**HAWKEYE CCTV**

**Object Detection API on the raspberry pi**

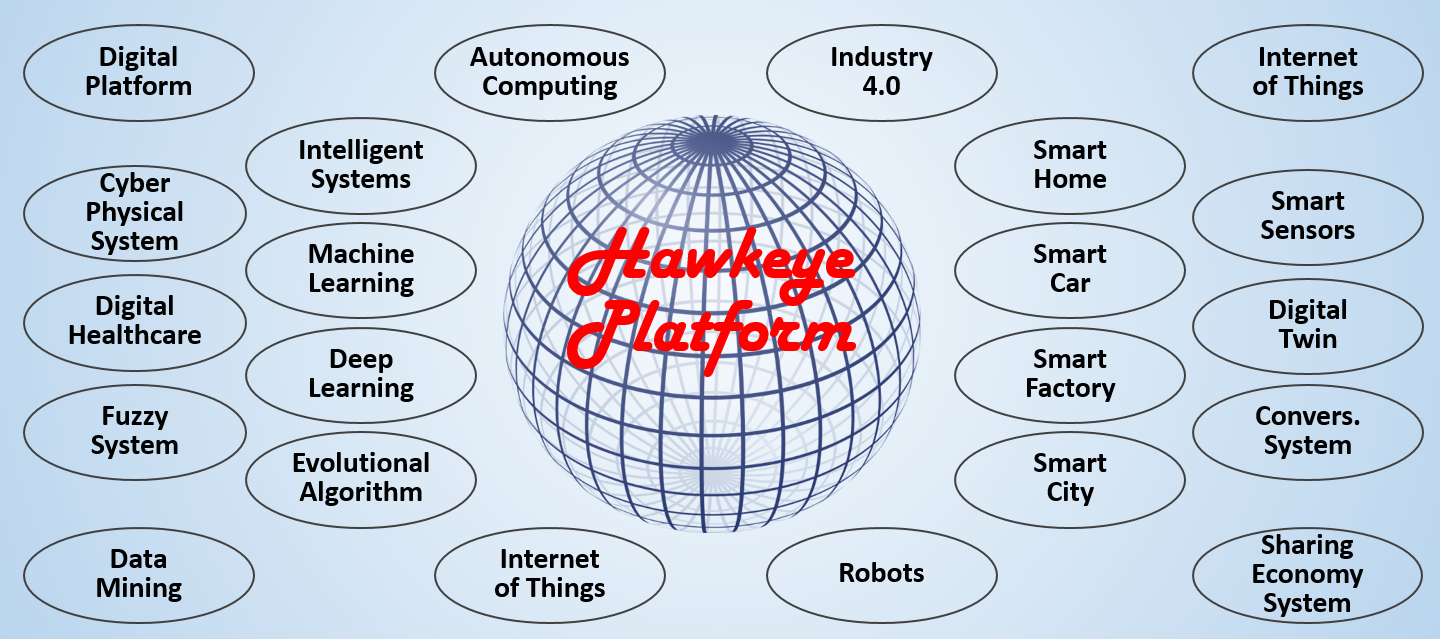
**Intelligent CCTV using HAWKEYE Platform**

**Version 0.9**

**Feb. 2019**

**Staffs**

Software Engineering Laboratory  
**Soongsil University**Seoul, Korea  
82-10-7392-2220  
sdkim777@gmail.com http://soft.ssu.ac.kr



**Table of Contents**

[1. Introduction 3](#_Toc885016)

[2. Required Package / Libraries 3](#_Toc885017)

[3. Installation 3](#_Toc885018)

[3.1. Ready Raspberry PI 3](#_Toc885019)

[3.2. Install TensorFlow 3](#_Toc885020)

[3.3. Compile and install Protobuf Package 3](#_Toc885021)

[3.4. Set up TensorFlow directory structure and the PYTHONPATH variable 4](#_Toc885022)

[4. Implement 4](#_Toc885023)

[4.1. Testing Model 4](#_Toc885024)

[4.2. Testing 4](#_Toc885025)

[4.3. Evaluation 5](#_Toc885026)

[5. APPENDIX. Code 6](#_Toc885027)

[5.1. Object\_detection\_picamera.py 6](#_Toc885028)

**Object Detection API on the Raspberry Pi**

# Introduction

# Required Package / Libraries

* Tensorflow
* Protobuf Package

The TensorFlow object detection API uses Protobuf, a package that implements Google’s Protocol Buffer data format. In order to install Raspberry Pie, there is a way to get the code and compile and install it directly. As of 2019-02-12, the most recent version is 3.7.0.

# Installation

## Ready Raspberry PI

* PICAMERA

Connect the camera to the raspberry pie and see if it works.

* OpenCV

The Object detection scripts use OpenCV. But, this is already installed.

## Install TensorFlow

* Step
* Step 1. Install LibAltas package to solve tensorflow’s requirement.
* sudo apt-get install libatlas-base-dev
* Step 2. Install Tensorflow
* pip3 install tensorflow
* Step 3. Install required libraries for Tensorflow Object Detection API
* sudo pip3 install pillow lxml jupyter matplotlib cython
* sudo apt-get install python-tk

## Compile and install Protobuf Package

* Step
* Step 1. Get the packages needed to compile Protobuf.
* sudo apt-get install autoconf automake libtool curl
* Step 2. Download the protobuf from Github repository.
* wget <https://github.com/google/protobuf/releases/download/v3.7.0-rc/protobuf-all-3.7.0-rc.tar.gz>
* Step 3. Unpack the file.
* tar -zxvf protobuf-all-3.7.0-rc.tar.gz
* Step 4. Configure the build in protobuf.
* ./configure
* Step 5. Build the package.
* make
* Step 6. Install the package.
* sudo make install

## Set up TensorFlow directory structure and the PYTHONPATH variable

* Step
* Step 1. Download the tensorflow repository from GitHub
* git clone --recurse-submodules <https://github.com/tensorflow/models.git>
* Step 2. Modify the PYTHONPATH environment variable to point at some directories inside the TensorFlow repository.
* export PYTHONPATH=$PYTHONPATH:/home/pi/tensorflow1/models/research:/home/pi/tensorflow1/models/research/slim
* Step 3. Compile the Protocol Buffer (.proto) files used by the Object Detection API.
* protoc object\_detection/protos/\*.proto --python\_out=.

# Implement

## Testing Model

* SSD\_Lite model (from the Tensorflow detection model zoo)

The model zoo is Google’s collection of pre-trained object detection models that have various levels of speed and accuracy.

* SSDLite-MobileNet model

The model is one of the SSD\_Lite model.

* Model that takes less processing power.
* The reason for using the lower performance model is that, Raspberry Pi has a weak processor.
* The model run faster than other although it shows lower accuracy.

## Testing

* Code
* python3 Object\_detection\_picamera.py
* Test Scenario

1. Detecting People

2. Detecting Objects

* Test 1. Detecting People
* Figure 1 show the result of detecting person.
* The left image’s detecting accuracy is 94%.
* The right image’s detecting accuracy is 94%.



Figure 1. Detect Person

* Test 2. Detecting Objects

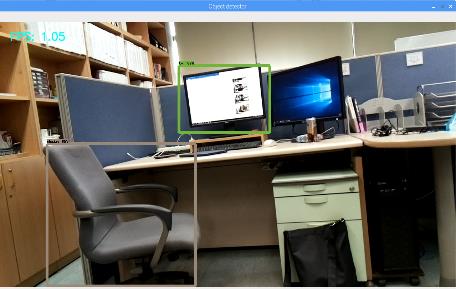


Figure 2. Detect Chair and TV : 85% , 67%



Figure 3. Detect TV : 78%

## Assessment

* Pros
* 1
* Cons
* 2

# APPENDIX. Code

## Object\_detection\_picamera.py

|  |
| --- |
|  |