IMMVP: An Efficient Daytime and Nighttime On-Road Object Detector

Cheng-En Wu
Institute of Information Science,
Academia Sinica,
Taipei, Taiwan
chengen@iis.sinica.edu.tw

Yi-Ming Chan
Institute of Information Science,
Academia Sinica,
Taipei, Taiwan
yiming@iis.sinica.edu.tw

Chien-Hung Chen
Institute of Information Science,
Academia Sinica,
Taipei, Taiwan
redsword26@gmail.com

Wen-Cheng Chen
Dept. of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan
dreamfantasy0@gmail.com

Chu-Song Chen
Academia Sinica & MOST Joint Research Center
for AI Technology and All Vista Healthcare
Taipei, Taiwan
song@iis.sinica.edu.tw

***Abstract*—It is hard to detect on-road objects under various lighting conditions. To improve the quality of the classifier, three techniques are used. We define subclasses to separate daytime and nighttime samples. Then we skip similar samples in the training set to prevent overfitting. With the help of the outside training samples, the detection accuracy is also improved. To detect objects in an edge device, Nvidia Jetson TX2 platform, we exert the lightweight model ResNet-18 FPN as the backbone feature extractor. The FPN (Feature Pyramid Network) generates good features for detecting objects over various scales. With Cascade R-CNN technique, the bounding boxes are iteratively refined for better results.**

***Index Terms*—Object detector, deep learning, vehicle detection, pedestrian detection, embedded system.**

I. Introduction

Deep leaning has demonstrates its great success on image classification and object recognition. Recently, various deep convolution neural networks (CNNs) have been proposed for object detection and achieve impressive performance, such as faster R-CNN [1], YOLOv2/v3 [2], [3], and SSD [4]. To further improve the detection accuracy, some promising techniques have been studied. For example, Feature Pyramid Networks [5] exploit the intermediate-level features for detecting small objects without a heavy computational burden. Cascade R-CNN [6] iteratively applies the process of bounding box regression and classification to sequentially refine the object detection results. On the other hand, to reduce the computational resource consumed and improve the inference speed, lightweight models such as MobileNet-v2 [7] and Pelee [8] have been introduced as well.

The purpose of this work is to design a lightweight model that can perform object detection on the road by using an edge-computing device (e.g. NVIDIA Jetson TX2). We target three classes of road object, **pedestrian**, **vehicle**, and **rider**. To fulfill the goal of the MMSP2019 Embedded Deep Learning Object Detection Model Competition (briefed as MMSP Competition below), we have to design an accurate-enough model (with the mAP at least 0.5), whereas the inference speed of the model is expected to be as faster as possible.

Fig. 1. An example of on road objects in the daytime.

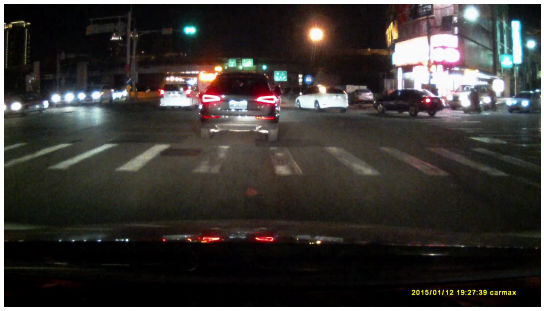

Fig. 2. An example of on road objects in the nighttime.

The rest of this paper is organized as follows. In Section II, we depict the rational of our design and the deep-learning model conducted for efficient object detection on road. In Section III, we present the experimental results on various settings of the object-detection models studied. Finally, conclusions are given in Section IV.

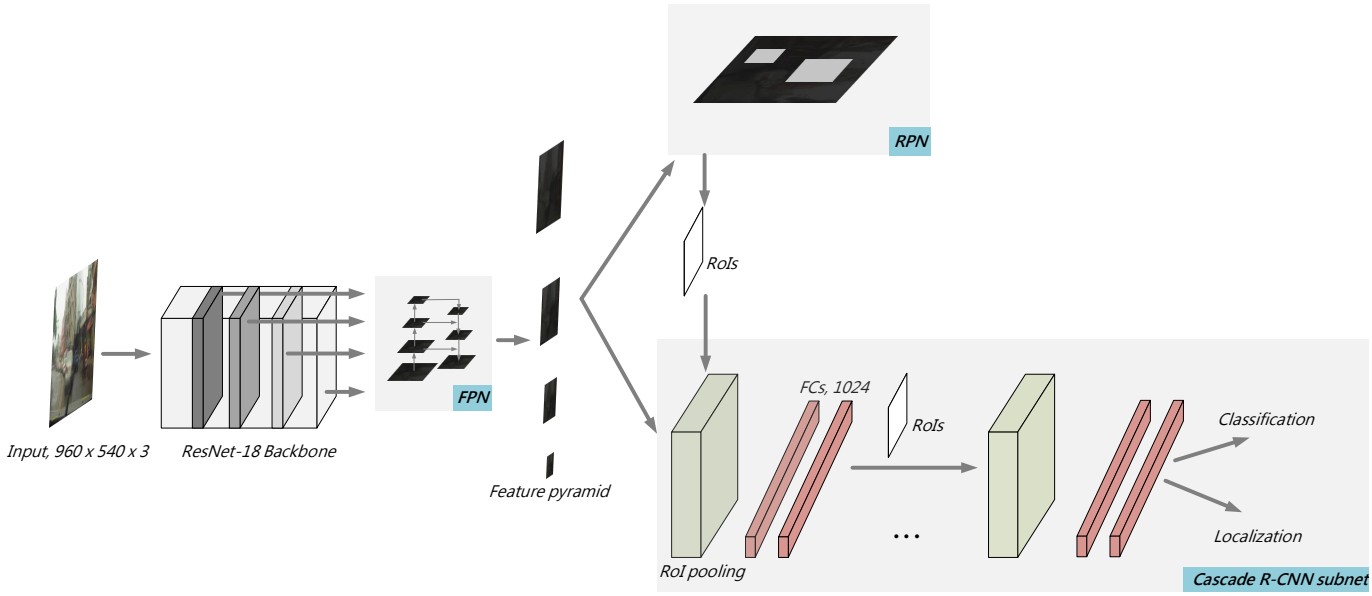

Fig. 3. The overall architecture of the proposed model. To meet the required accuracy of 0.5 mAP with inference efficiency, our model employs 960×540 input resolution. The model is based on Cascade R-CNN [6] but we replace the backbone with ResNet-18 [9] with Feature Pyramid Network (FPN) [5] added.

II. Design Rational and Proposed Object Detector

In this section, we elaborate on the object-detection model designed for this work. Our design compromises the inference speed and accuracy. First, we focus on the development of object-detection models that can achieve a high testing accuracy, as introduced in Section II-A. Then, to improve the inference efficiency, we reduce the model architecture selected above while maintaining the minimal level of accuracy required, as introduced in Section II-B.

A. Design of High-Accuracy Model

Our design strategy is to select the model of the highest accuracy from the existing state-of-the-art ones at first and then improve the efficiency of the model. Among the existing models, Cascade R-CNN [6] with ResNeXt-101 [10] backbone has the best accuracy on MS COCO dataset [11]. To further boost the performance, we add Feature Pyramid Network (FPN) [5] to the backbone of the Cascade R-CNN model so that features at different scales can be extracted better. However, this model has achieved merely the mAP of 0.40 on the 1st-stage public testing dataset announced by MMSP Competition, which is much lower than 0.5 (mandatory criterion of mAP). Hence, to enhance the accuracy, we adopt three strategies for the improvement in the training stage: sample frames selection, label expansion, and training data increments, as depicted below.

1) Sample Frames Selection: We found that there is much redundancy among the training data provided. Removing the redundancy can not only save the training time per trial but also increases the accuracy. Although multiple image sequences are contained in the training data, continuous frames in a sequence have monotonous and repeated information. When the data are directly applied to a typical batch-based learning procedure, the resulted models cannot generate comprehensive feature representations. To address this issue, we perform uniform sampling for each image sequence. We observed that after such pre-processing, a batch of training data owns more representative samples compared to that of utilizing full-sequence images. According to our experiments, the best accuracy of the model is achieved by uniformly sampling one frame from ten in a sequence. The training speed is considerably accelerated too.

2) Labels Extension: The task is to detect three types of objects including pedestrian, vehicle, and rider. The training dataset covers different times of days, including daytime and nighttime. Since the same object has quite different appearances on daytime and nighttime, it is difficult for a model to learn good feature representations of the objects across the times. For example, as shown in Figure 4, the vehicles during the nighttime are only observable via their headlights, causing the appearance to be highly dissimilar to those that are seen during the daytime. To address this issue, we propose a label-extension strategy, where six labels (daytime_vehicle, daytime_pedestrian, daytime_rider, night_vehicle, night_pedestrian and night_rider) are used instead of the original three labels (vehicle, pedestrian, rider) in the training stage. When calculating the accuracy in the training and inference stages, the day and night labels of the same kind of object are merged. That is, the final outcome is still merged into three categories, vehicle, pedestrian, and rider. With the proposed label-extension strategy, the deep-network model can focus on learning effective feature representations that can discriminate the dissimilar daytime and nighttime objects as two separated classes. The accuracy can then be

Fig. 4. An example of the vehicles with glare headlights during the nighttime.

considerably improved in our experience.

3) Training Data Increments: It is a common technique to enhance the accuracy by using more training samples. Accordingly, we extend the training data from an outside dataset, BDD100K [12]. This dataset contains 10 categories including bus, light, sign, person, bike, truck, motor, car, train, rider. We remove the four categories, light, sign, bike and train, and reorganize the remain six into three categories so that they are coincide with the MMSP Competition. To achieve this, we unify bus, truck, and car into the vehicle category, motor and rider into the rider category, and keep person as the pedestrian category, respectively. The training and evaluation sets of BDD100K are combined with MMSP training set for model learning.

B. Efficiency Improvement

After applying the strategies depicted above, the mAP of Cascade R-CNN with ResNeXt-101 backbone is boosted to 0.60 on the first stage public testing dataset, which is much higher than the original mAP of 0.40. However, the inference speed of the model is merely 0.17 FPS (frames per second) on NVIDIA TX2 (embedded computing device), which is non-satisfied yet. To further speed up the inference of the model, several techniques could be used. For example, network quantization and filter pruning are common strategies that can be exploited. Although many methods can be utilized to compress a model, they are time-consuming to training a compact model while keeping the required accuracy. To boost the inference speed in a limited period, we follow two design principles that are easy to be realized. The first is to choose an efficient backbone network and the second is to re-size the input image to a smaller resolution.

1) Backbone Net: In the above, we exploit ResNetx-101 FPN as the backbone network with the input resolution of 1920×1080 pixels. As mentioned, the model ahcieves 0.6 mAP on the public testing dataset, but is computationally expensive and memory intensive. The first improvement is to reduce the complexity of the backbone network under a fixed input size, 1920×1080 pixels. We gradually replace the backbone network with the CNN models that have fewer parameters, but keeps the resulted detector meeting the required accuracy. We conduct our backbone-replacement evaluation on the ResNet series. Our experiments reveal that employing RestNet-18 FPN as the backbone network can still achieve the mAP of 0.56 on the first stage public testing dataset. The inference speed is upgraded to 1.4 FPS.

TABLE I
ARCHITECTURE OF THE RESNET-18 BACKBONE NETWORKS.

Layer	Output Size	ResNet-18
Input	960×540	image
Conv1	480×270	7×7, 64, stride 2
MaxPool	240×135	3×3 max pool, stride 2
Stage1	240×135	3×3, 64 3×3, 64 ×2
Stage2	120×68	3×3, 128 3×3, 128 ×2
Stage3	64×37	3×3, 256 3×3, 256 ×2
Stage4	30×17	3×3, 512 3×3, 512 ×2
Complexity		56 GOPs
Parameters		11M

2) Input Resolution: To additionally improve the inference speed of the above model, we adopt a straightforward strategy: reducing the input image resolution. We re-size the input image to a smaller resolution before sending it to the model. This step takes only a very limited time and thus does not affect the inference speed much. We can then manipulate only a single parameter, the re-scaling size, for speeding up the model. According to our experimental study, the input resolution of 960×540 pixels can stil meet the minimum requirement of accuracy on the first stage public testing dataset, where the mAP on this resolution is 0.53, and the inference speed is boosted to 2.3 FPS.

Note that in the above, the inference speed (in terms of FPS) does not coincide with the officially announced results of MMSP Competiion (TABLE II). It is because that the inference speed presented above does not include the loading time of testing images and deep-learning model from disk.

We have tried to use a lightweight object detector, Pelee [8], in our study too. Pelee is more favorable than existing state-of-the-art computationally efficient models such as ShuffleNet [13] and MobileNet [14]. However, when we evaluate the accuracy of Pelee model on the first stage public testing dataset, only 0.23 mAP is attained, and thus we have not chosen to explore it in this study.

III. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our model on the MMSP Competition dataset. We also conduct ablation studies to evaluate our strategies.

Dataset: We evaluate the performance of our model on the dataset from MMSP Competition. MMSP dataset contains 89,002 annotated 1920×1080 images for training. During the competition, there are three kinds of testing sets including

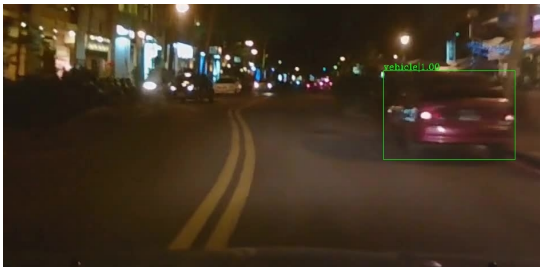

Fig. 5. The qualitative detection results of the model trained with three original labels (pedestrian, vehicle, and rider). The detector fails to recognize vehicles with obscure appearance in this example.

Fig. 6. The qualitative detection results of the model trained with the extended labels (adding another set of labels for objects in the nighttime). More vehicles in the nighttime can then be detected successfully.

1st-stage *public testing dataset* (1,500 full HD test images), 2nd-stage *testing dataset for qualification competition* (4,500 images), and 3rd-stage *private testing dataset for final competition* (3,000 images). The objects in images are annotated with three categories: pedestrian, vehicle, and rider. During the competition, the ground truth labels are not provided. After the competition ends, the annotations of these testing sets are available on the MMSP Competition website. In the following experiments, the accuracy of the model is evaluated on the private testing dataset for final competition.

A. Implementation Details

We implement the model with an open-source object detection toolbox based on PyTorch, mmdetetion [15]. Our model is trained end-to-end with 10 epochs. The learning rate is set to 0.02 at the beginning and then decayed by a factor of 0.1 at 80% of the total iterations. The four subnets of Cascade R-CNN are one RPN and three for detection with IoU @ $\{0.5, 0.6, 0.7\}$, respectively.

B. Results on the Competition Dataset

The officially announced competition results include the evaluation of model size, computation complexity, and speed on NVIDIA Jetson TX2, respectively. Table II shows the final results of the teams that achieve the accuracy greater than 0.43 mAP on final testing dataset. Among them, the inference speed of our model gets 2nd place, while the number of model parameters of our model is the fewest.

TABLE II
FINAL EVALUATION RESULT OF MMSP 2019 EMBEDDED DEEP LEARNING OBJECT DETECTION MODEL COMPETITION. THE SPEED IS EVALUATED BY THE AVERAGE EXECUTION TIME OF THE PROCESS TO DETECT 3,000 TESTING IMAGES (INCLUDING I/O TIME).

Team	mAP (IoU @ 0.5)	Model Size (MByte)	Complexity (GOPS/frame)	Speed (ms/frame)
R.JD	0.538	124	43	460
nctuai	0.476	195	490	1338
chenjiaqi	0.461	114	339	1195
IMMVP (ours)	0.460	**57**	724	510
NPUST-MIS	0.439	238	115	514

Improvement of Labels Extension. As shown in Fig. 5, when the original labels are adopted, the detector is struggled at detecting vehicles in the nighttime because of the strong appearance variations between nighttime and daytime. To address this problem, we extend labels as mentioned in II-A2. After the label extension, the detector can successfully recognize more vehicles in the nighttime. As shown in Fig. 6, a rider and two vehicles on the left side of the image can be detected after the label-extension improvement.

C. Ablation Studies on Temporal and Spatial Resolution

In the following, we explore the affection of sampling rate of frames to the accuracy on the final testing dataset. Then, we show the inference speed and the accuracy of the models with different image resolutions.

Affection of Frame Selection to the Accuracy. To figure out the affection of sampling rate to the accuracy on the final testing image set, we perform an ablation study with different frame selection rates. TABLE III lists the accuracy of Cascade R-CNN with ResNet-18 FPN backbone employing different sample rate. The best accuracy in terms of mAP is achieved when we sample one frame from ten in a sequence.

Affection different input resolution. We evaluate the inference speed of our model with different input resolution on PC (NVIDIA Tesla V100 GPU), and TX2 (embedding device with NVIDIA Pascal CUDA GPU). The results are shown in TABLE IV. When the input resolution is 960×540 and 1920×1080, our model achieves both higher than 0.5 mAP. Real-time detection is achievable on PC (with 24.3 FPS and 38.2 FPS, respectively).

Our mAPs shown above are all evaluated via VOC2012 mAP @ IoU 0.5 with the codes on the github https://github.com/yxlijun/Pelee.Pytorch/tree/master/data. One may note that the testing accuracy achieved by our IMMVP model (Cascade R-CNN w/ResNet-18 FPN shown in Figure 3 with the input resolution 960×540) is 0.507 on TABLE IV, which has a gap over 0.46, the officially announced mAP of our model on the final testing dataset (3,000 images) shown in TABLE II. It could be because that the officially announced mAP is evaluated via MS COCO metric@IoU 0.5 but not VOC2012 mAP @ IoU 0.5 adopted for our ablation study. According to our empirical testing with other datasets,

TABLE III
RESULTS ON THE FINAL TESTING DATASET WITH DIFFERENT SAMPLING RATES OF TRAINING. OUTSIDE DATASET IS USED IN THIS EXPERIMENT.

Model	Resolution	Sampling Rate	mAP_{voc}
Cascade R-CNN w/ ResNet-18 FPN	960×540	1	0.507
		1/10	**0.533**
		1/20	0.512
		1/30	0.511

TABLE IV
RESULTS OF DIFFERENT RESOLUTIONS ON THE FINAL TESTING DATASET. THE INFERENCE SPEED(FPS) ARE EVALUATED ON BOTH PC (NVIDIA TESLA V100 GPU) AND TX2 (NVIDIA PASCAL GPU).

Model	Input	mAP_{voc}	PC	TX2
Cascade R-CNN w/ ResNet-18 FPN	1920×1080	**0.546**	24.3	1.4
	960×540	0.507	38.4	2.3
	640×360	0.422	41.6	2.7
	320×180	0.403	**45.4**	**3.2**

the metrics of VOC2012- and MS COCO-mAP @ IOU 0.5 actually produce different evaluation results. To align the accuracy to the official announcement, we evaluate our model via the on-line testing service of the official website and the results are given as follows. The accuracy of different sampling rates from 1 to 1/30 become 0.466, 0.502, 0.460, and 0.453, respectively, on TABLE III, and that of the different input sizes from 1920×1080 to 320×180 become 0.511, 0.460, 0.377, and 0.341, respectively, on TABLE IV. Nevertheless, all of our ablation studies are evaluated via the VOC2012 mAP @ IoU 0.5 codes mentioned above for a fair comparison.

IV. CONCLUSION

In this paper, we introduce useful strategies to improve the accuracy of the daytime and nighttime on-road object detector. First, we remove the redundant information of the training data via sample selection. We then propose to use label-extension to address the problem of large appearance variations between daytime and nighttime objects. Finally, the training set is expanded with related outside data. To accelerate the inference speed of the model while maintaining accuracy, we choose the ResNet-18 FPN as the backbone of the Cascade R-CNN. As a result, the size of the model is 57 MByte, with the inference speed of 2.3 FPS on Nvidia Jetson TX2. It achieved 0.460 mAP (0.507 mAP by our evaluation) on the MMSP Competition final testing dataset.

ACKNOWLEDGMENT

This work is supported in part by the Ministry of Science and Technology of Taiwan under grants MOST 108-2634-F-001-004.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[2] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[3] ——, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.

[4] 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*. Springer, 2016, pp. 21–37.

[5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[6] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[8] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Advances in Neural Information Processing Systems*, 2018, pp. 1963–1972.

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

[10] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

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

[12] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[13] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[15] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.