Machine Learning Engineer Nanodegree

Capstone Proposal

Chandana Neerukonda
October 17 2017

Domain Background:

Autonomous Vehicles are one of the most in vogue today.Autonomous Driving touches many challenges in different fields, especially computer vision and machine learning. One of the most talked is pedestrian detection.

Pedestrian detection systems may be possible to implement without the use of spinning light detection and ranging (LIDAR) devices, which can often cost more than $10,000. Instead, cheap video cameras may be used for the same purpose of recognizing and tracking nearby pedestrians.[1] Research Papers show impressive results using transfer learning but the most common errors for detectors are as follows:

- Detecting tree leaves or traffic lights in background as pedestrian
- Detecting the same person twice
- Not detecting small persons

Other approaches like [2] use a real-time algorithm for the detection and trackcinf of image regions that possibly contain pedestrians. It then uses a classification algorithm based on the typical motion patterns of a pedestrian's legs. These approaches assume the visibility of the legs and works for walking people only!

There has been a lot of work done on Caltech Dataset[3] which is considerably the largest Pedestrian dataset.

Problem statement:

The problem is Pedestrian Detection on Caltech data set. I am personally motivated to explore the implementation of mobile nets for pedestrian detection to have this feature on an Android mobile. I believe this problem is solvable because of the new advances including SSD, YOLO and Faster RCNN and the support from Tensor flow for object detection => TFDetect (https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android)

Datasets and inputs:

Calltech Dataset (http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/)

The training data (set00-set05) consists of six training sets (~1GB each), each with 6-13 one-minute long seq files, along with all annotation information . The testing data (set06-set10) consists of five sets, again ~1GB each. Annotations for the entire dataset are now also provided.

Solution Statement

I aim to extract the images from the .seq files and the corresponding annotations from the .vbb files provided. I am planning to build tf_record files for both training and testing and use them to train my model using the pre-trained object_detection module on the coco_dataset.

Benchmark Model

http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/files/PAMI12pedestrians.pdf
This paper does benchmarking of several algorithms on the Caltech dataset.

But my approach would be entirely different from the bench_mark model because I am planning to use SSD_MobileNet / Faster_RCNN for training.

References:

[1].http://www.govtech.com/fs/Googles-New-Pedestrian-Recognition-Tech-Could-Quicken-Self-Driving-Cars-to-Market.html

[2] https://pdfs.semanticscholar.org/b406/6e7b3c227bcf8549af551edb163cedc1b931.pdf
[3] http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/files/algorithms.pdf

Evaluation Metrics:

The Evaluation metric I would propose would be the mAP value (detection and classification accuracy).

Project Design

Like I mentioned in the Solution_statement, I am planning to extract the images from the .seq files and the corresponding annotations from the .vbb files provided. I am planning to build tf_record files for both training and testing and use them to train my model using the pre-trained object_detection module on the coco_dataset.

The ground_truth annotations have different classes like Person, People, person(?) for the detection with lesser confidence and also occlusion_infomation. I am planning to not worry about the occluded detections but to consider them as true_positives. I am also planning to eliminate the person(?) class annotations and to merge the person and people detections. So, basically I will have just one class person. I might have false_negatives in the ground_truth data. But I would like to go with it just to simplify the process.

Thanks !!
Chandana.