Real-Time Object Detection

Software Requirement Specification Document

1. Synopsis

Object detection is the process of finding instances of real-world objects such as faces, buildings, and bicycle in images or videos. Object detection algorithms typically use extracted features and learning algorithms to recognize instances of an object category. It is commonly used in applications such as image retrieval, security, surveillance, and advanced driver assistance systems (Self-driving cars).

Tensorflow is an open-source deep learning framework created by Google Brain. “TensorFlow Object Detection API” which is an Open source framework built on top of TensorFlow that makes it easy to construct, train and deploy Object Detection Models and also it provides a collection of Detection Models pre-trained on the COCO dataset, the Kitti dataset, and the Open Images dataset.

It’s a deep learning approach based on **Convolutional Neural Networks** (CNN).

2. Python packages installed

- protobuf
- pillow
- lxml
- Cython
- jupyter
- matplotlib
- panda
- opencv-python
- tensorflow

3. List of Requirements

REQ-ID	**Requirement Title**	**Nature of Work**
01	Preparing Dataset	Software
02	Labeling the Dataset	Software
03	Generating Records for Training	Software

04	Configuring Training	Software
05	Training the Model	Software
06	Exporting Inference Graph	Software

$ - Unique ID (Sample: REQ1, REQ2, REQ3, … REQN)

4. Requirements Specification

REQID: REQ01

TITLE: Preparing Dataset

DESCRIPTION: Collect as many different and variety of images consisting of the objects. Create a directory named images inside research directory. Store 80% of the images into train directory and 20% of the images into test directory inside the images directory.

REASON / RATIONALE: The dataset is the most important thing in building a classifier. It's the basis of the classifier on which object detection is done

DEPENDANCY: None

REQID: REQ02

TITLE: Labeling the Dataset

PREREQUISITE: Install tool called as **labelimg.**

DESCRIPTION: Open the labelimg application and start drawing the rectangular boxes on the image wherever the object is present. Save each image after labeling which generates an xml file with the respective image's name.

DEPENDANCY: REQ01

REQID: REQ03

TITLE: Generating Records for Training

PREREQUISITE: Download two scripts from Dat Tran's Racoon Detector named as **xml_to_csv.py** and **generate_tfrecord.py**.

DESCRIPTION:

Convert all the XML files to CSV files and then generate the TFRecord files.

This creates test.record and train.record files in object_detection directory.

REASON / RATIONALE: TFRecords can be served as input data for training of the object detector.

DEPENDANCY: REQ02

REQID: REQ04

TITLE: Configuring Training

PREREQUISITE: Download the file **faster_rcnn_inception_v2_coco.**

ALGORITHM: faster_rcnn_inception

DESCRIPTION:

Create labelmap.pbtxtq file. The label map tells the trainer what each object is by defining a mapping of class names to class ID numbers.

DEPENDANCY: REQ02, REQ03

REQID: REQ05

TITLE: Training the Model

DESCRIPTION:

Copy the train.py file and paste it in the object_detection directory.Tensorflow creates a checkpoint for every 5 minutes and stores it. View the progress of the training job by using TensorBoard. Continue the training process until the loss is less than or equal to 0.1.

DEPENDANCY: REQ02, REQ04

REQID: REQ06

TITLE: Exporting Inference Graph

DESCRIPTION:

Navigate to the training directory and look for the model.ckpt file with the biggest index. Create the inference graph, a frozen_inference_graph.pb file in the \object_detection\inference_graph folder. The .pb file contains the object detection classifier.

REASON / RATIONALE: Inference graph is used to run the model.

DEPENDANCY: REQ05
