Automatic Object Detection in Oil Palm Plantation using a Hybrid Feature Extractor of YOLO-based Model

Mohamad Haniff Junos[1], Anis Salwa Mohd Khairuddin[1*], Muhammad Izhar Kairi[2], Yosri Mohd Siran[2]

[1]Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, 50603 Kuala Lumpur, Malaysia

[2]Department of Processing and Engineering, Sime Darby Plantation Research Sdn. Bhd, 43400 Selangor, Malaysia

[*]Corresponding Author
doi: https://doi.org/10.21467/proceedings.141.8

ABSTRACT

The current manual harvesting process is very laborious and time-consuming. Implementing a machine vision-based automated crop harvesting system may minimize operational costs and increase productivity. This paper aims to develop a one-stage object detection model with high accuracy, lightweight size, and low computing cost. A novel PalmYOLO model is proposed by modifying the architecture of the YOLOv3 tiny model to localize and detect oil palm tree, grabber and Fresh Fruit Bunch (FFB) in varied environmental conditions. The PalmYOLO model employed a lightweight-hybrid feature extractor composed of densely connected neural network and mobile inverted bottleneck module, multi-scale detection architecture, Mish activation function and complete intersection over union loss function. The proposed PalmYOLO model obtained an excellent mAP and F1 score of 97.20% and 0.91. Moreover, the proposed model generated a lower BFLOPS value of 26.732 and a lightweight model size of 46.7 MB. The extensive results demonstrate the PalmYOLO model's ability to accurately detect objects in palm oil plantations.

Keywords: Deep learning, YOLO, crop harvesting system.

1 Introduction

Numerous techniques have been adopted throughout the years to increase the efficiency and productivity of FFB harvesting [1]. Recently, a mechanical tractor with grabber (MTG) has been widely used for in-field FFB collection, contributing to effective evacuation operations [2]. Nonetheless, the use of MTG is still restricted, particularly in large-scale oil palm plantations. Hence, a machine vision-based automated harvesting system is a feasible alternative for reducing reliance on human labour, improving productivity, and lowering production costs [3].

Many researchers have attempted to develop robust algorithms for accurate crop classification and detection [3]. Over the years, the performance of crop detection systems has significantly improved, yet they are still inapplicable in the real world. In addition, the uncertainty and complexity of the environment in the orchards appear to be the primary obstacle to developing a crop detection system. In recent years, numerous crop detection systems have employed a hand-crafted features approach and machine learning classifiers such as support vector machine (SVM) [4,5], fuzzy [6] and artificial neural network (ANN) [7] classifiers. Besides, these classifiers used color, shape and texture information as feature extractors [6–9]. However, these approaches could not resolve the detection in extremely complex conditions. Thus, a state-of-the-art deep learning approach is applied to increase the accuracy of the detection system.

With the advancement of deep learning technology in machine vision applications, object detection based on deep convolutional neural networks (CNNs) has produced the state of the art results [10]. The CNNs

Proceedings DOI: 10.21467/proceedings.141; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-957605-4-1

can automatically extract features from the input image by self-learning. Several studies were conducted adopting CNNs for detection and counting tasks in agriculture [11–13]. Deep learning-based image segmentation approaches have produced good results in the segmentation of crop areas, which are important for crop localization and detection. However, these approaches cannot correctly segment the regions of each target in densely overlapped conditions.

This paper aims to develop an automatic detection system with high accuracy, low computing cost, and lightweight model size in order to address object detection problems in oil palm plantations. A PalmYOLO model is developed by replacing the original backbone of the YOLOv3 tiny with a hybrid network architecture that integrates DenseNet and mobile inverted bottleneck module. This feature extractor is linked to four detection scales to enhance the detection of small objects. Moreover, the Mish activation function and complete intersection over union (CIoU) loss function are adopted to further increase the detection accuracy. The remainder of the paper is organized as follows: Section 2 discusses related works on object detection based on the deep CNN method. Then, Section 3 introduces the data acquisition and describes in detail the proposed PalmYOLO. Next, Section 4 presents and discusses the experimental results. Finally, Section 5 describes the conclusion and future works.

2 Related Works

Two types of deep learning-based object detection methods are candidate region-based and regression-based models. Two-stage detection model generates region proposals in the first stage, followed by feature extraction from these proposals for bounding box and classification regression [14]. Faster R-CNN is state of the art for the two-stage object detection technique and is widely employed for crop detection [15–17]. However, despite its great classification and localization accuracy, the two-stage detection technique is inapplicable for real-time applications due to its poor detection speed.

Conversely, single-stage detectors approach object detection as a simple regression problem that generates class probabilities and multiple bounding boxes simultaneously using the entire image as the input making it faster than the two-stage object detectors [18]. Recent efforts have been focused on implementing a YOLO-based crop detection model. A novel YOLO model was developed based on the architecture of the YOLOv2 and YOLOv3 tiny models to improve speed and accuracy [19]. Moreover, a densely connected neural network (DenseNet) was used to enhance the performance of the YOLOv3 model for apple detection [20]. The developed YOLOv3-dense model combined the original Darknet53 feature extractor with a shallow DenseNet structure comprising two dense blocks with eight dense layers. The improved YOLOv3 model proposed in [19] and [20] has greatly enhanced the detection performance; nevertheless, a longer computing time is needed due to the network complexity. Additionally, a big model size is generated, which is unsuitable to be deployed on embedded devices. The DenseNet was also adopted into the YOLOv3 tiny model to improve the real-time performance, model size and computation costs [21,22]. These advantages are crucial for an automatic crop detection system based on machine vision.

3 Materials and Methods

3.1 Data Acquisition

The images were collected using a 12 MP EKEN H9R Ultra HD camera during harvesting season in the Sime Darby oil palm plantation in Selangor, Malaysia, during sunny and cloudy weather at various periods. In this paper, 5000 images were chosen to develop the palm dataset. Various scenes and challenging conditions were selected to improve the robustness and diversity of the dataset, including varying

Proceedings DOI: 10.21467/proceedings.141
ISBN: 978-81-957605-4-1

Series: AIJR Proceedings
ISSN: 2582-3922

illumination, overlapped, and occluded environments. The images were manually labelled into three classes: palm, grabber, and bunch. 70% of the images were used as train data and 30% as validation data.

3.2 Methodologies

The PalmYOLO model is developed by modifying the network structure of the YOLOv3 tiny model. The model integrates the following components: hybrid feature extractor, multi-scale target detection, Mish function, and CIoU loss function, as shown in Figure 1.

The DenseNet161 architecture was utilized as the main feature extractor replacing the YOLOv3 tiny's original backbone network. DenseNet ensures maximum and strong gradient flow by linking each layer directly to every other layer [23]. As a result, each layer obtains all the feature maps from the previous layers. Furthermore, DenseNet utilizes the network's potential through feature reuse and thus tends to have more diversified features. The network comprises four dense blocks, with 6, 12, 44 and 16 dense layers. Each dense layer consists of 1×1 Conv-BN-Mish with 128 filters and 3×3 Conv-BN-Mish with 32 filters. A growth rate of 32 expands the volume of feature maps in every layer. In addition, there are three transition blocks to downsample the number of feature maps and the volume size. The first block composes 1×1 Conv-BN-Mish with 128 filters and 2×2 max-pooling layers.

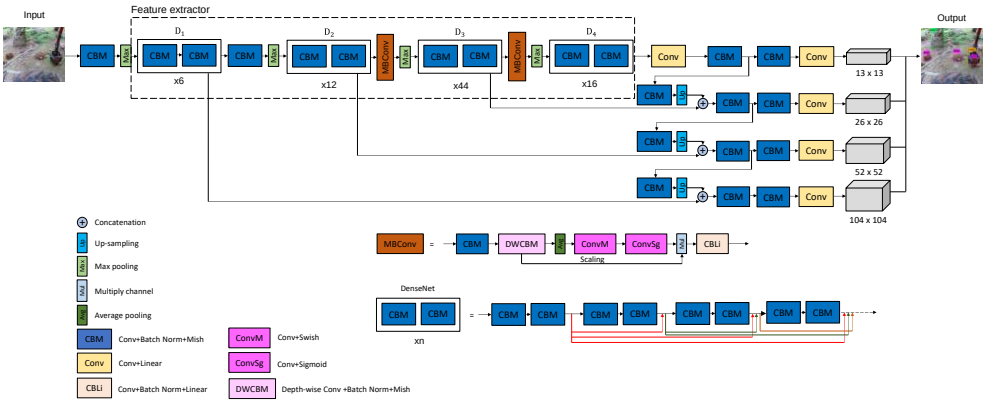

Figure 1: *Schematic architecture of the PalmYOLO model*

The second and third transition blocks integrated a mobile inverted bottleneck module (MBConv) [24] block to minimize the amount of computing in the network. The module performs three separate convolutions that follow a narrow, wide, and narrow approach. The optimized MBConv adopted a squeeze and excitation network (SENet) and the Mish function. The first layer of the first MBConv block widens the network by using a 1×1 Conv-BN-Mish layer with a 256 filter. Later, the number of parameters is reduced by applying a 3×3 DWConv-BN-Mish with a 256 filter. The SENet is then implemented using the average pooling layer to extract the global features from the channel dimension, followed by a 1×1 Conv-Mish with a 24filter size and a 1x1 Conv-Sigmoid with a 256 filter. Next, the final output is scaled with the compressed feature maps. Lastly, the number of channels is downsampled utilizing a 1×1 Conv-BN-Linear layer with a 80 filter. A similar structure is adopted in the second MBConv block but with different sizes of filter. The first and second convolution layers adopted 512-filter size, the SENet layers with 32 and 512 filter, and the last convolution layer with a 112-filter size. Lastly, the feature maps dimension is reduced by using a max-pooling operation.

In order to retrieve the location information of the targets with different sizes, four prediction scale is used using the feature pyramid network (FPN). The FPN merged the feature maps produced by multiple convolution layers in the feature extractor network based on its feature scales. The different feature map

dimensions from multiple layers are efficiently combined by adopting an up-sample operation followed by a 1×1, 3×3, and convolutional layers. As a result, the four prediction scales contain map sizes of 13×13, 26×26, 52×52 and 104×104 for bigger, medium, small, and smallest scales.

Finally, the final prediction feature map is produced using a 1×1 Conv-Linear layer, which is then used by the detection layers to locate and detect the targets. The final prediction is generated as a vector containing the coordinates of the predicted box, confidence score, and class label**.** The CIoU loss function quantifies the error of the overlap area, the distance between centre points, and the aspect ratio between the predicted and truth bounding box.

3.3 Evaluation Metrics

The detection accuracy was evaluated using mean average precision (mAP). mAP defines the average area under the precision (P_r) and recall (R_c) graph at different detection thresholds over all classes (Eq. 1). The P_r and R_c are formulated in Eq. 2 and 3. TP represents the number of correctly detected objects. FN indicates the number of undetected objects, whereas FP denotes the number of wrongly detected objects. The F1 score is the harmonic mean that considers the precision and recall, as shown in Eq. 4. Besides, several metrics were used to evaluate the computational performance. Floating point operations (FLOPs) describes the network's complexity. Then, the size of the model is determined by the number of generated parameters. Finally, the detection speed is measured for both detections on image and video.

$$\mathrm{m}AP = \frac{\sum_1^n \int_0^1 P_r(R_c) dR_c}{n} \tag{1}$$

$$P_r = \frac{TP}{TP + FP} \tag{2}$$

$$R_c = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = \frac{2 \times P_r \times R_c}{P_r + R_c} \tag{4}$$

4 Experimental Results and Analysis

4.1 Experimental Setup

The models were trained using Darknet framework in Windows 8 64-bits operating system, with Intel (R) core (TM) i7-4790 CPU @ 3.6 GHz processor with installed memory of 16 GB RAM and NVIDIA GeForce GTX 750 Ti, graphic card having 2 GB of GDDR5 memory type. The hyperparameters are standardized for a fair comparison. All experiments were trained with an input size of 416×416. The batch size was set to 64 with a subdivision of 32. In addition, a momentum of 0.9 was utilized to adjust network parameters, and a decay weight of 0.0005 was employed to minimize overfitting. The models were trained for 20000 training steps with 0.001 initial learning rate. According to the steps policy, the learning rate will be updated to 0.0001 and 0.00001 at 80% and 90% of training steps.

4.2 Experimental results

Table 1 shows the comparison of the detection results. Notably, the proposed PalmYOLO model achieved a mAP of 97.20 %, which outperformed the other models but was slightly lower than the YOLOv4 model (97.28%). In addition, the model shows mAP increments of 2.53 % over its original YOLOv3 tiny model. This result demonstrates the effectiveness of adopting the hybrid backbone, additional detection layer, Mish

Proceedings DOI: 10.21467/proceedings.141
ISBN: 978-81-957605-4-1

Series: AIJR Proceedings
ISSN: 2582-3922

activation and CIoU loss function. However, despite its excellent detection performance, the YOLOv4 model composes of complex backbone structures, CSPDarknet, which incorporates a cross-stage partial connections network leading to higher computational cost.

The proposed PalmYOLO model produced 73.35 % average IoU, 5.75 % and 2.13 % lower than the YOLOv4 and YOLOv3 models. However, it is considered a decent overlap percentage between ground truth and predicted boxes since the model achieved an excellent mAP value. In addition, the precision and recall values are slightly lower, at 0.88 and 0.94, respectively, yielding a slightly lower F1 score of 0.91 compared to the other two models. The precision-recall relation for all detection models is depicted in Figure 2. Additionally, the proposed model obtained comparable AP values for each class compared to the other models. The highest AP is achieved for the grabber class with 98.38 %. This result implies that the proposed hybrid backbone can successfully extract the deep features of the three classes

Table 1: *Performance comparison detection models*

Method	P_r	R_c	F1 score	Average IoU (%)	AP (%)			mAP (%)
					Bunch	Grabber	Palm	
YOLOv3 tiny	0.91	0.91	0.90	69.89	93.30	97.42	94.01	94.80
YOLOv4 tiny	0.84	0.92	0.88	69.01	96.05	94.24	98.22	96.05
YOLOv3	0.93	0.97	0.94	75.48	96.29	97.56	97.63	97.17
YOLOv4	0.93	0.97	0.94	79.10	95.93	98.35	97.55	97.28
PalmYOLO	0.88	0.94	0.91	73.35	95.80	98.38	97.42	97.20

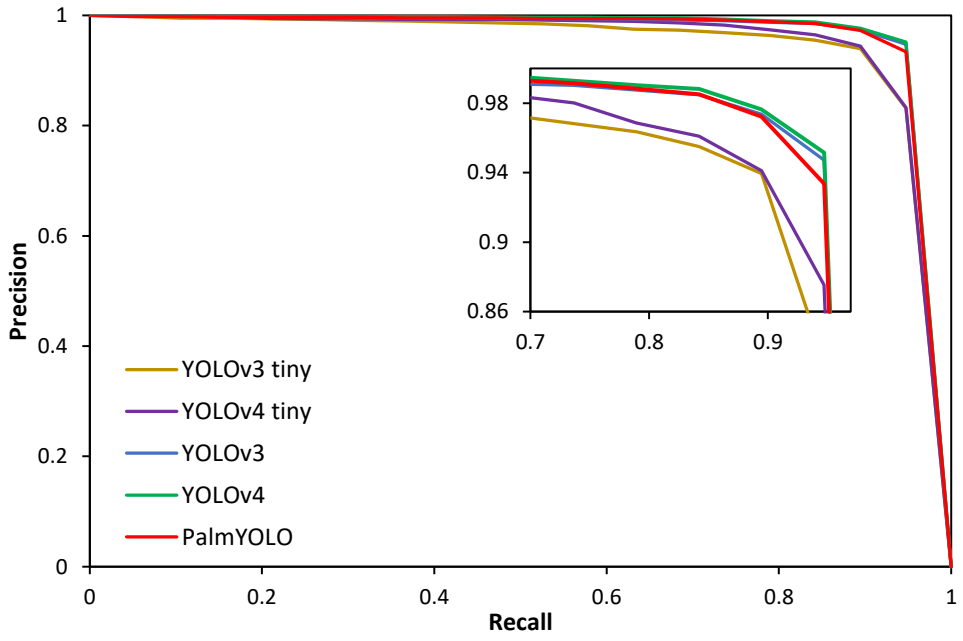

Figure 2: *Comparison of precision-recall graphs*

Figure 3 shows the example of visual detection under different complexities and scenarios. Notably, the proposed PalmYOLO model can distinguish the object in heavily occluded, overlapped, and illumination conditions.

Table 2 shows that the PalmYOLO model produced a small model size of 46.7 MB. The size is reduced by 80.86 % and 80.13 % compared to the YOLOv4 and YOLOv3 models. This result highlights the effectiveness of integrating the DenseNet and MBConv module, which helps develop fewer parameters. Besides, the YOLOv3 and YOLOv4 models obtained considerably bigger model sizes of 235 MB and 244 MB, which were mainly contributed by the highly complex backbone structure that produced a greater network's parameters. The models were also tested on the GeForce GTX 1650 to evaluate the real-time performance. As a result, the proposed model achieved 1.54 and 1.48 times faster inference time than the YOLOv4 and YOLOv3 models. Moreover, the proposed model outperformed the YOLOv4 (21.0 FPS) and YOLOv3 (21.3 FPS) models on the GeForce GTX 1650, achieving 28.8 FPS. In addition, the total BFLOPs generated by the proposed PalmYOLO model is 28.353 BFLOPs which is reduced by 55.18 % and 59.11 % compared to the YOLOv4 (59.585) and YOLOv3 (65.368) models. This suggests that the model is less computationally intensive and executes fewer operations than both models. Overall, the proposed model achieves the optimal balance between detection and computational performance.

Figure 3: *Visual detection of PalmYOLO model*

Table 2: *Comparison of results for computational performance*

Model	Inference time (ms)	FPS	Parameter (Million)	Model size (MB)	BFLOPS
YOLOv3 tiny	5.51	64.4	8.25	33.0	5.465
YOLOv4 tiny	5.97	64.2	5.63	22.5	6.804
YOLOv3	44.46	21.3	58.75	235.0	65.368
YOLOv4	46.32	20.9	61.00	244.0	59.643
PalmYOLO	29.98	28.8	11.68	46.7	26.732

5 Conclusion

This paper proposed a novel single-stage YOLO-based detection model to detect objects at palm oil plantations. The proposed PalmYOLO model modified the architecture of the YOLOv3 tiny model using several significant improvements to improve the detection accuracy. First, a lightweight-hybrid backbone based on DenseNet161 configurations and MBConv module was adopted, replacing the original backbone of the YOLOv3 tiny model. Then, an additional detection layer was added to improve the detection of small objects. Finally, the Mish function and CIoU were used as the activation and loss functions. The proposed model was validated on a novel dataset consisting of the bunch, grabber, and tree classes. The experimental results show that the proposed model achieved satisfactory detection performance with a mAP of 97.20%, F1 score of 0.91, and average IoU of 73.35%. In terms of real-time speed, the proposed model outperformed the state-of-the-art YOLOv4 model with 28.8 FPS when tested on the GeForce GTX 1650. Moreover, the proposed model significantly improves the BFLOPs value and model size with 26.732

Proceedings DOI: 10.21467/proceedings.141
ISBN: 978-81-957605-4-1

Series: AIJR Proceedings
ISSN: 2582-3922

and 46.7 MB. These benefits may greatly lower hardware implementation costs. In conclusion, the results demonstrated that the viability of the proposed PalmYOLO model for object detection in palm oil plantations and its applicability to automatic crop harvesting systems.

6 Declarations

6.1 Acknowledgements

The research funding was provided by Industry-Driven Innovation Grant (IDIG) by Universiti Malaya with project number PPSI-2020-CLUSTER-SD01.

6.2 Competing Interests

There is no conflict of interest.

6.3 Publisher's Note

AIJR remains neutral with regard to jurisdiction claims in published maps and institutional affiliations.

References

[1] S. Abd Rahim, K. Mohd Ramdhan, D. Mohd Solah, Innovation and technologies for oil palm mechanization, in: Furth. Adv. Oil Reserach, 2011: pp. 570–597.
[2] A.R. Shuib, M.R. Khalid, M.S. Deraman, Enhancing field mechanization in oil palm management, Oil Palm Bull. 61 (2010) 1–10.
[3] R. Mairon, Y. Edan, Computer vision for fruit harvesting robots — State of the art and challenges ahead, Int. J. Comput. Vis. Robot. 3 (2012) 4–34.
[4] Y. Song, C.A. Glasbey, G.W. Horgan, G. Polder, J.A. Dieleman, G.W.A.M. van der Heijden, Automatic fruit recognition and counting from multiple images, Biosyst. Eng. 118 (2014) 203–215.
[5] S. Sengupta, W. Suk, Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions, Biosyst. Eng. 117 (2014) 51–61.
[6] R. Hamza, M. Chtourou, Design of fuzzy inference system for apple ripeness estimation using gradient method, IET Image Process. 14 (2020) 561–569.
[7] F. Kurtulmus, W.S. Lee, A. Vardar, Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network, Precis. Agric. 15 (2014) 57–79.
[8] W. Maldonado, J.C. Barbosa, Automatic green fruit counting in orange trees using digital images, Comput. Electron. Agric. 127 (2016) 572–581.
[9] S. Kaur, S. Pandey, S. Goel, Semi-automatic leaf disease detection and classification system for soybean culture, IET Image Process. 12 (2018) 1038–1048.
[10] A. Koirala, K.B. Walsh, Z. Wang, C. McCarthy, Deep learning – Method overview and review of use for fruit detection and yield estimation, Comput. Electron. Agric. 162 (2019) 219–234.
[11] S.W. Chen, S.S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C.J. Taylor, V. Kumar, Counting apples and oranges with deep learning : A data driven approach, IEEE Robot. Autom. Lett. 2 (2017) 781–788.
[12] M. Dyrmann, R.N. Jørgensen, H.S. Midtiby, RoboWeedSupport - Detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network, Adv. Anim. Precis. Agric. 8 (2017) 842–847.
[13] P.A. Dias, A. Tabb, H. Medeiros, Apple flower detection using deep convolutional networks, Comput. Ind. 99 (2018) 17–28.
[14] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN : Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2017) 1–14.
[15] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, C. McCool, Deepfruits: A fruit detection system using deep neural networks, Sensors. 16 (2016).
[16] S. Bargoti, J. Underwood, Deep fruit detection in orchards, in: 2017 IEEE Int. Conf. Robot. Autom., IEEE, 2017: pp. 3626–3633.
[17] S. Madeleine, S. Bargoti, J. Underwood, Image based mango fruit detection, localization and yield estimation using multiple view geometry, Sensors. (2016).
[18] T.T. Le, C.Y. Lin, E.J. Piedad, Deep learning for noninvasive classification of clustered horticultural crops – A case for banana fruit tiers, Postharvest Biol. Technol. 156 (2019) 110922.
[19] A. Koirala, K.B. Walsh, Z. Wang, C. McCarthy, Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO,' Precis. Agric. 20 (2019) 1107–1135.
[20] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, Z. Liang, Apple detection during different growth stages in orchards using the improved YOLO-V3 model, Comput. Electron. Agric. 157 (2019) 417–426.
[21] M.H. Junos, A.S. Mohd Khairuddin, S. Thannirmalai, M. Dahari, An optimized YOLO-based object detection model for crop harvesting system, IET Image Process. (2021) 1–14.

[22] M.H. Junos, A.S. Mohd Khairuddin, S. Thannirmalai, M. Dahari, Automatic detection of oil palm fruits from UAV images using an improved YOLO model, Vis. Comput. (2021) 1–15.

[23] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 30th IEEE Conf. Comput. Vis. Pattern Recognit., 2017: pp. 2261–2269.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018: pp. 4510–4520.

Proceedings DOI: 10.21467/proceedings.141
ISBN: 978-81-957605-4-1

Series: AIJR Proceedings
ISSN: 2582-3922