

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

Report Submitted for Review I

Sai Sravan Manne
Roll No. EDM13B018

Guided by:
Dr. Binsu J Kailath

Abstract

Advanced driver assistance systems (ADAS) are one of the fastest-growing segments in automotive electronics, some currently available ADAS are adaptive cruise control, automatic parking system, collision warning, autonomous navigation etc. The pinnacle of an ADAS is auto pilot system i.e. a self-driving vehicle. Auto pilot system in automobiles is predicted to be the next disruptive technology in the upcoming decade. In auto pilot system real time object classification is one of the major modules.

A typical set of objects that are seen on road are dogs, cats, cow, pedestrians, motorbikes, cars, trucks, trees, dustbins, garbage. In the past several algorithms are proposed based on edge, pattern and texture recognition to identify these objects. But recently in 2012 a deep learning based algorithm called Convolutional Neural Network (CNN) [1] has demonstrated highest level of accuracy with very minimum computation time, this characteristic of CNN is very much required in ADAS because the reaction needs to be less than few mille seconds in order to avoid accidents. Hence, from then on CNN's are used for object classification in ADAS.

Objective

The chief objective is to develop a Convolutional Neural Network (CNN) and train it to classify on road objects with. This entire project involves the following stages:

1. Understanding the generic architecture of CNN.
2. Detailed study on existing CNN based algorithms like Alexnet, googlenet etc.
3. Design, development and training of a new CNN incorporating the recent advancements in cost and back propagation algorithms.
4. Adopting the neural network to classify objects from video.

5. Path estimation for the detected vehicles in the video.

Work Done

A detailed study is made on the complexities involved in using a CNN for object detection from videos [1][2][3] using Alexnet and VGG Net, and the inferences are listed below.

Problems faced due to direct implementation of CNN for object detection from videos:

1. Variable confidence levels and object detection boundaries:

Despite their effectiveness on still images, these still image object detection CNN's are not specifically designed for videos. One key element of videos is temporal information, because locations and appearances of objects in videos should be temporally consistent, i.e. the detection results should not have dramatic changes over time in terms of both bounding box locations and detection confidences. However, if still-image object detection CNN's are directly applied to videos, the detection confidences of an object show dramatic changes between adjacent frames and large long-term temporal variations, as shown by an example in Fig. 1 (a).

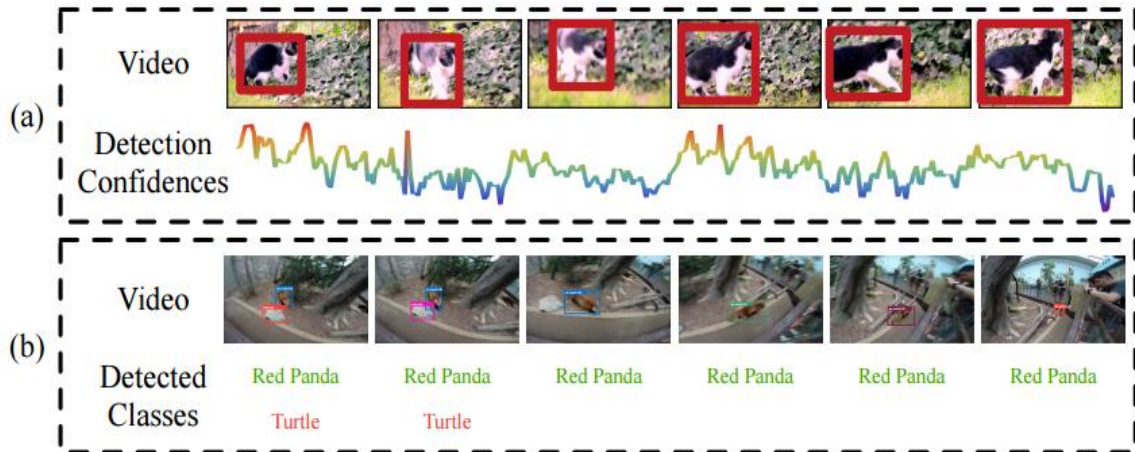


Fig. 1. Limitations of still-image object detection CNN's on videos. (a) Detections from still-image detectors contain large temporal fluctuations, because they do not incorporate temporal consistency and constraints. (b) Still-image detectors may generate false positives solely based the information on single frames, while these false positives can be distinguished considering the context information of the whole video.

2. Increased execution time:

Time required to classify 384*512*3 image using Alexnet and VGG Net individually, on a 2GB GPU is specified in Fig 2, Fig 3.

Profile Summary

Generated 04-Oct-2016 08:59:41 using cpu time.










Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
matconv	1	2.023 s	1.582 s	
run	1	0.282 s	0.005 s	
vl_setupnn	1	0.272 s	0.002 s	
addpath	5	0.262 s	0.007 s	
path	5	0.254 s	0.203 s	
vl_simplenn	1	0.069 s	0.009 s	
general\private\parsedirs	10	0.051 s	0.050 s	
graphics\private\clo	2	0.040 s	0.005 s	
vl_nnconv (MEX-file)	8	0.039 s	0.039 s	

Fig2. Execution time for Alexnet for a 384*512*3 image.

Profile Summary

Generated 04-Oct-2016 08:24:33 using cpu time.







Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
matconv	1	1.936 s	1.477 s	
run	1	0.289 s	0.010 s	
vl_setupnn	1	0.279 s	0.000 s	
addpath	5	0.279 s	0.010 s	
path	5	0.269 s	0.209 s	
vl_simplenn	1	0.070 s	0.000 s	
general\private\parsedirs	10	0.060 s	0.060 s	
graphics\private\clo	2	0.060 s	0.000 s	
setdiff	3	0.060 s	0.020 s	
setdiff>setdifflegacy	3	0.040 s	0.010 s	
vl_nnconv (MEX-file)	8	0.040 s	0.040 s	

Fig 3, Execution time for VGG Net for a 384*512*3 image.

For a single time frame, approximately 1.5 to 2 seconds is being consumed to classify the image, let us assume that final system would be implemented on a 8GB GPU, which will reduce the execution time to 0.3-0.6 sec.

However, in a typical video frame there will be at least 1 to 10 region proposals as shown in the Fig4 which have to be classified and labelled. This entire process for region proposal generation, and classification of the regions proposals should occur within less than 1 sec.

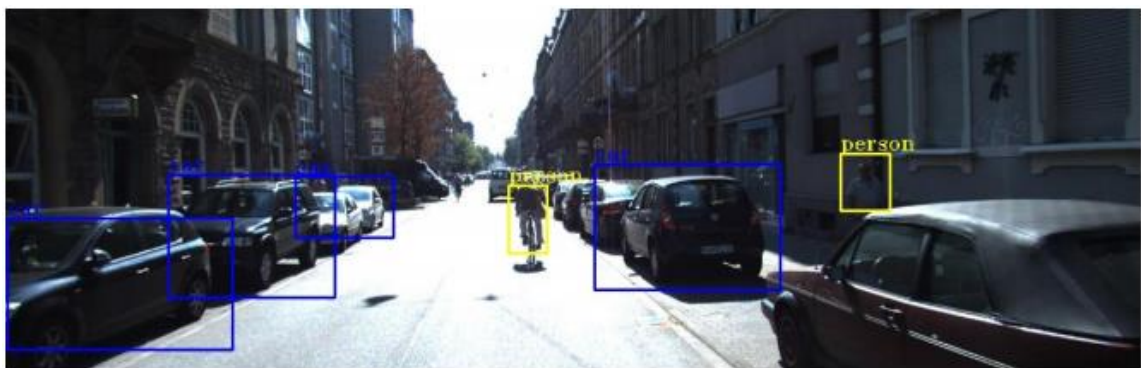


Fig 4.

So, in order to reduce the execution without any compromise on CNN complexity the following techniques can be used:

1. Instead of using a separate CNN for region proposal we can make use of a simple hack in our datasets i.e. as our datasets mainly deal with urban road conditions, it is easy to distinguish the road and the lanes from the vehicles if we calculate velocity vector for individual frames, in grey scale. This can be inferred from the Fig5. MATLAB function being used is: Optical Flow, this function calculates the relative direction and velocity of individual matrix elements (pixel value of the particular frame) from 1 frame to another. The pictures seen in the grey scale (left side) in Fig5

are the overlapped form of the individual frames with their relative velocities with respect to their previous frame.



Fig 5

2. The next technique is, if we can reduce the redundant information in video frames i.e. for every instant there will be 5 to 6 video frames depicting the same scene, so if we were to do region proposal and classification there will be considerable wastage of time. Instead, if we can compare individual frames and transfer the detection information from one frame to another, we can reduce the time period for computation.

This method can also lead to constant confidence scores and less variation in bounding boxes position.

Work to be done

1. The above mentioned methods need to be thoroughly tested to check their reliability.
2. Design, development and training of a new CNN for Indian road conditions, with incorporation of recent advancements in cost and back propagation algorithms.
3. Path estimation for detected vehicles.

References

- [1] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, and Wanli Ouyang, "T-CNN: Tubelets with Convolutional Neural Networks for Object Detection from Videos," published in CVPR 2016.
- [2] Andrej Karpathy, George Toderici, and Sanketh Shetty, "Large-scale Video Classification with Convolutional Neural Networks," published in CVPR 2014.
- [3] Sayanan Sivaraman, and Mohan Manubhai Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, December 2013.