Vulnerable Road User Detection using YOLO v3

Saranya.K.C[1]
School of Electronics Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu

Arunkumar Thangavelu[2]
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu

***Abstract*—Detection and classification of vulnerable road users (VRUs) is one of the most crucial blocks in vision based navigation systems used in Advanced Driver Assistance Systems. This paper seeks to evaluate the performance of object classification algorithm, You Only Look Once i.e. YOLO v3 algorithm for the purpose of detection of a major subclass of VRUs i.e. cyclists and pedestrians using the Tsinghua – Daimler dataset. The YOLO v3 algorithm used here requires less computational resources and hence promises a real time performance when compared to its predecessors. The model has been trained using the training images in the mentioned benchmark and have been tested for the test images available for the same. The average IoU for all the truth objects is calculated and the precision recall graph for different thresholds was plotted.**

Keywords—Yolo v3; Tsinghua-Daimler cyclist benchmark; cyclist detection; pedestrian detection; IoU

I. Introduction

The past decade has witnessed significant acceleration in the pace of development of automotive technologies which aim at making driving and commutation safe and facile. Deployment of autonomous driving vehicles and building Advanced Driver Assistance Systems (ADAS) to be used in hybrid vehicles are major steps in realizing this. Of the many fields related to these, systems related to improving the driving safety such as pre collision systems, crash imminent braking systems play a very crucial role. However, extensive research has been undertaken over the past few years to protect vulnerable road users (VRUs), including pedestrians, cyclists, motorcyclists. Nearly half of the world traffic deaths occur among vulnerable road users, and road traffic injuries are the eighth leading cause of death for all age groups, according to statistical data provided by WHO [1]. Among the many VRU categories, cyclists and pedestrians are the weakest and fall prey to most accidents because of the lack of protection devices. Hence the development of systems for the detection and identification of VRUs becomes an essential need of the hour, to make their commutes safer and for the ADAS to be practically and widely deployable.

Many approaches based on different sensors are employed in vehicle environment perception systems. The vision based sensors especially monocular cameras, are the most preferred as a standalone or in combination with other sensors when it comes to detection of VRUs, due to the availability of high resolution perception views. Vision based cyclist and pedestrian detection face several challenges due to the diversity in shape, posture, viewpoints, crowded backgrounds, etc. and several algorithms and methodologies have been implemented for the same keeping these considerations in account.

II. Background

Algorithms that are used for the purpose of feature extraction and classification can predominantly be handcrafted or Deep Learning based. The Haar-like feature detector which uses variations in intensities for the detection of the object [2], [3], the Viola and Jones (VJ) detector designed by Viola et al. [4] which uses a detection approach based on cascaded Haar-like features, which also considers the rapid pixel intensity changes ,and the Histogram of Gradients(HOG) detector, suggested by Dalal and Triggs which uses a linear Support Vector Machine for classification [5-8] to find an object's characteristics based on the intensities of the local gradients [2], [6] are some of the common hand-crafted features based methods used in general for pedestrian detection. However, hand-crafted methods which rely on low-level features which are manually designed to find the ROI's [9] are not very efficient as features which complex are arduous to handcraft. Au contraire, Deep Learning (DL) based techniques are highly autonomous by allowing the network to determine features.

Since the advent of DL, several approaches have been designed for pedestrian or cyclist detection. In the method described by Wei Tian [10] cyclists in different views and angles are located using cascade detectors. Together with trajectory planning, this model employs an ROI extraction derived based on geometry but achieves only 11 fps when employed in real time. Ren [11] realized an accuracy of 76.47% for an IoU threshold of 0.7 using a Recurrent Rolling Convolution (RRC) architecture employed on multiscale feature maps. Saleh in [12] use a Faster RCNN based network on synthetic image datasets to perform better than the HOG- SVM classifier by 21% in average precision. Felzenswalb [25] designed the Deformable Part Model (DPM) on the basis of HOG detector to undermine the distortions caused due to non-rigid objects. To ensure swift and accurate detection, Yang in [13] used convolutionary object detector with Scale based pooling and CRCs. The scale-dependent pooling allows the identification of tiny objects to be improved, and the CRCs help to enhance detection speed by rapidly removing false detections.

While all the previously cited works either concentrate on the detection of either the pedestrians or the cyclists, very less literature is available for the simultaneous detection of pedestrians and cyclists [19]. In [5] X. Li propose a unified framework for both cyclist and pedestrian detection using a UB-MPR based detection combined with Fast RCNN and Fu in [26], propose a system based on symmetry of objects to recognize the features of cyclists and pedestrians that appear in an image. However, this method still does not reach the real time speed requirements due to the complex isolated stages that required for the detection and classification.

As emphasized by Huang Ji [16] networks like YOLO which uses a relatively simpler framework for object detection and classification have a slight reduction in accuracy when set against methods like Fast RCNN [14] but expiate for it with real time performance [17] .With optimization, methods like YOLO and its modifications like, YOLO9000, fast YOLO, etc. [18] one can achieve better real time object detection and also makes proper use of the capabilities of the Graphical Processing Unit (GPU) [27].

This paper aims to present and examine the performance of the V3 variant of YOLO for the unified detection of cyclists and pedestrians using the Tsinghua- Daimler Benchmark [15] dataset. The work performed focused on analyzing the performance of the above model on the above dataset in order to establish its efficiency and reliability for the identification of cyclists and pedestrians in real time. A NVidia GTX 1080 Ti GPU was used for training the YOLO V3 network and the detection was run on a Intel Core i7 eighth generation processor CPU system.

The organization of the paper is as follows: Section 2 discusses the methodology elucidating about the dataset, the YOLOv3 network architecture and the procedure followed for training. This is followed by Section 3 and Section 4 which present results and conclusion respectively.

III. Methodology

A. Dataset

The Tsinghua-Daimler Cyclist Benchmark used consists of four subsets namely train, valid, test and non VRU *(Fig. 2)*. Train subset consists of 9741 images and only cyclist objects are annotated. This subset includes cyclists which are fully visible with an occlusion rate of lesser than 10% and greater than 60 pixels. Valid subset contains 1019 images and the objects annotated are pedestrian, cyclist, tri cyclist, moped rider, wheel chair user and motorcyclist. The objects that are labeled have more than 20 pixels. The test set has 2914 images and the same list objects annotated as the valid set. Like the valid set, the objects labeled here also have more than 20 pixels. Non VRU has 1000 images which do not contain any objects of interest such as pedestrian, cyclist, motorcyclist, tricyclist, wheel chair user, moped rider.

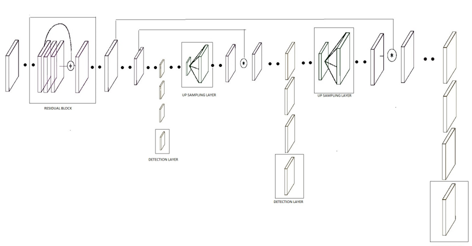


Fig. 1. YOLO V3 architecture

B. Network Architecture

YOLO V3 [20] *(Fig. 1)* uses dimension clusters as anchor boxes for the prediction of bounding boxes [17]. The network

Fig. 2. Sample images from the subsets of Tsinghua-Daimler Cyclist Benchmark: (a)Train (b)Valid (c)Non-VRU (d)Test

predicts 4 coordinates for each bounding box, l_x, l_y, l_w and l_h. If (O_x, O_y) is the offset of the cell from the top left corner of the image and pr_w and pr_h are the the width and height of the previous bounding box respectively, then the predictions correspond to:

$$b_y = \sigma(l_y) + Oy \tag{1}$$

$$b_w = pr_w e^{lw} \tag{2}$$

$$b_h = pr_h e^{lh} \tag{3}$$

Here (b_x, b_y) is the coordinate of the center of the bounding box relative to the location of filter application using a sigmoid function. b_w and b_h are width and height of the bounding box respectively.

The class that the bounding box may contain is predicted using multi label classification. This uses a combination of independent logistic classifier and binary cross entropy loss for class predictions during the training.

Yolov3 uses a feature extractor network called Darknet 53 whose architecture is shown in *(Fig. 3)* which is a hybrid variant of the network used in YOLOv2 , Darknet-19 [17]. This network has 53 convolution layers. The filters consist of

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 3. Darknet 53-network architecture [20]

successive 1x1 and 3x3 filters with shortcut connections. In YOLOv3, features are extracted from the 3 scales congruous to feature pyramid networks [21]. This majorly helps to identify small objects. The last layer predicts the bounding box, probability of box containing object and class predictions. The dimensions of the final prediction vector is

$$L \times L \times [3 \times (4 + 1 + No.of classes)] \tag{4}$$

Where three denotes the number of scales, four the number of bounding box offsets and one is the probability that an object is present is the bounding box. Here, the number of classes is two as this paper concentrates only on the detection of cyclists and pedestrians. $L \times L$ is the splitting of the picture into segments.

C. Training

The training was carried out using the parameters given in *Table I* with keras implementation of YOLOv3, running tensor flow as the backend [22]. For training the first subset labeled as train and a part of the pedestrian subset of the Tsinghua-Daimler benchmark was used. The validation dataset is obtained from a random subset of the above set, which is 10% of the total set. Thus 90% of the above set is used for training and 10% for validation. According to the model's architecture, the training images were divided into batches of 8, enabling the GPU to process 8 images at a time followed by the updation of the corresponding gradients and the related weights.

Initial weights for training were acquired from the weights of a YOLOv3-608 model.

TABLE I. TRAINING PARAMETERS FOR THE NETWORK

Dataset used	Tsinghua-Daimler Cyclist Benchmark
No. of Train samples	8767
No. of Validation samples	974
Initial learning rate	0.001 for 1-60 epochs 0.0001for 61-126 epochs
Beta 1	0.9
Beta 2	0.999
Pre-trained weights	YOLOv3-608
No. of Batches	8
No. of Epochs	126
Train size ($height \times width$)	416 × 416
GPU	Nvidia GTX 1080 Ti

The adam optimizer [23] which has an initial learning rate of 0.001 was used. Learning rate reduction on plateau with a factor of reduction of 0.1 was implemented and validation loss was monitored with early stopping. The beta rates of 0.9 and 0.999 were used.

IV. RESULTS

The results obtained after training 126 epochs *(Table II)* at a rate of 45fps for different scenarios are shown in *(Fig. 5)*. The performance was assessed using Intersection over Union i.e. IoU, Precision recall Graph, and F1 score as metrics *(Table III)*. The prediction was considered for account only when it had at least 0.5 confidence.

TABLE II. TEST PARAMETERS FOR THE NETWORK

Dataset used	Tsinghua-Daimler Cyclist Benchmark
No. of Samples used	2914
Threshold used	Detection :0.5 (50% confidence) IoU :50%
Size specs of test image	1024 × 2048 (height × width in pixels)

A. Intersection over Union (IoU)

Intersection Over Union (IoU), a measure based on Jaccard Index, that evaluates the overlap between the ground truth bounding box BB_g and the predicted bounding box BB_p, was used to identify the accuracy of detection.

$$IoU = \frac{area(BB_p \cap BB_g)}{area(BB_p \cup BB_g)} = \frac{area\,of\,overlap}{area\,of\,union} \tag{5}$$

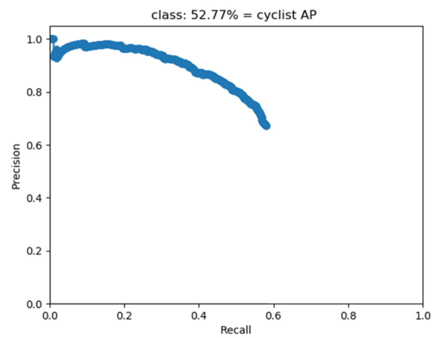
The predicted objects with an IoU of more than 50% are considered as true objects and the average IoU of all the true objects detected is 73.17%.

B. Precision-Recall Graph

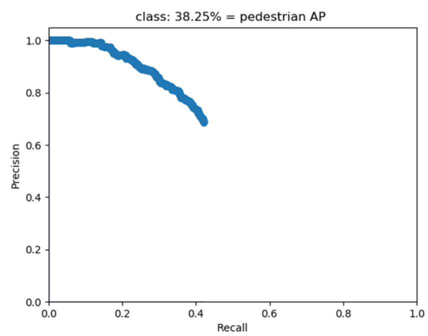
The Precision-Recall graph [24] as shown in the *(Fig. 4)* has been plotted over different threshold values. Here the precision which denotes a model's ability to identify only relevant objects is given by

$$Precision = \frac{T_P}{T_P + F_P} = \frac{T_P}{All\,detections} \tag{6}$$

And recall which is the percentage of true positive detected amongst all relevant ground truths and is given by



[]



[]

Fig. 4. The precision-recall graph over multiple thresholds.

TABLE III. CONSOLIDATED RESULTS

Class	Total	TP	FP	FN	Precision	Recall	F1	AP
Cyclist	1541	762	372	779	0.671958	0.494484	0.5697	52.77
Pedestrian	1314	649	297	665	0.686047	0.493912	0.574336	38.25

$$Recall = \frac{T_P}{T_P + F_N} = \frac{T_P}{All\, ground\, truths} \quad (7)$$

Where, True Positives (*TP*) and False Positives (*FP*) require predictions to have an IoU and confidence level greater than and less than the threshold value of 0.5, respectively. False Negatives(*FN*),which is calculated as the difference between the actual number of the object class present and the number of True Positives denotes the failure of the model to identify the ground truth of the particular object class.

By interpolating all precision and the corresponding recall points, the Average Precision (*AP*) was calculated.

C. F1 score

F_1 score an optimum mix between precision and recall was calculated as the weighted average between precision and recall, thus taking both the false positives and the false negatives into account

$$F_1 score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (8)$$

V. CONCLUSION

The implemented model shows good performance with regards to accuracy and efficiency with a real time performance at a rate of 45 fps for the scenarios found in the Tsinghua-Daimler benchmark. The deficiencies in the performance can mainly be attributed to situations where the objects of interest are very small or show high degree of occlusion or are limited by the characteristics of the camera used to capture images leading to reduced range of vision causing blind spots and limited field of view. Further studies can be carried out on the above-mentioned difficulties to improve the performance.

REFERENCES

[1] A. Dominguez-Sanchez, M. Cazorla, and S. Orts-Escolano, "Pedestrian movement direction recognition using convolutional neural networks," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 12, pp. 3540–3548, 2017.

[2] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[3] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *cvpr*, vol. 97, 1997, pp. 193–199.

[4] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[5] X. Li, L. Li, F. Flohr, J. Wang, H. Xiong, M. Bernhard, S. Pan, D. M. Gavrila, and K. Li, "A unified framework for concurrent pedestrian and cyclist detection," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 2, pp. 269–281, 2016.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 886–893 vol. 1.

[7] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *European Conference on Computer Vision*. Springer, 2014, pp. 613–627.

[8] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, and S. Tubaro, "Deep convolutional neural networks for pedestrian detection," *Signal processing: image communication*, vol. 47, pp. 482–489, 2016.

[9] W. Sun, S. Zhu, X. Ju, and D. Wang, "Deep learning based pedestrian detection," in *2018 Chinese Control And Decision Conference (CCDC)*, June 2018, pp. 1007–1011.

[10] W. Tian and M. Lauer, "Fast cyclist detection by cascaded detector and geometric constraint," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 1286–1291.

[11] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu, "Accurate single stage detector using recurrent rolling convolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 752–760.

[12] K. Saleh, M. Hossny, A. Hossny, and S. Nahavandi, "Cyclist detection in lidar scans using faster r-cnn and synthetic depth images," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[13] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2129–2137.

[14] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1440–1448.

[15] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila, "A new benchmark for vision-based cyclist detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 1028–1033.

[16] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3296–3297.

[17] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6517–6525.

[18] M. J. Shaifee, B. Chywl, F. Li, and A. Wong, "Fast yolo: A fast you only look once system for real-time embedded object detection in video," *Journal of Computational Vision and Imaging Systems*, vol. 3, no. 1, 2017.

[19] K. C. Saranya, A. Thangavelu, A. Chidambaram, S. Arumugam, and S. Govindraj, "Cyclist detection using tiny yolo v2," in *Soft Computing for Problem Solving*. Springer, 2020, pp. 969–979.

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

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 936–944.

[22] J. Redmon, "Darknet: Open source neural networks in c," 2013.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[25] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[26] L.-C. Fu, P.-Y. Hsiao, C.-E. Wu, Y.-M. Chan, and S.-C. Hu, "Vision based pedestrian and cyclist detection method," Jul. 21 2015, uS Patent 9,087,263.

[27] V. Kharchenko and I. Chyrka, "Detection of airplanes on the ground using yolo neural network," in *2018 IEEE 17th International Conference on Mathematical Methods in Electromagnetic Theory (MMET)*. IEEE, 2018, pp. 294–297.

Fig. 5. (a) Multiple object detection and slightly occluded pedestrian detection (b) Multiple object detection and far object detection (c) Multiple cyclist in different angles (d) Cyclist side view detection (e) Cyclist front view (f) Pedestrian side view detection (note: multiple pedestrian present but others are almost fully occluded)

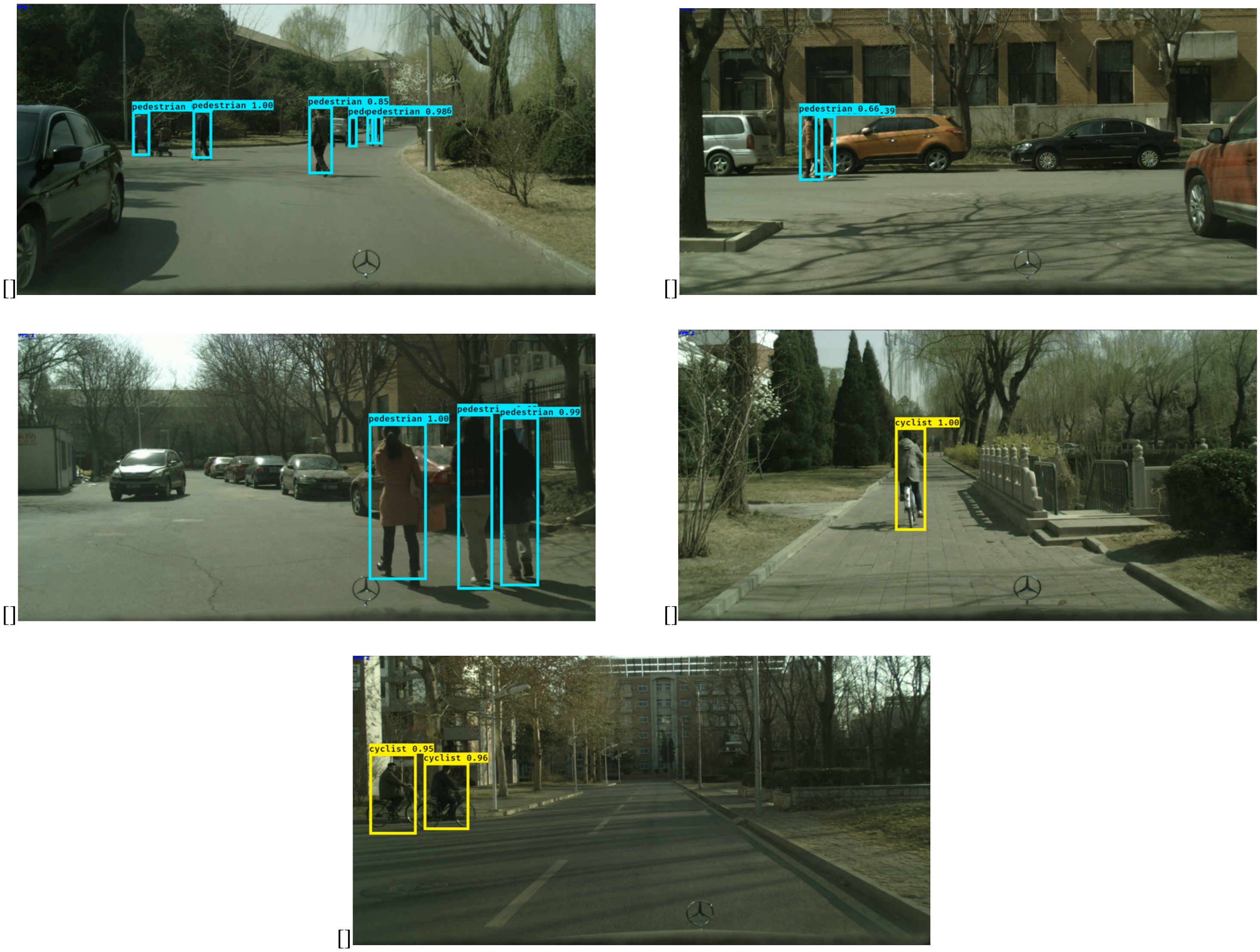


Fig. 5. *(contd.)* (g) Multiple pedestrian detection (h) Multiple pedestrians-side view- occluded (i) Multiple pedestrians-back view (j) Cyclist back view (k) Multiple cyclists side view detection