On the Performance of Extended Real-time Object Detection and Attribute Estimation Within Urban Scene Understanding

Khalid N. Ismail*,†
*Department of Computer Science
Durham University
Durham, UK
†Information Technology Dept., Faculty of Computers and Information
Menoufia University, Egypt
khalid.n.ismail@durham.ac.uk

Toby P. Breckon
Department of {Computer Science, Engineering}
Durham University
Durham, UK
toby.breckon@durham.ac.uk

***Abstract*—Whilst real-time object detection has become an increasingly important task within urban scene understanding for autonomous driving, the majority of prior work concentrates on the detection of obstacles, dynamic scene objects (pedestrians, vehicles) and road sign-age within the scene. By contrast, for an autonomous vehicle to be truly able to interact with occupants and other road users using a common semantic understanding of the environment it is traversing it requires a considerably extended scene understanding capability. In this work, we consider the performance of extended *"long-list"* object detection, via an extended end-to-end Region-based Convolutional Neural Network (R-CNN) architecture, over a large-scale 31 class detection problem of urban scene objects with integrated object attribute estimation for appropriate colour and primary orientation. We examine the extended performance of this multiple class object detection and attribute estimation task operating in real-time with on-vehicle processing at 10 fps. Our work is evaluated under a range of real-world automotive conditions across multiple complex and cluttered urban environments.**

***Index Terms*—object detection, region-based CNN, convolutional neural networks, autonomous vehicles, Human-like Guidance**

I. Introduction

Recent advances in efficient object detection architectures have played an important role within the rapid development of autonomous driving technologies [1]–[4]. With the development of deep convolutional neural networks (CNN), real-time object detection has become both robust and real-time [5]–[8].

Following the taxonomy of [9], object detection methods can be categorized into two types: region proposal based methods and regression/classification based methods. Region proposal based methods initially generate a set of candidate region proposals and subsequently classify each of these region proposals into either a number of discrete object classes or as background (i.e. not an object of interest) [9]. The region based CNN (R-CNN) work of [10] and its later versions Fast(er)-RCNN [6], [7] marked initial attempts at applying deep learning approaches to the joint task of both object detection and classification achieving significant performance improvement on benchmarks comparing to the traditional hand-crafted features based methods [11], [12]. Mask R-CNN [8], extends Faster R-CNN, additionally predicting an object segmentation mask in parallel with existing bounding box prediction.

R-CNN employ a stage-wise strategy combining region proposals with CNN based feature extraction and assigns a class-specific score to each region proposal using a support vector machine classifier. Fast R-CNN significantly accelerates the training and inference by introducing a Region of Interest

Fig. 1. Exemplar extended real-time multiple class object detection with attribute estimation within an urban environment.

The authors thank the Renault-Nissan Group for funding and experimental support.

(RoI) pooling layer so that feature extraction for multiple region proposals from the same image can be performed within one single forward pass through the CNN hence speeding up both training and inference significantly. Faster R-CNN [7] further reduces the running time of these detection networks by merging the region proposal network (RPN) and Fast R-CNN in a single network with shared convolutional features. Unlike the aforementioned object detection methods which predict a bounding box for each detected object, Mask R-CNN [8], which is extended from Faster R-CNN, has an additional branch predicting an object mask in parallel with the existing branch for bounding box prediction.

By contrast, combined regression and classification based object detection methods predict object detections directly from feature maps extracted within a deep CNN architecture. Within this paradigm, a detection is jointly characterized by both its location and class which can be predicted by learning a regression and classification models respectively. Typical object detection methods falling in this category include YOLO [13], YOLOv2 [14], YOLOv3 [15] and SSD [16].

Human-Like Guidance system for driving navigation should considerably decrease the cognitive load of the driver and minimize their navigational mistakes. Unlike the current turn-by-turn in-vehicle navigation systems, which can lead to confusion and distraction as drivers react to navigation instructions [17]. To enable the Human-Like Guidance system understand and perceive the surrounding environment, we could employ a region proposal based approach for "long-list" object detection and attribute estimation. Recognizing a long list of objects and its attributes will enable the Human-Like Guidance system to give landmark based navigation instructions such as "Follow the red Car", "Turn right after the shop", " Turn left after the parked black Car".

In this paper, following from recent comparative studies [18], [19], we leverage the Faster R-CNN architecture of [7] as a backbone for our extended detection and attribute estimation task. Overall, we report performance on the 31 object classes spanning vehicles, pedestrians, buildings and street furniture (see list in Table I, Section III). Furthermore, we extend the Faster R-CNN architecture to additional predict the colour (see list Figure 4), whether a vehicle is parked or in motion (Figure 5) and its discrete orientation to the camera (Figure 6, Section II).

The key contributions of this work are:

- an extended Faster R-CNN architecture for joint object detection and attribute estimation (Section II).
- performance evaluation of Faster R-CNN object detection over a complex urban environment specifically addressing the challenges of "long-list" object detection in the urban environment (Section IV).

II. Network Architecture

We extend the architecture of Faster R-CNN [7] to jointly perform object detection and attribute estimation.

Following the original Faster R-CNN architecture which divides the framework of detection in two stages. The first stage is the region proposal network (RPN) that generates region proposals. The second stage is the fast R-CNN network [6] which uses the proposed regions and performs classification and bounding box regression which adopts an established CNN classification architecture (e.g. VGG-16 [20], ResNet-101 [21]). The entire system remains a single and unified network for object detection and attribute estimation. The RPN consists of convolutional layers that generate a set of regions (anchors) with different scales and aspect ratios. The RPN then predicts the bounding box coordinates and object class probability scores for those anchors denoting whether the region is an object or not. Anchors are generated by spatially sliding a 3×3 window through the feature maps of the last shared convolutional layer. These features are then fed to objectness classification and bounding box regression layers. Objectness classification layer classifies whether a region proposal is an object or a background while bounding box regression layer predicts the coordinates of the area.

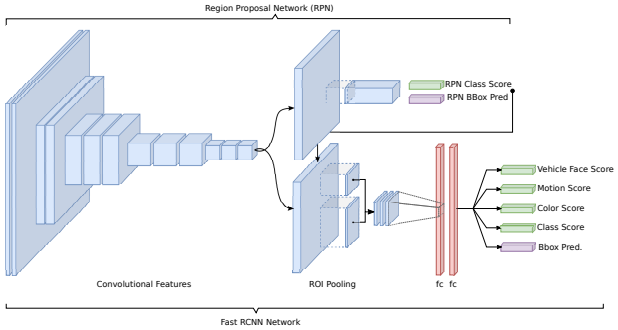

Fig. 2. Extended R-CNN architecture performing joint object detection and attribute estimation.

In the Fast RCNN stage, the whole image is first processed with convolutional layers to produce convolutional feature maps. These maps are fed into the Region of Interest (RoI) pooling layer which employs region proposals generated from the RPN as shown in Figure 2. Subsequently, this pooling layer extracts a fixed length feature vector associated with each region proposal. Each feature vector is then fed into a sequence of Fully Connected (FC) layers before finally branching into five output layers. One output layer is responsible for producing softmax probabilities for all object classes and background categories. The second output layer encodes refined bounding box coordinates with four real-valued numbers. The other three softmax layers are responsible for producing probabilities for color, motion and vehicle face attributes.

This extended architecture is trained end-to-end using a multi-task loss function as defined in Equation 1. The multi-tasks loss function L used to jointly train bounding box regression, object classification and attributes classification.

$$L(\{p_i, c_i, n_i, v_i, t_i) = \frac{1}{N}\sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N}\sum_i L_{clr}(c_i, c_i^*) + \frac{1}{N}\sum_i L_{mot}(m_i, m_i^*) + \frac{1}{N}\sum_i L_{veh}(v_i, v_i^*) + \frac{\lambda}{N_{box}}\sum_i p_i^* L_{box}(t_i, t_i^*) \quad (1)$$

Where where p_i is the predicted probability of the i-th anchor being an object. The ground truth label p_i is 1 if the anchor is positive, and is 0 if the anchor is negative. c_i is the predicted probability of the i-th anchor being a color. The ground truth label c_i is 1 if the anchor is positive, otherwise 0. m_i is the predicted probability of the i-th anchor being a motion (parked or moving). The ground truth label m_i is 1 if the anchor is positive, otherwise 0. v_i is the predicted probability of the i-th anchor being an orientation (front, rear or side). The ground truth label v_i is 1 if the anchor is positive, otherwise 0. t_i is a vector stores the 4 parameterized coordinates of the predicted bounding box while t_i^* is that of the ground-truth box associated with a positive anchor. The classification losses L_{cls}, L_{clr}, L_{mot} and L_{veh} are binary log losses. The regression loss L_{box} is a smoothed L_1 loss.

For the background RoI, there is no annotation of a ground-truth bounding box so L_{box} is ignored. Also, where there is no notion of ground-truth attributes for any detected object, L_{atr} is ignored.

Training is performed for the extended Faster RCNN architecture using a learning rate of 0.001 and a batch size of 1 is being adopted. All networks are trained on NVIDIA 1080 Ti GPU via PyTorch.

Fig. 3. The vehicle equipped with stereo camera and combined GPS/IMU system

III. Evaluation Dataset

In contrast to work primarily concentrating on pedestrian and vehicle object detection [22], [23], for our long-list object detection task we use a customized data set collected over two urban locations during daylight driving conditions (Versailles, France and Durham, UK) on differing vehicles. All imagery data is recorded using a Carnegie Robotics MultiSense S21 stereo camera mounted on top of the host vehicle (Figure 3).

A total number of 50,000 images were extracted from 6 hours of driving video footage, for which annotation for the 31 object classes (Table I) in addition to vehicle colour (Figure 4), motion (Figure 5) and orientation attribute sub-labels (Figure 6) was generated.

TABLE I
Detection results of extended Faster RCNN (ResNet-100 and VGG-16) for multiple object classes.

Class	Faster-RCNN	
	ResNet-101 (AP)	VGG-16 (AP)
car	0.964	0.937
bus	0.81	0.760
van	0.873	0.803
truck	0.86	0.759
bicycle	0.649	0.535
bike	0.689	0.550
roundabout	0.473	0.435
traffic light	0.544	0.532
direction sign	0.514	0.415
stop sign	0.36	0.275
one way sign	0.348	0.247
other sign	0.571	0.498
road light pole	0.796	0.763
safety wall	0.814	0.768
safety fence guard	0.683	0.646
safety pole	0.475	0.390
zebra crossing	0.838	0.770
cone	0.486	0.470
blocked road sign	0.457	0.415
bus stop shelter	0.587	0.579
billboard	0.63	0.542
garbage container	0.571	0.394
garbage bin	0.339	0.209
pedestrian	0.457	0.560
group of people	0.539	0.435
rider (on bike)	0.512	0.373
house	0.659	0.531
church	0.454	0.261
office building	0.744	0.671
monument	0.421	0.242
shop	0.508	0.365
mAP	**0.60**	0.52

IV. Evaluation

We evaluate using the image dataset outlined in Section III using a 9:1 train to test split (training: 45k, testing: 5k).

A. Object Detection

Detection performance is measured via average precision (AP) for each object class and as the mean average precision (mAP) over all classes following PASCAL VOC [24].

Table I shows our detection results for our ResNet-101 and VGG-16 classification network Faster R-CNN variants, trained for a fixed number of RPN proposals (300). Both classification networks have been pre-trained using ImageNet [25] for transfer into our Faster R-CNN architecture. We observe Faster R-CNN has a mAP of 0.60 with a deeper ResNet-101 architecture for the classification network whilst VGG-16 in the same role has a mAP of 0.52 (Table I).

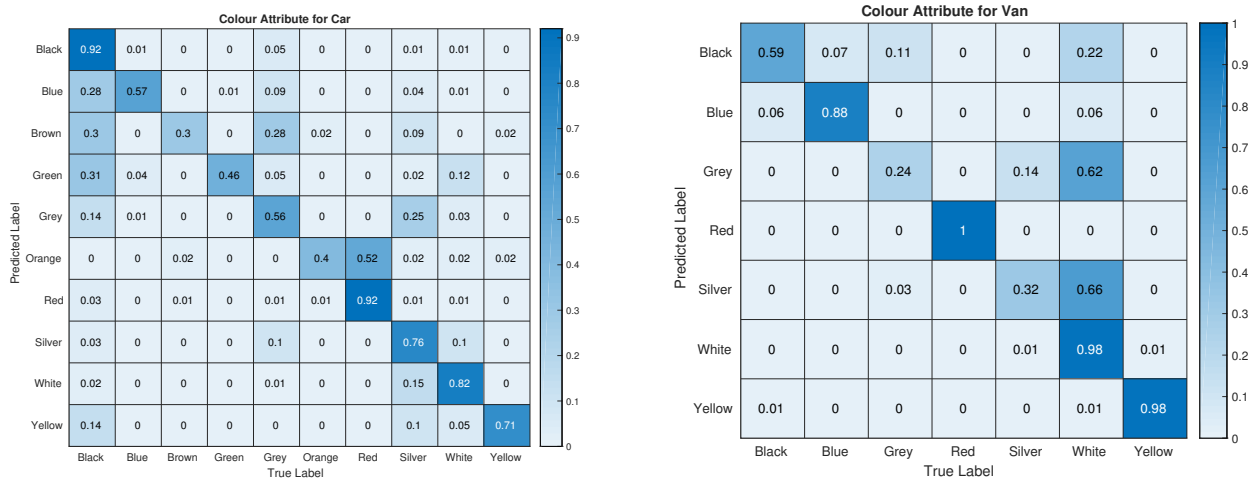

Fig. 4. Confusion matrix: colour - cars (left) and vans (right).

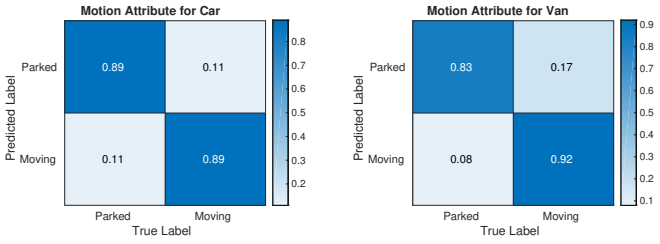

Fig. 5. Confusion matrix: motion - car (left) and van (right)

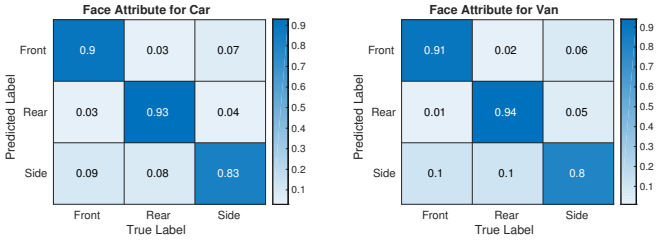

Fig. 6. Confusion matrix: orientation - car (left) and van (right)

Furthermore, we notice high AP performance for commonly occurring vehicle classes (car, bus, van, truck) due to imbalance, and hence strong presence of these classes, within the training dataset. In addition, classes comprising large-scale objects within the frame (e.g. office buildings, zebra crossing) also exhibit high AP performance. By contrast, smaller objects that occur less frequently (e.g. stop sign, one way sign, traffic light) exhibit lower AP performance attributable to both limited discriminating detail within the scene and limited training samples.

B. Attribute Estimation

In addition to primary object detection, object attributes for colour, motion and discrete vehicle orientation are been for all four wheel vehicles objects via the extended Faster R-CNN approach proposed (Section II).

The confusion matrices reported true positive (TP) and false positive (FP) performance for vehicle colour, motion and orientation are shown in Figures 4, 5 and 6 respectively. Overall we observe strong TP performance (confusion matrix, diagonal) and low FP occurrence (confusion matrix, off-diagonal) across all three attributes with the exception of common colour confusers for cars (e.g. grey to silver, white to silver) and additional outliers due to livery colours for vans (Figure 4, left/right).

Qualitative results are shown within Figures 1 and 7 (for Faster R-CNN + ResNet101) where we can see both object detection and attribute estimation from on vehicle processing achieved at 10 fps using a Nvidia 1080 GPU.

V. Conclusion

Our extended Faster R-CNN architecture is observed to provide performance over a "long-list" urban object detection task consistent with that of earlier work [7] with the addition of attribute estimation for a subset of objects within the same common architecture (maximal mAP: 0.6). Future work will consider the additional extension of attributes across additional object types and the integration of stereo depth. We will also consider using our extended Faster R-CNN architecture to support Human Like Guidance system in future.

Acknowledgment: The authors thank the Renault-Nissan Group for funding and experimental support.

Fig. 7. Exemplar extended real-time multiple class object detection and attribute estimation.

REFERENCES

[1] G. Payen de La Garanderie, A. Atapour-Abarghouei, and T. Breckon, "Eliminating the dreaded blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery," in *Proc. European Conference on Computer Vision*. September 2018, Springer, (to appear).

[2] N. Alshammari, S. Akcay, and T. Breckon, "On the impact of illumination-invariant image pre-transformation on contemporary automotive semantic scene understanding," in *Proc. Intelligent Vehicles Symposium*. June 2018, IEEE, (to appear).

[3] A. Atapour-Abarghouei and T. Breckon, "Depthcomp: Real-time depth image completion based on prior semantic scene segmentation," in *Proc. British Machine Vision Conference*. September 2017, pp. 208.1–208.13, BMVA.

[4] T. Guo, S. Akcay, P. Adey, and T. Breckon, "On the impact of varying region proposal strategies for raindrop detection and classification using convolutional neural networks," in *Proc. International Conference on Image Processing*. September 2018, pp. 3413–3417, IEEE.

[5] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[6] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, 2019.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 580–587.

[11] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, "Bottom-up segmentation for top-down detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3294–3301.

[12] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.

[14] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 7263–7271.

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

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision (ECCV)*. Springer, 2016, pp. 21–37.

[17] B. Wang, Q. Stafford-Fraser, P. Robinson, E. Dias, and L. Skrypchuk, "Landmarks based human-like guidance for driving navigation in an urban environment," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[18] S. Akcay and T. Breckon, "An evaluation of region based object detection strategies within x-ray baggage security imagery," in *Proc. International Conference on Image Processing*. September 2017, pp. 1337–1341, IEEE.

[19] S. Akcay, M. Kundegorski, C. Willcocks, and T. Breckon, "On using deep convolutional neural network architectures for automated object detection and classification within x-ray baggage security imagery," *IEEE Transactions on Information Forensics & Security*, vol. 13, no. 9, pp. 2203–2215, September 2018.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun 2010.

[25] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. Computer Vision and Pattern Recognition*, 2009, pp. 248–255.