Realtime analysis of vehicles rear lights

1st Sergey Ivanov
Institute of Robotics and Computer Vision
Innopolis University
Innopolis, Russia
r.ivanonv@innopolis.university

2nd Zanina Valeriya
Institute of Robotics and Computer Vision
Innopolis University
Innopolis, Russia
v.zanina@innopolis.university

***Abstract* — With the growth of technology, the requirements for intelligence cars have increased. In our project we proposed and realized a real-time vehicles rear lights detection and state analysis algorithm in order to achieve the intelligent-car's collision warning about the vehicle in front of it. For this task we solved such problems as vehicles detection and tracking, recognition and estimation of the state of taillights. Source code is at** *https://github.com/TechToker/CarLightSignalsDetection*

Keywords* — *Vehicle detection, YOLOv4, tracking, image processing, taillights recognition, computer vision

I. Introduction

The research is divided into several subtasks. The first task is to detect the vehicles on the video. We tried to find the best approach to detect the transports and selected to apply YOLOv4[1]. This method considers object detection as a regression problem. It predicts class probabilities and bounding box offsets from full images with a single feed forward convolution neural network the object detection task is simply identifying and locating all known objects in the frame. But in our case, we need to lock onto a particular moving object in a video. It is tracking task and for this we use centroid tracking algorithm. In the second step, we need to process the received frames to recognize the taillights. Each vehicle will be different in size, shape, and color, but rare lights of vehicles have some features by government's law which can help to do image processing. To extracting candidates in rare lights pairs we take to account that selected areas red in color, close to each other in pairs and they are also symmetrical. Intelligent cars can travel in a difficult environment that can provoke collision. That is one of the reasons why it is necessary to consider the security issues of self-driving vehicles and implement identification of the vehicle braking state in front of it. Such an improvement can help Intelligence vehicles to take appropriate action based on the conditions of the rear lights to further create safety of automatic driving. This algorithm can also be integrated in driver assistance systems for forward collision avoidance. [2] For analyzing stop signals end, we must compare the coordinates with the vehicle's headlights with the threshold of the difference in the colors of the rear brake lights, when the vehicles are not red or yellow. The breaking lights will be amongst the brightest objects in the image.

The paper is organized as follows. Section 2 describes the relevant work. The implemented model is presented in the third part and consists of all stages of the implemented algorithm. Section 4 presents experiments and estimates and finally Section 4 analyzes and observations.

II. Related work

We conducted the literature's research and found different approaches in this field. We looked through at different techniques for detecting a vehicle in the frame. One of the most interesting articles [3] describes an approach to detecting rear-view vehicles based on Gabor filters for feature extraction and Support Vector Machines (SVMs) for detection. Other techniques considered imply the presence of sensors [4-6], shadows [7] or color and texture [8]. Since we are considering a real-time vehicles detection approach, speed was the deciding factor when we choosing the model. We were looking towards of one stage detector in the sense of different types of YOLO [9-10].

Another work [11] describes a system which recognize vehicles at nighttime due to rear lights. It is focuses on close-range detection so that the distance. That is not good approach because it is not considered cases when transport is far or it's lights are broken. [12] Jian-Gang Wang in his work propose a two-stage approach which can robustly detect rear-lights from a single image. In his work he used deep learning technique based on appearance to recognize taillights. For training a classifier the rear rectangle of a vehicle detected by vehicle detector is used.

In our case we use the local feature extraction approach. It analyzes the information only in the current frame. It is based on filtering only in the YCbCr color space. We are looking at a reddish chroma Cr and luma Y channels, and this has the advantage of reducing the amount of data required for processing.

III. Implemented Model

A. *Vehicle Detection and tracking*

To choose the most suitable model for transport detection, we conducted an analysis and settled on YOLOv4. You can see the results of the analysis in chapter V of this article. Since we interested only in estimation on vehicle states, we need to crop every vehicle bounding box and processing it individually.

YOLO processes each frame separately and can't match the car's bounding box on the current frame to the same bounding box on the another (further or previous) frame. To solve this problem, we need to keep follow bounding boxes over the time. The result of tracking is on fig.1. Video tracking is aimed at associating target objects in consecutive video frames. [13] To solve this problem, we implement centroid algorithm with some improvements. It is based on the Euclidean distance between the centroids of the bounding box previous and current frames. The improvement this technique we added reducing the noise through comparison the sizes of boxes. If the box in current position and with the same size and in the previous frame there is a box in a similar position and with a similar size, then we can mark these box as the same box. In accordance with this, we associate the vehicle with its a unique ID.

Fig. 1. Tracking result

If a bounding box is not detected for a long time, then we clear it from the general list of objects. This allows to track an object even if it has disappeared from detection for several frames.

B. *Recognition of rear lights*

Another part of the project's work is to recognize the rear lights of vehicles. To do this we convert every frame to YCrCb Color-Space. Example of cropped image in YCrCb space on fig. 2.

Fig. 2. YcrCb image

Due to this format, it is easy to find high red areas on vehicles which presents the rare lights. Then we apply threshold to remove all non-pure-red pixels. The results shown on pic 3. We considering Y и Cr channels and use advance thresholding for it. To remove the noise, we apply Morphological operations also.

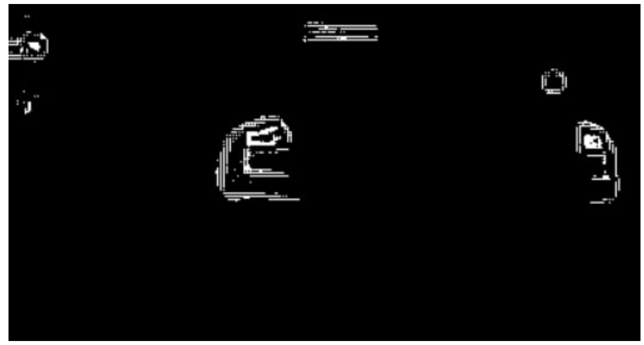

Fig. 3. Image after thresholding

Because the rear part of the car is symmetrical, we can remove the noise pixels by removing areas which don’t have the symmetric part on another side of the car. We perform the X-Y symmetry test and trying to find a pair of area which have the similar Y coordinates and at least one of the areas lie down on the edge of the image. If there are several such pairs, then choose the pair that has a large area. The result in the figure 5, as you can see, two pairs are highlighted in the same own color.

Fig. 4. Detected pair of taillights

C. *Estimation the state of the taillight*

To find the taillight we do a search on the Cr channel and to detect the brightness we use the Y channel. In order to determine the state of the taillight, the following operations are performed: on each frame stored the average value of both channels for detected taillights. When the bounding box of a particular car arrives then we compared the current value of the channels with the average value of these channels 0.2 seconds ago. To reduce random error, we take several frames from the past at once. If the current value is significantly higher than the value from the past, we indicate the state of this car as braking, and if it is significantly lower, then we switch to the status of driving. However, we cannot say anything when we see the first frame of the car and it is assigned the unknown status until the moment of a significant change in brightness. The result presented on fig.5.

Fig. 5. Final result

IV. Experiments and Evaluation

After conducting a significant number of experiments, we concluded that not all cars can be explicitly assigned the correct status due to the wide variation of design and headlights construction. Also, during the experiments, we found that tracking sometimes makes a mistake in traffic jams because there are too many cars close to each other and our tracker can not correctly assign an ID. Two different cars can be given the same identifier for the some period of time. Despite implementing works with the closest solutions [14,15], not all of them could operate during daytime. Due to combining speed detection and tracking we have lower noise and higher speed in real-time.

V. Analysis and Observations

We tried to find the best approach to detect the car in the video. Our first attempt was Haar ‘s cascade detection. The main advantage of the Haar detector is speed. Thanks to the fast image processing, we can easily process streaming video. The Haar detector can recognize most object classes. The classifier is based on AdaBoost.[16], Haarsfeatures are rectangular primitives we used the existing haar cascades that are in the file car.xml. The result you can see on fig. 6.

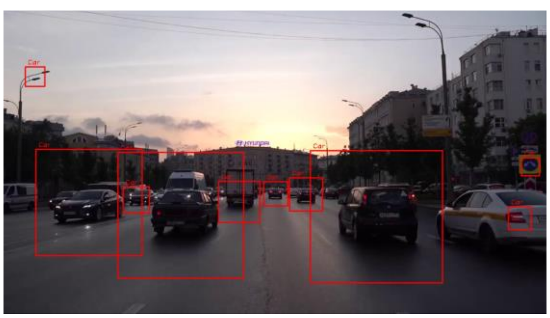

Fig. 6. Haar cascade implementation

That implementation did not give us a high-quality result, but it was quite fast. It gives us over 80 FPS (Tested on Nvidia GeForce 1050Ti). So to improve the results we decided to use YOLOv3. This is a very popular CNN architecture that is used to recognize multiple objects in an image.

In YOLOv3, CNN is applied once to the entire image at once. The network divides the image into a grid and predicts the bounding boxes and probabilities of the desired object for the plot. Result you can see on fig. 7.

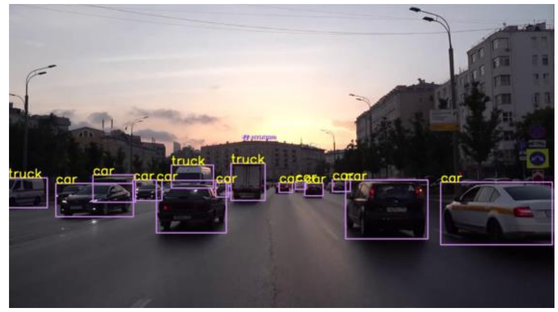

Fig. 7. YOLOv3 implementation

This approach gives us good result in detection, and it is a quite good for real-time streaming. The total result is over 50 FPS. But sometimes predictions of bounding boxes were not so accurate.

We want to do program more accurate, so we decided to use YOLOv4. This is more optimal for using in real-time. We use TensorFlow and it gives us 50 FPS also, but it is analytically looking more precise. The results you can see on fig. 8.

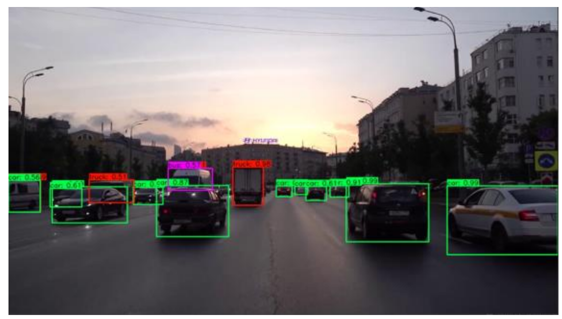

Fig. 8. YOLOv4 implementation

Next, we count each area of pixels as a car taillight. As you can see on the fig. 9 this approach gives quite good but not perfect results, and we chose the other method described before.

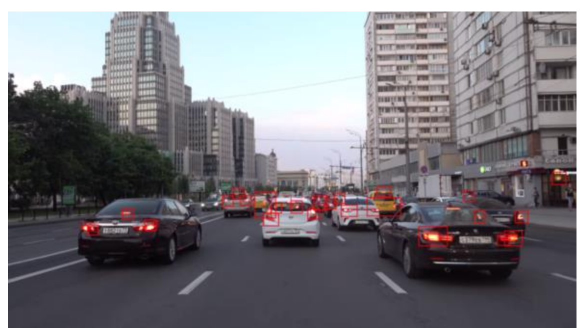

Fig. 9. Estimation taillights due to selection of red pixels

References

[1] Zhao, Liquan; Li, Shuaiyang ,(2020) Object Detection Algorithm Based on Improved YOLOv3. Electronics, 9(3), 537

[2] Skodras, Evangelos & Siogkas, George & Dermatas, E. & Fakotakis, Nikos. (2012). Rear Lights Vehicle Detection for Collision Avoidance. 2012 19th International Conference on Systems, Signals and Image Processing,

[3] Zehang Sun,; Bebis, G.; Miller, R. (2002). 2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 - On-road vehicle detection using Gabor filters and support vector machines. (2), 1019

[4] Chellappa, R.; Gang Qian, ; Qinfen Zheng, (2004) - Vehicle detection and tracking using acoustic and video sensors. (3),793-6.

[5] Matsuo, T.; Kaneko, Y.; Matano, M. (1999) - Introduction of intelligent vehicle detection sensors. 709–713.

[6] Sun, Z.; Bebis, G.; Miller, R. (2004). - Washington, WA, USA- On-road vehicle detection using optical sensors: a review, 585–590.

[7] Liu, Wei; Wen, XueZhi; Duan, Bobo; Yuan, Huai; Wang, Nan (2007). IEEE Intelligent Vehicles Symposium - Rear Vehicle Detection and Tracking for Lane Change Assist. , (), 252–257.

[8] Schiele, B. (2006). Model-free tracking of cars and people based on color regions. Image and Vision Computing, 24(11), 1172-1178.

[9] Ouyang, L., & Wang, H. (2019). Vehicle target detection in complex scenes based on YOLOv3 algorithm. IOP Conference Series: Materials Science and Engineering, 569,

[10] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement", arXiv preprint arXiv:1804.02767, 2018.

[11] O'MALLEY, R., GLAVIN, M., and JONES, E. 2008 Vehicle detection at night based on taillight detection. In Proc. 1st International Symposium on Vehicular Computing System

[12] Wang, Jian-Gang; Zhou, Lubing; Pan, Yu; Lee, Serin; Song, Zhiwei; Han, Boon Siew; Saputra, Vincensius Billy (2016) Appearance-based Brake-Lights recognition using deep learning and vehicle detection, (), 815–820

[13] Gilbert, Alton L.; Giles, Michael K.; Flachs, Gerald M.; Rogers, Robert B.; Hsun, U Yee (1980). A Real-Time Video Tracking System. , PAMI-2(1), 0–56.

[14] A. Akhan, C. Mauricio, and V. Senem. (2012) Autonomous tracking of vehicle rear lights and detection of brakes and turn signals.IEEE,

[15] *J. H. Park and C. S. Jeong*, (2008) “Real-time Signal Light Detection”, Proc. International Conf. on Future Generation Communication and Networking Symposia

[16] Wen-Chung Chang; Chih-Wei Cho (2010). Online Boosting for Vehicle Detection. , 40(3), 0–902.