Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 45 (2020) 971–978

Transportation Research Procedia

www.elsevier.com/locate/procedia

AIIT 2nd International Congress on Transport Infrastructure and Systems in a changing world (TIS ROMA 2019), 23rd-24th September 2019, Rome, Italy

Comparison of Vehicle Detection Techniques applied to IP Camera Video Feeds for use in Intelligent Transport Systems.

Mark Bugeja[a,b,*], Alexiei Dingli[b], Maria Attard[a], Dylan Seychell[b]

[a]*Institute of Climate Change and Sustainable Development, University of Malta, Msida, Malta*
[b]*Department of Artificial Intelligence, Faculty of ICT, University of Malta, Msida, Malta*

Abstract

Vehicle detection is an important area in Transport and Artificial Intelligence. Through vehicle detection techniques, vehicles can be located across different images. Some of these models are robust enough to identify parts of vehicles in images where the vehicle might be partially occluded. Recent advances in detection methods gave rise to a range of different techniques that can be used for recognition and detection of vehicles. Although each technique has its merits, it is not always the case that the adopted model works well for scenarios involving IP Cameras. The motivation for this study is to compare several state-of-the-art techniques, including deep learning models and computer vision approaches. A set of experiments are developed in order to test these models on a number of low quality IP camera footages set in the transport domain in order to measure detection and recognition accuracy. The final evaluation compares detection accuracy using mean average precision, the semantics of the recognised vehicle as well as recognition robustness when applied to a dataset that contains images with different light conditions. The study also looks at persistence in recognition across frames in video data and a detailed description of the dataset used to train the evaluated models. Finally, the paper also goes through some scenarios that applies the results obtained in this study to ITS systems that use IP camera feeds.

Keywords: Intelligent Transport Systems, Computer Vision, Artificial Intelligence, Vehicle Detection;

1. Introduction

One of the uses of Computer Vision (CV) and Artificial Intelligence (AI) is in conjunction with the areas of traffic management and Intelligent Transport Systems (ITS). There is already a large selection of different algorithms and

* Mark Bugeja.
E-mail address: mark.bugeja@um.edu.mt

2352-1465
10.1016/j.trpro.2020.02.069

intelligent models that are used to count cars, predict traffic flow, predict vehicle speed as well as identify dangerous drivers Engel et al. (2017). Notwithstanding the challenges faced when developing these transport systems, there are already several models that are considered state of the art. These models use a range of features to output results, including, vehicle size, colour and GPS location. These features and the process of selecting the best approach provides an opportunity to study the effect of applying object detection models to different datasets and scenarios. Thus, the primary contribution of this paper is not to develop a new pipeline for vehicle detection, but, to develop a methodology that allows transport experts to evaluate and find the best models for their particular Intelligent Transport System needs. This paper also focuses on feeds obtained from live data mined over the internet. Apart from the standard dataset used for the evaluation of this study, we introduce a new dataset in section 3.2 made up of live footage captured through IP cameras. This dataset offers a new set of challenges in itself as the video feed is of low quality. When developing ITS, the approaches described in McQueen and McQueen (1999) initially create isolated cases where the individual traffic control models work. For example, finding the best camera angle to capture incoming traffic and increase detection accuracy. Additionally, vehicle detection is improved by adding light fixtures and other video enhancing tools to optimise visibility. This process can be time-consuming and not always feasible. Thus, one of the main motivations of this study was to evaluate the different state of the art object detection models on datasets that contain low-quality footage mined over the internet. This process is done in order to introduce a more effective way of developing ITS systems that use vehicle detection algorithms and models that do not need custom implementations or a lot of resources to deploy. This paper evaluates vision models that are used for vehicle detection. As previously mentioned, these algorithms are often used in a different context when building ITS systems. This paper is split into four different sections; the background section identifies the current state of the art approaches to vehicle detection and delves into the configuration of these models. This section is then followed with the Methodology that expands on the evaluation metric used, a detailed explanation of the datasets and setup. The third section goes into detection accuracy results for both datasets, followed by a discussion that expands on possible usage for the results obtained and which models might better fit different ITS applications. The concluding section of the paper summarises the results obtained and future work in the area.

2. Background

Deep learning has become a state of the art approach to solving machine learning problems, given how effective these models are when successfully applied to various fields of study, such as speech recognition and computer vision. Machine learning is a field of study comprised of a collection of statistical models that are trained on datasets made up of different features in order to, classify or predict information Bishop (2006). For example, a model trained upon weather information would be able to predict the weather for a given date Cramer et al. (2017). This process is straightforward but requires fine-tuning in terms of data as well as network (model) configuration. Furthermore, with the advent of big data, companies such as Google, Microsoft and Apple that have access to large volumes of data have consistently pushed the field of Deep Learning and have applied it to well-known systems such as Siri for Apple and Google Translate for Google. Deep Learning is not a new concept LeCun et al. (2015) it has gained popularity in recent years due to advancement in technology. Processing units such as GPU have in the past years become cheaper, coupled with the significant increase in processing power of these chips has paved the way for more research in Deep Learning technologies Coates et al. (2013). Deep Learning Models are a set of techniques based on neural networks that exploit current technology to create network configurations built using a large number of layers as well as trained upon big data. This network design choice leads to the deep learning model's ability to exploit this architecture for use in unsupervised learning and pattern classification. Thus, significantly increasing the accuracy output of the models compared to their more shallow implementations Schmidhuber (2015). In the next section, a number of Convolutional Neural Networks (CNN) architectures are explored. The networks include, the You Only Look Once (YOLO), RetinaNet single stage detector as well as the Region Based Convolutional Neural Networks (RCNN).

2.1. Convolutional Neural Networks for Object Detection

CNN is a type of feedforward neural network model. Variations of this network have been successfully applied to several different computer vision problems. CNN's use the concept of convolution in order to create an architecture using several layers of convolution and nonlinear activation functions Krizhevsky et al. (2012) Karpathy et al. (2014). Yann LeCun pioneered one of the earliest convolutional networks in 1990 LeCun et al. (1990) . The resulting architecture called LeNet was used for character recognition for zip codes and numerical digits. What makes convolutional networks attractive is the fact that no features apart from the images are used when training the network. A property that makes CNN stand out with respect to other neural network model's is its ability to process and find patterns in images LeCun et al. (1995). There are several different CNN architectures used for object detection. A distinction that is important to note is that there is a difference between network configuration, that is, the number and type of layers used and the network architecture when dealing with object/vehicle detection models. The latter is usually compromised of several CNN's as well as other Machine Learning models to increase detection accuracy and speed. In the following sections, we explore a number of different state-of-the-art models used in object detection.

2.2. You Only Look Once

Most object detection models divide images into regions of interests (ROI) or segments. A detection algorithm is then applied to these ROI/segments in order to classify any object or part of an object. The YOLO) model, on the other hand, applies a single neural network to a whole image Redmon et al. (2016). It is then the job of the network itself to divide the image into classified segments and ROI. Another step is then applied by the model to sum the ROI with the same label to finally output the detected object label and location in the image, as seen in Figure 1. This significant change in procedure allows the network to detect and classify images at extremely high speeds allowing for real-time image detection Redmon and Farhadi (2017). The only problem, with this network, is that it does not handle the detection of tiny objects well because the network attempts to output detection results at one go. The latest version of YOLO, on the other hand, manages to more accurately detect smaller objects due to several classification steps introduced in the network **?**. The most salient feature of version three (v3.0) is that it makes detections at three different scales. The only downside to this is a slight reduction in processing speed; in fact, it is about 15 fps slower then YOLOv2.0.

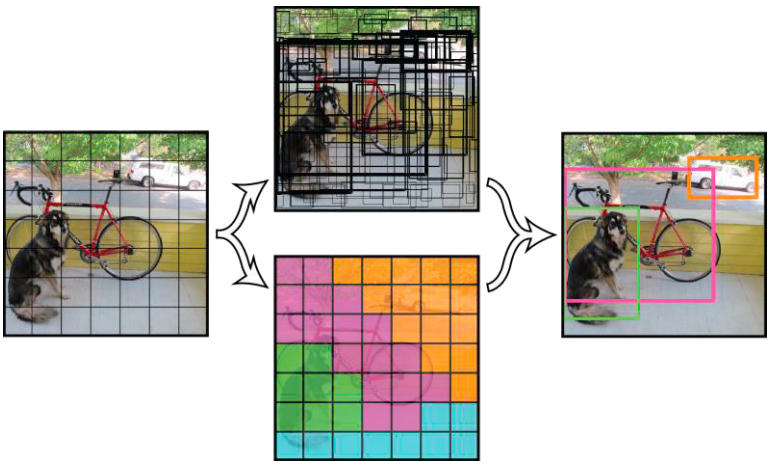

Fig. 1. A representation of how YOLO applies prediction on a single network and how final classification is realised from the amalgamation of the various detected segments.

2.3. RetinaNet

The RetinaNet model implements a similar algorithm to the YOLO detection model. In both cases, the models apply a neural network on a single image. These type of detection networks referred to as single stage detection

models tend to fail with respect to two-stage detectors in terms of detection accuracy. This failure is mostly due to extreme class in balance encountered when training Lin et al. (2017b). Lin et al. manage to solve this issue by applying a focal loss. Focal loss is the reshaping of cross entropy loss such that it down-weights the loss assigned to well-classified examples. The novel focal loss focuses on training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. RetinaNet also manages to compare around 100,000 different segments for one image in contrast to models such as YOLO that classifies between 98 to 2000 segments depending on the version. As shown in Figure 2 this model uses ResNet for deep feature extraction as well as a Feature Pyramid Network (FPN) Lin et al. (2017a) on top of the ResNet to build a rich feature set that can discern between different image patterns and correctly classify multiple objects.

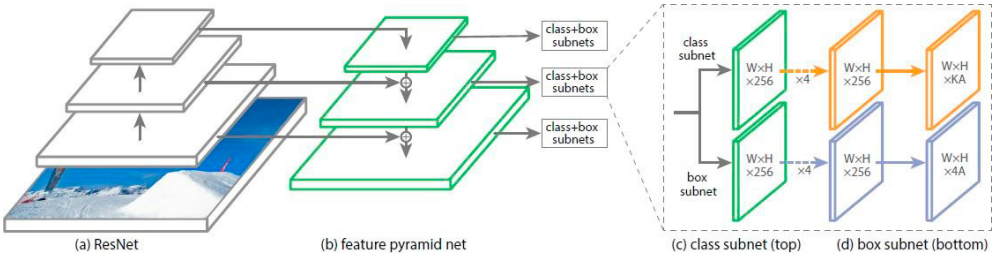

Fig. 2. The RetinaNet network architecture uses a Feature Pyramid Network (FPN) backbone on top of a feedforward ResNet architecture Lin et al. (2017b)

2.4. Region Based Convolutional Neural Network

The final detection model architecture evaluated in this study is the RCNN model. Unlike YOLO and Retinanet, RCNN splits the detection process into two separate stages. The first stage evaluates an image in order to extract possible ROI. In the first version of the RCNN model Girshick et al. (2016), the number of extracted ROI amount to 2000 regions. A CNN is then applied on each ROI as a feature extractor in combination with a Support Vector Machine (SVM) as the classification network, as shown in Figure 3. Additionally, the algorithm also outputs four different dimensions describing the presence of the detected object in the ROI. Thus, managing to localise the object better and find the border of the detected object. An essential feature for Autonomous Vehicle (AV) systems is the boundary that differentiates between incoming cars and traffic. The RCNN architecture has improved over the past four years with the latest version introducing two different processes to the original RCNN architecture Ren et al. (2015). The ROI extracted in the initial stage was done through a greedy search. Faster RCNN applies a pre-processing step that applies convolution to the whole image, followed by a network that is trained to find ROI. The last step substantially reduces computational time and improves upon the accuracy of the model Girshick (2015).

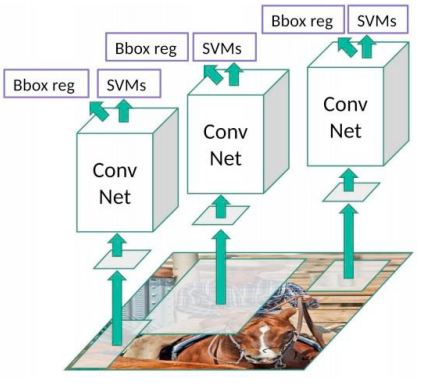

Fig. 3. Process diagram showcasing the breakdown of an image into segments and final detection for an RCNN network Girshick et al. (2016)

3. Methodology

3.1. Evaluation Methodology

The evaluation was conducted by first building a simple pipeline to convert the video clips into separate frames. Each video frame was individually annotated with, the number of vehicles in the frame, position of vehicles, type of vehicle (if available) a selection of consequent frames were selected upon two different criteria. The first criteria looked for difficulty based on occlusion (vehicles occluded each other); the second criteria were based upon difficulty due to weather or light conditions. The object detection models were then applied to the datasets, and the following metrics were measured:

- Detection accuracy
- Number of evaluated frames per second (FPS)

Detection Accuracy is computed by using the mean average precision metric (mAP). This metric is calculated by first measuring the precision value, that is, the percentage value of correct predictions, which in object detection is the number of pixel values that are correctly detected. The second metric is the recall value. This value is a measure that calculates the effectiveness of the True Positive (TP) values. Intersection over union (IOU) is the final measure used. This value determines the number of overlap between the bounding box for the detected object and the ground truth. The final mAP value is calculated by measuring the precision and recall values at different IOU for each frame (image) **?**. FPS is the number of frames (images) displayed per second it is a value that determines how smooth an animation or video is. The higher the FPS, the smoother the animation but results in a more computationally expensive video or animation. The FPS value is an important metric as it determines the speed of the detection algorithm. Faster algorithms are more suited for real-time applications but, usually results in a loss in detection accuracy (mAP) as shown in the results section of this paper. In addition, all models are trained on Common Objects in Context (COCO) dataset Lin et al. (2014). The dataset is made up of 123,287 images and contains 886,284 instances of 81 different categories of objects. This dataset as it is a standard detection dataset used in several different evaluations He et al. (2016). This step also ensures uniformity across training. Although not a transport detection network in itself, the COCO dataset contains numerous examples of vehicles split into cars, trucks, buses, bikes and motorcycles, as shown in Figure 4. The architectures identified for testing include YOLOv3.0, Faster RCNN and RetinaNet.

Fig. 4. a,c,b,d are a sample of vehicle detection images found in the COCO dataset. The coloured shapes mask the edge of the vehicles. This is used both for training as well as testing to measure the accuracy of the detection model.

3.2. Setup

The evaluation process undertaken in this study takes into consideration the speed of detection. Thus, it is also important to go through the setup of the machine used for evaluation:

- CPU: Intel Core i7-8750h CPU 2.20GHz
- RAM: 32.0GB
- GPU: GeForce GTX 1050 Ti
- OS: 64-Bit OS Windows 10

3.3. UA-DETRAC Dataset

UA-DETRAC Wen et al. (2015) is a challenging real-world multi-object detection and multi-object tracking benchmark. The dataset consists of 10 hours of video footage. The video is captured with a Canon EOS 550D camera at 24 different locations at Beijing and Tianjin in China as seen in Figure 6. Each video is recorded at 25 frames per seconds (FPS), with a resolution of 960540 pixels. There are more than 140 thousand frames in the UA-DETRAC dataset and 8250 vehicles that are manually annotated, leading to a total of 1.21 million labelled bounding boxes of objects.

Fig. 5. Sample images taken from the UA-DETRAC dataset

3.4. GFRT Dataset

The GFRT (Gozo ferry road transport) dataset is a unique dataset based upon the IP camera capture feeds mined from the internet. The dataset is made up of over 20,000 frame images of 1 min clips taken from 5 different IP cameras across the same road network at different times during the day. This challenging dataset contains 640x320 pixel images under sometimes extreme light invariance conditions. The dataset has been physically annotated by the authors and will be released online later on during the progression of the current research.

Fig. 6. Sample images taken from the GFRT dataset made up of traffic captured at different times during the day

4. Results

Table 1 compares mean average precision (mAP) Yue et al. (2007) values and FPS obtained using three different models and applied to two different datasets. Although the detection accuracy is not impressive on its own, one must keep in mind that the dataset chosen for training is not fine-tuned on the dataset itself. Furthermore, we also found that given that the models were trained on the COCO dataset which includes high-resolution images, the mAP for the GRFT was impacted as the quality of the images supplied was not as good. This meant that in some cases, object where incorrectly misclassified due to the resolution difference. As further discussed in the following section, several ITS apply some form of detection on video feeds. This means, that although a vehicle might not be detected in 1 frame, it can still be detected in other frames as it changes angle or increases in size, thus, increasing overall robustness of the system.

Table 1. Vehicle detection accuracy results

Object Detection Model	Dataset	mAP	FPS
YOLOv3.0-608	UA-DETRAC	56.5	12
YOLOv3.0-608	GFRT	54.3	8 22
YOLOv3.0-tiny	UA-DETRAC	25.8	120
YOLOv3.0-tiny	GFRT	18.9	144
Retinanet-101-800	UA-DETRAC	61.2	0.2
Retinanet-101-800	GFRT	59.5	1
RCNN	UA-DETRAC	58.4	2
RCNN	GFRT	57.8	3.3

5. Discussion

When developing ITS, different use cases require fine tuning in various aspects of the system. ITS use vehicle detection in order to process camera feeds that extract traffic information from IP camera feeds. These systems use techniques that calculate traffic flow, vehicle speeds as well as tracking of dangerous drivers. If the ITS aims to operate control structures such as traffic lights that change value depending on traffic flow, then, based upon the results and study conducted on this research YOLOv3.0 gives the best results. This is because such a system would require an IP camera footage to be processed in real time but remain relatively accurate. On the other hand, if traffic flow is needed for training other machine learning models, the mAP is more important, and Retinanet would, in that case, be the best option. YOLOv3.0 obtained the highest FPS, but it also had the lowest mAP. In the case of areas with heavy traffic, mAP might not be that important, especially if the tracking process results in a workable system. This trade-off is due to the accuracy of tracking algorithms if a vehicle is at least detected once (across different frames) than that vehicle can be tracked and counted in traffic monitoring ITS system. RCNN provided the most balanced result with high accuracy to frame-rate ratio. Finally, if the setup is upgraded to the latest hardware FPS as well as accuracy would increase. For example, in the latest benchmarks, YOLOv3.0 achieved an impressive FPS of 220 Redmon and Farhadi (2018). The results obtained can be used to drive the ITS implementation choice for different purposes. For example, smart parking systems do not require the detection to occur in real-time as availability of parking spots requires accuracy rather than speed. On the other hand, traffic monitoring systems need more real-time detection to fine tune traffic control structures such as traffic light the micro changes in traffic.

6. Conclusion

In this paper, a methodology for evaluating vehicle detection models for ITS is presented. The models were applied to two different datasets. An initial dataset used for vehicle detection as well as a new dataset used for the first time in this study. Finally, the results show that YOLOv3.0 and Faster RCNN are the most balanced models when comparing mAP and FPS. Further research needs to be done to measure the effect that the detection accuracy has on vehicle tracking models as well as possible pre-processing techniques that can be used to improve image quality when working

with low-quality IP camera feeds. Future work also includes how detection accuracy changes due to IP camera feed quality and resolution as well as the infrastructural costs in terms of processing and data rates needed for ITS systems using IP camera for data gathering.

References

Bishop, C.M., 2006. Pattern recognition and machine learning. springer.

Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., Andrew, N., 2013. Deep learning with cots hpc systems, in: International conference on machine learning, pp. 1337–1345.

Cramer, S., Kampouridis, M., Freitas, A.A., Alexandridis, A.K., 2017. An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. Expert Systems with Applications 85, 169–181.

Engel, J.I., Martin, J., Barco, R., 2017. A low-complexity vision-based system for real-time traffic monitoring. IEEE Transactions on Intelligent Transportation Systems 18, 1279–1288.

Girshick, R., 2015. Fast r-cnn, in: The IEEE International Conference on Computer Vision (ICCV).

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2016. Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence 38, 142–158.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725–1732.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436.

LeCun, Y., Bengio, Y., et al., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361, 1995.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network, in: Advances in neural information processing systems, pp. 396–404.

Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: European conference on computer vision, Springer. pp. 740–755.

McQueen, B., McQueen, J., 1999. Intelligent transportation systems architectures.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788.

Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 .

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region pro- posal networks, in: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neu- ral Information Processing Systems 28. Curran Associates, Inc., pp. 91–99. URL: http://papers.nips.cc/paper/ 5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks 61, 85–117.

Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.C., Qi, H., Lim, J., Yang, M.H., Lyu, S., 2015. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. arXiv preprint arXiv:1511.04136 .

Yue, Y., Finley, T., Radlinski, F., Joachims, T., 2007. A support vector method for optimizing average precision, in: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM. pp. 271–278.