. The stream will be divided into individual frames. These frames will be sent to the remote server for object detection. But we faced lag due to uploading the image/frame directly. So, we scripted a module that will convert the image/frame into base64 format. Once, the base64 format of image is received from the module, the app will send a JSON request to remote server with the base64 image data in it along with a unique frame id. The flask API which will be deployed in the remote GPU server will receive the JSON file and convert the image in base64 format to NumPy format. This will be further passed to the trained Deep Leaning model that will be able the detect the object and compute the location of the object in the image (ROI's: Region of Interest).

The input of the JSON:

{
FrameID : "1",
ImgData: "BASE64 IMAGE DATA"
}

The model will take the input image of size 300 x 300 so that the input image will be compressed to that format and will be sent to the trained deep learning model. The model will take compressed image as input and will give a vectorized output which will contain the label information that it has detected also the location of the object with respect to the image. The model is described as a function

$$y = f(x)$$

x – base64 encoded data of the image
f - the trained deep learning model
y – list of ROIs – [[label_1, x_{min}, y_{min}, x_{max}, y_{max}], [label_2, x_{min}, y_{min}, x_{max}, y_{max}]]

These computed data will be further converted into JSON file which have the information of all the detected labels and its location along with the frame ID. The JSON file will be sent back to the conversion module in smart phone, that will convert the JSON file into object detection ROI's on the image and further call the augmentation of description of each label that are detected on the image using ARKit or ARCore. In this way, the object detection can be seen on both the side of user and expert.

The output of the JSON:

{
ROI: [[label, x_{min}, y_{min}, x_{max}, y_{max}]]
}

Approach 2: Training and detection Locally

Training: The data collected is divided into 80% training, 10% validation and 10% test data. The model which will be trained should also be converted into mobile accessible models and these models are efficient even in low computation power environments. SSD (Single Shot Detection) models were chosen since these models were computation lite models and good on accuracy when trained on COCO dataset. These models were also easy to convert into the mobile accessible versions. As per the use case, trade-off of accuracy and inference time should be computed to select a preferable model. We chose SSD MobilenetV2, SSD MobileNet FPN, SSD Resnet50 models and we have trained dialysis data using these models. We also found best model which suits for our datasets. Alternatively, we have also tried various values for hyperparameters like learning rate, batch size, momentum to achieve the best model with the datasets.

Detection Locally:

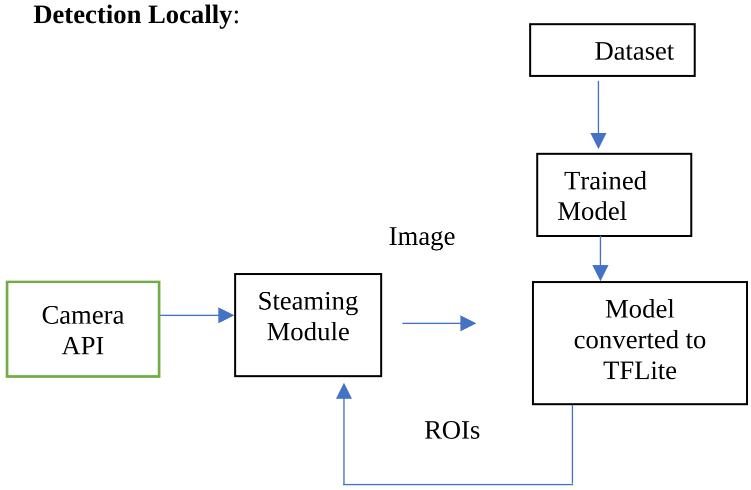

Figure 3: Local Detection using TFLite Model

The local detection using TFLite model is explained in Figure 3. In this approach, we will train the deep learning model using machine data. But the trained deep learning model will require heavy computation power and making it difficult to run the trained model on all devices. Most of the smartphones does not have the necessary resources to run the model except few high-end phones. These high-end phones may run the model, but the time taken for the computation will not be suitable for the remote assistance use case. Therefore, the deep learning model is converted to mobile accessible models for detecting the objects faster. This conversion will result in decreasing the accuracy slightly, but it will become more efficient even with low resources in the smart phones. Hence there is a reasonable trade-off to take for porting a deep learning model into the mobile app directly. The streaming module will receive the image/frame from the camera API on the user side which will be sent to the expert also. The trained TensorFlow model further optimized and converted into TensorFlow model file using TFLite. The .tflite file is ported into android mobile using android studio (see in Figure 4) and in iOS mobile devices the .tflite file ported using Xcode (see in Figure 4). The converted model file (.tflite) will be deployed on the streaming API at user side and expert side. The images streaming module on both the side of user and expert will be sent to the converted TFLite model for ROIs (Region of Interest) and class label prediction.
The model works as a function as described

$$y_1 = f_1(x_1)$$

x_1– base64 encoded data of the image
f_1 - the trained deep learning model
y_1– list of ROIs – [[label_1, $x_1$min, $y_1$min, $x_1$max, $y_1$max], [label_2, $x_1$min, $y_1$min, $x_1$max, $y_1$max]]

In this way, the object detection works on local devices faster and better predictions. Once the object detection is completed, the augmentation module built using ARCore and ARKit over the object detection module. We will use the ROIs and detected labels to perform augmentation that will help the expert to guide the user on various tasks and helps the expert with the description of all the detected labels augmented.

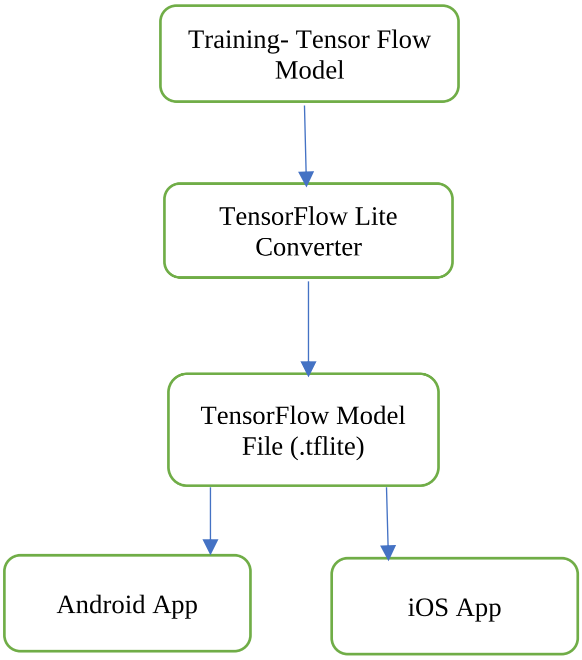

Figure 4: Local Detection on Android and iOS App

B. Streaming Module

The Streaming Module is the first step for the remote assistance system where the Streaming module will send the audio and video information from user side to expert side and expert to user. On the User side, the mobile application will use the camera API, to collect the images data. This data collected will be useful to expert for analyzing the device information/device state and guiding the user to perform the repair or periodic maintenance tasks. Therefore, the image data must be streamed to the expert along with the voice of the user. Similarly, the expert instruction to the user must be streamed from expert side to the user side. The expert side does not need any camera API as the image data comes from the stream. The expert side and user side solution features are shown in below figures 5 and 6.

To build Streaming module we have used the WebRTC streaming API for mobile phones. The module on expert side will capture the image and using the WebRTC API for peer to peer communication the image along with microphone will be encrypted on the user side and will be sent to the expert side mobile application. The data will be decrypted, and content will be shown on the expert side mobile application. The application will also show the objects detected on the device using object detection module. Similarly, at expert side the voice data will be captured and encrypted by the APIs and will be sent to the user. The streaming module was built and could be ported into the mobile app on both expert side and user side. The module was tested around various environments such as internet speed, image data size, frames per seconds and by varying various hyper parameters learning rate, batch size, number iterations. In the dialysis application, on normal 3g data, 640 x 480 size image could be captured and streamed with a speed of 10 frames per second.

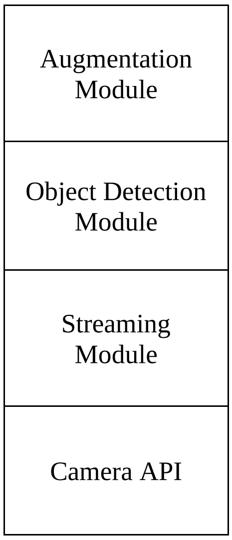

Figure 5: User Side Features

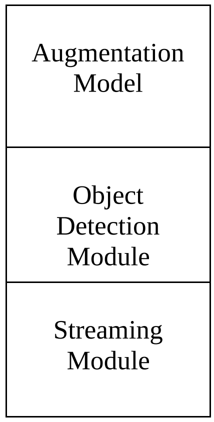

Figure 6: Expert Side Features

C. ***Augmentation***

The augmentation is the last step on the remote assistance system where the information about the device will be augmented and shown to the user and expert. Initially, the information that needs to be augmented is the description of the labels (machine parts) and device information. In our application, the information about the dialysis machine and its parts must be stored in the database and this information can be leverage along with object detection module. In the object detection module, the last or lower layer will predict the labels of the data and its location (ROIs) and this information will be send to the augmentation module. Based on the ROIs, the augmentation module will augment the description that has been stored in the database. These augmentations will help expert and user as they have the description of the labels and track the location of the object. The augmented information can also include the periodic maintenance details and repair details such as which part has anomaly or which part has to repair. The below figure 7 depict the Augmentation process on mobile devices.

For Android devices, we have used ARCore libraries and for iOS devices, ARKit libraries are used for Augmentation. Using the libraries, we have imported a 3d arrow figure which is designed in unity to provide the information about the devices and its parts. Once the ROIs (Region of Interests) from object detection module are received, the arrow will be augmented on the device along with the label name that is predicted. If user or expert needs to know more information about that labels, they will click on the arrow which will augment the description that is stored in the database. Once the objects are detected, the tracking of the labels will be started using the same ARCore and ARKit libraries. Therefore, even if the orientation changes, the augmentation will still visible and help the user with the repair or maintenance of device.

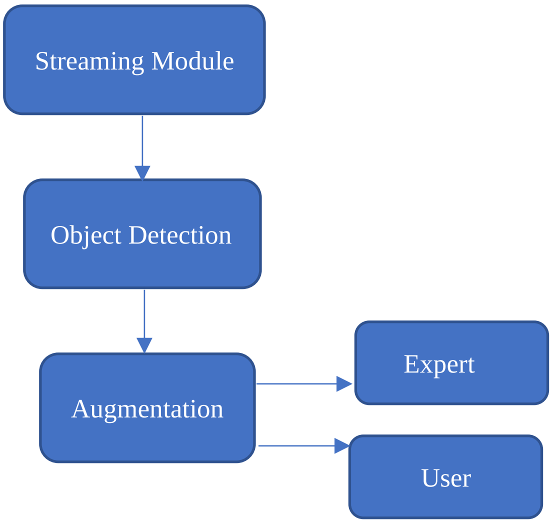

Figure 7: Augmentation process on Mobile Devices

4. APPLICATION

We have trained model with the dialysis machine (Given in Figure 8) on annotated data as explained in Section III which has 59 labels such as blood pump, optical detector, groove, gear box without motor, venous measuring head etc. The total image data set is 1079 labels and we have augmented on image data set and in total we have 6000 annotated image datasets. In each label, images are varying from 400 images to 1000 images.

Figure 8: Dialysis Machine

We have trained with different object detection methods as explained in Section III with different model architectures for the dialysis data. The results of training and inference of models for different approaches are given below.

In approach 1 (as explained in Section III, Remote detection), Yolo method with darknet architecture and TensorFlow object detection (TFOD) API with Faster RCNN model architecture was trained on the dataset of the dialysis machine. The faster RCNN used the inception ResNet v2 as the feature extractor. The results of the training were given in below table 1.

	YOLO	Faster RCNN
mAP @ .75IOU	64.3	73.38
Loss at convergence	1.8	1.2
Inference time	0.32s per frame	0.65s per frame

Table 1: Approach 1, Yolo Vs TFOD

The trained model was deployed on remote GPU server which has Nvidia k80 Tesla. From client side (mobile) we will connect to the remote GPU to obtain the object detection and the inference time of the models as stated in Table 1. But the bottle neck was the JSON file that must be sent to the remote GPU and receive the ROI's as the response from the GPU to the client.

Using approach 1, from the results of the object detection Yolo model has better inference speed than Faster RCNN model but RCNN model has better accuracy than YOLO model for the dialysis machine dataset. But there was significant lag of sending the JSON file containing the image data from mobile app to remote GPU and receiving the ROIs from remote GPU to client using mobile app. On an average, it was taking 2 seconds for the network lag making total inference time for YOLO model equal to 2.3 seconds and faster RCNN model equal to 2.6 seconds. But due to the network time lag, the response time may also go up significantly. But with respect to accuracy YOLO model and Faster RCNN was significant. Also, when tested on images YOLO model gave more false positives than Faster RCNN. Therefore, for the dialysis machine dataset, the Faster RCNN model was chosen and has been deployed in remote GPU server and used for object detection module in the remote assistance mobile application.

In approach 2 (as explained in Section III, local detection) the dialysis data is trained with the SSD MobileNetV2, SSD MobileNetV1 FPN, SSD ResNet50 were trained and the results of the three models are provided in Table 2.

As explained in Section III, these trained models were converted into mobile accessible models using TFLITE libraries. The .tflite file is ported in to the Android and iOS mobile devices for local detection. The models were tested on Redmi k20 (Android) and iPhone XR (iOS) devices. The inference time of these models are given in Table 3 below.

In approach 2, the mobile accessible deep Learning models, the SSD MobileNetV2 model on dialysis machine was performing good with low inference speed, but the accuracy when compared to other models was significantly low. The SSD MobileNetV1 FPN has better accuracy than MobileNetV2 but the inference time was increased. The SSD ResNet was theoretically thought to be best performing as it has better mAP value than MobilenetV1 FPN when trained on COCO dataset, but on the dialysis machine dataset the SSD ResNet was having mAP values slight lower than the MobileNetV1 FPN but inference time was more. Another observation is that, these models performed well on iOS device than on Android devices. For the dialysis application, MobileNetV1 FPN was chosen for the object detection and real time guiding of user with the expert help on machine repair and maintenance activities. In general, these models are chosen based on accuracy and inference time between MobileNetV2 and MobileNetV1 FPN.

	MobileNetV2	MobileNetV1 FPN	ResNet50
mAP@ .75IOU	0.57	0.71	0.69
mAP@ .5IOU	0.79	0.88	0.87
Average Recall	0.54	0.7	0.67
Steps for Convergence	40,000	80,000	60,000
Loss at convergence	1.8	0.41	0.42

Table 2: Approach 2, MobileNetV2 Vs MobileNetV1 FPN Vs ResNet50

	MobileNetV2	MobileNetV1 FPN	ResNet50
Android (Redmi k20)	250 ms	1250 ms	1800ms
iOS (iPhone XR)	50ms	650ms	1050ms

Table 3: Approach 2, Android Vs iOS devices detection time

5. CONCLUSIONS AND FUTURE WORK

The solution provided in this paper discusses remote assistance system that will enable the expert to guide the user to perform a repair or periodic maintenance tasks on the machine. The method has streaming module as the first step which will send the stream of audio and video data to the expert side and audio data from expert to user side. This module was built on the APIs from WebRTC peer to peer communications. On the streaming module, object detection module was built to guide the user and expert by detecting the labels from the video stream. Various models and approaches were followed which were compared and based on accuracy and inference time. The models were finally chosen based on trade-off between the inference time and accuracy. The solution approach is deployed on dialysis machine to detect different objects/parts of the machine and guide the user for repair or periodic maintenance of dialysis machine. The final step of the solution was the Augmentation, which will enable image tracking and necessary augmentations can be projected which will help the user and expert. We will be using ARCore and ARKit libraries for tracking and augmentations. All these modules were built, integrated, and ported into iOS and Android devices to guide the user and expert on machine activities.

In future works, the action and scene prediction on images can be added to the object detection intelligence which will point out the anomalies or errors of machine. We will also fuse the sensor information along with the image information for better action prediction. We will port the complete solution on hands free devices such as HMD (Head Mounted Devices) like google glass, HoloLens etc. This solution can also be extended to other domains like remote training where the user will be learning practically about the device from the expert. It can also be extended to consultancy in healthcare, retail, and e-commerce domain.

REFERENCES

[1] *Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie*; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117-2125

[2] *Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi*; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788

[3] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99. 2015.

[4] *Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, Stan Z. Li*; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4203-4212

[5] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[6] Kirillov, Alexander, Ross Girshick, Kaiming He, and Piotr Dollár. "Panoptic feature pyramid networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6399-6408. 2019.

[7] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[8] Lapointe, Jean-François, Heather Molyneaux, and Mohand Saïd Allili. "A Literature Review of AR-Based Remote Guidance Tasks with User Studies." In *International Conference on Human-Computer Interaction*, pp. 111-120. Springer, Cham, 2020.

[9] Huang, Weidong, Leila Alem, and Franco Tecchia. "HandsIn3D: supporting remote guidance with immersive virtual environments." In *IFIP Conference on Human-Computer Interaction*, pp. 70-77. Springer, Berlin, Heidelberg, 2013.

[10] Chen, Alvin I., Max L. Balter, Timothy J. Maguire, and Martin L. Yarmush. "Deep learning robotic guidance for autonomous vascular access." *Nature Machine Intelligence* 2, no. 2 (2020): 104-115.

[11] Anoop Joyti Sahoo, and Rajesh Kumar Tiwari “A Novel Approach for Hiding Secret data in Program Files” International Journal of Information and Computer Security. Volume 8 Issue 1, March 2016,

[12] Abu Salim, Sachin Tripathi and Rajesh Kumar Tiwari “A secure and timestamp-based communication scheme for cloud environment” Published in International Journal of Electronic Security and Digital Forensics, Volume 6, Issue 4, 319-332.

[13] Rajesh Kumar Tiwari and G. Sahoo, “A Novel Watermark Scheme for Secure Relational Databases” Information Security Journal: A Global Perspective, Volume 22, Issue 3, July 2013.

[14] Zillner, Jakob, Erick Mendez, and Daniel Wagner. "Augmented reality remote collaboration with dense reconstruction." In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 38-39. IEEE, 2018.

[15] Chen, Po-Hsuan Cameron, Krishna Gadepalli, Robert MacDonald, Yun Liu, Shiro Kadowaki, Kunal Nagpal, Timo Kohlberger et al. "An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis." *Nature medicine* 25, no. 9 (2019): 1453-1457.