Camera Based Object Detection for Autonomous Vehicles

Saurabh Amarnath Mahindre
University at Buffalo
smahindr@buffalo.edu

Sonal Vijay Kelwadkar
University at Buffalo
skelwadk@buffalo.edu

Dana Moukheiber
University at Buffalo
danamouk @buffalo.edu

***Abstract*—This paper addresses the issue of object detection for autonomous vehicles in different weather conditions. Various methods like Fast RCNN, Mask RCNN, Linear SVM, HOG & NSM as well as RetinaNet models have been studied. Our model has been trained on the open dataset released by Waymo and the Canadian Adverse Driving Conditions Dataset released by UWaterloo. Pre Trained object detection models on standard datasets like COCO (Common Objects in Context) datasets have also been used for comparison and transfer learning. These datasets assisted the model to detect other vehicles and objects commonly found while driving. The results compare various deep learning architectures and linear models for this task.**

I. Introduction

The safety and reliability of autonomous driving vehicles depends majorly on its capability of detecting objects in its surroundings and making appropriate decisions in a fraction of second. Lidar and camera are used together for detecting objects in the surroundings. However, in this paper we focus only on the input from the cameras. Most of research and industry work is focused on deploying vehicles in fair-weather locations. Currently, there is a gap in research and industry with the aspect of adapting self-driving cars in bad weather. In this paper, we study the methods to address this issue. We have studied object detection algorithms to be able to detect objects in different weather conditions. Snowfall and foggy weather are found to be particularly challenging.

We have used Waymo, YOLO and COCO datasets for training our algorithm. We will be working on one or combination of these public datasets to build a machine learning pipeline to be able to detect objects, depending on the normal or adverse weather conditions. Some variants of RCNN propose regions with possible candidates for object bounding boxes in a captured image. Then a classifier runs on these proposed bounding boxes. The bounding boxes are then refined by detecting and removing duplicates. Based on other objects in the captured image, the boxes are rescored.

YOLO uses direct regression to detect the objects and their bounding boxes and get their classification probabilities. YOLO which stands for You Only Look Once evaluates the captured image in just one glance. This makes YOLO models very fast even though their network is comparatively larger than other approaches. The YOLO model directly optimizes on detection performance and is trained on full images. Thus, YOLO considers the entire image which reduces its background error.

Waymo Open Dataset consists of high-resolution sensor data collected by Waymo self-driving cars [1]. The data consists of segments of sensor data, each of 20 seconds collected at 10Hz (200000 frames) in a variety of geographies and weather conditions. Waymo open dataset provides 1 mid-range lidar data, 4 short-range lidar data and 5 cameras (front and sides) data. However, for this paper, we consider only the camera data and do not integrate the lidar data with it. In Waymo dataset, 4 object classes, vehicles, pedestrians, cyclists and signs are labeled. High-quality labels are provided for camera data in 1000 segments and 11.8M 2D bounding boxes with tracking IDs. Waymo dataset has a rich variety of geographies from urban areas to small towns thus allowing the variety from larger crowded roads to smaller streets. Along with the variety in geographies, the dataset covers the images of these places in different weather conditions. Such a dataset is unavailable currently from any other source. Thus, for object detection in different weather conditions, we use the Waymo open dataset majorly.

The Canadian Adverse Driving Condition Dataset was used for training models with images taken in harsh climate and difficult visible conditions like heavy snowfall and rains [2]. This dataset was used in combination with the waymo dataset.

In this paper, we wanted to study and understand the methods that are efficient in detecting objects while an autonomous vehicle runs during different weather conditions. After implementing object detection algorithms using pre-trained models with tensorflow, we have implemented the algorithms to work with the Waymo dataset. The methods which are implemented in this paper are Fast RCNN, mask RCNN and a model with linear SVM, HOG & NSM model. We evaluate these methods on the Waymo dataset with varied conditions and observe the performance of each of these models.

The evaluation metrics section explains the metrics and how they are calculated for all of those which are used in the evaluation of models. The model evaluation section shows us the results for each of the implemented models, followed by the conclusion section in which the final observation is concluded.

II. Method

A. Fast RCNN

We looked into a transfer learning approach, whereby we used knowledge gained from solving object detection problems on COCO dataset and applied it to normal weather images to detect cars. The model used is a Fast RCNN model, pre-trained on COCO dataset.

Fast RCNN is an object detection convolutional neural network architecture [3]. This method offers certain flexibility and robustness along with fast training and inference time. Fast RCNN is composed of three components. The first component consists of convolution layers whereby filters are trained to extract necessary features of an image. The second component consists of Region Proposed Network (RPN), which is another small neural network that predicts the presence of an object and bounding box of the objects detected. The final component consists of a fully connected layer that takes the regions outputted from RPN and predicts the object class using softmax (classification) and the bounding boxes (regression). To train the model, Stochastic Gradient Descent (SGD) to optimize convolution layer filters, RPN weights and Fully Connected Layer weights as shown in Figure 1.

However, Fast RCNN does not provide for pixel-to-pixel alignment between network inputs and outputs. This can be observed by how coarse spatial quantization is performed by RoIPool for feature Extraction. To fix this issue of misalignment, a quantization-free layer called ROIAlign is proposed in Mask RCNN. This layer preserves exact spatial locations thus improving the accuracy of the model substantially.

B. Mask R-CNN

Mask RCNN is extended from Faster RCNN by adding a branch for predicting segmentation masks [4] [7].
The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner.

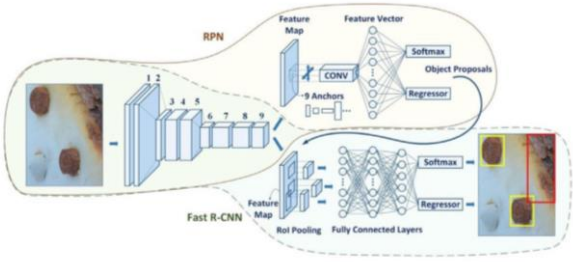

Fig 1. Fast R-CNN Architecture

The masked RCNN algorithm builds on the faster RCNN architecture with two main changes:

1. ROI(Region of Interest) Pooling module is replaced by ROI Align module
2. An additional branch out of ROI Align module is inserted

This ROI Align improves mask accuracy by 10% to 50% showing better gains under stricter localization metrics. It uses Region Proposal Network (RPN) to generate the regions of the image which have higher probability to contain the object.

In this module, each selected region of interest goes through three parallel branches of the network: Label prediction, Bounding Box prediction and Mask prediction.

Each of these Regions of Interest (ROIs) go through non-maxima suppression during prediction. The top N number of the detection boxes are saved. Thus, we get a finer spatial layout of the object.

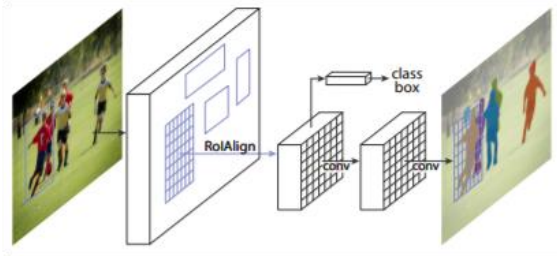

Fig 2. The Mask RCNN framework for instance segmentation

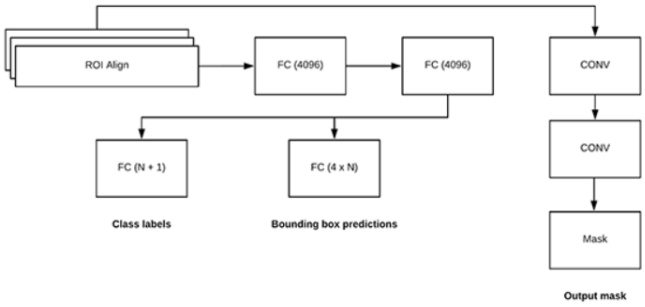

Fig 3. Mask R-CNN Architecture

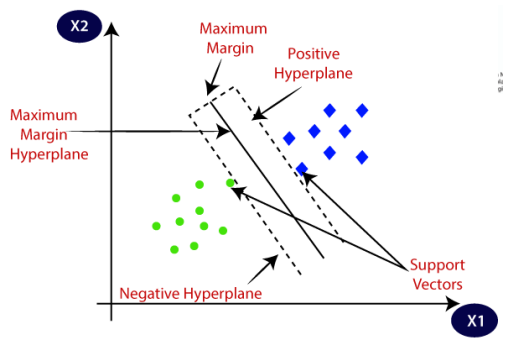

Fig 4. Support Vector Machine

C. Linear SVM, HOG & NSM Model

SVM combined with HOG have been used for object detection, however such methods return multiple bounding boxes. NSM (Non-Maximum Suppression) comes into play to form more compact bounding boxes by ignoring bounding boxes that overlap each other. It does this by clustering bounding boxes by spatial closeness determined by IOU, and keeping the highest confidence score [5].

Linear SVM is used for objects detected where each data point is a point in n-dimensional space, where n represents the number of features. Classification of two different classes is done by finding the max distance from the hyper-plane to the nearest point of either class.

HOG methods work by extracting the gradient and orientation of edges. The orientations are calculated in localised portions and finally histograms are generated for each of the small regions using gradient and orientation of the pixel values.

Object detection using HoG is almost always followed by classification using linear SVM. While training a model with HoG, positive images (the images with objects of interest) and negative images (the images which do not contain the object of interest) are considered. The HoG uses a sliding detection window to compute a HoG descriptor. The linear SVM model then classifies this HoG descriptor as the desired object or not from the labels 1 or 0 that were given to the positive or negative images respectively.

D. RetinaNet

RetinaNet is a combination of networks made of three components [6]:

- Feature Pyramid Net, a backbone network that is built on top of ResNet. This network is responsible for computing convolutional feature maps of an entire image.
- A subnet for performing object classification on the output of Feature Pyramid Net.
- A subnet for performing bounding box regression on the output of Feature Pyramid Net

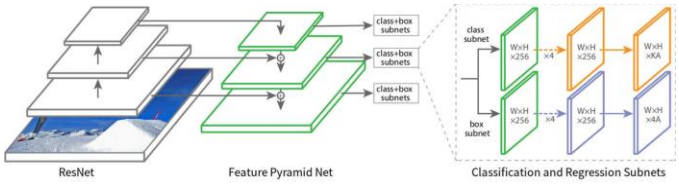

Fig 5. RetinaNet Architecture

The feature pyramid network (FPN), the backbone of RetinaNet network is built in a fully convolutional fashion. RetinaNet uses feature maps at different resolutions to accommodate input images of various sizes and resolutions. This scale-invariant property makes RetinaNet training faster. The higher level feature maps contain graid cells that can cover larger regions in an image and detect larger objects, while lower level grid cells are responsible for detecting the smaller objects in the image.

The classification subnet is a fully connected network (FCN) attached to each FPN level. The subnets consist of 4, 3x3 convolutional layers with 256 filters. And another one with a 3x3 convolutional layer with KxA filters. This is followed by ReLU and sigmoid activations respectively. The Regression Subnet is also attached to each feature map of the FPN parallel to the classification subnet. Its design is similar to classification subnet only difference being the last convolutional layer is 3 x 3 with 4A filters

During prediction, for each of the grid cells in a grid output by a FPN of size n x n, the RetinaNet defines n x n boxes called as anchor boxes. The classification subnet predicts K numbers showing probability distribution of object classes. The regression Subnet predicts 4 numbers which shows the offset between each of the anchor boxes and the corresponding bounding boxes. The objects are then predicted by the anchor boxes. However, multiple anchor boxes may predict the same object. This redundancy is removed by applying non-maximum-suppression (NMS) to each class. box with highest IoU is retained while others are discarded.

III. Evaluation Metrics

A. MAP (Mean Average Precision)

In object detection there are two main tasks summarised under MAP metric:

1. Object Classification to find if an object exists (achieved by AP).
2. Object Localisation to find location of an object (achieved by IOU).

Mean Average Precision is an estimator of the area under the precision-recall curve (AUCPR).The mean Average precision takes the average of AP overall all classes and or over all the IOU thresholds.

B. AP (Average Precision)

Precision (false positive rate) is the ratio of true objects detected over the total number of objects the classifier predicted. (TP/TP+FP). Value close to one indicates whatever the classifier predicts as a positive detection is in fact a correct prediction.

Recall (false negative rate) is the ratio of true objects detected over the total number of objects in the data set. (TP/TP+FN). Value close to one indicates almost all objects that are in your dataset will be positively detected by the model

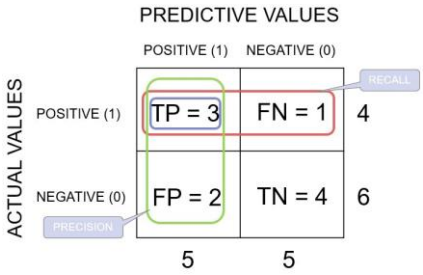

Fig 6. Confusion Matrix

These 2 parameters have an inverse relationship and depend on the model threshold set by the user.

AP is defined as the mean precision at the set of 11 equally spaced recall values, [0, 0.1,..., 1]:

$$AP = \frac{1}{11} \sum_{\text{Recall}_i} \text{Precision}(\text{Recall}_i) \quad (1)$$

B. IOU (Intersection Over Union)

Intersection over Union is an evaluation metric for segmentation(a score greater than 0.5 is considered a good prediction) and similar to dice loss(one wants to minimize dice error). In both evaluation metrics, sensitivity and specificity are combined.

IoU= Area of Overlap  / Area of Union  (2)

The area of overlap is between the predicted and ground truth bounding box and the area of union is the total area covered by predicted and ground truth bounding box.

IV. Data Pre-Processing

The data is a collection of records from one or more TFRecord files. The whole dataset is over 1 TB in size. After the TFRecord files were extracted, we separated image, object labels and bounding boxes for each frame in a video. Each frame has 4 views: Front, Back, Right and Left images and corresponding labels.

V. Preliminary Results

We performed 2D Object detection on a subset of Waymo's 2D Detection open dataset. Following are the results below. Sample results are presented using bounding boxes.

A. Linear SVM, HOG & NSM

The ground truth bounding boxes are the hand-labelled bounding boxes from the testing set.

The predicted bounding boxes are from the model.

Fig 7. Ground truth and prediction using SVM

B. Mask RCNN

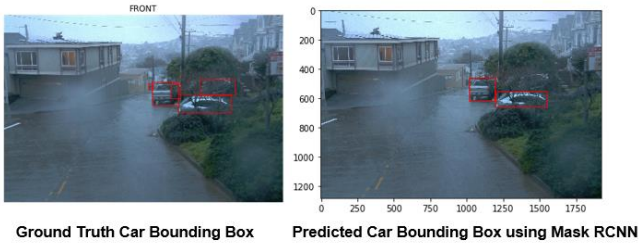

Fig 8. Ground Truth and prediction using Mask RCNN

VI. Model Evaluation

Table 1. Model Evaluation

Model	Evaluation Metric	Result
Mask RCNN	mAP + IOU	0.2187
Mask RCNN + Soft NSM	mAP + IOU	0.2187

Linear SVM + HOG + NSM	mAP + IOU	0.078
RetinaNet	mAP + IOU	0.061

Models were evaluated by comparing them to the ground truth labels and using a IOU threshold of 0.5. Then the mean average precision value was calculated and is displayed in the following table.

We experimented with different IOU thresholds to check Average precision with respect to Intersection over union values. Figure 9 shows that the precision decreases for high thresholds.This is expected since the images contain a large number of objects and types given the data includes various objects in driving settings.

In order to get a sense of the model performance, we also implemented a user-interface that facilitates visualising detected objects over different frames. The user-interface was implemented using the Streamlit framework, which provides easy to use and accessible APIs for building data science visualisation applications. Figure 10 and 11 show snippets from the UI.

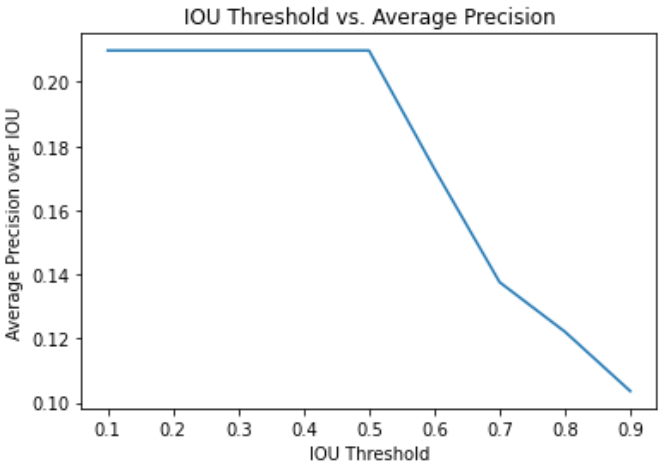

Fig 9. IOU threshold vs Average Precision

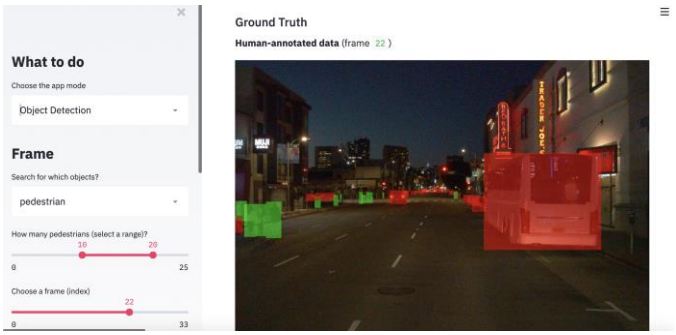

Fig 10. Ground Truth images for Object Detection in UI

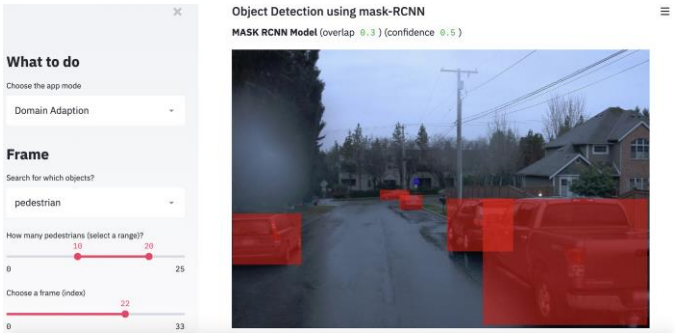

Fig 11. Object Detection with Domain Adaptation

VII. Implementation

The implementation and demo code has been pushed to github at :
https://github.com/Saurabh7/waymo-2d-object-detection

VIII. Dataset

Classical Machine Learning Model, where we use some part of CADC data set to train the model and the other part of the data set to detect objects in the field of view.

After we check the performance using CADC data set as the ground truth, we will incorporate the Normal Weather and Canadian dataset and train 2 data sets. This will be under object detection for the transfer Learning. We will use Waymo Dataset for normal Images and CADC Dataset for winter Images.

The Waymo dataset (https://waymo.com/open/data/) consists of images captured from different camera angles.

- It is a labelled dataset consisting of labels which are bounding boxes for each object detected in the image.
- The dataset is ~1TB in size, we will be working on ~10 GB subset of the data.

- Waymo has suggested using Mean Average Precision (MAP) to evaluate the performance of the algorithm

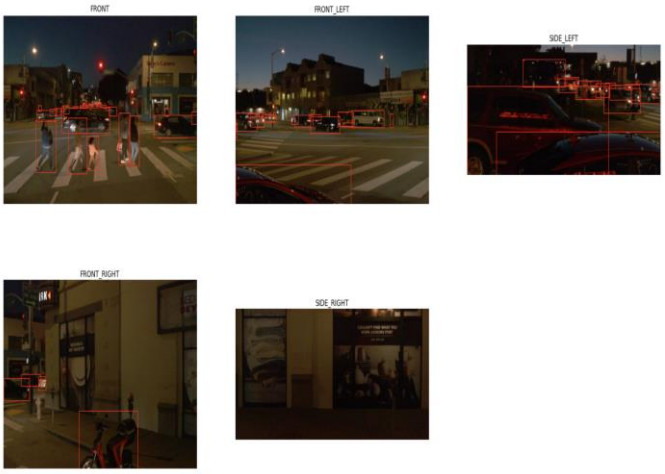

Fig 12. Waymo dataset ground truth images with different camera angles

Sample of different camera angles and ground truth of objects from the Wamo Open Dataset as in Figure 12.

The CADC dataset consists of the following parts:

- 56,000 camera images (images from 8 cameras).
- 75 scenes of 50-100 frames each.
- It also includes 10 annotation classes of cars, pedestrians, trucks, buses, garbage containers on wheels, traffic guidance objects, bicycles, pedestrians with objects, horses and buggies and animals.
- Adverse weather driving conditions, including snow.

Sample of different images from the CADC dataset with adverse weather conditions:

Fig 13. Cloud Cover Weather Condition

Fig 14. Snow Weather Condition

IX. Conclusion

We performed object detection on autonomous driving datasets. Multiple models like Mask RCNN, Mask-RCNN with Soft NSM, Linear SVM, HOG and NSM, and Retina Net were used. Mask RCNN performed the best with Mean Average Precision of 0.2187 and similarly when Soft NSM was added.

Mask-RCNN with Soft NSM gave the same performance as NSM chooses the best bounding box score and suppresses the bounding boxes with significant overlap.[8]

Linear classification models like SVM along with HOG performed comparatively badly. This is expected since object detection is a highly non-linear task.

This demonstrates the efficiency of Deep Learning Architectures using convolutional layers like Fast-RCNN and Mask RCC, which combines both Fast-RCNN and FCN, in capturing spatial relationships in an image.

Retinanet architecture uses a backbone network and 2 subnetworks, one for classification of backbone output and regression of bounding boxes. It was fine-tuned for 1 epoch. This did not perform well in comparison to other architectures, possibly due to the fact that it has more parameters and sub-networks and additional fine tuning is needed [9].
Transfer Learning enables using pretrained models on a different, larger dataset consisting of generic images and objects (COCO Dataset) to perform well on a specific task of object detection for autonomous driving.

In the future, we will use what we learnt from camera object detection for autonomous vehicles, we incorporated domain adaptation into the model using DA-fast RCNN [10].
Domain Adaptation trains on a general object detection data set, like COCO dataset, fine tunes on a small number of adverse weather images and tests on different adverse weather images.

We will adapt mask R-CNN to a new domain (harsh weather conditions), which is different from the training domain that uses normal weather conditions and urban driving settings. Domain Adaptive Fast R-CNN will adapt to different image scale, style and illumination on the image level as well as adapt to different object size and appearance on the instance level.

We will also factor in driver's behavior using the DBNet [11], a large scale driving behavior data set, to take into account additional confounding variables such as lane change, vehicle speed and steering angle.

ACKNOWLEDGEMENTS

We would like to acknowledge the feedback and guidance we received from Juxuan Huang and Dr. SREYASEE DAS BHATTACHARJEE. We would also like to highlight everyone's contribution including Saurabh Amarnath Mahindre(Mask R-CNN, Data Extraction), Dana Moukheiber(NSM+SVM+HOG) and Sonal Vijay Kelwadkar(Keras RetinaNet) for working on data extraction, testing, validating and training on the respective models outlined above. We made equal contributions for literature review, evaluation, user interface and compiling the results.

REFERENCES

[1] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon , Amy Gao , Aditya Joshi , Sheng Zhao , Shuyang Cheng , Yu Zhang∗ , Jon Shlens, Zhifeng Chen, and Dragomir Anguelov, "Scalability in Perception for Autonomous Driving: Waymo Open Dataset "

[2] Matthew Pitropov, Danson Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki1 and Steven Waslander, "Canadian Adverse Driving Conditions Dataset"

[3] Ross Girshick, Microsoft Research, "Fast R-CNN "

[4] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, Facebook AI Research (FAIR) "Mask R-CNN"

[5] Hilton Bristow, Simon Lucey, Queensland University of Technology, Australia, Carnegie Mellon University, USA, "Why do linear SVMs trained on HOG features perform so well?"

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar, Facebook AI Research (FAIR), "Focal Loss for Dense Object Detection"

[7] Xiao Lu, Chen Luo, Michelle Zhang, "Mask RCNN Application: Instance Segmentation in Driving Scenes

[8] Bodla, N., Singh, B., Chellappa, R. and Davis, L., 2020. *Soft-NMS -- Improving Object Detection With One Line Of Code*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1704.04503> [Accessed 6 May 2020].

[9] Hoang, Toan Minh, et al. "Deep retinanet-based detection and classification of road markings by visible light camera sensors." *Sensors* 19.2 (2019): 281.

[10] Chen, Yuhua, et al. "Domain adaptive faster r-cnn for object detection in the wild." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[11] Dang, Yonghao, et al. "DBNet: A New Generalized Structure Efficient for Classification." *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019.