International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-3S, February 2019

Phonic Eye for the Visually Impaired

Meenatchi KV,Mahima S,Poornima T, P Kumar M.E., Ph.D.

Abstract: ***Visual disability is a physical abnormality that deters a person from performing day to day activities due to the lack of awareness about the surroundings. The greatest challenge faced by the visually impaired people is to identify the things around them and they need to be dependent on other people when they require usage of a particular product or identifying any obstacle on their path. Though there are numerous applications built for the visually impaired, the goal of the proposed solution is to create an interactive mobile application based on the object detection technology to report the user about the vicinity. This mobile application serves as an effective tool in enhancing the experience of the user by capturing the image of the scene in the environment and identifying the objects in the frame with the application of Convolutional Neural Networks based on faster Region based Convolutional Neural Network model and the results are delivered to the user in the form of speech signals thereby enabling the users to be self-reliant and phonetically observe the locale.***

Keywords: ***Tensorflow, faster_rcnn, flask, TFServing, TTS, Blind, Visually Impaired, Object Detection***

I. INTRODUCTION

According to the survey conducted by the world health organization, around the world approximately 1.3 billion people live with some form of vision impairment and 36 million people are completely blind. People suffering from visual impairments have to depend on others for performing their everyday activities. The most arduous thing faced by them is determining the type of objects present in front of them. Guide cans help people only to detect the presence of an obstacle on their way and they do not describe the obstacles. In the recent years, with the advent of video surveillance, autonomous vehicles, facial recognition and people counting applications in crowded areas, fast and accurate object detection systems are up surging in demand. Object detection is a very powerful aspect of computer vision .Object detection refers to the potential of the software systems to locate objects in a scene or image and to identify each type of object regardless of its size and orientation, thereby increasing the awareness with regards to the surroundings.

LITERATURE SURVEY

In Aabi A's proposal [1], the spy camera is used for capturing the objects. Histograms of the images are compared with the one in the database to perform recognition. The histograms are generated by converting the captured image to grey scale and then applying Local Binary Pattern algorithm to recognize the objects. The output is conveyed to the user via Bluetooth speaker. Aabi's work identifies the objects which are a part of the MATLAB database.Various image processing algorithms were used, like the use of Principle Component Analysis (PCA), proposed by J Prakash [6] or the use of Thresholding techniques for background - foreground separation and extracting the necessary objects [2]. But these techniques are not much efficient when compared to the Neural Networks based classifiers.Some systems like the one proposed by VikkyMohane [3], the text patterns are localizedusing SIFT technique instead of Optical Character Recognition (OCR). Camera based system which will help blind person for reading text printed on hand held objects and converts to speech output. Rui (Forest) Jiang's [6] work uses YOLO model based on Neural Networks for object detection, but requires a portable camera like GoPro Hero 3 to capture images. Though object detection has been done by various techniques, there are some or the other fall backs. Some techniques makes use of a database, which limits the number of objects the system can detect while the other makes use of outdated techniques for image processing. Usage of microprocessors like Raspberry pi may limit the processing capability of the technique, while other efficient object detection techniques failed to cater visually impaired.

II. PROPOSED SYSTEM

The proposed system, PHONIC EYE, enables the visually impaired to carry a visual assistant along with them in their smart phone. PHONIC EYE can be invoked by using a voice command, which opens the camera to take photo. This image contains the objects which the user want to know. The image is sent to the server for further processing and the output is delivered to the user via speakers. The server contains two modules, the first one is a Flask server, which is used to receive the input image and reply back the objects in the image to the user. While the second module is the Object detection server which is served using TFServing, for detecting the objects in the image. The TFServing serves Faster_RCNN model trained on the COCO dataset. The COCO dataset is a collection of 90 objects, each containing at least 300 images. The Faster_RCNN is a CNN based Neural Network model for Object Detection.

Manuscript published on 28 February 2019.
* Correspondence Author (s)
Meenatchi KV,Student, Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, TN, India. (meenatchi.kv.2015.cse@rajalakshmi.edu.in), +91 7397281875
Mahima S,Student, Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, TN, India.mahima.s.2015.cse@rajalakshmi.edu.in, +91 8668095051
PoornimaT,Student, Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, T.N, India. (poornima.t.2015.cse@rajalakshmi.edu.in) +91 9444185366
Dr. P Kumar, Professor, Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, TN, India (kumar@rajalakshmi.edu.in), +91 9840573702

A region of interest (ROI) is generated by a Region Proposal network before performing image classification. YOLO model is faster than Faster_RCNN but lacks accuracy. This is the reason why the Faster_RCNN is more efficient than other models.

III. SYSTEM OVERVIEW

The system consists of the following modules as shown in the *Figure 1*.

1. Mobile Application
2. Server
   a. Flask Server
   b. TFServing Server

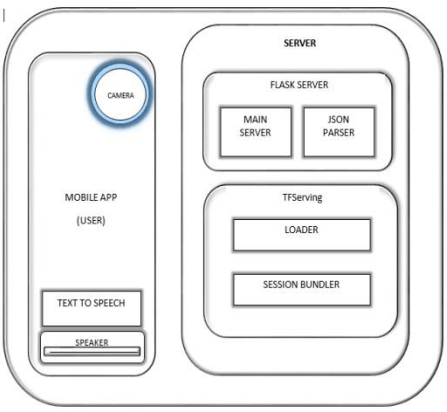

Figure 1. Phonic Eye Architecture

Figure 2 represents the use case of PHONIC EYE and the flow and order of data transfer in the system. First, the Mobile Application captures the image and sends it to the flask server as base64 string. Then the flask server receives the string, converts it back to image and forwards the image to the Object Detection model in the TFServing as a numpy array. The TFServing responds back to the flask server as json response. The flask server parses the json response and extracts the required result. Finally, the Mobile application receives the list of objects as a string from the flask server and conveys the output via the speaker by using an efficient TTS module.

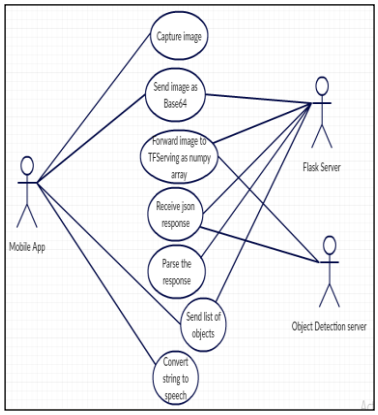

Figure 2. Use case of PHONIC EYE

OBJECT DETECTION USING FASTER_RCNN

Steps followed by Faster_RCNN for detection of objects is as follows:

1. Take an input image and pass it to the Convolution Network which returns feature maps for the image.
2. Apply Region Proposal Network (RPN) on these feature maps and get object proposals.
3. Region of Interest pooling layer is applied to resize.
4. The proposals are passed to a FC layer to generate bounding boxes and to identify the objects in the image.

The captured image is an array of pixels which are represented as h(height)*w(width)*d(depth). The image will then pass through Convolutional layers with filters, RPN, ROI Pooling layers, fully connected layers and softmax function for classification.

Convolution layers with filters/ kernals

Features are extracted from the image that is fed as input. It takes two inputs in the form of image matrix and a filter/ kernal. Convolution of an image with different filters can perform operations such as Histogram equalization, Canny edge detection, normalization and Gaussian filter is applied to the image which sharpens the picture and reduces noise. According to *formula 1*, the neural network accepts the image (h x w x d) as input and computes fixed feature map Y by convoluting it with various filters of size (f_hx f_wx d). The Convolution of a sample image is shown in the *figure 3*.

$$\textbf{Output } Y = (h - f_h + 1) * (w - f_w + 1) \qquad 1$$

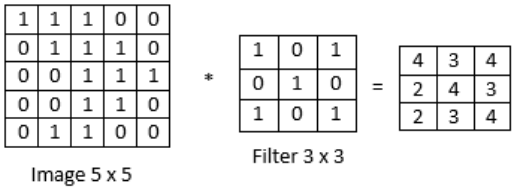

Figure 3. Sample Convolution of Image and Filters
RegionProposalNetwork

At the end layer of the convolutional neural network, a 3x3 sliding window moves across the feature map and maps it to a 256-d dimension. For each sliding-window location, *multiple* possibleregions, anchor boxes or the bounding boxes are generated**.** Each proposal consists of an accuracy score and coordinates of the anchor box.

ROI Pooling Layers

Pooling is introduced to perform down sampling and to send only the required data to the CNN layers. It is a type of max-pooling where its pool size is dependent on the input, so that the output always has the same size.

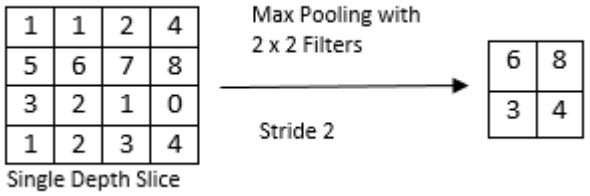

Figure 4. Sample Max Pooling of a Single slice of image.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Hence the fully connected layer always expects the same input size. The input of the ROI layer becomes the proposals. *Figure 4* shows an example of Max Pooling with 2 x 2 filters and Stride 2.

Fully Connected Layers

FC's are layers of neurons at the end of CNN. These neurons have full connections to all activations in the previous layer. It computes the exponential sum of the weights and fixed feature map. The input to the fully connected layer is the falttened matrix of the image. The flattened matrix is generated by converting the n-dimensional matrix to a single dimension. Output from the FC layer is fed to an activation function like Softmax function.

Activation Function

Apply Softmax function*formula 2* to classify objects with probabilistic values between 0 and 1. As mentioned in *formula 2*, given an input parameter, objective is to predict if the trained set of features with X each having its own weight and the matrix consisting of binary values.

$$Y = softmax(X * W + b) \qquad 2$$

Where Y - predictions, X - images, W - Weights, b - Biases

As mentioned in *formula 3*, Rectified Linear Unit (ReLU) introduces non-linearity in the CNN.

$$f(x) = max(0,x) \qquad 3$$

Training the Data

The pre-trained Faster RCNN model was trained with COCO dataset. The COCO dataset is a collection of 90 objects each containing at-least 300 images. Once we have sufficient images available, the next step is to annotate those images to describe the specific object in that image. For this purpose, a graphical annotation tool called LabelImg is used where the output will be in the form of Pascal VOC XML format. The images along with their XML files must be placed in the test and train folders for testing the data and training them respectively. The XML files are to be converted to CSV format. Once CSV files are ready, TFRecords are generated from the CSV files. Next is the creation of a label map where each object is identified by mapping the class names to class ID numbers. Either new models are created or existing models are configured to trainthe data. When the training is successful, inference graph can be extracted to perform object detection.

Steps involved in training the data:
1. Prepare the model.
2. Set the path to inference graph.
3. Set number of classes used in the training.
4. Load the label map.
5. Use frozen checkpoint to run a test feed.

IV. DESIGN AND IMPLEMENTATION

Android Device

The user's mobile application is developed using Android Studio. The initial layout is the camera screen which is opened through the voice prompt by the user. The image is captured at a particular position and the captured image in jpg/ png format is encoded to Base64 string and is immediately sent to the server. The response from the server is received by the client's application in the form of string and is fed to the Text-To-Speech Engine. Thetext-to-speech engine tokenizes the text into the written words and then the synthesizer converts the words into sound. The speech signal is delivered to the end user through the mobile device's speaker.

Object Detection Service

The server receives the encoded Base64 string from the client's application and it is converted to numpy array in the flask server. The flask server is a micro framework for python and the flask packages are easily available on the public repository. The numpy array is sent as an input to the Convolutional Neural Network and where faster RCNN is used as the model and Region proposal network is applied to generate the region of interest based on which the class names and the accuracy detection scores are obtained. The model is served using TensorFlow Serving (TFServing) which is a flexible, high-performance system for ML models. It loads the model and performs session bundling to deliver the production to gRPC client. The results are written to a file in the server and the resultant string is sent to the user's mobile device for further process of speech synthesis.

V. EXPERIMENTAL RESULTS

User accesses the application in the android device with a simple voice command like "capture" or by a single click. The images are subjected to object detection process in the server and the output is displayed only when the accuracy is greater than a minimum threshold of 80%. It could even recognize the objects that are overlapping one another and it does not require much of focusing.

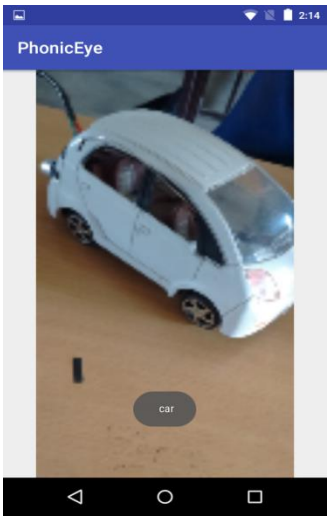

Figure 5. Sample output of PHONIC EYE App

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The obtained results include the object name which is converted to a voice prompt in the user's device. In case of multiple objects in the frame, then all the names will be delivered to the user with a time lag of about 0.5 seconds one after the other object. *Figure 5* shows the sample output of the PHONIC EYE App, where the predicted images are conveyed via speech.

VI. CONCLUSION

The proposed system acts as a phonic eye for the visually impaired people. Since it has been implemented with simple interactive mobile application, it meets the basic needs of the blind people thus making it an efficient and reliable system. This application provides greater affordability and scalability thereby enhancing the experience of the user and can also be customized as per the requirements.

REFERENCES

1. Aabi A, Dhivyalakshmi T. Joan Kanishka S, Ms.S.Jaipriya, "Hand-Held Object Recognition for a Blind Person Using Raspberry Pi", The International Journal Of Engineering And Science (IJES), Volume 5, Issue 4, 2016.
2. SarveshAthawale, Javed Ali, TejalBirajdar, Deepak Patil, Prof.SaurabhSaoji, "Object Detection in a Smartphone for Visually Impaired Users", International Journal of Advanced Research in Computer Science & Technology, 2015.
3. VikkyMohane,"Object Recognition for Blind people Using Portable Camera", World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), 2016
4. David Stutz, seminar report on "Understanding Convolutional Neural Networks", 2014.
5. J.Prakash, P.Harish, 3 Ms.K.Deepika, "Android based object recognition into voice input to aid visually impaired", International Journal of Advanced Technology in Engineering and Science , Volume No 03, Special Issue No. 01, March 2015.
6. Rui (Forest) Jiang,Qian Lin, ShuhuiQu, Standford University, "Let Blind People See: Real-Time Visual Recognition with Results Converted to 3DAudio", 2017.