OBJECT DETECTION AND SOUND RECOGNITION FOR BLIND PERSON

Mobile Application

21-BSSE-P46

BACHELOR OF SCIENCE
IN
SOFTWARE ENGINEERING

SUBMITTED BY

MOHSIN IQBAL	**17-NTU-1088**
MUHAMMAD TALHA	**17-NTU-1099**
BEHZAD AHMED	**17-NTU-1079**

SUPERVISED BY
Dr. Muhammad Asif Habib

CO SUPERVISED BY
Dr. Muhammad Adeel

DEPARTMENT OF COMPUTER SCIENCE
NATIONAL TEXTILE UNIVERSITY, P.O-37610 FAISALABAD
SESSION 2017-2021

Abstract

Blind persons, face a lot of challenges with respect to their autonomous movement. Most of the time they need someone for assistance, especially in the indoor environment, which is also not a feasible thing. There are already very good GPS based products available but these products fail when it comes to an indoor environment. So, in order to tackle these issues, a computer vision-based system was proposed to assist a blind person throughout an unknown indoor environment by detecting the hurdles coming on his way during the movement. The basic proposed objects were Doors, Stairs, Chairs, Tables, and Vehicles. This report contains different methodologies, their comparisons, summary, and future depictions.

Introduction

Strong and productive indoor object detection can help individuals with extreme vision hindrance to autonomously get to new indoor conditions and evade threats [1]. While GPS-guided electronic way-finding helps show a lot of guarantee in outdoor situations, there are not many indoor direction and route helps. Computer vision innovation on a fundamental level can possibly help blind people to autonomously get to, comprehend, and investigate such situations. However it stays a test for the accompanying four reasons: First, there are huge intra-class varieties of appearance and plan of items in various structural conditions. Second, there are generally little between class varieties of various item models. Third, comparative with lavishly finished and colored objects in the regular scene or open-air situations, most indoor texture are man-made and have little surface. Highlight descriptors that function admirably for outside situations may not adequately portray indoor items. At last, object with huge view varieties, and regularly only parts of objects (inside the field of view) are caught when a visually impaired client moves. A compelling indoor way-finding help should deal with object impediment and view varieties. This report consists of an introduction and the main idea of the whole project, the portion of the work to be completed, following by the literature review, comparison of different methodologies used with their results, and the future plans.

Description of the work

The purposed system consists of two parts, one is the object classification model and the other one is the mobile interface. The image captures taken from the Mobile camera will be sent to the model which will then return the prediction results, and later on which will be conveyed to the blind person through a voice message. The communication has made possible with the help of Tensorflowlite API which is working as a tool of the integration between the model and the mobile interface.

Technologies Used

In this project, the following technologies were used:

Python (used for training the model)

Python is one of the most popular general-purpose programming languages. It is among the world's fastest-growing programming languages and is used by software engineers, mathematicians, data analysts, scientists, network engineers, students, and accountants.

The features that make Python such a powerful language are:

- It is an Interpreted, object-oriented, and a high-level programming language. Python is called an interpreted language as its source code is compiled to bytecode which is then interpreted. CPython usually compiles Python code to bytecode before interpreting it.

- It supports dynamic typing and Dynamic binding. In languages like Java, C and C++ you cannot initialize a string value to an int variable and in such cases, the program will not compile.Python does not know the type of the variable until the code is executed.
- Python has an easy syntax which enhances readability and reduces the cost of code maintenance. The code looks elegant and simple.

- Automatic memory management. Memory management in Python involves a private heap(a data structure that represents a queue) containing all Python objects and data structures. On-demand, the Python memory manager allocates the heap space for Python objects and other internal buffers. The management of this private heap is ensured internally by the Python memory manage

Keras (for building the model))

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation.

Being able to go from idea to result as fast as possible is key to doing good research.

- Simple – but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.
- Flexible – Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you’ve already learned.
- Powerful – Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

Colab Notebook (Cloud ML Platform)

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

As a programmer, you can perform the following using Google Colab.

- Write and execute code in Python.
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks.
- Import/Save notebooks from/to Google Drive
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

Tensorflowlite (Used to convert the model for the mobile use)

A TensorFlow Lite model is represented in a special efficient portable format known as

FlatBuffers (identified by the .tflite file extension). This provides several advantages over TensorFlow’s protocol buffer model format such as reduced size (small code footprint) and faster inference (data is directly accessed without an extra parsing/unpacking step) that enables TensorFlow Lite to execute efficiently on devices with limited compute and memory resources.

A TensorFlow Lite model can optionally include metadata that has human-readable model description and machine-readable data for automatic generation of pre- and postprocessing pipelines during on-device inference.

We can generate a TensorFlow Lite model in the following ways:

- Use an existing TensorFlow Lite model: Refer to TensorFlow Lite Examples to pick an existing model. Models may or may not contain metadata.

- Create a TensorFlow Lite model: Use the TensorFlow Lite Model Maker to create a model with your own custom dataset. By default, all models contain metadata

- Convert a TensorFlow model into a TensorFlow Lite model: Use the TensorFlow

Lite Converter to convert a TensorFlow model into a TensorFlow Lite model. During conversion, you can apply optimizations such as quantization to reduce model size and latency with minimal or no loss in accuracy. By default, all models don’t contain metadata.

Android Studio with Java (Used to implement the android app)

- Mobile phones have become a very common platform for communication and personal use.

- Android is an development platform for the application and it is becoming the most widely used platform among the mobile technologies.

Results

During training, the best model on the basis of the validation accuracy was achieved on the 37th epoch. The table shows all the training and testing results followed by the graphs.

Data	Accuracy
Train	0.9219
Validation	0.9375
Testing	0.9208

Table : Results of MobileNet Model

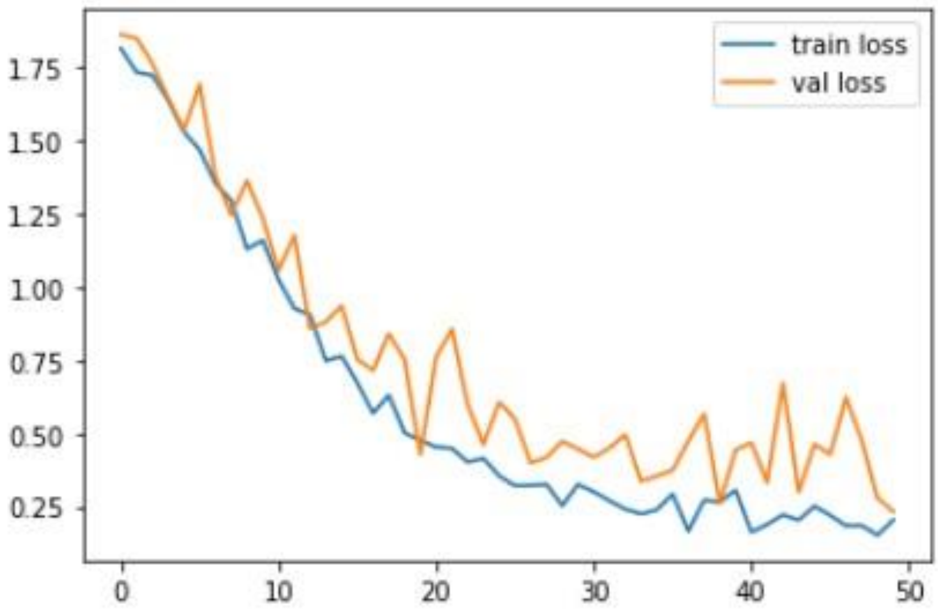

Figure : Loss Graph of MobileNet Model