Vehicle Detection using Satellite Images

Priyam Saikia (UFID 9414 5292) *

***Abstract*—In this project, I propose to use supervised learning approach for object detection in an image or in a video. I surveyed various deep learning methods for object-detection and concluded to use two broadly used methods – Faster R-CNN (region-based convolutional neural networks) and Single Shot Detection (SSD) to train my data. The objective of the project would be use images from available data sets, prepare the data and train them with object detection neural networks to detect vehicles in the images. I will also analyze and observe results from both the methodologies used and compare their performance based on their accuracy and other parameters. This project is aimed to be useful in areas such as traffic surveillance control, spying, disaster management and intelligent transportation systems.**

***Index Terms*— vehicle detection, R-CNN, SSD, TensorFlow**

I. Introduction

The task of object detection is one of the most fascinating problems in pattern recognition. Various machine learning and deep learning approaches are being continuously researched upon for better object detection.

I started this project with an extensive literature review on the various approaches on object detection. Considerable research has been conducted on the area of deep learning approaches, special using region-based neural networks. I pinned down on two such approaches – Faster R-CNN and SSD.

Apart from initial literature survey and final report preparation, this project will consist of two main phases – data preparation and data modeling. In the data preparation phase, I use available datasets to get images containing vehicles and use labelling tools to prepare some of the data that is not properly annotated. In the modeling phase, I would implement the mentioned deep learning methods to train the data to detect vehicles. For this phase, I would use already available code-base for neural networks and build on top of that.

Eventually, I plan to prepare a report, a PowerPoint presentation and a video walk-through of the completed project. If time permits, I also plan to host a webpage that would accept input image from a user, run the neural network process in the background, and output the input image with the detected vehicles marked clearly.

I have provided a brief timeline and description of the various steps of my project plan in Figure 1.

*This work is supported by University of Florida

P. Saikia is with the Department of Computer and Information Science and Engineering, University of Florida (e-mail: priyam.saikia@ufl.edu).

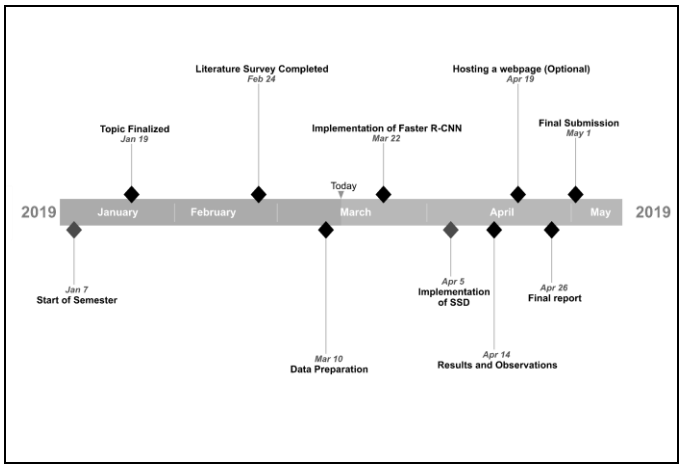

Fig 1. Timeline of Project Plan

II. Project Execution

A. Topic Selection

Object detection tends to be more difficult than object classification since the former requires not only recognizing an object of a given class in an image but also finding the coordinates of the object in the image and marking a bounding box around the recognized object [1]. On top of that, it is imperative to prepare a properly annotated dataset to train the data for specific class of objects – in this case, vehicles [2]. This is one of the reasons I selected object detection as a topic because it would give me an exposure to the complete data-to-insight pipeline – including dataset preparation and data modeling. Also, the potential application of this project could be in traffic surveillance control, spying, disaster management, intelligent transportation systems, and designing logistics using drones.

The decision on what topic to select took me around two weeks and concluded on around 19th January.

B. Literature Survey

I went through various journal papers, blogs and articles that discussed end-to-end processing of data for object detection. Hao Gao's blog [3] in Medium was the final push that made me decide to work on a vehicle detection project. Hao Gao has explained Faster R-CNN in a very detailed manner. I also went through various articles on how to implement Faster R-CNN and SSD. I also surveyed about the best sources to get image data from and came across sources such as GTI Vehicle Image Database, Visual Genome, Open Images Dataset V4, and Microsoft COCO datasets [4]. I also surveyed the collected annotated data but realized that creating custom data sets of annotated data makes the process more efficient.

The literature survey took me more than a month and I finished it around 24th of February.

C. Preparation of Annotated Data

During the literature review, I surveyed about the best sources to get data from and concluded to use data from Google's Open Images and Microsoft COCO [5]. As I went through the collected data, I decided to annotate some of them myself. I used Labellmg [6] to generate annotation records in XML format and then used an open source code to convert the XML formats to TensorFlow records [7]. As of now, I have reviewed that all the images that I will use are properly annotated with vehicles clearly marked with bounding boxes. These data set would later be fed to Faster R-CNN and SSD to help detect vehicles.

The preparation of the data took me around two weeks. Currently, on March 10th, I am completed with the process.

D. Implementing Neural Networks

I pinned down two methodologies to train my data in. A paper published by researchers of National Laboratory of Pattern Recognition [8] goes extensively into layers of Deep Convolutional Neural Networks (DNN) and hybrid DNNs. In another paper [9], the vehicle detection problem is solved using two stages of object classification by merging RPN and Faster R-CNN. Another object-detection approach that intrigued me was the implementation of SSD (Single Shot MultiBox Detector) [10] [11]. It is generally expected to be faster than Faster R-CNN.

I am planning to use TensorFlow's object detection API as the base code to train my data in, especially for Faster R-CNN. It is capable of localizing and identifying multiple objects in a single image [12]. For SSD, PyTorch is a probable tool. I will also be using Google's Protobuf (3.7) for Tensor Records. I am planning to use Model Zoo to get the pertained models for object-detection which are already trained in MS COCO, and Open Images datasets [13].

I am planning to be done with the implementation and training of the data by the 5th of April.

III. PROJECT RESULTS

A. Study of results from both methods

Once the training is completed, my implementation would be successfully able to detect vehicles in given images (and possibly videos too). I would analyze as well as verify my results and try to optimize the model. Furthermore, I would compare results from both the methods and observe the similarity and differences in their performances. I expect to get helpful insights from these results.

I am planning to be done with this part of the project by first 14th of April.

B. Hosting a web page for custom test input

Going forward, if time permits, I am planning to host a simple webpage in a local server in which a user can upload an image and then click a button to fed it to my neural network and get an output image with all the detected vehicles.

If I complete step II.D and III.A on time, I will provide myself one week's time to create this basic webpage. The planned deadline is 19th of April.

IV. FINAL REPORT

Finally, I would prepare a final report on the vehicle detection deep learning project that I am working on. I would also create a PowerPoint presentation and prepare/upload a YouTube video discussing the end-to-end project. I have surveyed a few tools which gives good desktop recording including single and double clicks.

This should take no more than a week and should be done by 26th April. This also gives me a buffer time in case any of the steps in II or III takes longer than anticipated. If everything goes well, this gives me a few days' time to make any final finishing touches to any part of the project before the final submission on May 1st.

ACKNOWLEDGMENT

I would like to thank Professor Dapeng Wu for his guidance and all the authors and researchers whose work enables me to create essential progress in my project.

REFERENCES

[1] Hulstaert, Lars, "A Beginner's Guide to Object Detection", *Datacamp*, April 19, 2018, www.datacamp.com/community/tutorials/object-detection-guide
[2] Yang, Bo, et al. "Vehicle detection from low quality aerial LIDAR data", IEEE Workshop on Applications of Computer Vision, 2011.
[3] Gao, Hao, "Faster R-CNN Explained", *Medium*, Sept 27, 2017, medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8
[4] Yikang, Li, "Dataset gallery", CVBoy, Sept 20, 2017, cvboy.com/post/dataset/
[5] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham, 2014.
[6] Tzutalin," LabelImg", Git code (2015), github.com/tzutalin/labelImg
[7] Velde, Dion v., "How to train a Tensorflow face object detection model", *Medium*, Sep 26, 2017, towardsdatascience.com/how-to-train-a-tensorflow-face-object-detection-model-3599dcd0c26f
[8] Chen, Xueyun, et al. "Vehicle detection in satellite images by hybrid deep convolutional neural networks." IEEE Geoscience and remote sensing letters, 2014.
[9] Ren, Shaoqing, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.
[10] Gao, Hao, "Understand Single Shot MultiBox Detector (SSD) and Implement It in Pytorch", *Medium*, June 6, 2018, medium.com/@smallfishbigsea/understand-ssd-and-implement-your-own-caa3232cd6ad
[11] Liu, Wei, et al. "SSD: Single shot multibox detector", European conference on computer vision, Springer, Cham, 2016.
[12] TensorFlow, "Tensorflow Object Detection API", Git code (2018), github.com/tensorflow/models/blob/master/research/object_detection
[13] TensorFlow, "Tensorflow detection model zoo", Git code (2018), github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md