Object Classification using Convolutional Neural Network(CNN) for Advanced Driver Assistance Systems (ADAS)

Project Report

Submitted by

Sai Sravan Manne
EDM13B018

in partial fulfillment for the award of the degree of

Bachelor of Technology

in

Electronics Engineering – Design and Manufacturing

Indian Institute of Information Technology Design and Manufacturing, Kancheepuram, India

December 2016

BONAFIDE CERTIFICATE

This is to certify that the thesis titled "**Object Classification using Convolutional Neural Network(CNN) for Advanced Driver Assistance Systems (ADAS)**" submitted by **Mr. Sai Sravan Manne(EDM13B018)** to the Indian Institute of Information Technology Design and Manufacturing, Kancheepuram, for the award of **Bachelor of Technology in Electronics Engineering – Design and Manufacturing,** is a *bona fide* record of the project work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr.Binsu J Kailath

Project Guide

Asst. Professor
Indian Institute of Information Technology
Design and Manufacturing, Kancheepuram
Chennai 600 127
India

Place: Chennai

Date: 01-12-16

ACKNOWLEDGEMENTS

I would like to thank the Director of the Institute Prof. R. Gnanamoorthy for providing all infrastructural facilities required for my project as well as graduate study here. I wish to express my sincere gratitude to my Project Guide Dr. Binsu J. Kailath, Assistant Professor, IIITDM Kancheepuram, for the continuous support of my project work, for her patience and motivation.

I also wish to express my sincere gratitude to my Co-guide Dr. Sudha Natarajan, Autonomous Vehicle Program - R&D, Tata Elxsi Limited, Chennai, for her support and guidance for my project.

ABSTRACT

Obstacle detection and classification is one of the key tasks in the perception system of ADAS and self-driving vehicles. Vision-based approaches are popular due to cost effectiveness and appearance information associated with the vision data. In this paper, an alternative algorithm for region proposals is discussed. The designed algorithm has higher frame rate compared to existing CNN based algorithms and requires less GPU capacity for implementation.

In the future, this algorithm can be further improvised, such that it can tune automatically for specific brightness condition ,and also, for increased precision in generating region proposals.

Contents

Contents ix
List of Figures xi
Nomenclature xv
Chapter 1 Introduction 1
1.1 Motivation 1
1.2. Problem Statement 1
1.3 Overview of the Project 2
1.4 Contributions of the Thesis 2
1.5 Organization of the Report 2
Chapter 2 Background 3
2.1 Convolutional Neural Network (CNN) 3
2.2 LIDAR(Light Detection and Ranging) 3
2.3 Region-based CNN(RCNN) 4
2.4 Faster RCNN 4
Chapter 3 Proposed Design for Optimizing the Processing Time 7
3.1 Object detection and Classification 7
3.2 Proposed Algorithm 11
Chapter 4 Implementation and Performance Evaluation 13
Chapter 5 Conclusion and Future Work 19
References 21

List of Figures

Figure 2.1: Output of a LIDAR…… .4

Figure 2.2: Flow chart for working of RPN……...…5

Figure 3.1: Flow chart of proposed algorithm…… 11

Figure 4.1: Video frame……...13

Figure 4.2: Optical Flow vector output……14

Figure 4.3: Threshold operation output of fig 4.2……14

Figure 4.4:Area filtration, Erosion, Dilation Output……15

Figure 4.5: Road colour filtration output…….15

Figure 4.6: Area filtration output…….16

Figure 4.7: Logic inversion and masking output…….16

Figure 4.8: AND operation output of fig 4.4 and 4.7……...17

Nomenclature

ADAS - Advanced Driver Assist Systems

CNN - Convolutional Neural Network

RCNN - Region-based Convolutional Neural Network

GPU - Graphic Processing Unit

LIDAR - Light Detection and Ranging

RPN – Region Proposal Network

Chapter 1 Introduction

1.1 Motivation

Today, wherever we go, we can definitely find a person operating a smart phone, this is the extent by which the smart phones have penetrated into the market, and also, into our lives. Similarly, in the next 10 years this will be the exact scenario with the self-driving vehicles. The technology required to achieve this feat is currently in the testing phase and world's leading automobile manufacturers like Benz, BMW, Volvo etc have invested huge amounts in its development and are competing to release their autonomous driving vehicles by 2017.

Though this technology can completely change the phase of automobile industry, in the initial phase it is very costly. A lot of research is being done in the search for new algorithms and methods to reduce the system cost and to make it feasible. So, this project is one such effort to reduce the cost of the system by minimizing the usage of GPU and it is being done in collaboration with Tata Elxsi.

1.2 Problem Statement

- To develop an indigenous system to detect and classify on road objects with minimal usage of GPU
- To design, develop and train a CNN incorporating recent advancements in cost and back propagation algorithm

1.3 Overview of the Project

This project presents a vision based solution for object detection and classification. The reason behind going for a vision based approach instead of LIDAR is that, it reduces the overall cost of the system, making it commercially feasible. The existing solutions for object detection and classification [1],[2] are developed mainly for datasets constituting images, and also, they require high end GPU for their implementation. To overcome this problem is the chief aim of the project.

1.4 Contributions of the Thesis

In this project, a new algorithm for region proposal is presented that has higher frame rate compared to existing CNN based solutions and requires lower GPU capacity for implementation.

1.5 Organization of the Report

This report is organized as follows. The following chapter briefly describes the conventional CNN and its variant RCNN. Chapter 3 explains the proposed algorithm for region proposal. Its implementation and performance analyses is given in Chapter 4. Chapter 5 concludes the report.

Chapter 2 Background

2.1 Convolutional Neural Network (CNN)

In the living era of Autonomous vehicles, there are several techniques to detect and classify the on-road obstacles. The state-of-the-art techniques for object detection and classification use CNN as the basis. CNNs are made up of neurons which have weights and biases. These neurons receive inputs from an image, perform a dot product with the weights and generate a scoring output. The entire network expresses a single score function which is always differentiable. These functions take the raw image pixels as input and output the class scores. The weights are updated by means of backpropagation with reference to the loss function. The loss function like Support Vector Machines (SVM)/Softmax is generally used in CNNs on the last Fully Connected (FC) layer. To build a ConvNet we use three layers namely convolutional layer, pooling layer and fully-connected layer. Each layer can input a 3D data and transform it to a 3D data output.

2.2 LIDAR(Light Detection and Ranging)

The LIDAR instrument fires rapid pulses of laser light at a surface, some at up to 150,000 pulses per second. A sensor on the instrument measures the amount of time it takes for each pulse to bounce back. Light moves at a constant and known speed so the LIDAR instrument can calculate the distance between itself and the target with high accuracy. By repeating this in quick succession the instrument builds up a complex 'map' of the surface it is measuring.

region proposals. Fig 2.1 displays the output from a LIDAR that is mounted on a car.

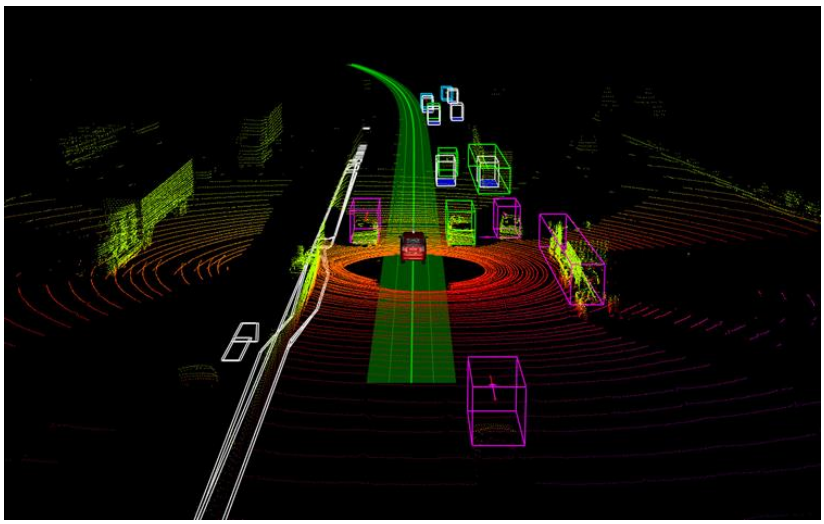

Figure 2.1: Output of a LIDAR

2.3 Region-based CNN(RCNN)

CNNs are able to give better mean Average Precision (mAP) in object detection and classification, but take a lot of time in training. In order to optimize the training time as well as detect1 1ion time, a modified version of CNN called RCNN [3] was proposed which gave 58.5% mAP on PASCAL VOC 2007. It consists of three modules. The first module generates a set of category-independent region proposals. The second module is a large CNN that extracts a fixed-length feature vector from respective region proposals.

2.4 Faster RCNN

Faster RCNN [4] is a new method to realize RCNN for better performance on mAP as well as execution time. In this method, region proposals are made by a separate convolutional network called Region Proposal Network (RPN). This network shares the convolutional features with the Detection Network (RCNN). RPNs are trained to generate better region proposals, which

are used for detection by the detection network. RPN and RCNN are together trained which share a set of convolutional features. The sliding window size is chosen as 3x3 and is mapped to a lower-dimensional vector (256-d). This vector is sent to teo layers called box-regression layer (reg) and box-classification layer (cls). The k region proposals are predicted such that the reg layer has 4k outputs (coordinates of k boxes) and the cls layer has 2k scores (object or not). These proposals are relative to k reference boxes, called anchors. The working of RPN is described in the Figure. 2.2. Faster RCNN achieved 73.2% mAP on PASCAL VOC 2007 [5] using 300 proposals per image.

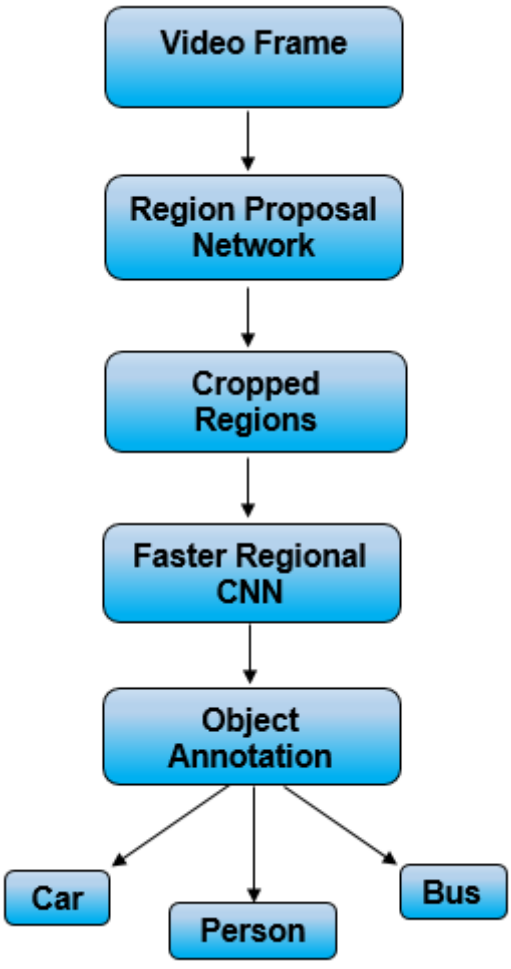

Figure 2.2: Flow chart for working of RPN

Chapter 3 Proposed Design for Optimizing the Processing Time

3.1 Object detection and Classification

The system discussed in the previous chapter was published in 2016, is by far the most effective one in object detection and classification. The region proposal network can produce over 300 region proposals per network, but, the only problem with its practical implementation is that it requires a high end. On a typical 2GB GPU it processes over 2~3 frames per second. Inorder for it to be implemented over ADAS, a frame rate greater than 30 frames per second has to be achieved, with minimal usage of GPU.

So, in order to reduce the processing time and the overall cost of the system, we are proposing a non-neural network based solution for object detection. The initial design of this function is implemented in MATLAB. This function will generate the region proposals which will be given as input to the CNN in the next stage to classify the objects. The function is developed by utilizing the built in functions in MATLAB Computer Vision Library. It is made sure that the built-in functions that are utilized are available in Open CV, so that, shifting to a real time platform will be easy.

The following is the brief description of the built in functions being utilized:

1. **OpticalFlow System object**:
   This estimates object velocities from one image or video frame to another. It uses either the Horn-Schunck or the Lucas-Kanade method.

a) Horn-Schunck Method:

By assuming that the optical flow is smooth over the entire image, the Horn-Schunck method computes an estimate of the velocity field, $[uv]^T$, that minimizes this equation:

In this equation, $\partial u/\partial x$ and $\partial u/\partial y$ are the spatial derivatives of the optical

$$E = \iint (I_x u + I_y v + I_t)^2 dx dy + \alpha \iint \left\{ \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \right\} dx dy \quad (3.1)$$

velocity component, u, and α scales the global smoothness term. The Horn-Schunck method minimizes the previous equation to obtain the velocity field, [u v], for each pixel in the image. This method is given by the following equations:

In these equations, $[u_{kx,y} v_{kx,}]$is the velocity estimate for the pixel at (x,y),

$$u_{x,y}^{k+1} = \bar{u}_{x,y}^{k} - \frac{I_x [I_x \bar{u}^{k}_{x,y} + I_y \bar{v}^{k}_{x,y} + I_t]}{\alpha^2 + I_x^2 + I_y^2}$$

$$v_{x,y}^{k+1} = \bar{v}_{x,y}^{k} - \frac{I_y [I_x \bar{u}^{k}_{x,y} + I_y \bar{v}^{k}_{x,y} + I_t]}{\alpha^2 + I_x^2 + I_y^2} \quad (3.2)$$

and $[u_{kx,y} v_{kx,y}]$is the neighbourhood average of $[u_{kx,y} v_{kx,y}]$ For $k = 0$, the initial velocity is 0.

b) Lucas-Kanade method:

To solve the optical flow constraint equation for u and v, the Lucas-Kanade method divides the original image into smaller sections and assumes a constant velocity in each section. Then, it performs a weighted least-square fit of the optical flow constraint equation to a constant model for $[uv]T$ in each section Ω. The method achieves this fit by minimizing the following equation:

$$\sum_{x \in \Omega} W^2 [I_x u + I_y v + I_t]^2 \quad (3.3)$$

W is a window function that emphasizes the constraints at the centre of each section. The solution to the minimization problem is

$$\begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum W^2 I_x I_t \\ \sum W^2 I_y I_t \end{bmatrix} \tag{3.4}$$

2. **BlobAnalysis System Object:**

   The BlobAnalysis object computes statistics for connected regions in a binary image. It computes the following parameters for the connected regions: area, centroid, bounding box, major-axis, minor-axis, orientation, eccentricity, equivalent diameter, perimeter.

3. **Dilation:**

   The binary dilation of A by B, denoted A⊕B, is defined as the set operation:

$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \emptyset \right\}, \tag{3.5}$$

   where B is the reflection of the structuring element B. In other words, it is the set of pixel locations z, where the reflected structuring element overlaps with foreground pixels in A when translated to z. Note that some people use a definition of dilation in which the structuring element is not reflected.

   In the general form of *gray-scale dilation*, the structuring element has a height. The gray-scale dilation of $A(x,y)$ by $B(x,y)$ is defined as:

$$(A \oplus B)(x,y) = \max \left\{ A(x-x', y-y') + B(x', y') \middle| (x', y') \in D_B \right\}, \tag{3.6}$$

   where D_B is the domain of the structuring element B and $A(x,y)$ is assumed to be $-\infty$ outside the domain of the image. To create a structuring element with nonzero height

values, use the syntax strel(nhood,height), where height gives the height values and nhood corresponds to the structuring element domain, D_B.

Most commonly, gray-scale dilation is performed with a flat structuring element ($B(x,y) = 0$). Gray-scale dilation using such a structuring element is equivalent to a local-maximum operator:

$$(A \oplus B)(x,y) = \max\{A(x-x', y-y') \mid (x', y') \in D_B\}. \quad (3.7)$$

4. **Erosion:**

The *binary erosion* of A by B, denoted $A \ominus B$, is defined as the set operation $A \ominus B = \{z|(B_z \subseteq A\}$.

In other words, it is the set of pixel locations z, where the structuring element translated to location z overlaps only with foreground pixels in A.In the general form of *gray-scale erosion*, the structuring element has a height. The gray-scale erosion of $A(x, y)$ by $B(x, y)$ is defined as:

$(A \ominus B)(x, y) = \min \{A(x + x', y + y') - B(x', y') \mid (x', y') \in D_B\}$,

where D_B is the domain of the structuring element B and $A(x,y)$ is assumed to be $+\infty$ outside the domain of the image. To create a structuring element with nonzero height values, use the syntax strel(nhood,height), where height gives the height values and nhood corresponds to the structuring element domain, D_B.

Most commonly, gray-scale erosion is performed with a flat structuring element ($B(x,y) = 0$). Gray-scale erosion using such a structuring element is equivalent to a local-minimum operator:$(A \ominus B)(x, y) = \min \{A(x + x', y + y') \mid (x', y') \in D_B\}$.

3.2 Proposed Algorithm

The following figure3.1 is the flow chart of the proposed algorithm

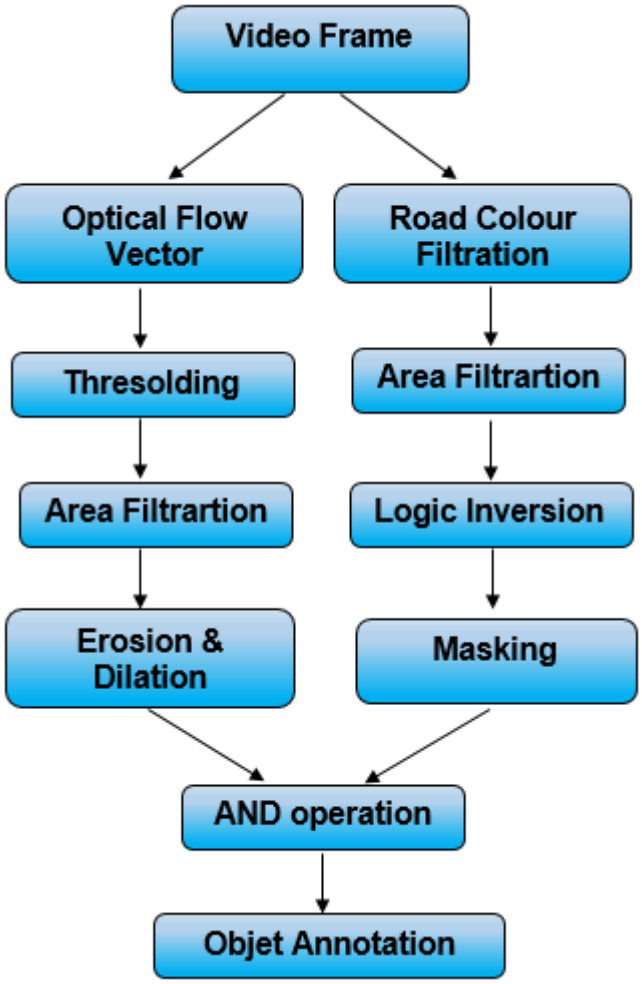

Figure 3.1: Flow chart of proposed algorithm

The main goal of the algorithm is to separate the on road objects from the relatively moving backgrounds. There are many object detection and tracking algorithms in MATLAB, it has to be noted that, all those algorithms are meant for stationary backgrounds.

Here, the central idea on which the algorithm is constructed is that :

- All the objects that lie on the black surface(road) can be considered as on-road objects.

- Since the colour of the road is uniform, this gets undetected when relative velocity vector matrix is calculated.
- In any video frame the level of the road extends only till the centre of the frame, because, generally the camera is mounted to the inner rear view mirror of the car.

In the next chapter, we can see how these ideas adhere each other in producing the required output.

Chapter 4 Implementation and Performance Evaluation

4.1. System Configuration:

The proposed system is implemented on windows workstation with AMD GPU in MATALB. The GPU has 2GB graphics memory and the workstation is powered by Intel i5 with 8GB RAM.

This algorithm is processing frames at rate of 10 frames per second.

4.2. Level wise output of the Algorithm:

- Fig 4.1 is the input video frame to the algorithm

Figure 4.1: Video frame

- Fig 4.2 displays the magnitude of the velocity vector generated by the Optical Flow function

Figure 4.2: Optical Flow vector output

- Fig 4.3 displays the output generated by filtering the velocity magnitudes that are less than the average value

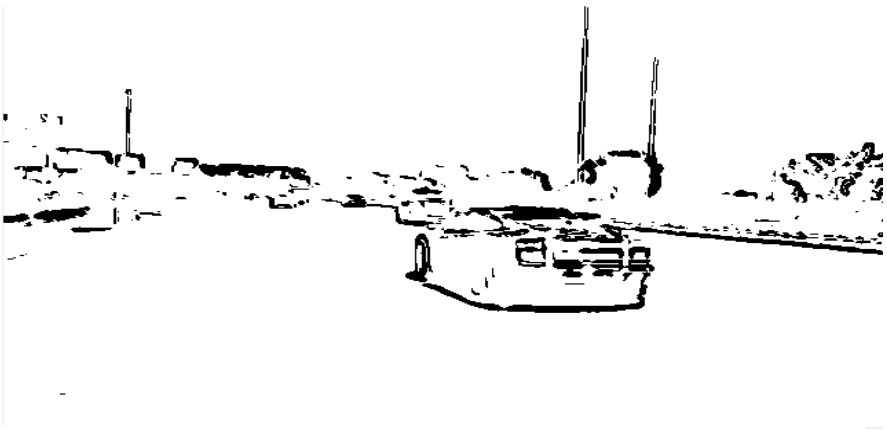

Figure 4.3: threshold operation output of fig 4.2

- Fig 4.4 displays the output of the erosion, dilation and area filtration operations performed on fig 4.3

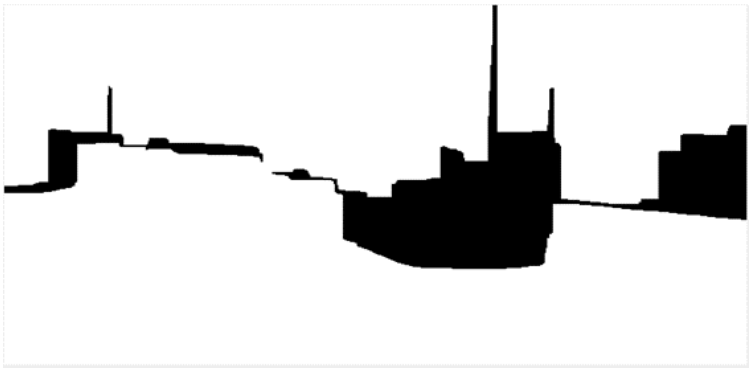

Figure 4.4: Area filtration, Erosion, Dilation Output

- Fig 4.5 is the output obtained from filtering the road colour from fig 4.1

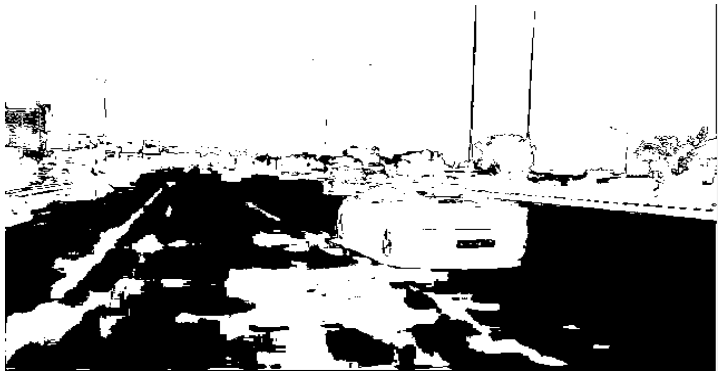

Figure 4.5: Road colour filtration output

- Fig 4.6 is the output obtained after filtering the white areas that are less 30 percent of the total image area in fig 4.5

Figure 4.6: Area filtration output

- Fig 4.7 is the output from logic inversion and masking operation performed on fig 4.6

Figure 4.7: Logic inversion and masking output

- Fig 4.8 displays the output obtained by performing an ‘AND’ operation between fig4.4 and 4.7

Figure 4.8: AND operation output of fig 4.4 and 4.7

- Fig 4.9 displays the overlapped form of the generated bounding box and fig 4.1

Figure 4.9: Object Annotation Output

Chapter 5 Conclusion and Future Work

In this Project, an optimized algorithm for generating region proposals is discussed. The developed algorithm produces a higher frame rate compared to existing methods for region proposals, and also, requires lesser GPU capacity for implementation. Although the proposed algorithm detects almost all vehicles on the road, it sometimes detects the trees, signboards, and similar unconcerned objects. Yet, these redundant region proposals can be filtered by the object classifier.

In future, this algorithm can be further improvised such that, it can track the detected objects. Also, the CNN based object classifier has to be developed incorporating the latest advances in cost and backpropagation algorithms.

References

[1] A. Mukhtar, L. Xia and T.B. Tang, “Vehicle detection techniques for collision avoidance systems: A review”, IEEE Transactions on Intelligent Transportation Systems, Vol. 16, No. 5, Oct. 2015.

[2] S. Sivaraman and M.M. Trivedi, “Looking at vehicles on the road: A survey of visionbased vehicle detection, tracking, and behavior analysis”, IEEE Transactions on Intelligent Transportation Systems, Vol. 14, No. 4, Dec. 2013.

[3] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142-158, Jan. 1 2016. doi: 10.1109/TPAMI.2015.2437384

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sunar “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” arXiv: 1506.01497v2 [cs.CV] 13 Sep 2015

[5] PASCAL VOC 2007 Dataset. [Online]. Avilable: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html