Towards an Optimised Vehicle Detection Algorithm for Multi-Object Tracking in Traffic Surveillance

Soumi Mitra[3], Nandhini Reddy Aileni[3], Irina Tal[1,3], and Malika Bendechache[1,2,3]

[1] Lero – The Irish Software Research Centre
[2] ADAPT – Science Foundation Ireland Research Centre
[3] School of Computing, Dublin City University, Ireland
{soumi.mitra2, nandhini.aileni2}@mail.dcu.ie
{irina.tal, malika.bendechache}@dcu.ie

Abstract. Smart cities implementation has been increasing in recent years, using computer vision techniques which help to reduce the problems of traffic congestion and also monitor traffic. Computer vision-based detection and tracking methods using convolutional neural networks are being successful in implementing smart cities, using cameras on the streets and at a low cost for the internet of things. Convolutional neural networks are deep learning algorithms used to analyse visual imagery. However, the same vehicle is detected multiple times in a particular frame at a particular timestamp, which thereby increases the time complexity of the algorithm by detecting the same vehicle multiple times. This paper proposes a new optimised algorithm that solves the vehicle duplication problem. Our proposed approach is based on extending and optimising the state of the art Point-RCNN (Region-based Convolutional Neural Network) algorithm by combining it with the D-Hash (Difference hash) algorithm. The D-Hash is a robust image hashing algorithm used for the identification of duplicate images. These images are then processed onto a 3D multiple object tracking system called Point-RCNN which is used for the bounding and identification of vehicles. The proposed algorithm was tested on the KITTI 3D object detection benchmark data set. Our experiments show that the vehicle duplication issue is eliminated without sacrificing the accuracy of data object detection results. In addition, our proposed approach decreases the time complexity by passing 70 more frames per second (FPS) when compared to the Point-RCNN baseline. The execution time (speed) of the proposed algorithm is also improved by almost 34% compared to the baseline.

Keywords: Vehicle Detection, Point-RCNN, Vehicle Duplication, D-Hash.

1 Introduction

Transportation is one of the most significant domains where actionable insights drawn from data gathered by camera sensors can be beneficial [10]. In the last few

years, various analytical approaches and video data have been processed together with the increased computing power, which has enabled new applications as well. Traffic analysis can be a very important aspect of smart city implementation as it helps in reducing various traffic related issues like traffic congestion, accidents, etc.

Deep learning is a technique where insights from data can be driven by understanding the patterns of the data, thereby classifying it [20][1]. A technique that helps computers to understand digital images or videos is defined as computer vision [3]. Deep learning models, in particular, convolutional neural networks, help with object detection. Object Detection is defined as a combination of image classification and object localization. Image classification helps in the classification of images by assigning a class label. Object localization helps by bounding a box around the objects in that image. Object detection helps in bounding boxes around the objects of interest in an image [2]. Unique Id's are generated from object detection, which furthermore helps in tracking objects moving around frames in a video [13]. Deep learning models and computer vision help in monitoring the traffic rate.

3D object detection and Multiple Object Tracking are considered essential applications for autonomous driving [18]. 3D object detection helps in capturing an object's size, position, and orientation in the world. 3D object detection is more accurate when compared to 2D object detection, especially in real-time systems. The LiDAR (Light Detection and Ranging) point cloud helps in obtaining 3D detection [13]. 3D object detection automatically produces semantic masks for 3D object segmentation, whereas 2D provides weak semantic segmentation [12]. The visual system that tracks multiple moving objects in a dynamic environment is defined as Multiple Object Tracking.

However, the existing approaches and algorithms suffer from vehicle duplication issues where the same vehicle is being identified multiple times in a particular timestamp.

In this paper, we have presented a new algorithm by merging the baseline Point RCNN and D-Hash algorithm which helps to resolve the issue of vehicle duplication without sacrificing the accuracy of the baseline.

The paper is aligned as following. Section 2 introduces related works about object detection and tracking using deep learning techniques. Section 3 gives a brief description of the data set used, pre-processing techniques and the algorithms used for our proposed approach. Section 4 introduces our proposed approach. Section 5 presents the results and their evaluation metrics. We make concluding remarks in Section 6.

2 Literature Review

This section summarises machine learning, particularly deep learning algorithms such as regions convolutional neural networks (RCNN) were extensively used for traffic analysis using video and imaging data. This section also summarises the KITTI dataset benchmark.

2.1 Deep Learning Algorithms Used

In [11], a new method was proposed for Vehicle detection with sub-class training, where data was trained on convolutional neural networks based on faster RCNN architecture. Vehicle detection is performed on Sub-Classes categories learning using RCNN in order to improve the performance of vehicle detection by identifying different categories of vehicles in different orientations as well as different climatic conditions. The transfer learning approach is also evaluated for fine-tuning a pre-trained RCNN model. Coco dataset is used for training the neural networks and features are extracted from them. The extracted features are tested on the UA-DETRAC dataset. Average precession is the evaluation metric used for vehicle detection. Where the proposed method with various variations was evaluated on the validation set comparing to the four baseline methods which are Deformable Part Model (DPM), Aggregate channel features (ACF), RCNN, and CompACT. The accuracy of vehicle detection using Faster RCNN was 93.43%. However, the issue of multiple bounding boxes overlaps from different subclasses on the same object.

In 2019, another paper was published in [8] where the authors proposed a low confidence track filtering extension on the Deep SORT tracking algorithm, which can significantly reduce false-positive tracks generated by Deep SORT. Tracks with low average detection confidence in their initial several frames will be deleted. In this way, the detection confidence can be set to a lower value and even zero to avoid missing detections. They also generated a vehicle re-identification dataset from the UA-DETRAC dataset to train the Deep SORT for vehicle data association. Experiments on the UA-DETRAC test dataset show that the proposed extension can achieve promising results by notable margins against state-of-the-art trackers. The evaluation metric used in this paper was PR-MOTA and average detection confidence threshold. But, this paper does not address the issue of vehicle duplication.

In 2019, [9] proposed a new method, a multi-scale detector for accurate vehicle detection using UA-DETRAC data set. Where additional prediction layers are integrated into conventional Yolo-v3 using spatial pyramid pooling. Initially, for vehicle detection vehicles, MS-COCO pre-trained Yolo-v3 model is used to initialize the Darknet-53 network which is trained in an end-to-end manner with Stochastic Gradient Descent (SGD). This later generates feature maps, which acts as an input for Feature Pyramid Network (FPN). 2 more prediction layers and 5 more SPP networks with batch normalization are used to accounting various object scales. Feature maps from layers at different stages have different dimensions, sampling operation is performed to combine them effectively. The pooling layer is used to progressively reduce the dimension of feature representation from the convolution layer, inserted in-between successive convolution layers. Overall mean Average precession (mAP) was calculated for DPM (Deformable Part-based Models), ACF (Aggregate Channel Features), RCNN, Faster RCNN2, SA-FRCNN, NANO, CompACT (Complexity Aware Cascade Training), EB (Evolving Boxes), R-FCN (Region-Based Fully CNN), GP-FRCNN (Geometric Proposals for Faster RCNN), HAVD, SSD-VDIG and

conventional Yolo-v3 using spatial pyramid pooling with 2 additional prediction layers which is the proposed method. The proposed method outperformed the existing algorithms with an accuracy of 85.29%. However, vehicle detection speed was low when compared with RCNN.

In 2020, another paper was published [14] where they proposed the motion priors embedded parallel architecture for surveillance vehicle detection. The key is to properly leverage motions by decoupling moving objects from overall vehicles, in order to enhance vehicle appearance while carefully suppressing false positives in the background. Following the protocol of UA-DETRAC, they had submitted the results of their detector with the input size of 512×512 to the public testing server for evaluation. The evaluation metrics used here were true positives and false positives. They achieved an overall accuracy of 80.76% AP while maintaining the fastest speed of 14 FPS among these detectors. In terms of the performance under different weather conditions, their approach obtained competitive results on the cloudy and sunny subset and outperforms the other methods on the night and rainy subset. They attributed the stable performance under various conditions especially the bad weather to the proper use of motion priors. Detectors only use geometric features are susceptible to unexpected environments when detecting vehicles in real traffic, thus motions are very critical to generate robust predictions in surveillance vehicle detection.

Traffic congestion and occlusion are some of the major challenges in vehicle detection. A new methodology was developed in 2020 by [15] for vehicle detection under complicated conditions. A combination of the MOG2 (Mixture of Gaussians) Algorithm and H-Squeeze Net Algorithm were used to accurately identify vehicles and their respective categories. MOG2 acts as a background subtraction model which generates ROI's (Region of Interest) from a set of video frames. These generated ROI's helped in avoiding the bounding box problem. Whereas the H-Squeeze Net algorithm is used for identifying various vehicle categories and the complete classification of vehicles is determined by using Softmax classifier. Evaluation is performed on traffic data from a traffic intersection in Suzhou, China, CDNet 2014 and UA-DETRAC data sets. The proposed model is compared with the H-SqueezeNet model, with original SqueezeNet and other state-of-the-art networks such as VGG16, VGG19, Inception-v3, ResNet and Darknet-53. Out of all the metrics calculated accuracy and model size has helped in determining that this proposed model has outperformed the state-of-the-art models. In addition, the problem of false positives is reduced by using MOG2 and H-Squeeze Algorithms. However, different types of vehicle categories except for cars, trucks and busses are not identified and vehicle detection speed is comparatively low.

A new methodology was proposed in [12], where 3D objects were detected using Point-RCNN from the raw point of the cloud. 3D objects are detected and bounded in a combination of two stages. Where in the first stage global semantic features are generated by a bottom-up approach is used where high-quality 3D proposals are generated from the point cloud and these help in separating the foreground and background points. Whereas in the second stage local spatial

features are generated and by refining the proposal of canonical coordinates. Later global semantic features of stage 1 and local spatial features of stage two are combined, thereby accurately bounding 3D objects. The evaluation was performed on the KITTI dataset and obtained a 96.01% recall value.

A 3D multiple object tracking was proposed in [18], where objects were detected using LiDAR point cloud on a KITTI dataset. 3D Kalman Filter in combination with Hungarian algorithm is used for data association and state estimation [17]. The state-space of the Kalman filter is defined in the image plane thereby extending the state of objects to 3D including location, size, velocity, and orientation. The proposed algorithm was evaluated on 2D and 3D MOT systems, where MOTA, MOTP, IDS, and FPS were used as evaluation metrics. The algorithm outperformed the 3D MOT system with 207.4 FPS (frames per second), by achieving the highest speed.

In this paper, some image hashing algorithms are introduced and compared [4], which helps in detecting similar images from a large social network dataset. A-Hash, P-Hash, D-Hash, and W-Hash algorithms were considered for evaluating the robustness of large data set using precision, recall, and F1 score as their evaluation metrics. Experiments were conducted to infer that the P-hash algorithm outperformed the remaining three algorithms with an F1 score of 0.864 at precision and recall values of 0.926 and 0.81 respectively, for a distance threshold N = 16. Followed by the D-hash algorithm, with an F1 score of 0.846 at precision and recall values of 0.952 and 0.761 respectively, for a distance threshold N = 14. W-hash and A-hash algorithms obtained lower f1 score values.

2.2 KITTI Dataset

A KITTI dataset benchmark was developed [6] for stereo, optical flow, visual odometry, 3D object detection, and 3D tracking. 194 training and 195 testing images are used with a resolution of 1240 x 376 pixels for stereo and optical flow estimation benchmark, which includes difficulties such as gunman version and reflecting surface. Evaluation provides results for all non-occluded as well as all ground truth pixels. 3D odometry dataset consists of 22 stereo videos with a total length of 40 kilometres, this video provides GPS ground truth trajectory. The proposed evaluation metrics minimize bias by computing errors over all possible sequences for a given trajectory length or driving speed our online evaluation server, evaluate submit results as a function of these two variables capturing different sources of error. 3D Object detection, object orientation, and tracking benchmarks provide accurate 3D information in the form of 3D bounding boxes for object classes such as cars, vans, pedestrians, and cyclists. 3D object ground truth values are generated by annotating 3D bounding box trackers to all objects visible in the image.

The goal of our proposed approach is to improve the baseline algorithms by solving the vehicle duplication problem without sacrificing the accuracy of data object detection.

3 Methodology

3.1 Dataset Description

KITTI data set has been obtained from a moving platform in Karlsruhe, Germany [7]. This dataset is created by capturing the visuals on highways and rural areas using high-resolution stereo cameras with both greyscale and colour systems. Velodyne HDL-64E, laser scanner which produces more than one million 3D points per second, and OXTS RT 3003 localization system which combines GPS, GLONASS, an IMU, and RTK correction signals. The cameras, laser scanner, and localization system are calibrated and synchronized, thereby providing accurate preliminary values. Figure 1 shows one frame of the KITTI 3D Vision Benchmark Suite dataset.

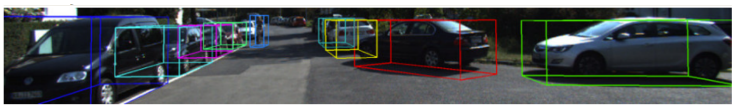

Fig. 1. KITTI 3D Vision Benchmark Suite

In our approach, a part of the KITTI dataset, a 3D object detection benchmark is adapted. Where a total of 7481 training images and 7518 testing images with 80.256 labeled objects have been gathered from corresponding point clouds.

Object detection and 3D orientation estimation are the important characteristics of the KITTI 3D object benchmark [6]. Accurate 3D bounding boxes are bounded for different object classes such as cars, cyclists, pedestrians, and these bounding boxes are obtained by manual labels of objects in 3D point clouds which are produced by the Velodyne system. Benchmark data is chosen by employing a greedy approach which uses 100 non-occluded objects per class along with 16 entropy object-orientation classes. Average Orientation Similarity (AOS) is an evaluation metric used where it is defined as True Positives over True Positives and False Negatives.

$$AOS = \frac{TP}{TP + FN}$$

True positives are to be overlapped by more than 50% and the multiple detections of the same object are considered as false negatives.

3.2 Data pre-processing

In our proposed algorithm, we have merged the D-Hash algorithm with Point-RCNN to remove the vehicle duplication issue and also for better detection and tracking. To implement the D-Hash algorithm, firstly, we had to convert the entire dataset images to grayscale format and also resized and flatten the images to calculate hamming distance. This step increases the efficiency of the

algorithm, as the resized images reduce the time complexity and also helps to calculate the hamming distance which is an important factor in our proposed algorithm. Figure 2 shows the data preprocessing steps.

Fig. 2. Data Pre-processing Steps

3.3 Algorithms Used

In this paper, we propose a new algorithm that uses point cloud technology along with the perceptual image hashing technique. We have merged the hashing or D-Hash algorithm with Point-RCNN. The following sections discuss the two algorithms used.

Point-RCNN (The Baseline): Point-CNN is a 3-dimensional framework used for object detection from raw point clouds. A point cloud is a bunch of data points that generate a 3-dimensional form or shape. Each point in the point cloud has its own 3-dimensional X, Y, and Z coordinates. Methods like remote sensing or photogrammetry are used to generate point clouds. In photogrammetry, a bunch of photos is taken in many dimensions to generate point clouds. And in remote sensing, satellites or aircraft are used to collect pictures or data from the globe. LiDAR (Light Detection and Ranging) sensors are also used in this process of collecting data from the earth's surface and are later used to generate the point cloud [5].

We have considered this 3D object detection approach as it generates a more accurate output than the conventional 2D methods. Point-RCNN uses a 2-stage method to generate the detection. The first stage uses a bottom-up approach to generate 3D proposals and the second stage refines the proposals into canonical coordinates, which are the X, Y, Z coordinates from the point cloud, to achieve the final detection, which is more accurate than the detections found in 2D methods. Also, this process achieves a higher speed than conventional methods. This technique avoids the use of a large number of 3D anchor boxes throughout the 3D environment, saving time and effort. The KITTI 3D object

detection benchmark is used to evaluate this algorithm and it outperforms all the conventional methods in terms of time complexity and effort. Figure 3 shows the architectural diagram of Point-RCNN.

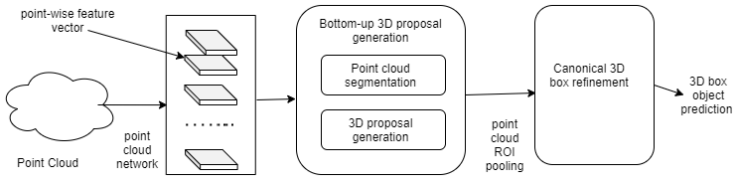

Fig. 3. Point-RCNN architectural diagram

D-Hash: There are so many perceptual hashing algorithms for image hashing, but we have considered D-Hash as it provides the best accuracy and speed [4]. In the D-Hash procedure, before hashing, the images are converted into grayscale and then reduced in size to compute the difference between two pictures. It basically focuses on the picture structure. This principle uses Hamming Distance as a comparison parameter to detect duplicate images in any dataset. Duplicate images detected from D-Hash algorithm are removed from our dataset by applying the remove function os.remove().

It is a very simple method to implement. Firstly, all the images are converted into grayscale and reduced to a block size of 9x8, which results in a total of 72 pixels. Then, the difference between each adjacent pair of pixels is calculated in a row, for a total of 8 differences in a row. The output of 8 rows with 8 differences produces a result of 64 bits. Then bits are assigned to them. Each bit is set in such a way that the left or right pixel will be the brighter one. Two images are considered to be the same if the hamming distance between them is less than 5.

To compare two binary data strings of equal length, Hamming distance is used. The XOR operation is used here to calculate the distance. The Hamming distance is mostly used in computer networking and coding theory as an error detection and correction metric [19].

The KITTI 3D object detection benchmark is used to evaluate D-Hash and it helps in eliminating vehicle duplication issues in vehicle detection and tracking.

4 Our Proposed Approach

Figure 4 shows an overview of the proposed approach. Our proposed algorithm merges the D-Hash algorithm with Point-RCNN to eliminate the vehicle duplication issue and also for better detection and tracking. We have merged a 3D Kalman filter with the Hungarian technique from the baseline study [4], which uses a 3D object detector to extract 3D detections from the LiDAR point cloud. Kalman filters are used mostly in dynamic systems where the data is uncertain. It can sometimes determine what may happen next in a real-time system. It is

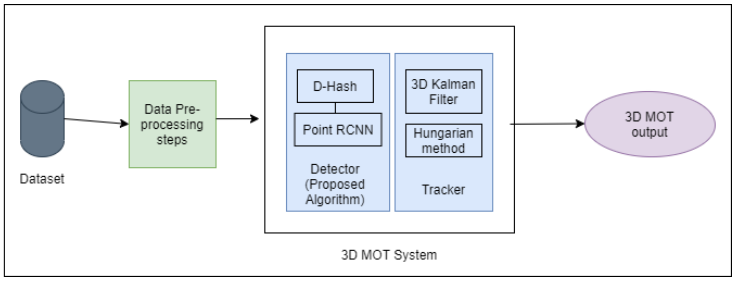

Fig. 4. Architectural Diagram of proposed 3D MOT system

widely used in systems where data is continuously changing over time. That's why it can be used very efficiently in multi-object tracking systems. As Kalman filters don't store historical data, they are memory efficient and also very fast to execute, which makes them very suitable for real-time MOT systems [16].

The Hungarian technique is a combinational optimized algorithm which is used to solve problems in polynomial time. This technique works on an iterative basis and in a very optimized way. Hence, it is very useful in real-time applications like 3D MOT (multiple object tracking) systems.

For car and cyclist divides, we employed Point-RCNN detections on the KITTI 3D object detection dataset [4]. In addition, to reduce time complexity, we combined the D-Hash technique to delete duplicate photos from track sequences. Thereby removing the duplicate images in the track sequence. Unlike other filter-based MOT systems, which define the filter's state space on the image plane, the state space of the objects is expanded to three dimensions, including three dimensions of location, size, velocity, and orientation [4]. Figure 5 shows how 3D objects are getting detected using our approach.

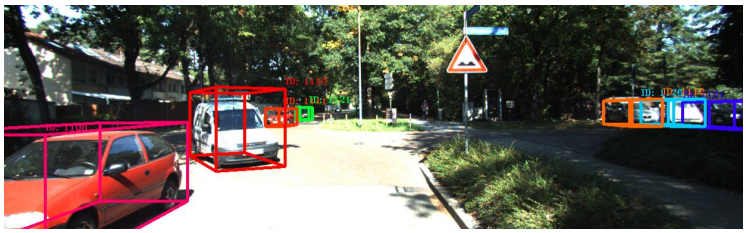

Fig. 5. 3D Object Detection using Proposed Algorithm

We have merged the Point-RCNN and D-Hash algorithms and applied them to the KITTI 3D object detection dataset to evaluate them. The data is passed through the 3D MOT system along with the D-Hash layer so that the duplicate images are removed and unique objects are bound in 3D bounding boxes. This technique achieves the same accuracy as the baseline Point-RCNN and also works

at faster FPS (Frames per second) than the baseline one. The KITTI 3D object detection dataset has three object types: cars, pedestrians and cyclists. For our approach, we have trained and tested cars and cyclists.

5 Evaluation & Results

The KITTI 3D object detection dataset, which includes LiDAR point clouds and ground truth 3D bounding box trajectories, is used to evaluate our proposed approach. We have used the KITTI validation set for 3D MOT assessment because the KITTI test set only allows 2D MOT evaluation and its ground truth is not available to users [4]. We compared the car and cyclist subsets of the KITTI dataset, based on previous research. We have used the same evaluation metrics used in the baseline Point-RCNN to evaluate our proposed algorithm. We have used MOTA (Multi Object Tracking Accuracy), MOTP (Multi Object Tracking Precision), MODA (Multi Object Detection Accuracy), MODP (Multi Object Detection Precision), sMOTA (scaled MOTA), AMOTA (average MOTA), AMOTP (average MOTP) along with other CLEAR metrics like Precision, Recall, F1 to evaluate our algorithm.

We have run our tracker on the KITTI 3D object detection validation set with our proposed algorithm detection and achieved sMOTA 93.28%, AMOTA 45.43%, AMOTP 77.41% for the car object and sMOTA 72.94%, AMOTA 37.95%, AMOTP 63.03% exactly the same as per the baseline.

We have summarised the results in Table 1 which shows that the proposed algorithm generates the same metrics as per the baseline.

Table 1. Comparing Point-RCNN and Proposed Algorithm

	Car		Cyclist	
Evaluation metrics	Point-RCNN	Proposed Algorithm	Point-RCNN	Proposed Algorithm
MOTA	86.24%	86.24%	79.82%	79.82%
MOTP	78.43%	78.43%	76.55%	76.55%
MODA	86.24%	86.24%	79.82%	79.82%
MODP	83.11%	83.11%	95.70%	95.70%
sMOTA	93.28%	93.28%	72.94%	72.94%
AMOTA	45.43%	45.43%	37.95%	37.95%
AMOTP	77.41%	77.41%	63.03%	63.03%
Recall	92.17%	92.17%	84.49%	84.49%
Precision	96.22%	96.22%	95.55%	95.55%
F1	94.15%	94.15%	89.68%	89.68%

Our proposed algorithm speed is 277.7 FPS (i.e., number of frames passed per second in a sequence. In other words, it is the number of distinct images captured in a second.) while the speed of the baseline one is 207.4 FPS which

is almost 34% more than the baseline one. Figure 6 shows the FPS comparison between Point-RCNN and proposed algorithm.

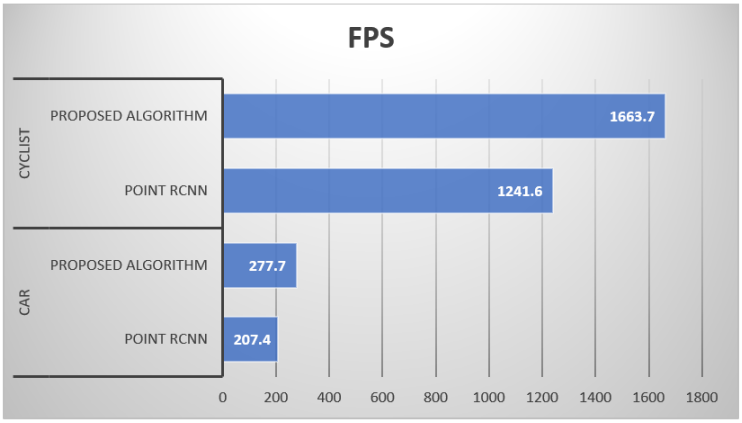

Fig. 6. Comparing FPS of Point-RCNN and Proposed Algorithm

6 Conclusion & Future Work

The existing baseline algorithm suffered from a vehicle duplication issue where the same vehicle was identified multiple times in a particular timestamp. This issue thereby increased the time complexity of the baseline algorithms. Using our algorithm, vehicle duplication issue in multi object detection and tracking in traffic surveillance is eliminated. The baseline Point-RCNN algorithm merged with the D-Hash algorithm which also gives a multi object detection and tracking solution with similar accuracy as the baseline and reduces time complexity by increasing the speed to almost 34%.

To evaluate our algorithm, we only used the KITTI 3D object detection benchmark suite. In the future, we hope to evaluate our algorithm using other similar datasets, such as the UA-DETRAC benchmark suite. We will also attempt to improve the algorithm's performance in light of the various weather conditions (like sunny, rainy, etc).

Acknowledgement

This work was supported in part by the Science Foundation Ireland grants 13/RC/2094_P2 (Lero) and 13/RC/2106_P2 (Adapt).

References

[1] Sweta Bhattacharya et al. *A Review on Deep Learning for Future Smart Cities*. May 2020. DOI: 10.1002/itl2.187.

[2] Jason Brownlee. *A Gentle Introduction to Object Recognition With Deep Learning*. en-US. https://machinelearningmastery.com/object-recognition-with-deep-learning/. May 2019. (Visited on 08/05/2021).
[3] *Deep Learning for Computer Vision*. en-US. https://machinelearningmastery.com/deep-learning-for-computer-vision/.
[4] Andrea Drmic et al. *Evaluating robustness of perceptual image hashing algorithms*. 2017. DOI: 10.23919/MIPRO.2017.7973569.
[5] *FME Community*. https://community.safe.com/s/article/what-is-a-point-cloud-what-is-lidar.
[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. *Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite*. 2012.
[7] Andreas Geiger et al. *Vision meets robotics: the KITTI dataset*. Sept. 2013. DOI: 10.1177/0278364913491297.
[8] Xinyu Hou, Yi Wang, and Lap-Pui Chau. *Vehicle Tracking Using Deep SORT with Low Confidence Track Filtering*. 2019.
[9] Kwang-Ju Kim et al. *Multi-Scale Detector for Accurate Vehicle Detection in Traffic Surveillance Data*. 2019. DOI: 10.1109/ACCESS.2019.2922479.
[10] Posted on November 4 and 2020. *How Computer Vision is shaping smart cities*. en. https://www.phase1vision.com/blog/how-computer-vision-is-shaping-smart-cities. (Visited on 08/05/2021).
[11] Sitapa Rujikietgumjorn and Nattachai Watcharapinchai. *Vehicle detection with sub-class training using R-CNN for the UA-DETRAC benchmark*. Aug. 2017. DOI: 10.1109/AVSS.2017.8078520.
[12] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. *PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud*. 2019. arXiv: 1812.04244 [cs.CV].
[13] Shaoshuai Shi et al. *From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network*. 2020. arXiv: 1907.03670 [cs.CV].
[14] Xiaolian Wang et al. *Illuminating Vehicles With Motion Priors For Surveillance Vehicle Detection*. 2020. DOI: 10.1109/ICIP40778.2020.9190727.
[15] Zhiyuan Wang et al. *A Robust Vehicle Detection Scheme for Intelligent Traffic Surveillance Systems in Smart Cities*. 2020. DOI: 10.1109/ACCESS.2020.3012995.
[16] Greg Welch, Gary Bishop, et al. *An introduction to the Kalman filter*. 1995.
[17] Xinshuo Weng et al. *3D Multi-Object Tracking: A Baseline and New Evaluation Metrics*. 2020. arXiv: 1907.03961 [cs.CV].
[18] Xinshuo Weng et al. *AB3DMOT: A Baseline for 3D Multi-Object Tracking and New Evaluation Metrics*. 2020. arXiv: 2008.08063 [cs.CV].
[19] *What is Hamming Distance*. https://www.tutorialspoint.com/what-is-hamming-distance.
[20] *What is Object Tracking - An Introduction*. en-US. https://viso.ai/deep-learning/object-tracking/. July 2021. (Visited on 08/05/2021).