PEDESTRIAN DETECTION IN DIFFERENT LIGHTING CONDITIONS USING DEEP NEURAL NETWORKS

Jason Nataprawira, Yanlei Gu, Koki Asami, Igor Goncharenko

College of Information Science and Engineering, Ritsumeikan University, Japan, guyanlei@fc.ritsumei.ac.jp

ABSTRACT

Pedestrian safety is one of the most significant issues in the development of advanced driver assistant systems and autonomous vehicles. DNN (Deep Neural Network) or deep learning has been effectively implemented through many applications, especially on object classification. In addition, several famous DNNs, e.g. Faster R-CNN (Faster Region Convolutional Neural Network), YOLO (You Only Look Once) and SSD (Single Shot Detector), are applied for pedestrian detection recently. However, most pedestrian detection research only dealt with the detection at the daytime or nighttime. A few research focused on pedestrian detection at both daytime and nighttime environments. This paper evaluates and compares the performance of DNN-based pedestrian detection algorithm YOLO at both daytime and nighttime environment. The evaluation was conducted on a pedestrian dataset which includes RGB images captured from both daytime and nighttime conditions. The experiment result indicates that the performance of DNN-based pedestrian detection is significantly affected due to the lighting conditions. In the daytime condition, 45% precision on person detection could be achieved, but only 20% precision is obtained in the nighttime condition.

Key words: Pedestrian Detection, Lighting conditions, Autonomous Driving.

1. INTRODUCTION

Pedestrian safety is one of the most significant issues in the development of modern transportation. In 2017, European Commission released a report that implies about 21% of traffic accidents were caused by pedestrian (*2017 Road Safety Statistics: What Is behind the Figures?*, 2017). Advanced driver assistant systems and autonomous vehicles are intensively developed to reduce accidents and improve the effectiveness of transportation. However, the current achievements are still inadequate, e.g. about 65 traffic accidents of Tesla autonomous vehicles involve pedestrians ("Tesla Deaths," 2020). As a result, pedestrian detection becomes an extremely important task before the autonomous vehicles are commercialized.

Recently, Deep Neural Network (DNN) or deep learning has been effectively used for many applications, especially for object detection (Zhao *et al.*, 2019). Since pedestrian detection is a part of object detection tasks, researchers have studied in applying DNN to pedestrian detection (Zhang *et al.*, 2016). Similarly, for some specific tasks, researchers needed to propose a new DNN in order to fit it into pedestrian detection task as what Tian *et al.* (2015) introduced.

Despite the successful implementation of DNN towards pedestrian detections, there are still some obstacles in pedestrian detection domain. Hwang *et al.* (2017) mentioned one of them is lighting condition. Most pedestrian detection research only dealt with pedestrian detection at daytime or nighttime. Only a few research focuses on pedestrian detection at both daytime and night environments. Nonetheless, autonomous vehicle should behave perfectly in all light conditions. In addition, it is better to develop the unified algorithm and system for pedestrian detection in all light conditions to avoid the switching between the daytime and nighttime model, because the correct switching is also a challenging problem.

This paper attempted to evaluate and compare the performance of DNN-based pedestrian detection algorithm at nighttime and daytime environment. This paper shows how the performance of DNN-based pedestrian detection is affected due to different light conditions. The results of this research can inspire the further development of pedestrian detection for autonomous vehicle, e.g. the relevancy of DNN-based pedestrian detection, and the necessity of hardware improvement for the pedestrian detection in different lighting conditions.

This paper is organized in five sections. Firstly, the backgrounds are introduced at the beginning. Then, related works of pedestrian detection and DNN techniques are explained in the second section. Following that, the methodology is explained. The fourth section presents the experimental results. The final section concludes the paper and discusses possible future works.

2. RELATED WORKS

To begin with, Deep Neural Network (DNN) or deep learning has been developed rapidly for a decade. Particularly, the suitable DNN network for object detection is known as Convolutional Neural Network (CNN). The CNN method for object detection was firstly proposed by Girshick *et al.* (2014). It is called R-CNN (Region Convolutional Neural Network). The method works by generating 2000 proposals and then obtains the features to be fed into the network. However, the first establishment never performed well due to the repetition of region proposals, resulting in slow computation time. Consequently, Girshick (2015) again improved the method by creating Fast R-CNN. A novel method from Fast R-CNN is customizing the output layer by branching it to two layers: "cls" regressor for classification task and "bbox" regressor for regression task. This results in the network being capable of running classification and regression tasks simultaneously, hence the faster computation time. Additionally, Fast R-CNN is perfected by Faster R-CNN (Ren *et al.*, 2017). It introduced a Region Proposal Network (RPN) layer and an anchor. The former is responsible of generating region proposals or features from the input image, allowing the network to learn where to locate region proposals by itself. The latter, however, is the center of each sliding window. The illustration of anchor is depicted in figure 1. The anchor becomes the foundation of YOLO (You Only Look Once) (Redmon *et al.*, 2016) method. The R-CNN family networks and YOLO have been widely used for pedestrian detection in daytime (Lan *et al.*, 2018; Tomè *et al.*, 2016; Zhang *et al.*, 2016).

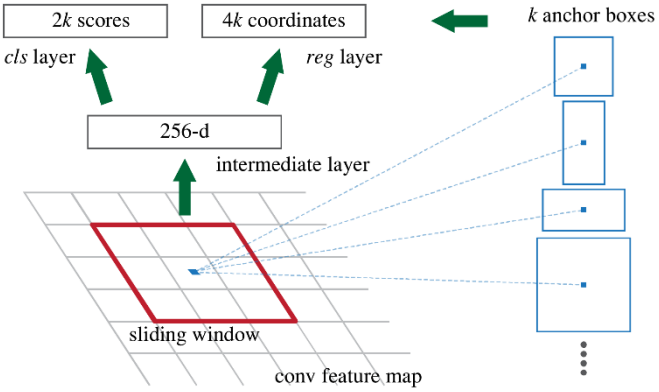


Fig. 1. Constructing anchors at sliding window in Faster R-CNN (Ren *et al.*, 2017)

In terms of pedestrian detection for nighttime environment, a few researchers have inspected this problem. First, multispectral method is one of the famous methods for detecting pedestrian detection at nighttime. When the multispectral pedestrian dataset was published by Hwang *et al.* (2015), a research on this multispectral dataset followed immediately. Choi *et al.* (2016) implemented CNN by inputting both RGB images and FIR (far-infrared) or thermal images to the CNN at the same time. Similarly, SSD (Single Shot Detector) (Liu *et al.*, 2016) was also applied for multispectral pedestrian detection (Hou *et al.*, 2018). Despite applying SSD directly, the authors applied pixel-level image fusion where it alters the pixel-level to obtain the best feature information. Furthermore, RPN was applied for multispectral method (Konig *et al.*, 2017). Additionally, boosted decision trees (Zhang *et al.*, 2016) was utilized for the classification task. By combining RGB and thermal images into RPN, it proved to produce better results.

In contrast, Kruthiventi *et al.* (2017) proposed a method which can extract multi-modal like features of thermal images. They only used RGB images, but they were capable of extracting features from them. They utilized ResNet50 as the base network to produce two networks called "ResNet-teacher" and "ResNet-student". The overview of the network is shown in figure 2. They claimed their "ResNet-student" network has the best average miss rate compared to other proposed pedestrian detection at nighttime environment by using only RGB images.

While most researchers focused more in leveraging the model in multispectral method, Chebrolu and Kumar (2019) applied Faster R-CNN (Ren *et al.*, 2017) for pedestrian detection at daytime and nighttime. They proposed "brightness awareness model, where it is capable of detecting the light environment whether it is day or night, and also detecting pedestrian afterwards. For daytime, they used RGB camera, whereas for nighttime they used thermal images.

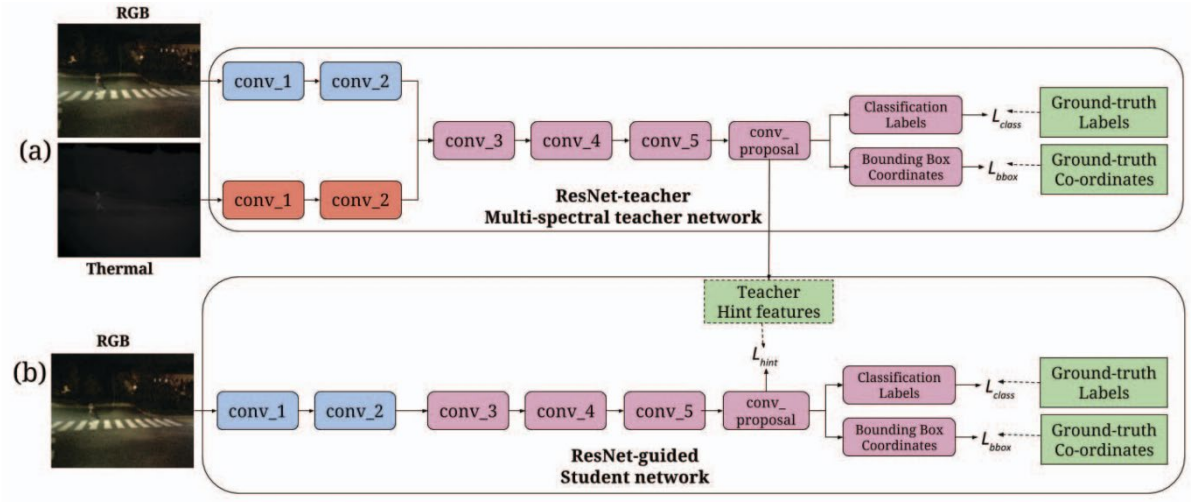


Fig. 2. ResNet-teacher and ResNet student networks architecture (Kruthiventi *et al.*, 2017)

This research focuses on pedestrian detection at both daytime and night environments. An unified algorithm YOLO v3 (Redmon and Farhadi, 2018) is used for pedestrian detection in all light conditions. By comparing the performance of pedestrian detection in different light conditions, this paper shows how the performance of DNN-based pedestrian detection is affected by different light conditions. The result of this paper can be used as a reference for the development of the pedestrian safety function of autonomous vehicles. In this research, the usage of RGB image is the main focus in assessing the performance of the pedestrian detection task. An open source code (Packyan, 2019) of single-stage detector of YOLO v3 (Redmon and Farhadi, 2018) was adopted to complete this research as it has a better performance in terms of processing time than Faster R-CNN.

3. METHODOLOGY

3.1 Algorithm

YOLO (Redmon *et al.*, 2016; Redmon and Farhadi, 2017, 2018) was used in this experiment. Different from the aforementioned R-CNN family methods, YOLO is classified as a single-stage detector. In other words, it means classification and regression tasks are run simultaneously.

Three YOLO versions have been published from 2016. The YOLO v1 (Redmon *et al.*, 2016) pioneered the single-stage detector classifier. As depicted in figure 3, YOLO divides an image into S × S grids. Next, each grid detects 2 bounding boxes whose parameters are x, y, width, height, and confidence. The role of x and y in YOLO is similar to that of the anchor in Faster R-CNN. Confidence is needed for declaring whether an object exists at the image by comparing IoU (Intersection over Union) to the ground truth bounding box.

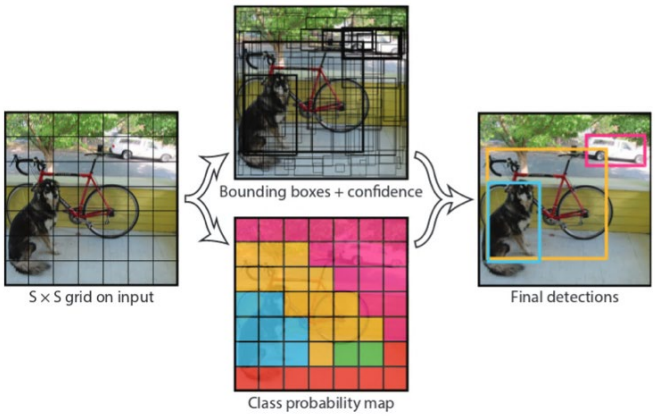


Fig. 3. General Flow of YOLO (Redmon *et al.*, 2016)

YOLO utilizes Darknet (Redmon, 2016), specifically Darknet19 as the network architecture. It is developed by 24 convolutional layers and 2 FC layers. Similarly, 1×1 reduction layers are used to lessen the features space, followed by 3×3 convolutional layers. To calculate the loss, it is detailed in equation (1):

$$
\begin{aligned}
&\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{\omega_i} - \sqrt{\hat{\omega}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
\quad \textbf{\textit{Fehler! Textmarke nicht definiert.}}(1)
$$

where it associates coordinates error, objectness score, and classification error. All notations follow the explanation in the previous paragraph. *C* represents the confidence score, $\mathbb{1}_i^{\text{obj}}$ marks the object exists, and $\mathbb{1}_{ij}^{\text{obj}}$ is extended from the previous notation where *j* is the bounding box predictor in cell *i*. In addition, the symbols with a hat represent the ground truth values, and the symbols without hat are the predicted values.

Following the success of YOLO v1, YOLO v2 (Redmon and Farhadi, 2017) was then published. It tried to improve YOLO v1, where it was still incapable of detecting occlusion objects or many objects aside. Several improvements in YOLO v2 are batch normalization, new architecture, and the new anchor boxes approach. Through batch normalization, it improved the mAP for more than 2%. In addition to that, YOLO v2 used Darknet-19 which consists of 19 convolutional layers and 5 max-pooling layers. The critical improvement was introducing the anchor boxes for classification tasks. The anchor box in YOLO v2 is the center of the bounding box which functions to predict bounding box, similar to the one introduced in Faster R-CNN (Ren *et al.*, 2017).

Finally, YOLO v3 was released in 2018 (Redmon and Farhadi, 2018). Compared to its predecessor, there were no significant changes. First, they used a new architecture called Darknet-53. As its name implies, it has 53 convolutional layers, which makes it deeper than Darknet-19. Additionally, it uses 3×3 sizes with 1×1 layer. Meanwhile, YOLO v3 also improved the performance by scoring the bounding box prediction then applying logistic regression towards the prediction. If the bounding box prediction covers the ground truth object more than any previous bounding box prediction, it is scored as 1. Otherwise, it refuses the prediction. In addition, YOLO v3 implements 3 different scales of predictions. This method was adopted from the Feature Pyramid Networks (FPN) concept (Lin *et al.*, 2017). For every detection, it detects three parts: boundary box, objectness, and 80 class predictions. Afterwards, it upsampled the previous 2 layers, then through several convolutional layers, it predicts a similar tensor. At last, the same method is applied to determine the final result. In this research, YOLO v3 is used for the evaluation of the performance of the pedestrian detection in different light conditions.

3.2 Dataset Preparation

KAIST Multispectral Pedestrian dataset (Hwang *et al.*, 2015) is one of the famous datasets for the evaluation of pedestrian detection in different light conditions. It is a dataset produced by Korea Advanced Institute of Science and Technology in South Korea. It has two types of pictures, one is captured from an RGB camera and the other is captured from an infrared camera. There are 3 places recorded, campus, downtown, and road. Each place has day and night scenarios.

This dataset follows annotation format as used in Caltech Dataset (Dollar *et al.*, 2010). All annotations used pixels format where the object locations are. They are saved in .txt file for each file, where one file is associated with the same filename for both an RGB image and an infrared image. This dataset has labelled three objects: person, people, and cyclist. The label “people” refers to a group of several persons, although there is not any clear explanations in defining a group of several persons. Additionally, if a group of several persons have been labeled as “people”, “person” label is not labeled again on each person object. However, sometimes there are some images which have several “person” label but not categorized as “people” label. In the training dataset, the objects are contained in 14100 images of daytime scenario and in 8058 images of nighttime scenario. In addition, about 2800

daytime images and 1600 nighttime images in the test dataset are used for the evaluation of the performance of the pedestrian detection.

Table 1. Training and Validation Configuration.

Parameter	Value
Epoch	50
Batch Size	10
Weights	Darknet-53.conv.74
Learning Rate	0.001
IoU threshold	0.5
Confidence threshold	0.8
NMS threshold	0.4

3.3 Training and Validation

The PyTorch open source code used in this research was created by Packyan (2019). The configuration was the same as the original YOLO v3. The architecture of the network was modified to adapt to the number of classes. As KAIST Multispectral Pedestrian Dataset has three objects, some convolutional layers follow this filter equation:

$$\text{filter} = [3 * (4+1+\text{number of classes})] \quad (2)$$

where 3 stands for 3 prediction boxes, 4 stands for 4 bounding box offsets, and 1 objectness prediction. From equation 2, it yields 24 filters for specific convolutional layers.

For training configuration, the parameters are explained in table 1. Number of epochs was 50. Batch size was 10. Darknet-53 weights, which is provided by Redmon (Redmon, 2016), were loaded. Learning rate was set to 0.001. In terms of hardware configuration, it was trained using GPU NVIDIA RTX 2070 Super and CPU Intel Xeon E5-1650 v4 3.60GHz.

For validation configuration, mAP was used to evaluate the performance. Additionally, some detected sample images will be shown. The threshold can be seen in detail in table 1. IoU (Intersection over Union) threshold was 0.5. Confidence threshold was set to 0.8, and NMS (Non-Maximum Suppression) threshold was set to 0.4.

4. EXPERIMENTAL RESULTS

After each training epoch, a validation was conducted on the test dataset and all mAPs were collected for each epoch. They have been plotted in figure 4. The solid line denotes the performance in daytime, whereas the dashed line indicates the performance in nighttime. In the daytime training, the detection performance for the "person" was around 45%. As for the "people" and "cyclist" detections, the precision values were relatively low. This occurred due to the problem of class imbalance among "person" label, "people" label, and "cyclist" label, and the training images which include "people" label and "cyclist" label were very few. As a result, this behavior affected the average of all classes mAP. Thus, this paper focuses on the discussion about the detection of the "person." Figure 4 clearly implies the pedestrian detection algorithm performed better in the daytime environment compared to nighttime environment.

Figure 5 shows sample pictures of the ground-truth and the detection result in the daytime environment. Each row shows two pairs of pictures, where the left picture of the pair is the ground-truth and the right picture of the pair is the detection result. In the ground-truth picture, "person", "people", and "cyclist" bounding box are colored green, pink, and yellow respectively. The detection result has a bounding box and a text label for visualization. In fact, the last two images of figure 5's first row present the problem that "people" is incorrectly recognized as several "persons". This also proves to be the main issue in the daytime experiment. In the future, labels will be optimized to overcome this issue.

Figure 6 visualizes the experiment results in nighttime environment. Basically, the pictures are dependent on the environment: darkness and over-exposure. In both situations, "person" could not be detected correctly as shown in the both pairs of the first row of figure 6. Especially, in the over-exposure environment, pedestrian's appearance is blurred with the background. Consequently, it causes misdetection in the over-exposure environment. For the darkness environment, the pedestrian is sometimes not visible the RGB images as presented in the first pair of the first row. In the future, the improvement may focus on darkness and over-exposure.

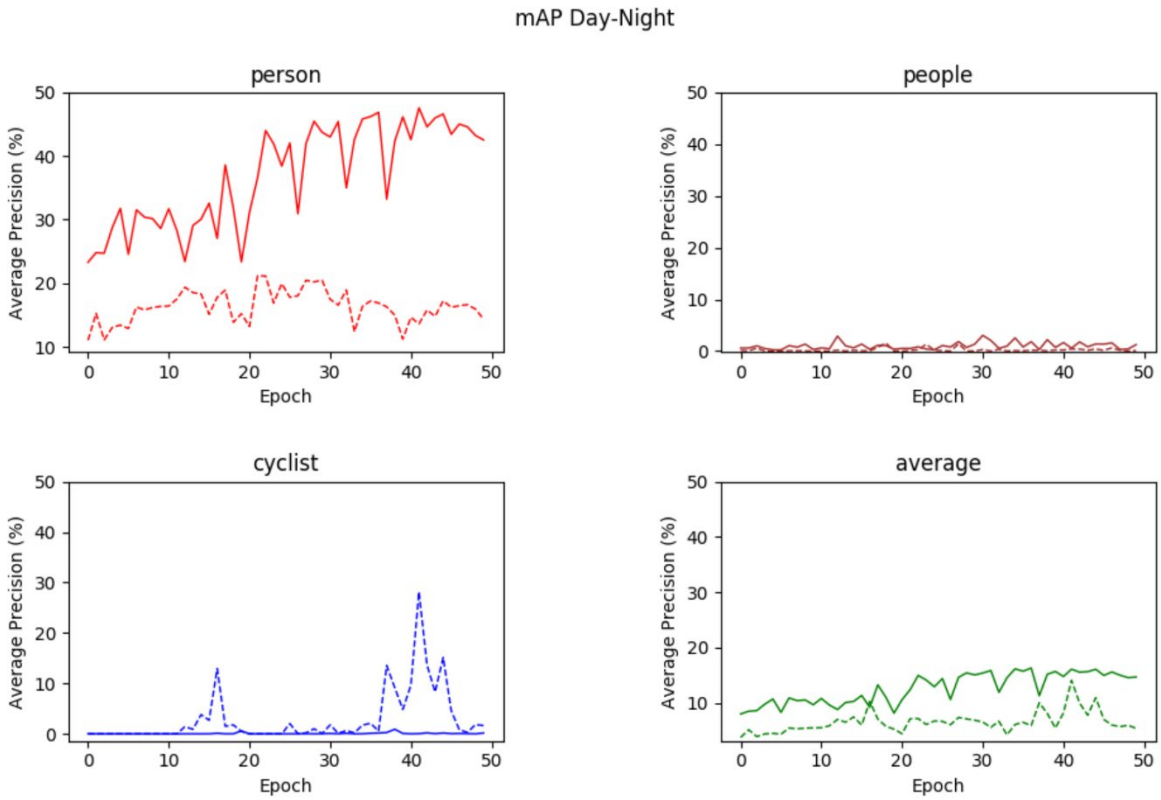


Fig. 4. Mean Average Precision for both daytime (solid line) and nighttime test (dashed line)

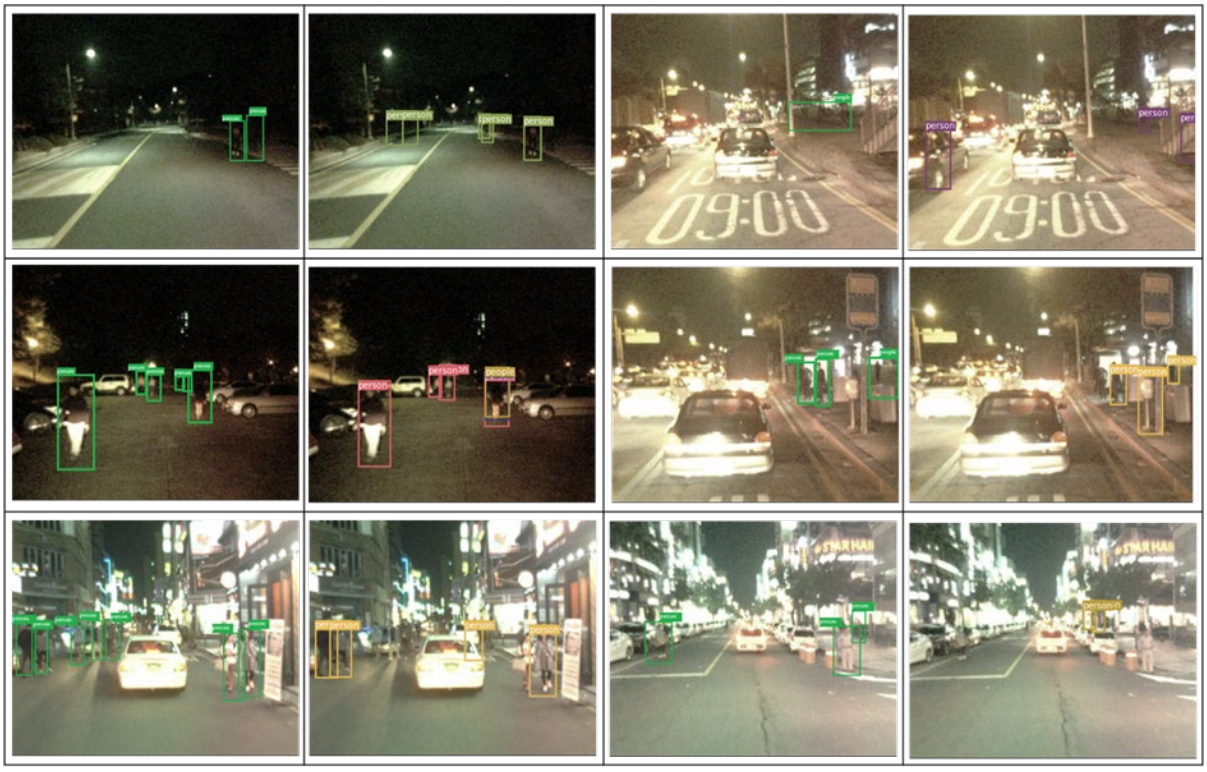
5. CONCLUSION

This paper we have presented a performance comparison of the DNN-based pedestrian detection in different lighting conditions, in order to answer the research question: how much the performance of DNN-based pedestrian detection is affected by the lighting conditions? This research adopted YOLO as the pedestrian detection algorithm and assessed the performance of YOLO on KAIST Multispectral Pedestrian dataset. The experimental results indicated that the performance of DNN-based pedestrian detection was significantly affected by the lighting conditions. In the daytime condition, 45% precision for person detection could be achieved, but only 20% precision was obtained for person detection in the nighttime condition. One reason for the incorrect detection results in the daytime experiment is because of the type of labels in dataset. By comparing the detection results in daytime and nighttime environments, this research found that both darkness and over-exposure could affect the performance of DNN-based pedestrian detection in nighttime environments.

In the future, the infrared camera may be considered to improve the problem caused by darkness, and the brightness suppression and adaption on RGB camera may be studied for reducing the incorrect detection in over-brightness environments. In addition, the re-labeling of the dataset may also be conducted for more accurate evaluation.

Ground truth Detection Ground truth Detection

Fig. 5. Sample pictures of ground truth and detection results at daytime environment



Ground truth Detection Ground truth Detection

Fig. 6. Sample pictures of ground truth and detection results at nighttime environment

REFERENCES

2017 Road Safety Statistics: What Is behind the Figures? (2017). Brussels, Belgium.

Chebrolu, K.N.R., and Kumar, P.N. (2019). Deep learning based pedestrian detection at all light conditions, In: *Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019*, 838–842. https://doi.org/10.1109/ICCSP.2019.8698101

Choi, H., Kim, S., Park, K., and Sohn, K. (2016). Multi-spectral pedestrian detection based on accumulated object proposal with fully convolutional networks, In: *Proceedings - International Conference on Pattern Recognition*. https://doi.org/10.1109/ICPR.2016.7899703

Girshick, R., Donahue, J., Darrell, T., Malik, J., Berkeley, U.C., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 5000. https://doi.org/10.1109/CVPR.2014.81

Girshick, R. (2015). Fast R-CNN, In: *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2015.169

Hou, Y.L., Song, Y., Hao, X., Shen, Y., Qian, M., and Chen, H. (2018). Multispectral pedestrian detection based on deep convolutional neural networks. *Infrared Physics and Technology*, 94, 69–77. https://doi.org/10.1016/j.infrared.2018.08.029

Hwang, S., Park, J., Kim, N., Choi, Y., and Kweon, I.S. (2015). Multispectral pedestrian detection: Benchmark dataset and baseline, In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 1037–1045. https://doi.org/10.1109/CVPR.2015.7298706

Konig, D., Adam, M., Jarvers, C., Layher, G., Neumann, H., and Teutsch, M. (2017). Fully Convolutional Region Proposal Networks for Multispectral Person Detection, In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2017-July*, 243–250. https://doi.org/10.1109/CVPRW.2017.36

Kruthiventi, S.S.S., Sahay, P., and Biswal, R. (2017). Low-light pedestrian detection from RGB images using multi-modal knowledge distillation, In: *2017 IEEE International Conference on Image Processing (ICIP)*, 4207–4211. https://doi.org/10.1109/ICIP.2017.8297075

Lan, W., Dang, J., Wang, Y., and Wang, S. (2018). Pedestrian detection based on yolo network model, In: *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*. https://doi.org/10.1109/ICMA.2018.8484698

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection, In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 936–944. https://doi.org/10.1109/CVPR.2017.106

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science*, *9905 LNCS*, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

Packyan. (2019). PyTorch-Yolov3-kitti. Retrieved from https://github.com/packyan/PyTorch-YOLOv3-kitti

Redmon, J. (2016). Darknet: Open Source Neural Networks in C. Retrieved from http://pjreddie.com/darknet/

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection, In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., and Farhadi, A. (2017). YOLO9000: Better, faster, stronger, In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., and Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. Retrieved from http://arxiv.org/abs/1804.02767

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2016.2577031

Tesla Deaths. (2020). Retrieved January 31, 2020, from https://www.tesladeaths.com/

Tian, Y., Luo, P., Wang, X., and Tang, X. (2015). Deep learning strong parts for pedestrian detection, In: *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2015.221

Tomè, D., Monti, F., Baroffio, L., Bondi, L., Tagliasacchi, M., and Tubaro, S. (2016). Deep Convolutional Neural Networks for pedestrian detection. *Signal Processing: Image Communication*, 47, 482-489. https://doi.org/10.1016/j.image.2016.05.007

Zhang, L., Lin, L., Liang, X., and He, K. (2016). Is faster R-CNN doing well for pedestrian detection? *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-46475-6_28

Zhao, Z. Q., Zheng, P., Xu, S. T., and Wu, X. (2019). Object Detection with Deep Learning: A Review, In: *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2018.2876865