Vehicle Identification With Speed/Lane Detection

Kaan Apaydin
Computer Science
UNSW
Sydney, Australia
z5075670@ad.unsw.edu.au

Surya Avinash Avala
Computer Science
UNSW
Sydney, Australia
s.avala@student.unsw.edu.au

Mohan Kagita
Computer Science
UNSW
Sydney,Australia
z5124393@ad.unsw.edu.au

***Abstract*—Identify cars in a real time environment and detect the speed of the vehicle as well as the lane it is travelling on.**

I. Background

Vehicle detection and identification systems have become increasingly common in the computer vision community over the last few years [1]–[5]. The key processing steps involved in these systems are extracting the features of images in video frames and classifying them using trained component classifiers to determine the class of the object.

Speed detection has traditionally been achieved via directly measuring the distance from the camera to the vehicle in order to calibrate the speed of the vehicle across frame [6]. For speed cameras specifically, they use Doppler radar technology [15] will measures the changes of microwaves reflected from vehicles in order to obtain the speed.

Many approaches have been applied to lane detection, which can be classified as either feature-based or model-based [7], [8].Feature-based methods detect lanes by low-level features like lane-mark edges [9]–[11]. The feature-based methods are highly dependent on clear lane-marks, and suffer from weak lanemarks, noise and occlusions. Model-based methods represent lanes as a kind of curve model which can be determined by a few critical geometric parameters [12]–[14].

II. Definition of Project Goals

The goals of our project have been defined as follows:

- Vehicle Detection: Being able to locate and identify vehicles in a video sequence.
- Speed Detection: Detecting the speed of the identified vehicles (km/hr) in a video sequence.
- Lane Detection: Being able to detect the lane in which the vehicle is travelling on.

III. Scope of Project

Following on from the previous proposed report, in which we aimed to detect the cars using deep convolutional networks and extract information relating to the vehicle from Roads and Maritime Services website, we have since updated the scope of the project. This is due to upon conducting research, the Roads and Maritime services did not have an API to query in order for us to obtain information about the vehicle such as the model, insurance information, expiration date etc.

The project now centres on identifying vehicles in real time video environment, by training a neural network to spot vehicles and subsequently place a bounding box over any vehicles detected in the frame. We have added in two new features for our project, one is calculating the speed of the vehicle using the relative pixel co-ordinates of the vehicles detected in subsequent frames divided by the focal length and the other is detecting the lane the vehicle is travelling on by analysing the white lane markers on the road, relative to the vehicles position in the video.

IV. Problem Decomposition

The project was decomposed into the following sub-tasks:

A. Vehicle Detection

In this section, Vehicles/vehicle boundaries (primarily Cars) were identified automatically using Deep Learning via neural networks. We have trained Convolutional Neural Network models in order to identify the car/vehicle boundaries on frames of the video. This involved training and optimizing the model to be specific to vehicle classification.

B. Speed Detection

This part mainly focused on detecting the speed of the vehicle in real time by keeping track of the relative pixel positions in every single frame and identifying the changes in positions of the cars detected by using pre-trained vehicle classifier and converting them into real object parameters.To generalize the direction in which the vehicle is progressing subjecting to change in pixel values, Manhattan distance is being calculated between the centroids of the vehicles in subsequent frames and resulting pixel value is converted from image plane to object plane.

C. Lane Detection

This involves obtaining Sobel gradients of the image in order to reveal the edges surrounding the white lane markers of the road. From here, a mask image is needed in order to extract the white lane markers. The space between the lines in the mask is then coloured in order in order to visually demonstrate to the user that the lane has been identified.

V. Implementation

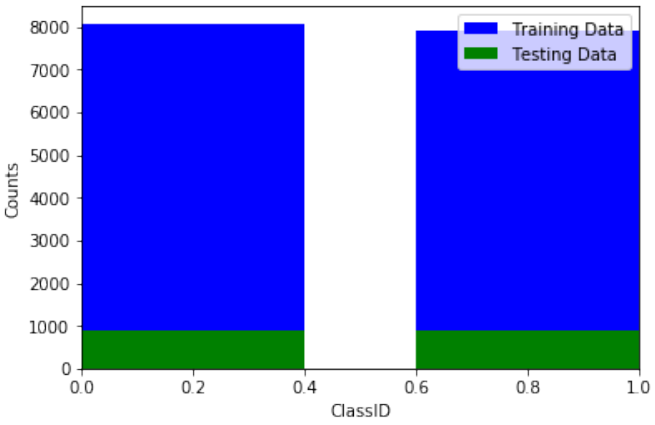
In our approach to solving the problems mentioned in the problem decomposition, we have utilized a combination of neural networks and traditional computer vision processing techniques in order to develop our prototype.

A. Design & Algorithms

Vehicles are detected using a deep learning approach with a fully convolutions network, details of which are explained below.

The data-set is explored here. It is comprised of images taken from the GTI vehicle image database [16], the KITTI vision benchmark suite [18], and the examples were extracted from the test video itself. The data-set is labelled with two classes, cars and non-cars. Cars have a label of 1.0, whereas non-cars have a label of 0.0, as can be seen from the following figure:

A total of 17760 samples were acquired from the above mentioned data-sets, each image is colored and has a resolution of 64x64 pixels. The data-set was then further split into the training set (90%, 15984 samples) and validation set (10%, 1776 samples). From the following distribution, it can be seen that the data-set is nicely balanced, which is important for training the neural network. This balance ensures that the neural network is not biased towards either of the classes. The distribution looks as follows:



The Neural Network's architecture is explained here. This is essentially a binary classification problem, where the neural network has to decide whether the given image is a vehicle or not. The model has 10 layers excluding the input and output layers. The architecture of the model is as follows:

- Input layer, where the 64x64 pixels were fed with all the 3 color channels
- Convolutional Layer 1 & dropout, a 2-dimensional conv layer with a 128 filter, 50% dropout and ReLU activation
- Convolutional layer 2 & dropout, a 2D conv layer with a 128 filter, 3x3 each, 50% dropout and ReLU activation.
- Convolutional layer 3, dropout & maxpool, a 2D conv layer with 8x8 max-pooling, 50% dropout and ReLU activation.
- Dense convolutional layer, 128 dense neuron layer with a ReLU activation
- Dense neuron layer, with hyperbolic tangent (tanh) activation

The full Neural network architecture is summarized below:

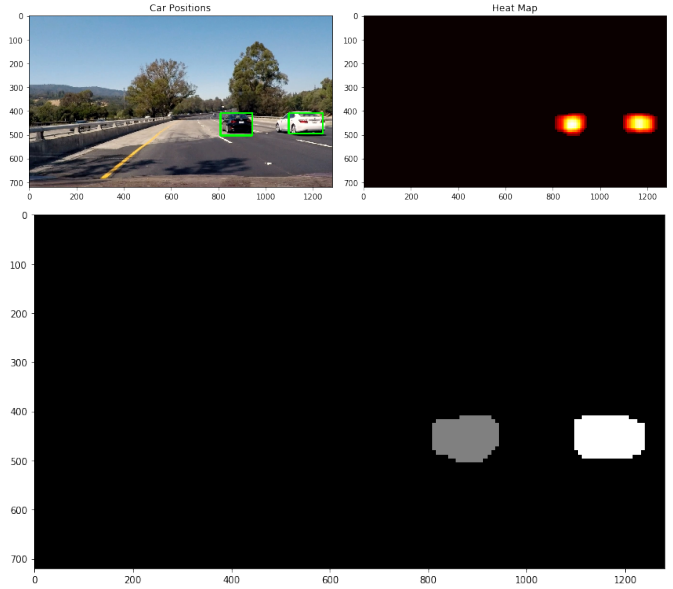
Layer (type)	Output Shape	Param #
lambda_1 (Lambda)	(None, 64, 64, 3)	0
conv1 (Conv2D)	(None, 64, 64, 128)	3584
dropout_1 (Dropout)	(None, 64, 64, 128)	0
conv2 (Conv2D)	(None, 64, 64, 128)	147584
dropout_2 (Dropout)	(None, 64, 64, 128)	0
conv3 (Conv2D)	(None, 64, 64, 128)	147584
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 128)	0
dropout_3 (Dropout)	(None, 8, 8, 128)	0
dense1 (Conv2D)	(None, 1, 1, 128)	1048704
dropout_4 (Dropout)	(None, 1, 1, 128)	0
dense2 (Conv2D)	(None, 1, 1, 1)	129

Total params: 1,347,585
Trainable params: 1,347,585
Non-trainable params: 0

Each frame from the video was fed into the Neural Network, which has been used to detect vehicles everywhere in the frame. This network scales up to be compatible with whatever the input size is, as there is no fully-connected layer at the end, but just a conv layer with max pooling and dropout. Since there is no single-neuron output, the neural network outputs an image like a virtual heat-map. The bounding boxes were then drawn on the hot positions as can be seen below:

A heatmap was created and a small threshold was added

to it to avoid false positives. And a single bounding box can be drawn over every detected heat source, which essentially identifies the vehicles in the images, as can be seen below:



The network parameters (weights) were frozen and stored for future use, since the neural network has to be trained only once. These weights were then merged with Google Tensorflow's pre-trained Object detection models [18] and were later used in predictions.

Once the vehicle is detected using the above trained neural network, the subsequent part that needed to be solved was the speed of the vehicle. In order to achieve this, the video was read using Scikit video package and all video frames with vehicle being detected and bounding boxes drawn around with the help of the Cascade Classifier, were output to a folder. Subsequently, the video frames were iterated over one by one and their coordinates or absolute pixel values of the positions of the detected vehicles were obtained, using the HOG Classifier in order to track the ROI of the vehicle/s present in every frame.

Once the regions of interest of all the vehicles in the frames were obtained, the midpoints of the bounding boxes surrounding the vehicles were obtained, which is held as the "position" of the vehicle which will be key for future frames. This process was repeated for all vehicles to create a list of midpoint values which were updated every frame.

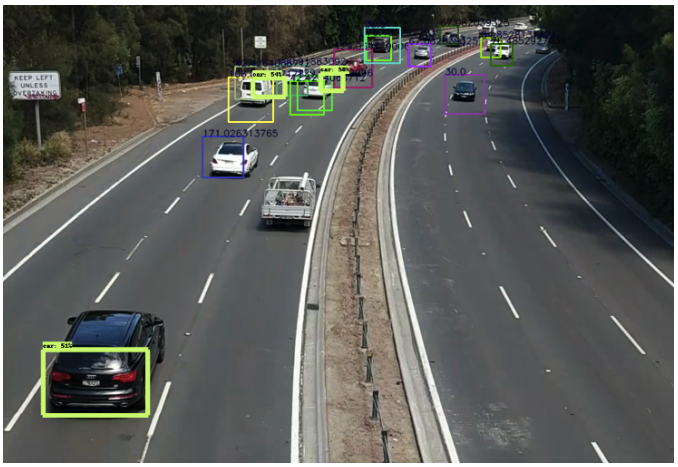
In order to find out the speed of any object the basic parameters that will be needed are distance and time. In this project, we instead are making all calculations in the image plane and then later projecting them in the object plane. In this case, the distance a particular vehicle has travelled in the image plane would be the relative differences in pixels at various vehicle positions in subsequent frames. Time would be the rate at which the frames were being processed, which we assumed to be 30 frames per second for the purposes of this project.

From the above description, **Time = Number of pixels progressed by vehicle/frames per second** and **Distance = sqrt(mid1*mid1+mid2*mid2)** where mid1 and mid2 are the positions of the vehicle in frame 1 and 2 respectively. If the difference is less than 10 pixels, then the car is assumed to be stationary and is neglected when updating values.

Once the speed of the vehicle in the image plane is calculated, we then convert the pixel values obtained into km/sec which is representative of the object plane, using the distance of the object from the camera in the given formula **exact distance to object (mm) = focal length (mm) * real height of the object (mm) * image height (pixels)/object height (pixels) * sensor height (mm)**

However, the trained classifier is unable to detect all the cars in every single frame and thus the obtained speed values are not always accurate and subject to deviate from original value depending on the following assumptions on the parameters. The height of the object in object plane is assumed to be 4meters in general approximating to median values for all cars which is not the same in every case. The distance between the camera and the road is approximated which is not accurate due to practical constraints and hence approximated.

The detected cars and their speeds are best demonstrated in the below figure:



One scenario faced in this project is that the classifier was able to detect the car in one frame and then unable to detect the vehicle several subsequent frames before finally detecting the vehicle once more. This obviously led to erroneous speed measurements as the vehicle is seen travelling one position from one frame to another position in the n'th frame as a transition between two frames. This was rectified by introducing a variable check to determine whether the vehicle which discarded speed readings based on the difference between frames.

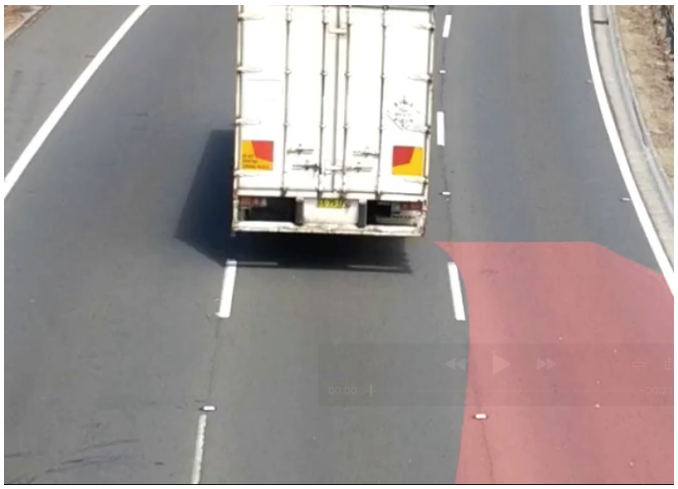
For lane detection, a combination of colour and gradient were used in order to obtain a binary image of the edges. Firstly, the image was converted to a HSV image and the S channel is used to filter the image as this channel is effective at locating yellow and white lines under different lighting conditions. Sobel gradients of the x and y plane are used in order to remove any horizontal line from the image.

Once the binary image of the road markers is obtained, the image is then warped using a perspective transform in order to obtain a "birds eye" view of the road. The road markers that slowly vanish off into the horizon in the original image will now appear to be completely straight in the new warped image. Now depending on whether we are operating on the first video frame or not, two approaches were taken in order to get the positions of lane markers:

- Detection (first frame): Using the binary image, the base of the image is scanned in order to identify the x co-ordinates of the start of the lane markers. This is easily achieved by creating a histogram of the pixel values and then selecting the x positions that have the highest count of white pixel values. Following on, approximately 3 search windows at y positions above the interval between the base x positions are scanned in order to obtain positions of road markers that are further up the road. These search windows are slightly bigger (in terms of width) than the base road marker positions to account for left/right curvature of the road.

- Tracking (N'th frame): Since the road has not changed significantly between the last frame and this frame, most (if not all) x positions obtained previously are still within margins of the road markers. Thus, a search near the top of the image for any new white lines is conducted in order to obtain locations of any new road markers.

Once the x positions of road markers have been obtained, these positions were translated to x co-ordinates back to the original non-warped image and bounding boxes which span the area within the road markers are created and filled in with colour. This image is then shown as an output image to the user in order to demonstrate that the lane has indeed been detected and tracked.



B. Implementation Issues

There were a few implementation issues when it came to developing the project. Firstly, training the neural network with the data-sets available took quite some time which interfered in achieving deadlines for completing tasks and prolonged testing.

When it came to integrating lane detection, we had issues with the laptops not having enough GPU processing power in order to process the video sufficiently. Unfortunately, we were unable to integrate this feature properly and thus did not include it in the project code. During the presentation we were only able to show output videos rendered in Google cloud to demonstrate the feature.

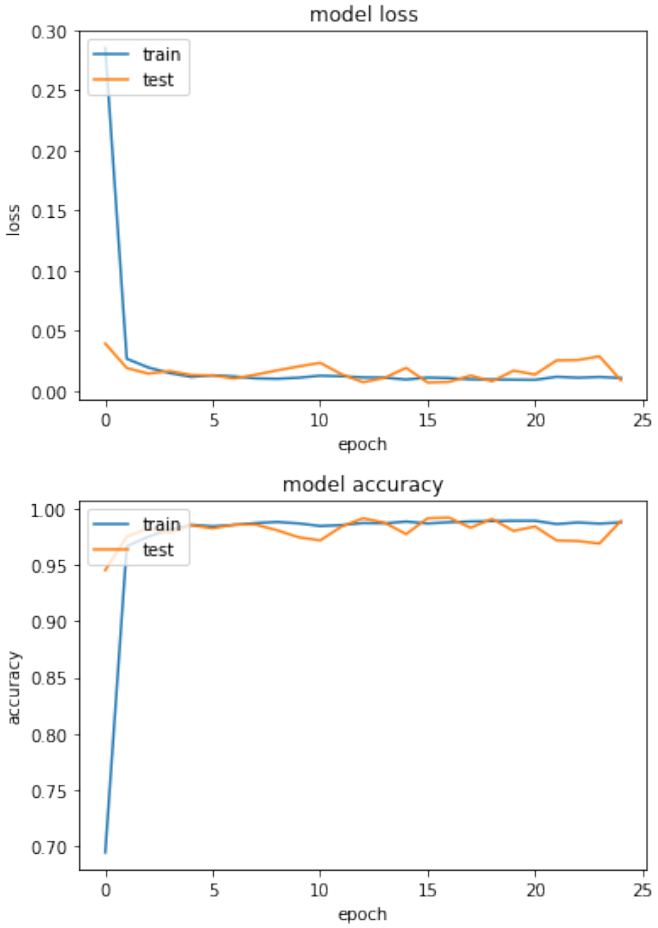
In order to obtain accurate speed detection readings, the focal length parameter is manually changed for each video to reflect different approximate distances between vehicles and the camera, which is obviously prone to inaccuracies.Also the classifier is not able to detect cars in every single frame which is needed to obtain more accurate speeds in object plane.

Furthermore, Number Plate Recognition & Vehicle detail could not be implemented as New South Wales Road and Maritimes association does not officially allow developers to integrate their service. NSW RMS on their server-side, blocks any requests that were made by computer programs.

VI. Design Justifications, Testing & Results

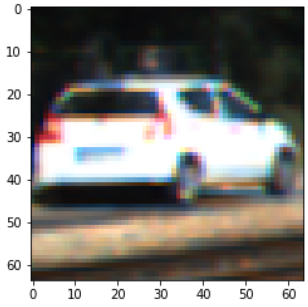
Previous research [19], [20] has shown that Convolutional Neural Networks perform extremely well in Image classification problems because of their ability to exploit feature locality. They do it at different granularities, therefore being able to hierarchically model higher level features. And they can be made translation invariant with pooling units. They are not rotation-invariant per say, but they usually converge to filters that are rotated versions of the same filters, hence supporting rotated inputs.

The Neural Network architecture was finalized because of it's overall accuracy for the task, after training and testing various other models. The current model achieves an accuracy of approximately 99% after 20 epochs of training. The loss and accuracy over time can be seen from the figure below:

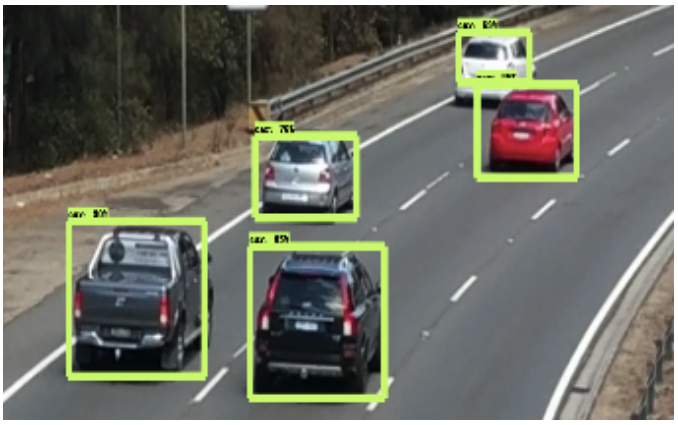


A prediction made by the model on a random sample test image can be seen:

```
NN Prediction: CAR with value 1.0
Ground-truth: CAR with value 1.0
```



Next, a prediction on a full 1920 x 1080 real-world image/frame can be seen:



In order to test our prototype,samples of videos were collected across different locations around Sydney to capture vehicles with varying speeds and backgrounds. The testing phase allowed us to fix few bugs in our implementation which needs improvements in efficiency of the Classifier to detect vehicles in every detected frame and work on removing false positives and subtracting stationary objects etc.. one such case where false positives occur is demonstrated in below figure.

The inefficiency of the classifier not detecting vehicles in every frame can be best demonstrated in below figure in which the car is not detected.

Drawing a comparison between predicted vs obtained values,we are successful in most of the cases except for the few cases mentioned in the Implementation issues which are limited by many real time constraints such as fps of camera,height of vehicles,focal length etc.

VII. Conclusion & Future Extensions

In conclusion, although there were few issues with false positives and the inability to detect cars, the prototype generally performed well during the lab demonstration.

This project being one of the most focused topics in real time vehicle identification systems can be extended to detect collisions. One obvious extension would be to design a program to detect if there any accidents depicted in real time video and notify emergency services if so. Another possible extension is to perhaps solely focus on lane detection, and adding features to detect abrupt lane departures.

VIII. Individual Contributions

Since the project had 3 core modules, the tasks were split into three and assigned to the team as follows:

- Vehicle Identification - Surya Avinash Avala
- Speed Detection - Mohan Kagita
- Lane Detection - Kaan Apaydin

References

[1] Bernd Heisele, Ivaylo Riskov, and Christian R. Morgenstern, Components for Object Detection and Identification.

[2] Bileschi, S., Wolf, L.: A unified system for object detection, texture recognition, and context analysis based on the standard model feature set. In: British Machine Vision Conference (BMVC) (2005)

[3] Heisele, B., Serre, T., Pontil, M., Vetter, T., Poggio, T.: Categorization by learning and combining object parts. In: Neural Information Processing Systems (NIPS), Vancouver (2001)

[4] Poggio, T., Edelman, S.: A network that learns to recognize 3-D objects. Nature 343, 163266 (1990)

[5] Weber, M., Welling, W., Perona, P.: Towards automatic dscovery of object categories. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (June 2000)

[6] Yong-Kul Ki, Doo-Kwon Baik, "Model for Accurate Speed Measurement Using Double-Loop Detectors", The IEEE transactions on Vehicular Technology, vol. 55, no. 4, pp. 1094-1101, July 2006.

[7] J.C. McCall and M.M. Trivedi, Video-based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation, IEEE Transactions on Intelligent Transportation Systems, vol.7, pp.20-37, 2006.

[8] Y.Wang, E. K.Teoh and D. Shen, Lane Detection and Tracking Using B-snake, Image and Vision Computing, vol. 22, pp. 269-280, 2004.

[9] A. Broggi and S. Berte, Vision-based Road Detection in Automotive Systems: a Real-time Expectation-driven Approach, Journal of Artificial Intelligence Research, vol.3, pp. 325-348, 1995.

[10] M. Bertozzi and A. Broggi, GOLD: A Parallel Realtime Stereo Vision System for Generic Obstacle and Lane Detection, IEEE Transactions of Image Processing, pp. 62-81, 1998.

[11] S.G. Jeong, C.S. Kim, K.S. Yoon, J.N. Lee, J.I. Bae, and M.H. Lee, Real-time Lane Detection for Autonomous Navigation,IEEE Proc. Intelligent Transportation Systems, pp. 508513, 2001.

[12] Y.Wang, E.K. Teoh and D. Shen, Improved Lane Detection and Tracking Using B-snake, Image and Vision Computing, vol. 20, pp. 259-272, 2005.

[13] C. R. Jung and C. R. Kelber, A Lane Departure Warning System Using Lateral Offset with Uncalibrated Camera, Proc. IEEE Conf. on Intelligent Transportation Systems, pp.102-107, 2005.

[14] D.J. Kang, J. W. Choi and I.S. Kweon, Finding and Tracking Road Lanes Using Line-snakes, Proceedings of Conference on Intelligent Vehicle, pp. 189-194, 1996.

[15] S. M. Paing, S. S. Y. Mon, H. M. Tun, "Design And Analysis Of Doppler Radar-Based Vehicle Speed Detection", INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 5, ISSUE 06, JUNE 2016.

[16] J. Arrspide, L. Salgado, M. Nieto, Video analysis based vehicle detection and tracking using an MCMC sampling framework, EURASIP Journal on Advances in Signal Processing, vol. 2012, Article ID 2012:2, Jan. 2012 (doi: 10.1186/1687-6180-2012-2)
[17] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, "Vision meets Robotics: The KITTI Dataset", International Journal of Robotics Research, 2013.
[18] Google, "Tensorflow pre-trained Object Detection Models", https://github.com/tensorflow/models/tree/master/research/slim, 2017.
[19] Y. LeCunn, L. Bottou, Y. Bengio, P. Haffner, "Gradient-Based Learning Applied to Document Recognition", IEEE, 1998.
[20] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", AlexNet, 2012.