INDIAN INSTITUTE OF INFORMATION TECHNOLOGY VADODARA
GOVERNMENT ENGINEERING COLLEGE, SECTOR-28
GANDHINAGAR, GUJARAT - 382028

Design Project Report

on

Raspberry Pi Based System for Object Detection and Recognition for Blinds

AALOK KUMAR (201551012)
AMAN CHOUDHARY (201551019)
MONISHA WAMANKAR (201551073)
SAKSHI DUBEY (201551096)

supervised by
Dr. Kamal Kishor Jha
and
Dr. Jignesh Bhatt

January 4, 2018

***Abstract*—Nowadays the demand of low cost well-trained embedded devices that can be applied in real world is increasing. Our efforts in this project are determined to achieve a system which can render one of such application to benefit the blind people. In this project video detection and recognition is presented based on a single board computer represented by Raspberry PI as an embedded solution. The aim is to make a smart system which detects the object for the blind user, measures its distance , and report the output in the form of audio signals to alert the blind user of the obstacle ahead. This entire work is done on raspberry pi with Raspbian (Jessie) operating system ported on it.**

Keywords: Raspberry Pi, TensorFlow, OpenCV, Ultrasonic sensor (HC-SR04), Arduino Uno

I. Introduction

People are witnessing the dawn of the new era of Deep Learning and embedded devices. Real life applications whether it be in the domain of Image recognition, Driver less cars, better preventive health care are all here today or on the horizon. The primary motivation which drives our team is how we can build a real-world application, in a cost effective manner which can benefit a section of society namely-”blinds”.

A. Problem Statement

One of the major problems faced by Blind people is detecting and recognizing an obstacle in their path. The projects approach lies in developing a system based on Raspberry Pi 3 , which is capable of labeling objects with the help of OpenCV and TensorFlow libraries and converting the labeled text to speech and producing output in the form of audio signals to make the blind person aware of the object in front of him. The scope also includes measurement of the distance of the object from the person and reporting the same. Most of the Object Detection algorithm has been tested on GPU with high computation abilities and are less likely to achieve same speed and accuracy with less powerful devices with microprocessor only, which are in high demand in current scenario. We choose Pi 3 as our platform because it is a standard representative of embedded device and is widely being used for devising low cost-system. We would like to have a prototype that can successfully perform real-time detection in about 5-10 fps on Pi, with decent accuracy.

B. Major Challenges

- Exploring OpenCV and TensorFlow libraries, discovering, selecting and applying their those aspects which will help in achieving our target (Object Detection and Recognition)
- Installing TensorFlow on Raspberry Pi, embedding the code in it and working with Raspbian (Jessie) OS.
- Making command over Advanced Linux Sound Architecture (ALSA), setting up Text-to-speech engine (eSpeak) and making use of alsa-utils package and alsamixer program to produce audio signals.

C. Hardaware and Software used

- Raspberry Pi 3 model B
- Zebion webcam
- Arduino Uno
- Ultrasonic sensor (HC-SR04)
- TensorFlow library
- OpenCV library

II. Literature Survey

Reference [1] was the first work we came across which was quite similar to our approach. It aimed at deploying a real-time object detection system that operates at high FPS on resource-constrained device such as Raspberry Pi and mobile phones.The primary use case was-detecting traffic objects, including pedestrians, cars, bus and so on. In this report the contributors tested a variety of detection models, including the state-of-art YOLO2 and concluded that current object detection methods, although accurate, is far from being able to be deployed in real-world applications due to large model size and slow speed.
Reference [2] was a IJAREEIE research paper which aimed at developing a video surveillance system based on Raspberry Pi. Authorized user can access to their monitoring system remotely via internet with the use a mobile phone and monitor the situation on application. If an intruder enters, the face of the intruder is captured with the help of camera and it is sent as an email to the users email-id. It is also sent as a text message to the user’s mobile so that the user can know that someone else is trying to use the system without proper authentication. The proposed system is simple in design, easy to install and is highly reliable.
Reference [3] is another research paper that aims at object detection in Raspberry Pi .The application is a desktop application in that client gives a summon to catch a picture. This picture is put as goal picture. The robot then catch picture and then the caught picture will be compared to goal picture. This utilizes region wise correlation for recognizing.

III. The Present Investigation

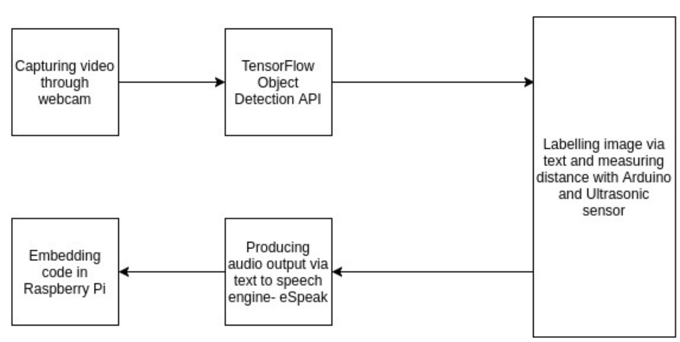

Fig. 1: Flow chart representing the methodology

A. Object Detection and Recognition

The procedure begins with installing required dependencies and environment on the system. Fig (2) depicts the image of the experimental system .Video is captured from the Webcam using OpenCV libraries. With an object detection model , not only we can classify multiple classes in one image but we can specify where the object is in the image with the bounding box framing the object. The TensorFlow models Github repository has a large variety of pre-trained models(of which we are using SSD) for various machine learning tasks and one excellent resource is their - Object detection API. It makes it extremely easy for your own object detection model for a large variety of different applications. API makes it easy to work on live stream high frames per second (fps) applications on high accuracy desktop model.

Fig. 2: Experimental setup

B. Producing output through audio jack via eSpeak

Raspbian OS uses Advanced Linux Sound Architecture(ALSA) for managing audio devices. We have to install a package to test sound device through ALSA i.e. alsa-utils. Thereby load sound driver. Then test audio configuration using amixer, volume can be adjusted using alsamixer utility.
Fig (3) depicts alsamixer volume adjustment.

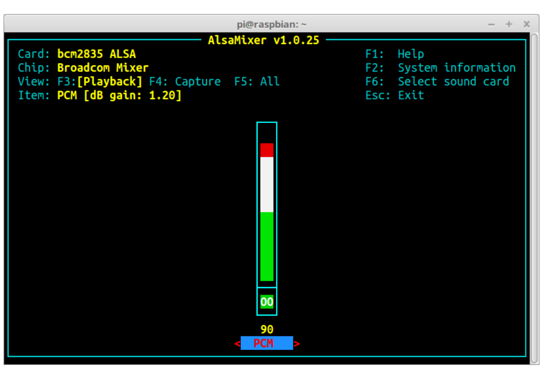

Fig. 3: Alsamixer volume adjustment

C. Training and Testing for our own custom object

TRAINING

Collection of dataset is a major issue to create a good dataset for training purpose, so first step is collecting images and labeling them with any existing software(in our case - LabelImg).This software actually bound those pixels of image where object is situated. At the end of this label process we get XML files and we have to convert those XML files of images into CSV file format. CSV files actually contain the object's X and Y coordinates with labels.
These csv files are required for generating TFRecord file that contains all the classes with corresponding labels and gives the machine readable binary file. Their are two important files that must be required for training on the dataset

1) Config file
2) LabelMap file

A config file actually contains total number of classes present in the training dataset. For a config file the input is TFRecord file and also we can control here the number of iterations in training of object.
Another file is LabelMap (pbtxt extension) that contains a dictionary format for wrapping labels with their unique ids. After this we train the model with GPU (in laptop) and CPU (in Raspberry Pi) and by using Tensorboard we see the loss graph. We stop training when the graph converges to loss of about 1 (or lower) on average.

TESTING

In order to use the model to detect things ,we need to export the graph and put the test images for testing. After some modification in image detection source code we are able to capture and detect objects successfully in live stream video from Webcam.

We trained and tested the system for Zebra crossing and states of traffic light-red, yellow, green.

IV. Results and Discussions

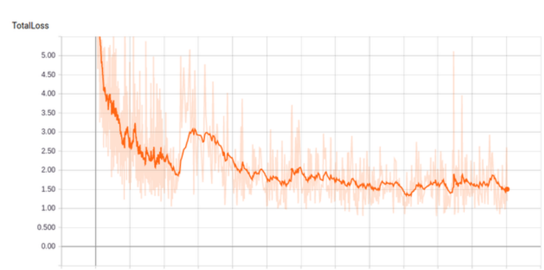

Fig. 4: Total loss graph

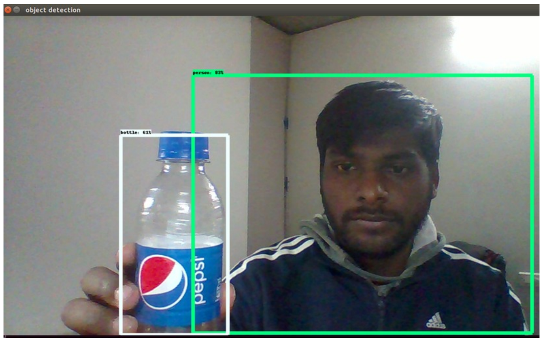

Fig. 5: Object Detection

Fig. 6: Zebra crossing recognition

Fig. 7: Yellow state recognition of traffic light

Fig. 8: Red and Green state recognition of traffic light

Evaluation Criteria

1) FPS - Frames per second is the number of consecutive images that appear on a display. FPS depends on computational ability and hardware.The larger the FPS , the lesser the delay.
2) mAP - The mean average precision has to be large but if you increase precision system model will perform slow or it takes more time to converge.
3) Model size - Model size is neither too bulky nor to simplified.
4) System cost- System cost is reduced to a great extent since we are trying to approach the problem with a simple cost-effective embedded approach and CPU rather than heavy machinery and GPU .

The project has contributed a foundation to work on embedded devices for object detection. It has contributed a lot to our knowledge as beginners in the fields of deep learning and embedded devices. From using the TensorFlow and OpenCV libraries , training and testing the system for zebra crossing and states of traffic light, embedding the code in Raspberry Pi, measuring the distance, to finally giving out the output to the blind person in the form of audio so that he can make himself aware what object is at what distance ahead, we have successfully build a real-world application from which someone can benefit.

The above evaluation criteria and to the extent they have been fulfilled leads to an inference that we were able to meet our expectations to a great extent but at the same time the detecting speed (FPS), precision and accuracy of our work can be enhanced in the future and their lies scope for improvement of our model whether it be in portability, accuracy, speed or measurement of distance.

V. Conclusions

- The usage of Raspberry Pi has optimized the system and brought down the cost drastically compared to conventional systems.
- The proposed system is simple in design, easy to install and is reliable to a great extent.
- Blind person is able to detect and recognize the object in front (within a specific range) through audio output which otherwise is mostly done through either his blind stick or by touching from hands.
- Before the code is embedded in Raspberry Pi, it is perfectly designed, coded and tested on laptop which reduces errors and delays.
- The larger the dataset used during training, the more accuracy you get while testing.
- The distance measurement with Arduino Uno and Ultrasonic Sensor gives proper feedback which alerts the blind person about the obstacle ahead.
- In all, the developed system is able to develop a technical approach for solving a real- life problem in a time and cost-effective manner to a great extent.

VI. Future Work

- Reducing the bulkiness and making the system portable (eg. making a kit or through IC ,which the blind person can attach to his stick) for the application to come into picture in real life.
- Modifying speed , accuracy , and reliability of the system.
- Making the Raspberry Pi speak and respond to instructions given by blind person (eg. Navigation).

Acknowledgment

We are thankful to our institute, IIIT Vadodara, for providing the computational and infrastructure facilities.

Our supervisors- Dr. K K Jha, for providing his guidance over hardware part and helping us in assembling them and Dr. Jignesh Bhatt for building our concepts in training the system.

We are extremely thankful for your support and enlightenment and it would have been very difficult for us to achieve this model in the given duration without your effective directions.

A special thanks to Mr. Jitu Sharma, Jr. Technical Superintendent, for your efficient guidance while working with hardware components and eminent support.

Along with this we would like to deliver our heartfelt gratitude towards some of our seniors Ashish Dubey (DA-IICT) for his hardware assistance , Prashant Maurya (IIITV) - for helping us with his knowledge from his own design project, Hari pad Rajak (IIITV) for providing us with Webcam.

References

[1] Yundong ,Zhang Haomin Peng ,Pan Hu, A report on Towards Real-time Detection and Camera Triggering

[2] International Journal of Advanced Research in Electrical , Electronics and Instrumentation Engineering(IJAREEIE) , Motion Detection Using IoT and Embedded System Concepts,Vol. 5, Issue 10, October 2016

[3] Onkar R. Kirpan, Pooja I. Baviskar, Shivani D. Khawase, Anjali S. Mankar, Karishma A. Ramteke, Object Detection on Raspberry Pi, Volume 7 Issue No.3