Vision-based Human Detection System for Internet of Things

Ryan Reynolds (2693018)
Brett Dawson (2655543)
Mina Kamel (2667766)
Botros Shenouda (2639271)

Fall Semester 2018

EEC 493

Cleveland State University

Abstract

Real-time human detection and tracking is a vast, challenging and important field of research. It has wide range of applications in human recognition, human computer interaction, video surveillance etc. This project is aimed to develop a low-cost system that can be used as a form of public surveillance. The software identifies, tracks, and counts people within a street, store, and other places through monocular cameras attached to a microcontroller. The tracked frames are then transferred to a cloud based storage container to be further analyzed by traffic controllers. After the software has been developed, it will be implemented on a raspberry pi microcontroller by the hardware side of the team. The microcontroller will be able to communicate the number of people within a given area in real time to a cloud-based system using IoT communication. Accuracy, tracking, and path projection can be improved and implemented if time permits. The hardware operating expenses were set at $139 including all the hardware and software required for this project. An extensive research phase has been conducted in the span of nine months to insure the highest results presented to launch the project.

Introduction & Background

Recent hardware advancements have thrown Artificial Intelligence into the focus of popular science and researchers alike. The field of computer vision is at the forefront of this rise in interest. One of the critical surges of research inside computer vision, that researchers attribute a considerable measure of need, is the comprehension of human action from within a video[1]. The expanding enthusiasm for human movement investigation has been complemented by late enhancements in computer vision due to the accessibility of minimal effort equipment- low cost camcorders, smartphones, etc.- and an assortment of new encouraging applications- distinguishing individuals, distinguishing gestures, and visual reconnaissance. The point is to consequently figure the movement of a human or a body part from monocular or multi-see video pictures. The fascinating examination for different application in human body movement investigation are physical execution, assessment, computer generated reality, and human-machine interface[3]. Hence in general, three parts of research headings are considered in the investigation of human body movement: following and evaluating movement parameters, breaking down of the human body structure, and perceiving of movement action.

Artificial intelligence has become an integral part of today's society. With technological advancements from healthcare to entertainment, AI's role will only increase as time goes on. The purpose of AI is to create a solution to a problem that humans encounter, and have it fix this problem at a level that is superior to what a human could do. Our project is an example of the broad explanation of the purpose of AI[2]. Because humans are unable to efficiently perform mass surveillance, creating algorithms to count and track humans in this space is a fitting solution to this problem. Human tracking using AI can create a force of good or evil in society. Along with exposing us to the nature of creating and implementing surveillance algorithms and hardware, it is also our responsibility to adhere to our ethical responsibilities when dealing with this powerful technology[4]. The following is an explanation of how we will accomplish the creation of accurate human tracking software implemented on a Raspberry Pi.

Project Objectives:

1) The human detection is performed and transferred in real time via IoT communication with a minimum framerate of 10 fps.
2) The software consistently runs on the raspberry pi and successfully interfaces with the camera.
3) The collected data is successfully transferred to cloud storage with a concise file management system.

Technical Approach

We will achieve our objectives using agile methodology. Agile methods will work best for our project because our requirements are going to change. Our obtainable requirements are limited by the raspberry pi hardware. Furthermore, we will not be able to get an accurate measure of how limited we are going to be by the hardware, until we have the raspberry pi up and running. We are going employ agile development by using three detection techniques- one resource heavy (YOLO), one resource light (Haar), and one in between (Hog). This will allow us to react quickly to changes and ensure our objectives will be met.

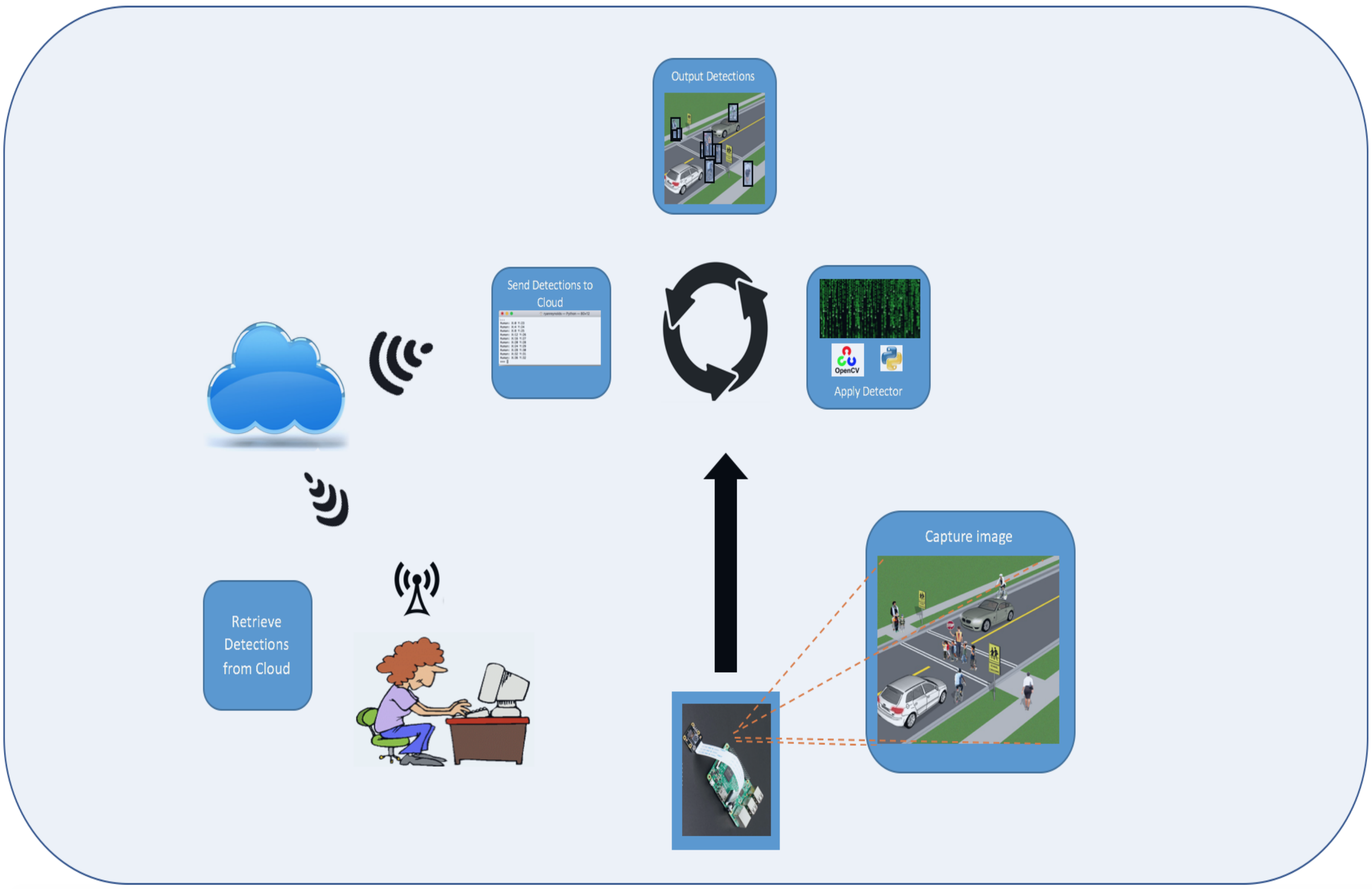


Figure (A): Project Model

Research Phase

The research phase of the project is a substantial portion of the project. The research phase includes three main tasks- determining low cost and accurate methods of object detection, researching the raspberry pi architecture, and picking a real time communication method.

Researching the raspberry pi architecture will allow us to control and interface the raspberry pi with our software. For example, in this phase we will learn how to install and use the camera. Additionally, the raspberry pi architecture determines the limits on the types of detection methods we are able to implement because of hardware limitations.

The determination of low-cost and accurate detection methods is detrimental to achieving the project objectives. We settled on three detection methods Hog, Haar, and Yolo human detection. Hog uses the idea that an object can be described by the distribution of intensity gradients. Essentially, hog counts the occurrences of gradient orientation in a portion of an image. Haar looks at neighboring rectangular regions at a location in an image. Haar sums the pixel intensities in a region then takes the difference between the sums to categorize subsections of the image. Yolo applies a single a convolutional neural network to the entire image. The network divides the image into regions and uses activation functions on each region to predict weighted bounding boxes. Each method will be evaluated for detection performance using the standard precision formula True Positives / (True Positives + False Positives) and runtime performance by measuring fps.

The final research task is to determine a real time communication method that can be implemented on a raspberry pi. For this task, we will evaluate Microsoft's Azure IoT Hub and Amazon AWS. The main evaluation criteria for this step is cost.

The software side of the project is going to require constant refactoring. In order to respond to the changes, we are going to rely on agile development. is going to require constant. Separate the tasks into software and hardware related.

Hardware Installation Phase

The hardware installation phase involves setting up the raspberry pi components, operating system, and software dependencies. The first step will be installing the OS. Once the OS is installed then we will set our monocular camera and interface it with the OS. Additionally, a makeshift mount to house and protect the camera will be fashioned during this phase. Then an SSH client will be set up to allow us to control the pi without a monitor. This is needed in case we run the detector in an area where it would be difficult to mount a monitor. We classified the installation of software dependencies as hardware installation to better distribute the work between the software and hardware members. The dependencies must be installed in a virtual environment in case our dependencies change later in the project. The following dependencies will be installed with all their required dependencies: Python 3.5+, OpenCV 3.4.2 and Azure IoT hub.

Software Implementation Phase

The software implementation phase will create the actual script that runs the detection model and sends the detections to the cloud. All three detection algorithms will be implemented in python using intel's computer vision library OpenCV. An initial classifier will be developed for each of the algorithms. Performance scripts will be developed to measure the run time frames per seconds and precision for each algorithm. Then each algorithm will be run on a test video to get base performance metrics on the actual raspberry pi. The IoT communication will be implemented by having the detector script open a real-time message ingestion link to Azure IoT Hub or an AWS container, then caching the detections either based on time delay or number of frames. Then finally sending the data to the storage container based on the packet format previously stated.

Training and Testing Phase

In this phase we will run our three models on live feeds from the raspberry pi using our performance scripts to evaluate each model. If precision is poor, we will train the model(s) for the area the camera is placed. The method of training for each of the models varies based on the classifier but the ideology is the same for all three. A better trained model will be achieved by adding images of humans at the distances and angles expected from the detection location and/or adding non-human images the classifier encounters at the detection location to the model. We will also monitor its performance on different locations to ensure that we do not over-train the model. If frame rates are low we will refactor to lighter models. Specifically, for Hog and Haar we will tweak the detector parameters. For Yolo, we will reduce the number of classes trained. Then if the framerate is still low, we will retrain on a neural network with less convolutional layers. This process will be repeated until precision is maximized, while maintaining the maximum fps allowed by the raspberry pi. After this is achieved, we will begin to explore lightweight tracking methods using a similar test and training cycle.

Budget Justification

The hardware supplies are the main objects needed for this project. The software used for the project are open source libraries.

Project Expenses	Cost
Raspberry Pi 3 B+ Kits Model 2018	$80
Camera module 5 MP 1080 p	$25
Raspberry pi 32 GB preloaded SD Card	$19
Lithium Battery pack Expansion board power Supply Switch	$15
Total	**$139**

Table 1: Project Expenses

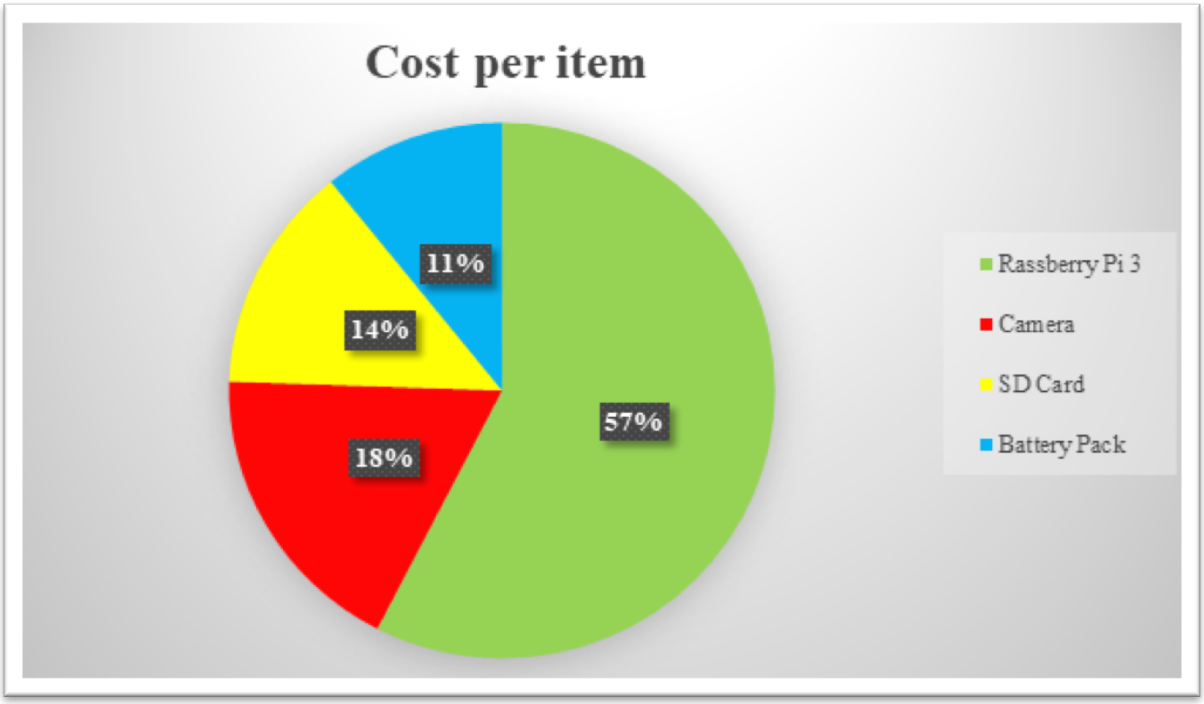


Figure (B) Pie Chart

Professional Awareness

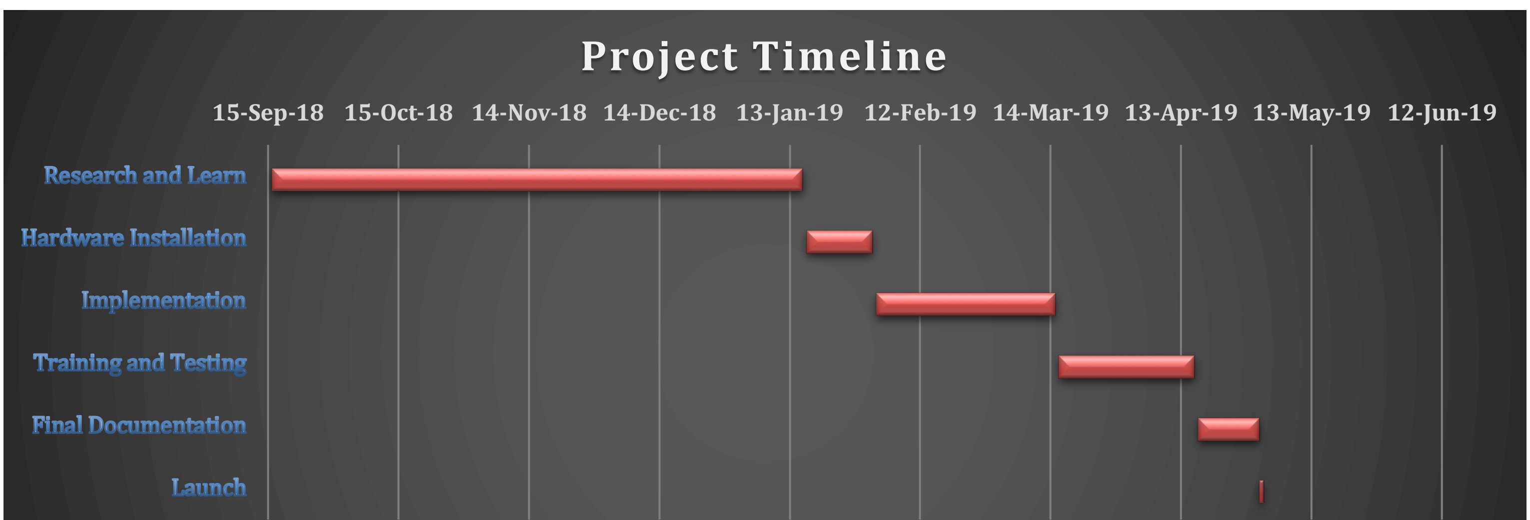
With the ever-growing presence of artificial intelligence in our world today, it is important to consider the ethical responsibilities of those who implement this technology. This is especially important with human tracking software. By using a mounted camera and a variety of algorithms to analyze objects in the camera's view, we have access to biometric data. It is our responsibility to keep their identity private for their own safety and security. We accomplish this by testing our algorithms using available test data from reputable sources such as MIT. When the project arrives at the stage of using our own camera to track, we will keep our produced test data secure and make no attempts to further identify those in the video feed.

This project broadens our understanding of artificial intelligence and allows us to apply what are learning in the classroom to real world situations. This is especially important for a field such as computer science where the easiest way to learn new information is to directly work with the technology.

Project Timeline

The Project timeline for this project will be over the span of 2 semesters, divided into the following phases as shown below:

- **Phase one:** Research and Learn
    - Initial Research
    - Group meetings
    - Faculty meetings
    - Install CV libraries and update Python to the latest version on Computer
    - Research detection libraries
    - Project Specifications (Proposal)
    - Midterm presentation
- **Phase two:** Hardware Installation
    - Set up the camera
    - Set up hardware
    - Install dependencies on PI
- **Phase three:** Implementation
    - Software Implementation
- **Phase four:** Training and Testing
    - Software testing
    - Hardware testing
    - Final testing
- **Phase five:** Final Documentation
    - Final Proposal
    - Final Presentation
- **Phase six:** Launch



	Research and Learn	Hardware Installation	Implementation	Training and Testing	Final Documentation	Launch
Start Date	16-Sep-2018	17-Jan-2019	2-Feb-2019	16-Mar-2019	17-Apr-2019	1-May-2019
Days to Complete	122	15	41	31	14	1

Figure (C) Gantt Chart

Conclusion

The initial goals of this project have been met up to this point in the project's implementation. As a group, we have successfully installed OpenCV (library for detection) along with testing the different algorithms on a laptop. We have received the Raspberry Pi and have used this to begin testing the detection algorithms that were initially implemented on the laptop. The next steps are to continue successful implementation of the required software and algorithms to perform human detection on the raspberry pi, followed by tweaking these algorithms and loading the data in real time into the cloud. As mentioned in the Technical Approach section, the software portion of this project may see changes in the requirements based on what the Raspberry Pi is capable of handling. This is due to the fact that the laptop this was run on has a more powerful CPU and GPU than the raspberry pi. With that being said, we should accomplish all three of our goals with a possible exception in number 1. The Pi may not be able to run the detection software at 10 frames per seconds, but we will work to accomplish this goal in the coming months. If it is determined this goal cannot be met, we will provide information on an alternative microcontroller that our software could run on at that framerate.

References

[1] Dagless, E.L.; Ali, A.T.; Cruz, J.B. Visual Road Traffic Monitoring and Data Collection. In Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference, Ottawa, ON, Canada, 12–15 October 1993; pp. 146–149.

[2] Mahapatra, A.; Mishra, T.K.; Sa, P.K.; Majhi, B. Background Subtraction and Human Detection in Outdoor Videos Using Fuzzy Logic. In Proceedings of the IEEE International Conference on Fuzzy Systems, Hyderabad, India, 7–10 July 2013; pp. 1–7.

[3] Rajaei, A.; Shayegh, H.; Charkari, N.M. Human Detection in Semi-Dense Scenes Using HOG descriptor and Mixture of SVMs. In Proceedings of the International Conference on Computer and Knowledge Engineering, Mashhad, Iran, 31 October–1 November 2013; pp. 229–234.

[4] Rastegari M.; Ordonez V.; Redmon J.; Farhadi A. "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," Computer Vision – ECCV 2016 Lecture Notes in Computer Science, pp. 525–542, 2016.

[5] Redmon J.; Divvala S.; Girshick R.; Farhadi A. "You Only Look Once: Unified, Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) June 2016.

[6] Shokrolah Shirazi, Mohammad, "Vision-Based Intersection Monitoring: Behavior Analysis & Safety Issues" (2016). UNLV Theses, Dissertations, Professional Papers, and Capstones. 2741. https://digitalscholarship.unlv.edu/thesesdissertations/274, pp. 34-37, 39, & 42-45