People Counting and Space Utilization Detection

Yaxin Chen, Qi Wu, Wenyu Dai, Minxuan Qin, Yiming Bi
{yc715, qw85, wd120, mq71, yb127}@scarletmail.rutgers.edu
Advisor: Prof. Vishal M Patel

Motivation

Many study rooms are available on campus. Some of them are crowded while others are pretty empty. Our design brings the computer vision into one study room (EE105) and provides a crowd level of that room for students to check online.

Goal and Objectives

❖ Objectives:
- ○ Bring object detection onto portable Raspberry Pi through Pi-camera with computation library of TesorFlow and create standalone object detector.
- ○ The Web page where students can easily access and check the population and crowd level of the study room.

❖ Goal:
- ○ Students will be able to check real-time study room population anywhere and anytime.

Background

❖ TensorFlow: open source software library for high-performance numerical analysis, originally developed by Google Brain team. comes with strong support for machine learning and deep learning.[1]

❖ TensorFlow Models: machine learning models developed by researchers in TensorFlow. Object detection model, localizing and identifying multiple objects in a single image, is used in this project. [2]

❖ OpenCV: library of programming functions mainly aimed at real-time computer vision, supports deep learning framework like TensorFlow, Torch/PyTorch and Caffe.[3]

Discussion

❖ Due to the privacy issue, the images captured by Raspberry Pi will not be saved: they will be used to analyze. Only the numerical data will be saved and stored in the database.

❖ The TensorFlow Object Detection API is able to detect 90 objects, but we only want certain objects to be detected. Therefore, we modified the codes to make sure that the program will filter out only the objects we need.

❖ Since we want the Raspberry Pi detector to be a standalone detector, we need to set Pi on SSH connecting, thus, we can remotely control Pi without adapting to a display.

Methodology

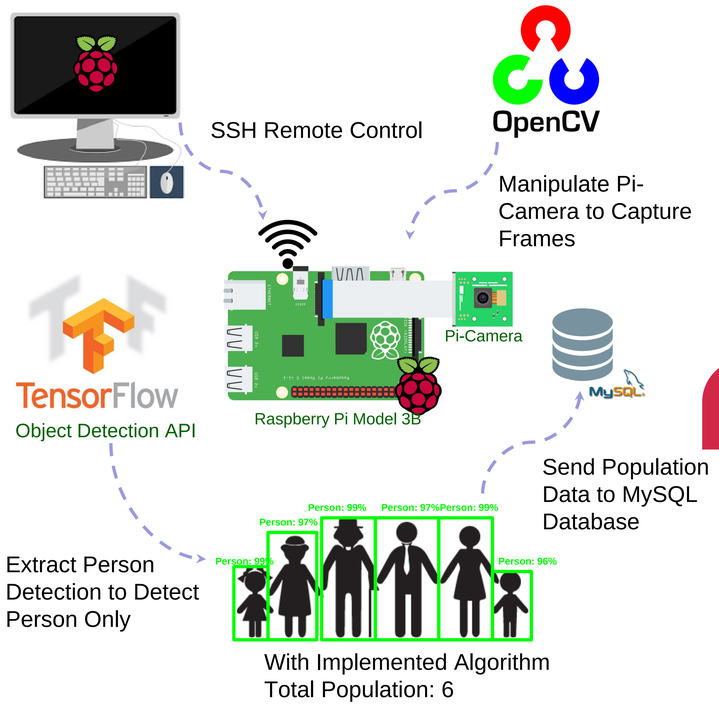
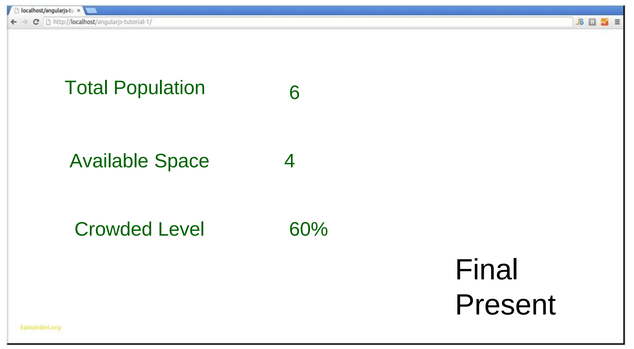


Figure 1: Diagram of System Connections

❖ Initialize object detection with a pre-trained model (a frozen inference graph). Stream camera frames. Run object detection on some frames.

❖ Since regular TensorFlow does not run on Raspberry Pi 3, Sam Jabrahams TensorFlow on Raspberry Pi 3 will be used: a fully featured TensorFlow or Bazel on a Raspberry Pi 3.



Research Challenges

❖ As we began this project, we had a hard time to find an appropriate approach towards population detection. We thought about a lot of approaches like body temperature sensing, face detection, and head-shoulder pattern recognition; however, none of them satisfied our needs or was easy to implement. For facial detection, it was hard to catch people's face when they turn to sideways.

❖ Most object detection algorithms work really well on computers, but it's hard to implement them on Raspberry Pi due to memory capability and processing rate. For example, YOLO (You Only Look Once), a real-time object detection system, could be one of the most accurate object detection system existing so far. Raspberry Pi cannot afford large memory allocation and takes much longer time to process the detection than computers do.

❖ About network setting-up, Rutgers WiFi is WPA2-enterprise encryption, it's really hard for us to try to configure.

Results and Future Work

❖ After collecting some data for some time, we plan to build a model and provide a predicted number of people in the room.

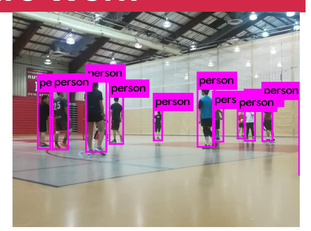


Figure 2: Detection Testing

Acknowledgement

❖ Special thanks to Tim King, blogger of "Object Detection on a Raspberry Pi" on Theta bog.
❖ Special thanks to Sam Jabrahams, who modified TensorFlow so it can work on Raspberry Pi.
❖ Special thanks to TensorFlow research team on building a perfect tutorial on using Object Detection API.

References

[1] TensorFlow home page: https://www.tensorflow.org/
[2] TensorFlow Object Detection API github page: https://github.com/tensorflow/models/tree/master/research/object_detection
[3] OpenCV Wikipedia page: https://en.wikipedia.org/wiki/OpenCV