Fast Person Detector with Efficient Multi-level Contextual Block for Supporting Assistive Robot

Muhamad Dwisnanto Putro, Duy-Linh Nguyen, Adri Priadana, and Kang-Hyun Jo
Department of Electrical, Electronic, and Computer Engineering, University of Ulsan
Ulsan, Korea
Email: dputro@mail.ulsan.ac.kr; ndlinh301@mail.ulsan.ac.kr; adripriadana3202@gmail.com; acejo@ulsan.ac.kr

***Abstract*—The robotic demand a vision method to work in real-time on embedded devices. Besides, an assistive robot requires person detection, which is widely used to help automatically interact with the user. This work presents a fast real-time person detection (Fast-PdNet) to localize human areas implemented on a Jetson Nano. This device has been commonly used as an embedded system and is suitable for synchronizing sensors and actuators. The proposed architecture contains layers of Convolutional Neural Network consisting of two main modules: backbone and detection. An efficient extractor module with a multi-level contextual block is employed to extract the spatial features quickly. It avoids high-cost computing to distinguish interest features of the human body and background features. The lightweight learning attention selects suspected specific features area without generating excessive parameters. The end-to-end training was conducted on MS COCO 2017 to generate efficiently weighted models. The Fast-PdNet achieves competitive performance with other light detectors evaluated on the MS COCO 2017, PASCAL VOC 2007, and 2012 datasets. Moreover, this detector can run 35 frames per second when working in real-time on Jetson Nano.**

***Index Terms*—Person detector, efficient multi-level contextual, jetson nano, real-time.**

I. Introduction

Nowadays, robotics have spread rapidly in society globally to improve the welfare of humankind. Instead of only using in industrial areas, robots have been applied in residents, offices, streets, schools, shops, department stores, and public areas [1]. They tend to be assigned to difficult and dangerous tasks. Even specific robots can help with human tasks, one of which is an assistive robot. This robot is employed in public areas to serve humans [2]. Besides, interaction activities always occur between robots and users. Human-robot interaction has been present and widely implemented to prevent misunderstanding of user actions. Therefore, a computer vision method is needed for a robot to detect and recognize humans. It is the initial process of a robot to generate the perception of its interaction with the user [3]. The primary motivation of computer vision in the robotics field is to obtain a perception level that is as close as possible to the human visual system. As an essential part of visual perception, computer vision in robotics is mainly used for object detection and recognition. More specifically, to support interaction activities between robots and users, pose estimation, human action recognition, and person identification require this method as the beginning of the step to precisely localize the human body area.

The human body contains distinctive features, and it is easily distinguished from the background using the human eye. The computer vision method adopted this visual technique to localize the person area in an image. Several works have introduced computer vision methods to identify these essential features [4]–[6]. Blair et al. [4] have presented a pedestrian detection approach using a histogram of oriented gradients (HOG) implemented in an embedded system. This method explores a combination platform with multiple heterogeneous accelerators to investigate the trade-off characteristics and performance. Other work also utilizes HOG as a features extractor that quickly separates important features [5]. This study applied SVM (Support Vector Machine) to classify human body features from raw features generated by HOG. Logic Inference is employed sequentially to combine the elements selected in the final classifier. The short execution time showed that these studies have detectors that can operate quickly by avoiding over-computation costs. However, conventional methods are weak to identify partially occluded persons and extreme positions in low-level brightness.

The deep learning approach has shown excellent results in discriminating against specific features and backgrounds [7]. Convolutional Neural Network (CNN) has been implemented by benchmark framework to solve person detection tasks [7]–[11]. They are supported by the anchor method, which precisely predicts the location of small objects. Faster-RCNN [7] applied a two-stage approach to predict the Region of Interest (RoI), identify the class of objects, and refine their location. Meanwhile, YOLOV3 [8], YOLOV4 [9], and YOLOV5 [10] explored the feature pyramid network approach to fuse features with different frequencies. SSD (Single Shot Multibox Detector) [11] predicts object localization and classification in a single forward pass of the network. Although the frameworks have implemented several mobile and tiny versions to reduce the number of parameters, the detectors still have problems running smoothly on edge devices. The previous architecture employed deep layers to discriminate human features. Therefore, they tend slowly operate at real-time processing speed on inexpensive hardware. This weakness hinders the reliability of a vision method in the practical application aspect.

A robot needs a vision method that can work in real-time

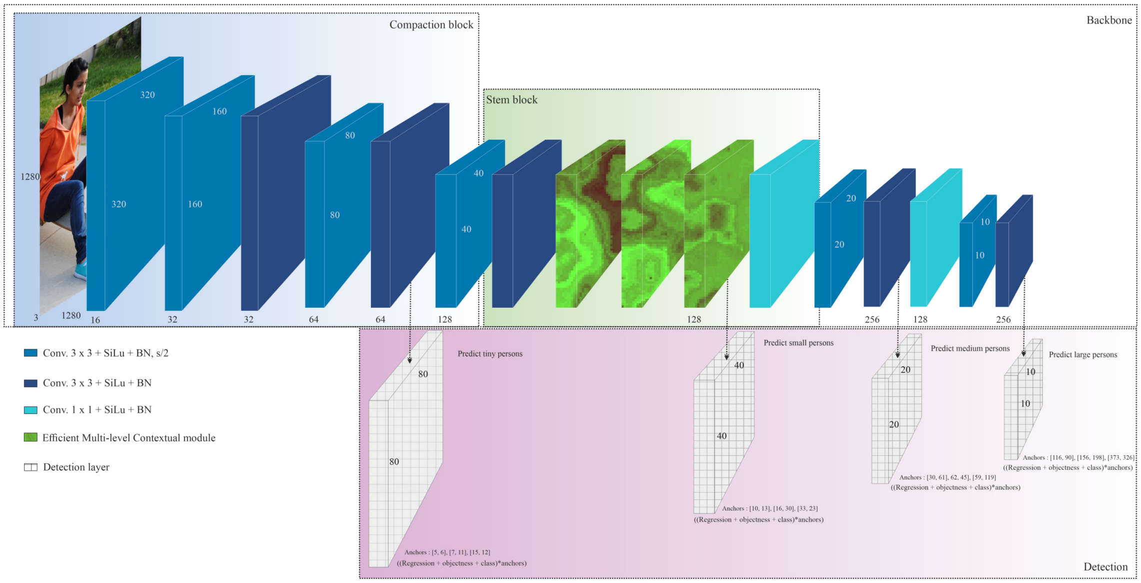

Compaction block
Backbone
Stem block
1280
3
1280
320
320
160
160
80
80
40
40
20
20
10
10
16
32
32
64
64
128
128
256
128
256
Conv. 3 x 3 + SiLu + BN, s/2
Conv. 3 x 3 + SiLu + BN
Conv. 1 x 1 + SiLu + BN
Efficient Multi-level Contextual module
Detection layer
Predict tiny persons
80
80
Anchors : [5, 6], [7, 11], [15, 12]
((Regression + objectness + class)*anchors)
Predict small persons
40
40
Anchors : [10, 13], [16, 30], [33, 23]
((Regression + objectness + class)*anchors)
Predict medium persons
20
20
Anchors : [30, 61], 62, 45], [59, 119]
((Regression + objectness + class)*anchors)
Predict large persons
10
10
Anchors : [116, 90], [156, 198], [373, 326]
((Regression + objectness + class)*anchors)
Detection

Fig. 1. The Fast-PdNet architecture. An efficient backbone module is consists of compaction and stem module to sequential extract the human body features. A hierarchy detection layer is applied on four feature maps to predict person bounding boxes of varying sizes. Best viewed in color.

to acquire a response from the user [12]. The information from the object must be received directly without suspension. This issue can be overcome by minimizing the algorithm computation to reduce the delay time. On the other hand, robotics uses low-cost devices to process the input and output data [13]. Embedded hardware has been commonly used as the main processor of robots to decide actions from physical sensor information and synchronize them. Jetson Nano is an embedded system used to perform computing at the edge of the system included by a low-cost accelerator [14]. Therefore, a person detector must be required to work smoothly on this device that encourages a robot's ability. It increases the capability of the vision method to be implemented in practical applications. This work presents a new real-time detector that can efficiently detect a person's area without compromising performance.

A Fast person detection (Fast-PdNet) proposes a light architecture that offers a multi-level contextual block to discriminate against specific features of the human body. This network avoids the computational overhead and produces fewer weighted parameters than standard detectors. So the slim structure produces a lightweight detector that can speedily operate on edge devices. Based on this description, the main contributions of the study are summarized as follows:

1) A novel fast person detection (fast-PdNet) is offered to efficiently find the location of multiple humans using hierarchy features detection. It can run smoothly on a Jetson Nano that is suitable to implement for assistive robots.
2) An efficient multi-level contextual module is proposed to quickly distinguish person features by assigning gradual attention to improving the detector's accuracy. The performance result achieves competitive accuracy with other architectures on MS COCO 2017 [15] and PASCAL VOC [16].

II. Proposed Architecture

The proposed architecture employs a series of convolution layers consisting of two main modules, as shown in Fig. 1. A backbone efficiently extracts person features by applying compaction and stem blocks sequentially. Furthermore, hierarchy features detection is applied to four layers to serve various object sizes by adjusting them to anchor sizes.

A. The Backbone

The backbone module extracts interest features by discriminating components assumed as target objects from trivial features. The convolutional operation employs a weighted kernel at each input pixel to produce spatial extracted features affected by neighboring source pixels. The CNN architecture generally works to reduce the size of the feature map incrementally for saving computing from subsequent layers. The number of channels from each layer will increase, which provides rich information about the object's features. The compaction block employs 3 × 3 convolution with two strides to shrink the feature map. It is more robust than the pooling layer [17]. It consists of four stages which reduce 32 times of the input image to produce a feature map of 40 × 40 with 128 channels at the end of the block. In addition, 3 × 3 convolution with a stride of one was employed in the third and fourth stages to improve the quality of the low-level features. In order to overcome the gradient problem, it applies ReLU (Rectified Linear Unit) activation and Batch Normalization after convolution operations. The compaction block is assigned to rapidly reduce the feature map in stages while generating mid-level features that contain elements of the human body.

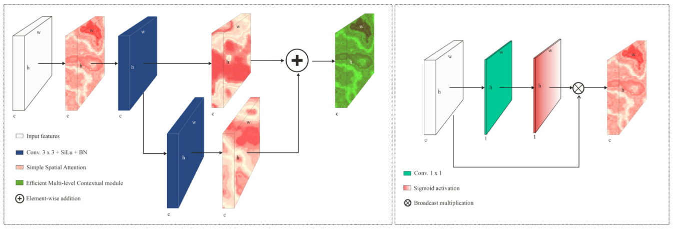


Fig. 2. An efficient multi-level contextual block (left side) consist of 3 × 3 convolution and simple spatial attention (right side). Best viewed in color.

The series block provides efficient computing that supports the ability of the detector to operate quickly.

The Fast-PdNet proposes the central feature extractor as a stem block to comprehensively filter human-specific features from unimportant features. This block offers an efficient multi-level contextual module to help the detector operate quickly without compressing its performance. Fig. 2 shows this module applies two 3 × 3 convolutions at different frequency levels. It generates multiple receptive fields, thereby providing a variety of information. Additionally, contextual blocks are inserted before and after the CNN layer to enhance specific features. This combination block is described as:

$$C_i = S_2(W_1[S_1(x_i)]) + S_3(W_2[W_1[S_1(x_i)]]), \tag{1}$$

where x_i is the input features of each cell pixel $i-th$, W is the weighted kernel for convolution operations, while S is a simple spatial attention module illustrated as follows.

$$S(z_i) = z_i \otimes \sigma(W[z_i]). \tag{2}$$

An attention module affirms the important elements of the features map (z_i) from the previous process. It uses 1 × 1 convolution to generate a single layer, then sigmoid activation (σ) establishes the probability score of each pixel. Updating features assert quality improvement by applying the broadcast multiplication $(\otimes)$ operation to each input cell. It encourages a better feature map output, which improves the intensity value of each input feature. A low probability will reduce the intensity of the input features, while a weight with a high score enhances the input features and describes the component as an interset feature. On the other hand, the multi-update system promotes improvement in extraction quality by reducing the intensity of trivial features to a low score. Therefore, only human body features are present to ensure the vital information supports the prediction system for precise and accurate results. In addition, simple attention does not produce a significant number of parameters and computations. It is due to using a single channel kernel that only performs one filter-based operation. Furthermore, a transition block is applied to produce feature maps of different sizes at high-level frequencies. It generates feature maps of 20 × 20 and 10 × 10 for predicting medium and large-sized persons. Sequential convolution represents the extracted features by applying 1 × 1 and 3 × 3. This combination is more efficient than vanilla convolution.

B. Hierarchical Features Detection Module

Object detectors require the detection layer to predict the person area at the network's end. Generally, the locations of suspected persons are marked with bounding boxes. The prediction features map estimates the coordinates and dimensions (x, y, h, w). Besides, the network also provides class probabilities for each object (person and none). Fast-PdNet implements a Hierarchical pyramid network to generate different map features. The variety of map sizes increases the ability of the adjustment process at the anchor assignment. Instead of using a pyramid features network [8], it implements a more efficient structure to encourage detectors to work faster. Four detection stages are used to predict tiny, small, medium, and large person. Anchors are employed as initial bounding boxes, making it easier for the network to adjust and fit the size. Each stage employs three anchors. Fig. 1 shows that each anchor level is adapted to the size of the feature map to predict different person scales. The scale-based assignment employs a map of 80 for predicting tiny people, 40 for small, 20 for medium, and 10 for large scale. The transition model helps the detector produce medium and small feature map sizes. So, the detector can explore the ability to detect full-body human objects at multi-scale. Additionally, this strategy also enhances the network performance.

C. Multi-box Loss Function

The CNN detector requires a feedback process to evaluate the detection and update the kernel weights to produce accurate predictions. Therefore, the loss function is used to measure the inaccuracy of a prediction compared to the ground truth box. It also predicts the presence of an object (objectness) and the probability of class (pedestrian and none). Firstly, the predicted localization (L_{coord}) was evaluated by employing a complete IoU loss [18]. It assesses the difference of intersection, center point distance, and aspect ratio of the predicted box and ground truth. Then, to evaluate the presence of an object in each cell (L_{obj}), it applies a confidence loss [9]. Meanwhile, the binary cross-entropy [9] is utilized to measure the error of the probability of the predicted class (L_{cls}). Multi-box loss is applied to each cell g-th and anchor a-th. It accumulates all these losses is expressed as follows:

$$\begin{aligned} L_{MB} = \lambda_{coord} \sum_{g=1}^{G^2} \sum_{a=1}^{A} \mathbf{g}_{ga}^{obj} L_{coord} + \lambda_{obj} \sum_{g=1}^{G^2} \sum_{a=1}^{A} \mathbf{1}_{ga}^{obj} L_{obj} \\ + \lambda_{cls} \sum_{g=1}^{G^2} \sum_{a=1}^{A} \mathbf{1}_{ga}^{obj} L_{cls}, \end{aligned} \tag{3}$$

where A is the number of anchors, G^2 is cell area. λ_{coord}, λ_{obj}, and λ_{cls} are balancing parameters in regression, objectness, and classification loss, respectively. 1_{ga}^{obj} is equal to one if there is an object in the grid and 0 otherwise.

III. Implementation Setup

The model was trained using GTX 1080Ti as a GPU accelerator on MS COCO 2017 that contains 118,287 images.

Then, it was tested on Intel Core I5-6600 CPU @ 3.30GHz, 32GB RAM. The training dataset provides complex instances, so it can help the model to learn various conditions. The proposed detector applies several augmentations: random color distortion, crop, vertical and horizontal flipping to enrich the variety of data. Then, a mosaic frame with 1280 × 1280 is generated in the last process to help the model learn a variety of object scales. In the training stage, it uses the initial learning rates of 10^{-2} and updates by $2 \cdot 10^{-1}$ in the final OneCycleLR learning rate. To optimize the updating weight, it applies Stochastic Gradient Descent with weight decay of $5 \cdot 10^{-4}$, and the momentum is 0.937. The entire images on the dataset are inserted in 32 mini-batches. In addition, it sets 0.5 as IoU (Intersection over Union) threshold to establish the best bounding box with the highest confidence score. The whole structure of Fast-PdNet was implemented on the PyTorch framework. Implementation in real-world scenarios was conducted indoors under different lighting conditions and at nighttime.

TABLE I
EVALUATION RESULTS ON MS COCO 2017, PASCAL VOC 2007 AND PASCAL VOC 2012 DATASETS.

Evaluation on MS COCO 2017		
Model	**Average Precision (%)**	**Backbone**
ResNet18 (0.25)	40.1	ResNet18
ShuffleNetv2(0.5)	33.8	ShuffleNetv2
MobileNetV2(0.33)	39.1	MobileNetV2
PeleeNet(0.5) [12]	41.9	PeleeNet
Tiny model-Bai et al [12]	42.9	Manually-designed
Fast-PdNet	**41.6**	**Manually-designed**
Evaluation on PASCAL VOC 2007		
Improved Faster R-CNN [7]	75.65	VGG16
TinyYOLOV2	63.88	Darknet19
Tiny-YOLOV3	68.54	Darknet19
Improved Tiny-YOLOv3 [8]	73.98	Darknet19
Enhanced Tiny-YOLOv3 [8]	78.64	Darknet19
YOLOV5-nano [13]	86.2	CSP Bottleneck
YOLOV5-small [13]	88.8	CSP Bottleneck
Fast-PdNet	**82.7**	**Manually-designed**
Evaluation on PASCAL VOC 2012		
Faster R-CNN	62.9	VGG16
SSD512 [11]	39.4	VGG16
RefinedDet320	58.5	VGG16
RefinedDet320+	61.6	VGG16
RefinedDet512	63.6	VGG16
RefinedDet512+	66	VGG16
RFBNet300	29	VGG16
RFBNet512-E	32.6	VGG16
RFBMobileNet	23.8	MobileNet
RetinaNet	60.7	ResNet-50
YOLO-AF-MS [19]	77.3	CSPDarkNet53
YOLOV5-nano [13]	86.6	CSP Bottleneck
YOLOV5-small [13]	88.9	CSP Bottleneck
Fast-PdNet	**83.6**	**Manually-designed**

IV. EXPERIMENTAL RESULTS

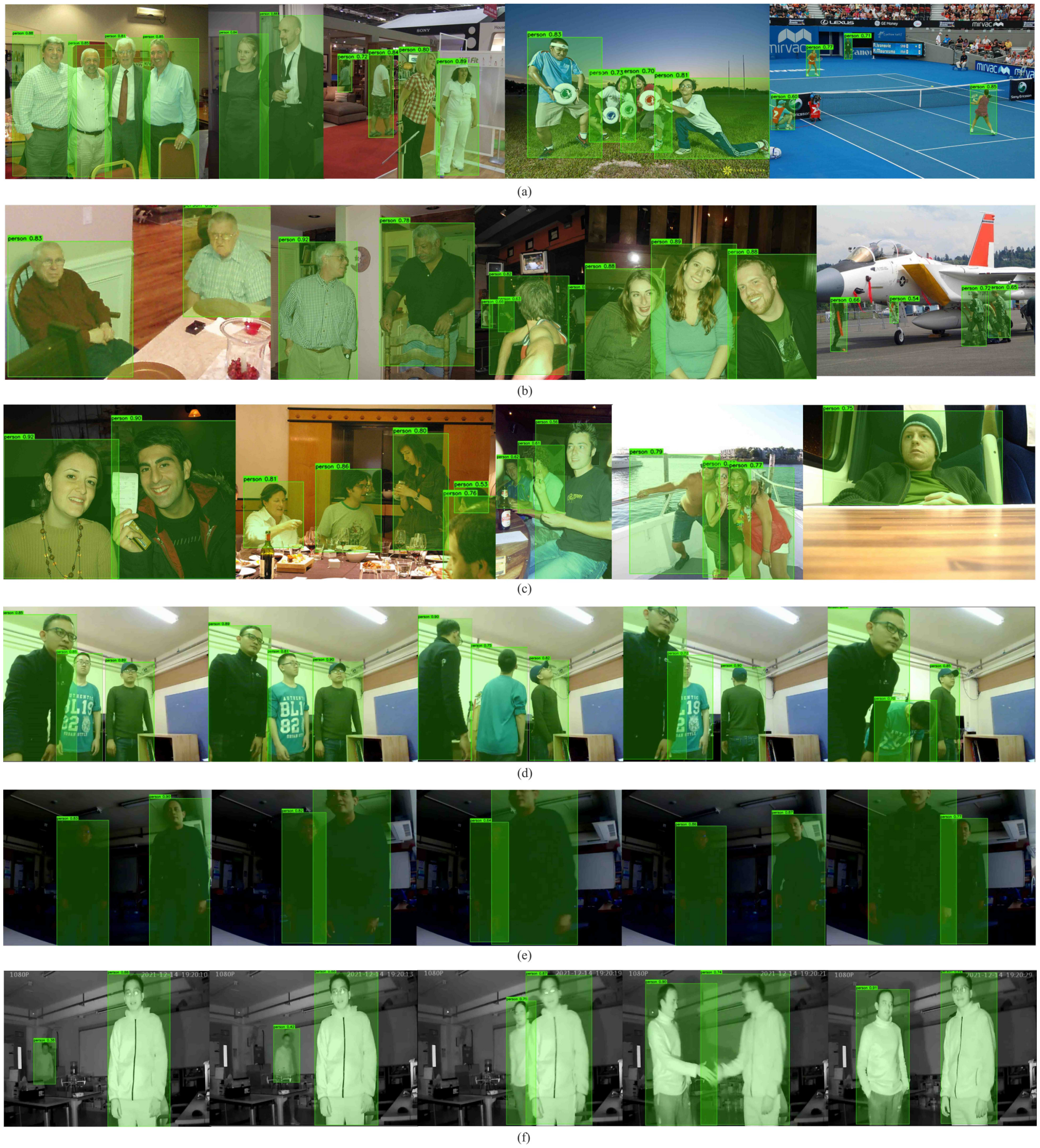
This section examines the proposed architecture on a benchmark consisting of MS COCO 2017, PASCAL VOC 2007, and PASCAL VOC 2012 datasets. It also evaluates the efficiency of the detector tested on a Jetson Nano and compares it to other competitors.

A. *Evaluation on Datasets*

1) MS COCO 2017: The proposed detector is evaluated in MS COCO 2017 to test the performance of the person detector on a wide variety of data. The dataset consists of 122,218 labeled images and 80 object classes containing many objects with complex backgrounds. It also includes many challenges with different poses, the object scale, and the occlusions. The Fast-PdNet uses 64,115 human images for the training processing, and 2,693 were used for testing. The image with person class is used as a knowledge detector to learn and evaluate the specific features of the human body. In the evaluation stage, it applies Average Precision (AP) with the primary metric (IoU=.50:.05:.95) to measure the accuracy of the bounding box prediction. As a result, Fast-PdNet achieved 41.60% AP, outperforming detectors with light backbone Resnet18, ShuffleNet, and MobileNet. In addition, the proposed detector was below the performance of PeleeNet and Bai et al., which differed by 0.3 and 1.3, respectively. Although the detectors show weakness in detecting tiny persons, Fast-PdNet is assigned to work more efficiently in person detection to support service robot systems. Fig. 3 (a) shows the detector's qualitative results, which can localize multiple person areas in conditions such as occluded human body parts and varying backgrounds.

2) PASCAL VOC 2007: The PASCAL VOC benchmarks are contained 20 object classes that consist as follows: Person, Bird, Cat, Cow, Dog, Horse, Sheep, Airplane, Bicycle, Boat, Bus, Car, Motorbike, Train, Bottle, Chair, Dining table, Potted plant, Sofa, and TV/Monitor. The PASCAL VOC 2007 has total images is 9,963 with different backgrounds, human postures, scale, and occlusion. It also provides various illuminations of each object. The proposed detector uses 2,007 images labeled as a person to evaluate its performance with considered a predicted bounding box is to be true detection if it has an IoU 0.5 with a ground-truth annotation. As a qualitative result, Fast-PdNet can detect persons for various challenge datasets, even with different lighting intensity frequencies, as shown in Fig. 3 (b). In addition, Table I shows that PdNet achieves an AP of 82.70% that is 3.5% different from the YOLOV5 small version. In contrast, it outperforms Tiny-YOLOV2, Tiny-YOLOV3, and their improvements.

3) PASCAL VOC 2012: This dataset is extended from the 2007 version containing 11,540 images with 20 classes. The proposed detector uses 2,093 images with a ground truth label to evaluate the model. This configuration is the same as PASCAL VOC 2007 that only uses person class to examine the proposed detector. Fig. 3 (c) shows that the proposed detector can detect a person with occluded objects and various scales that are optimized for robot needs. In comparison, the quantitative results show that this achieves an AP of 83.60% in this evaluation dataset. It is under the accuracy of 5.3% of YOLOV5 small architecture.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3. The prediction results of the Fast-PdNet detector on the MS COCO 2017 (a), PASCAL VOC 2007 (b), PASCAL VOC 2012 (c), Color video on VGA-resolution (d), low-illuminance level on VGA-resolution (d), and infrared video on VGA-resolution (e).

B. Runtime Efficiency

The Fast-PdNet focus quickly localizes the person area to support assistive robot performance. This robot is assigned to provide services to public users by interacting with humans at close range. Therefore, the detector is designed to have a high performance for medium and large objects and optimize the speed. Fig. 3 (d) shows the precise results to detect persons on these scales. The real-case testing can localize multiple occluded objects. In addition, the detector is also reliable in working in low illuminance conditions, as illustrated in Fig. 3(e). The model can carefully learn human body features at low brightness and contrast intensities and distinguish trivial features reliably. On the other hand, the robot is also required to work all the time, so it must be able to detect people at night. Fig. 3 (f) shows that the robot can recognize person features and localize them in bounding boxes on infrared video. This reliability represents that Fast-PdNet is suitable for assistive robots to support human-robot interaction.

The detector's efficiency also increases the model's ability to be implemented in practical applications. The proposed detector generates 2,218,545 parameters with 1.4 GFLOPS. These results describe that the light model uses an efficient number of kernels and computations. It encourages Fast-

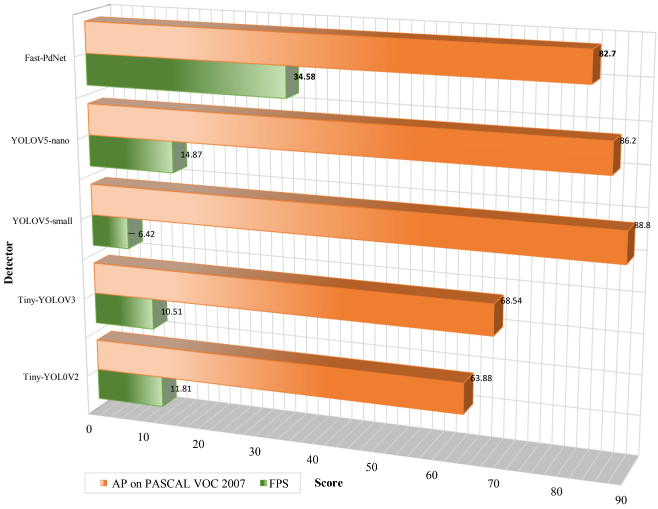


Fig. 4. Comparison of detector speeds on a Jetson Nano on VGA-resolution video.

PdNet to operate in real-time on a low graphics accelerator device. The proposed detector is the fastest detector that achieves 34.58 FPS on a Jetson Nano, as illustrated in Fig. 4. Although YOLOV5-nano [10] outperforms its accuracy by 3.5%, the proposed method is 2.3 times faster than this competitor. The YOLOV5-small [10] only achieved a speed of 6.42 on the Jetson Nano device. Additionally, these results indicate that the proposed detector uses a smaller number of algorithm operations than other models. The proposed detector comprehensively learns person features from complex data and instances. This learning capability does not compromise the model's efficiency, generating low-cost computations that enable the detector to work on an edge device in real-time.

V. Conclusion

This paper presents a fast person detector that uses CNN structure to localize human areas supporting assistive robots. The Fast-PdNet is designed to operate in real-time on a Jetson Nano without compromising performance. The entire network consists of a backbone and multi-layer detection. The extractor features block employs an efficient multi-level contextual module to comprehensively discriminate against specific features of the human body from trivial features. In addition, this module also avoids computational overhead resulting in the fewer number of parameters and computations of standard detectors. Hierarchical features detection helps the network predict multi-scale objects with anchor assignments adjusted to the feature map size. The proposed detector achieves competitive performance with other light detectors on MS COCO 2017, PASCAL VOC 2007, and PASCAL VOC 2012. The detector's capability shows that it can run 34.58 FPS in real-time on a Jetson Nano, faster than other competitors. The integration of the detector with a robot will be explored in the future to assess the reliability in real-case applications.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT). (No.2020R1A2C200897212)

References

[1] M. M. Blankenship and C. Bodine, "Socially assistive robots for children with cerebral palsy: A meta-analysis," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 1, pp. 21–30, 2021.

[2] M. D. Putro and K.-H. Jo, "Real-time face tracking for human-robot interaction," in *2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT)*, 2018, pp. 1–4.

[3] W.-Y. Hsu and W.-Y. Lin, "Ratio-and-scale-aware yolo for pedestrian detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 934–947, 2021.

[4] C. Blair, N. M. Robertson, and D. Hume, "Characterizing a heterogeneous system for person detection in video using histograms of oriented gradients: Power versus speed versus accuracy," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 236–247, 2013.

[5] M. J. Flores Calero, M. Aldás, J. Lázaro, A. Gardel, N. Onofa, and B. Quinga, "Pedestrian detection under partial occlusion by using logic inference, hog and svm," *IEEE Latin America Transactions*, vol. 17, no. 09, pp. 1552–1559, 2019.

[6] D. Wang and R. Yang, "A new descriptor for pedestrian detection based on feature fusion," in *2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2018, pp. 37–42.

[7] X. Shao, J. Wei, D. Guo, R. Zheng, X. Nie, G. Wang, and Y. Zhao, "Pedestrian detection algorithm based on improved faster rcnn," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, 2021, pp. 1368–1372.

[8] C. B. Murthy and M. Farukh Hashmi, "Real time pedestrian detection using robust enhanced tiny-yolov3," in *2020 IEEE 17th India Council International Conference (INDICON)*, 2020, pp. 1–5.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *ArXiv*, vol. abs/2004.10934, 2020.

[10] J. B. Glenn Jocher, Alex Stoken, "ultralytics/yolov5: v3.0," Aug 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3983579.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[12] M. D. Putro, D.-L. Nguyen, and K.-H. Jo, "Lightweight convolutional neural network for real-time face detector on cpu supporting interaction of service robot," in *2020 13th International Conference on Human System Interaction (HSI)*, 2020, pp. 94–99.

[13] D. Shi, H. Mi, E. G. Collins, and J. Wu, "An indoor low-cost and high-accuracy localization approach for agvs," *IEEE Access*, vol. 8, pp. 50 085–50 090, 2020.

[14] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge, "Real-time apple detection system using embedded systems with hardware accelerators: An edge ai application," *IEEE Access*, vol. 8, pp. 9102–9114, 2020.

[15] 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 *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[16] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, p. 303–338, 2010.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "Yolo-firi: Improved yolov5 for infrared image object detection," *IEEE Access*, vol. 9, pp. 141 861–141 875, 2021.

[19] W.-Y. Hsu and W.-Y. Lin, "Adaptive fusion of multi-scale yolo for pedestrian detection," *IEEE Access*, vol. 9, pp. 110 063–110 073, 2021.