J Inf Process Syst, Vol.18, No.3, pp.428~442, June 2022
https://doi.org/10.3745/JIPS.04.0246

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Development of a Real-Time Automatic Passenger Counting System using Head Detection Based on Deep Learning

Hyunduk Kim, Myoung-Kyu Sohn, and Sang-Heon Lee*

Abstract

A reliable automatic passenger counting (APC) system is a key point in transportation related to the efficient scheduling and management of transport routes. In this study, we introduce a lightweight head detection network using deep learning applicable to an embedded system. Currently, object detection algorithms using deep learning have been found to be successful. However, these algorithms essentially need a graphics processing unit (GPU) to make them performable in real-time. So, we modify a Tiny-YOLOv3 network using certain techniques to speed up the proposed network and to make it more accurate in a non-GPU environment. Finally, we introduce an APC system, which is performable in real-time on embedded systems, using the proposed head detection algorithm. We implement and test the proposed APC system on a Samsung ARTIK 710 board. The experimental results on three public head datasets reflect the detection accuracy and efficiency of the proposed head detection network against Tiny-YOLOv3. Moreover, to test the proposed APC system, we measured the accuracy and recognition speed by repeating 50 instances of entering and 50 instances of exiting. These experimental results showed 99% accuracy and a 0.041-second recognition speed despite the fact that only the CPU was used.

Keywords

Automatic Passenger Counting, Deep Learning, Embedded System, Head Detection

1. Introduction

Currently, computer vision is broadly applied to practical applications, such as security, human behavior, and object detection and recognition [1,2]. In relation to this, the market for video based automatic passenger counting (APC) systems is exploding. Transport systems use statistical information about the number of passengers as part of the effort to plan routes and schedules. By identifying passenger flows, transport companies can reasonably utilize their resources, enhance their service quality, and reduce transport costs. Reasonable transport schedules based on passenger flow information can help the company prevent empty routes and reduce social and environmental pollution. Recently, the development of the APC system has become an important issue, as these systems can accurately and reliably count passengers. Various methods are used in APC systems. Most current APC systems consist of sensors and algorithms to count the number of passengers in the transport vehicles.

Manuscript received February 3, 2021; first revision November 29, 2021; accepted January 8, 2022.

***Corresponding Author:** Sang-Heon Lee (pobbylee@dgist.ac.kr)

Division of Automotive Technology, Daegu Gyeongbuk Institute of Science & Technology, Daegu, Korea (hyunduk00@dgist.ac.kr, smk@dgist.ac.kr, pobbylee@dgist.ac.kr)

Various types of sensors, such as laser scanners, infrared, 3D sensors and RGB cameras, have different advantages and disadvantages when used in an APC system. Laser scanners allow accurate profiling of the surrounding area, but at a high cost. Infrared (IR) based technologies are cost effective, but the accuracy is low in crowded environments. The 3D sensors are robust to illumination changes but are expensive and computationally costly as well. Finally, RGB cameras are inexpensive and easy to handle but cannot accurately count passengers in environments with low level of illumination. Although, prices of 3D sensors have been going down recently, they are still expensive to commercialize. Meanwhile, the APC system based on the RGB camera has been widely developed recently because convolutional neural network (CNN)-based approaches were achieved outstanding performance.

Various people counting algorithms which are related to the APC system were proposed. The people counting algorithms can be mainly categorized into two approaches. One is feature-based approach, and the other is people detection. The feature-based algorithms statistically model a complex scene using various features. On the other hand, the people detection algorithm detects everyone who appears in the image. If a person is detected, the people counting system increases the number of people and tracks the person to prevent counting it again in the next scene. However, these approaches work well in less crowded scenes but it does not work well in more crowded scenes due to occlusions [3-9].

In this study, we proposed a real-time APC system using head detection. Head detection is useful in many real-world applications based on computer vision. Especially, accurate head detection algorithm is a most important process for vision-based APC systems. Recently, many researchers have proposed reliable and effective algorithms, which are able to predict head bounding boxes in practical applications. However, the field of head detection still has several challenging problems owing to various viewpoints and occlusions. Especially, it is difficult to predict head bounding boxes in crowded scene or low illumination environment.

Generally, the image pyramid concept has been exploited to predict object boxes across various scales and aspect ratios. Moreover, several handcraft feature descriptors, including the histograms of oriented gradients (HOG), the scale-invariant feature transform (SIFT), and variants of local binary pattern (LBP), have been applied to extract feature of objects. After feature extraction, several traditional classifiers, including the AdaBoost, support vector machine, random forests, and hidden Markov model, have been applied to obtain object information, such as the identification and location of the object. These traditional feature extraction models can only describe low-level feature and they are limited when used to detect multiple objects in crowded scenes owing to their poor generalization performance.

Recently, CNN has been widely used for object detection [10-20]. CNN-based object detection algorithm can successfully achieve efficient and accurate results. However, while the models often need to be perform efficiently in embedded system with limited computational resources in the real world, CNN-based object detection methods need a GPU to make them performable in real-time. Consequently, these methods cannot be performed in real-time on embedded systems with non-GPU environments. In this paper, we develop a lightweight head detection network which is performable in real-time on embedded devices with non-GPU environments. Starting with the Tiny-YOLOv3 network, which is a simplified version of the YOLOv3 network, we apply certain strategies to speed up the network and to make it more accurate in non-GPU environment. Moreover, we demonstrate an implementation of the real-time APC system using proposed head detection algorithm on standard non-GPU devices, in this case a Samsung ARTIK 710 device (Samsung Inc., Seoul, Korea).

The rest of this paper is organized as follows: we will briefly review some existing APC and head detection algorithms in Section 2. We will describe the details of the proposed APC system using the proposed head detection network in Section 3. We will demonstrate the experimental results for comparing the detection accuracy and efficiency of the proposed network to the YOLOv3 and the Tiny-YOLOv3 networks in section 4. Finally, the conclusion of our work and a discussion of possible future work will be presented in Section 5.

2. Related Works

2.1 Passenger Counting based on Computer Vision

Recently, several passenger counting algorithms based on computer vision have been introduced. Lengvenis et al. [4] presented a passenger counting system using computer vision techniques. They used four algorithms to calculate the number of passengers in the transport vehicles. They got promising results with ABSZ (algorithm of barrier simulation for zones) which is effective and has low false rate for passenger counting. Nasir et al. [5] proposed an APC system based on image processing techniques. In detail, they applied several image processing techniques to calculate the segmented region of interest (ROI) of passenger and the number of passengers in the transport vehicles. First, the APC system calculates the segmented ROI, which is related to the segmented passenger in the public transport, using skin color detection. Subsequently, the system calculates the number of passengers by counting the number of ROIs. Bernini et al. [6] introduced a passenger counting system using stereo vision for counting passengers. The main idea was to calculate the number of passengers entering or exiting the transport vehicles. Hence, they can compute reliable estimations of passenger's flow to control reasonably the door of the transport vehicle. They used a stereo vision camera, which was installed over bus doors. Moreover, they applied people counting and object tracking algorithms to calculate the number of people entering or exiting a predefined specific region. Lefloch et al. [7] proposed a real-time people counting system using single video camera. They performed a background subtraction based on an adaptive background model and automatic thresholding. Moreover, they performed a segmentation in the HSV color space to remove shadows. Chen et al. [8] introduced an effective people counting system based on zenithal video camera, which can detect moving people in crowded scene. They used two-stage segmentation to extract each person from a crowded scene. First, they applied morphological processing, region growing, and frame-difference technique to segment a crowd. Subsequently, they applied the connected component labeling method to create people-patterns from the results of crowd segmentation. Then, they correctly segmented each person from generated people-pattern by analyzing the area, height, and width of each people-pattern. Finally, they calculated bounding box from segmented person and tracked it. If each person reached to specific line, the system increases the number of person.

2.2 Head Detection based on Deep Learning

The object detection algorithm using deep learning can be grouped into two categories [12]. The first contains those region proposal-based methods. One of the pioneering studies was R-CNN [13], which was the first to replace the conventional object detection approaches and achieved an outstanding in the field of object detection. Later, Fast R-CNN [14] and Faster R-CNN [15] were introduced in the same

research group. Fast R-CNN computed the convolution feature map from input image only once and mapped the region proposals on feature map using ROI projection. As a result, this approach could detect object faster than R-CNN. Faster R-CNN used a fully CNN to get high quality regional proposals, unlike traditional region proposal methods. The other approach is regression-based methods. Liu et al. [16] presented the single shot detector multibox (SSD) method, which had base network and extra network. They computed multiple convolutional feature maps from extra network and detected multiple bounding boxes from these multiple feature maps across various scales. Redmon et al. [17] proposed You Only Look Once (YOLO). They generated a 7×7 feature map and detected bounding boxes and categories of target objects on each cell in these feature maps. Subsequently, YOLOv2 was introduced in the same researcher group, which improved the accuracy and speed notably [18]. They also introduced Tiny-YOLO, which is a simplified version of YOLOv2. Finally, they proposed YOLOv3, which is the latest version. They applied several strategies to improve accuracy and speed, such as multi-scale predictions, data augmentation, batch normalization and a better backbone network [19].

3. Methodology

The proposed APC system consists of two steps: head detection and passenger direction recognition. For head detection, we introduce a head detection algorithm using a CNN, which is modification of the Tiny-YOLOv3 network. To make the network faster and more accurate, we apply several strategies. For human direction recognition, we define two lines and determine whether a passenger is boarding or departing based on the direction by which the passenger passes these lines. In this way, the system can calculate the number of passengers entering or exiting the transport vehicles. The details are introduced in the following sections.

3.1 Head Detection based on Deep Learning

In this study, we introduce a head detection algorithm using deep learning, which is able to real-time performance in embedded systems to develop an APC system. As mentioned in Section 2, the YOLOv3 algorithm used a concept similar to a feature pyramid networks (FPN) to predict bounding boxes and categories of target objects on various scales. Moreover, they utilized the DarkNet-53 network, which is a 53-layered CNN with shortcut connections, as a backbone network and a better object detector with feature map upsampling and concatenation. While the YOLOv3 achieved state-of-the-art detection accuracy, detection speed was slow in non-GPU environment. On the other hand, the Tiny-YOLOv3 algorithm improved a detection speed by simplifying the backbone network. In detail, the Tiny-YOLOv3 algorithm used seven convolutional layers and six max-polling layers as the backbone network. Moreover, it has one branch to predict object on various scales. However, it still requires essentially a GPU to perform in real-time. Fig. 1 shows the architecture of the Tiny-YOLOv3 network.

In this study, we modify a Tiny-YOLOv3 network using following strategies to make a proposed algorithm be operable in real-time on non-GPU environment. First, we lighten the proposed network by removing and reducing several layers and the number of convolution filters of the backbone network of the Tiny-YOLOv3. In detail, we remove the last one convolutional layer and max-polling layer and halve the number of convolution filters in all convolutional layers after the first four convolutional layers.

Moreover, we add two branches to enhance the detection accuracy. Using this method, the proposed network can obtain more significant information and more detailed information from the upsampled feature maps and the previous feature maps, respectively. In conclusion, the proposed head detection network can extract features from three different scales using the concept similar to a FPN and can predict head boxes on those scales. To reduce the computational complexity, we utilize depthwise separable convolution technique, which was introduced in MobileNet [20]. The depthwise separable convolution is a depthwise convolution followed by a pointwise convolution, which replaces a traditional convolution. In detail, the depthwise convolution is a channel-wise spatial convolution to extract features along input channels, and the pointwise convolution is a 1×1 convolution to merge the results of the depthwise convolution. Fig. 2 shows the architecture of the proposed head detection network.

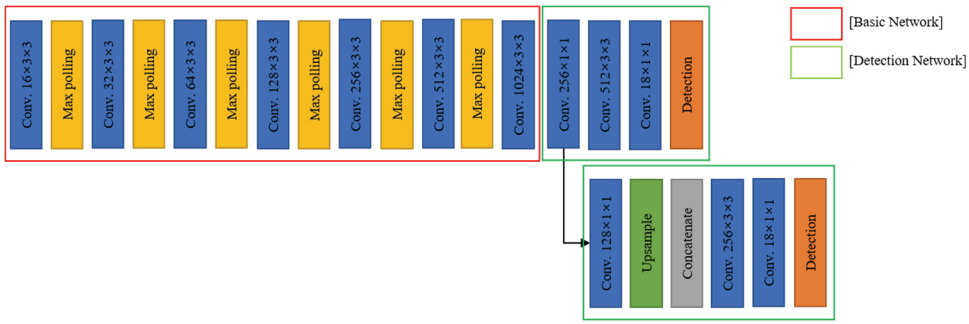

Fig. 1. Architecture of the Tiny-YOLOv3 network.

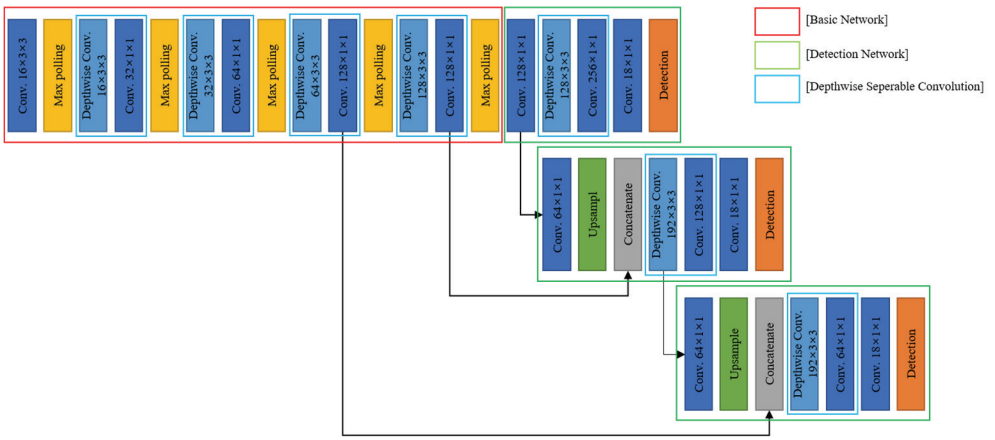

Fig. 2. Architecture of the proposed head detection network.

Generally, decreasing the size of the input image makes the network faster, but the network may not detect small objects. In fact, it can almost double the network speed by halving the size of the input image. However, because we already added an additional branch to enhance the detection accuracy when we designed the head detection network, we decrease the size of the input image to make the proposed network faster. In detail, while the size of the input image of the Tiny-YOLOv3 is 416×416, we resize the input image to 224×224 and feed it to the proposed network. Moreover, a batch normalization has been commonly utilized in all convolutional layers because it significantly improves stability and

convergence of training. On the other hand, as mentioned in [21], because the batch normalization process slows down the entire feedforward process and it may not need in simplified network, we eliminate the batch normalization in all convolutional layers of the proposed network.

3.2 Passenger Direction Recognition

To count the number of passengers, we determine whether a passenger is entering or exiting by means of passenger direction recognition. Generally, people counting systems set only one specific line and check the moving directions of people. However, this approach may not be accurate if people move

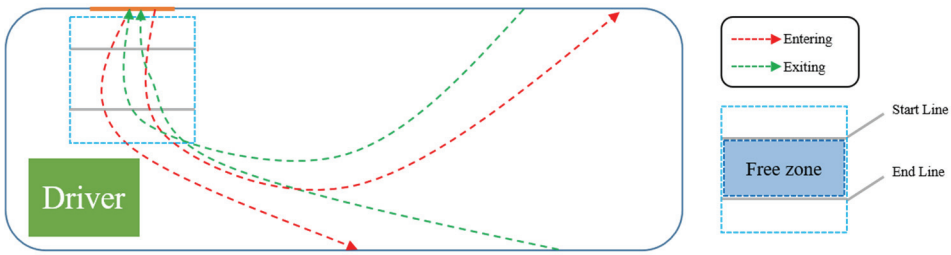

Fig. 3. Proposed passenger direction recognition scenario and flow chart.

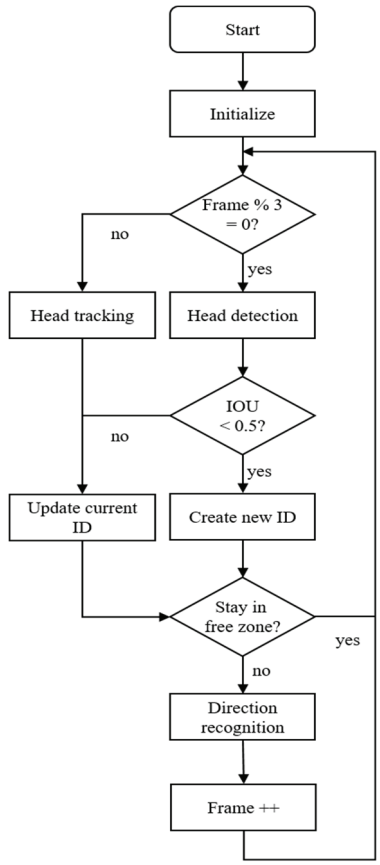

Fig. 4. Proposed passenger direction recognition scenario and flow chart.

around the specific line. Hence, we set two specific lines, a start line and an end line, and a free zone. Subsequently, we determine whether each passenger is entering or exiting using these lines by analyzing passenger's moving trajectory. If a passenger passes through the start and end lines in sequence, the system determines that the passenger is entering, while in the opposite case, the system determines that the passenger is exiting. Moreover, if the passenger stays in the free zone, the system reserves its decision. Fig. 3 shows the proposed passenger direction recognition scenario.

For passenger direction recognition, we initially detect the head box using the previously mentioned head detection network once every three frames to detect new passengers, comparing the current head box to the previously detected head boxes. If the intersection over union (IoU) value exceeds 0.5, we update the head box to the current head box. If not, we assign a new ID. Subsequently, we apply simple tracking algorithms, such as the kernelized correlation filter (KCF) tracker [22] to reduce the processing time. Fig. 4 shows the proposed passenger direction recognition and flow chart.

4. Experimental Results

4.1 Datasets

In this section, we discuss three public head datasets, such as HollywoodHeads, SCUT-HEAD, and CrowdHuman, and one collected head dataset, named by DGIST-HEAD. Fig. 5 shows sample images from the DGIST-HEAD and public head datasets.

The HollywoodHeads dataset [23] includes total 224,740 frames labeled with 369,846 human heads from 21 Hollywood movies. In detail, the HollywoodHeads dataset contains 216,719 frames from 15 movies, 6,719 frames from three movies, and 1,302 frames from another set of three movies for training, validation, and testing, respectively. The SCUT-HEAD dataset [24] contains total 111,251 heads annotated in 4,405 images and consists of a PartA and a PartB. The PartA contains 67,321 heads annotated in 2,000 images, which are collected from CCTV of classrooms in a university. The PartA includes 1,500 images and 500 images for training and testing, respectively. The PartB contains 43,930 heads annotated in 2,405 images, which are crawled from Internet. The PartB includes 1,905 images and 500 images for training and testing, respectively.

The CrowdHuman dataset [25] includes total 470k human instances with head bounding box annotated in 24,370 images, which are crawled from Google image search engine with a keyword considering crowded scene. In detail, the CorwdHuman dataset contains 15,000 images, 4,370 images, and 5,000 images for training, validation, and testing, respectively. Moreover, we collected total 65,000 head images, referred to as the DGIST-HEAD dataset. In detail, we installed a camera on the top of the door of a bus and acquired 13,000 head images. Moreover, we installed a camera on the ceiling of a laboratory and acquired 50,000 head images.

4.2 Implementation Details

To evaluate the proposed head detection network, we trained the proposed network on ImageNet and fine-tuned it on each public head dataset. Also, the YOLOv3 and Tiny-YOLOv3 networks were initialized by pre-trained weights and fine-tuned them on each public head dataset. Multi-scale training approach was commonly applied to improve the detection accuracy of various sizes of the images and the heads. We trained the proposed, the YOLOv3 and the Tiny-YOLOv3 networks for 500,200 iterations with

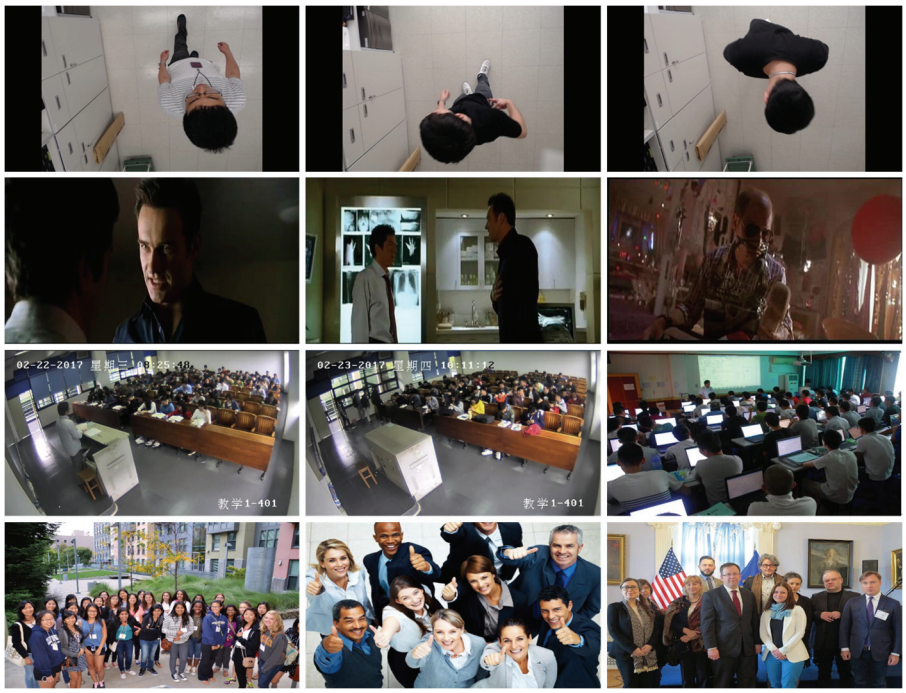

Fig. 5. Sample images. The first row is from the DGIST-HEAD, the second row is from the Hollywood-Heads, the third row is from SCUT-HEAD, the fourth row is from the CrowdHuman dataset.

stochastic gradient descent (SGD) optimizer, batch size of 64, learning rate of 0.001, momentum of 0.9, and weight decay of 0.005. We implemented the proposed network using the DarkNet framework [26] on a computer with NVIDIA TITAN RTX GPU.

Finally, we implement an APC system using the head detection network on a Samsung ARTIK 710. The proposed APC system mainly consists of two parts. The first step is head detection and tracking. The APC system detects the head region in each frame obtained from RGB camera installed in transport using previously mentioned the head detection network. We use the OpenCV DNN module to inference the proposed head detection network and the OpenCV Tracking API, which contains the KCF tracker to track the detected head box. The second step is passenger counting. The system determines that passenger is getting in or out using the direction by which the passenger passes specific predefined lines. In this way, the system can calculate the number of passengers entering or exiting the transport vehicle. Fig. 6 describes the architecture of the proposed APC system.

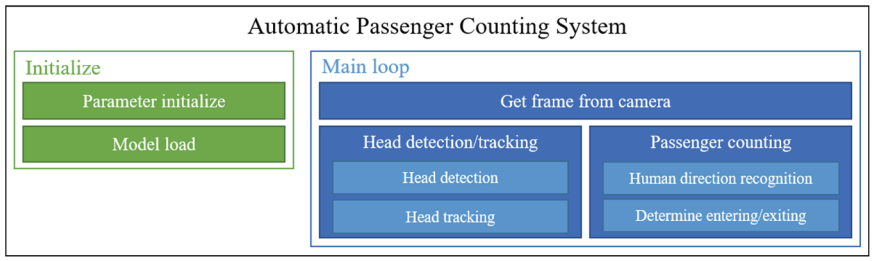

Fig. 6. Architecture of the proposed APC system.

4.3 Experimental Results

We train and evaluate the proposed head detection network using the DarkNet framework [26] and three public head datasets. Fig. 7 shows the validation set loss value on three public head datasets. The vertical and horizontal axes represent the loss values of the validation set and the number of iterations. The decrease in the loss value according to the progress in each iteration can be seen. For each iteration during which the proposed head detection network is being learned, the loss value for the validation set is calculated. The calculated loss value in each iteration helps predict the model's stability and performance in real-world applications. To compare the accuracy of the proposed network, we used the average precision (AP), which is a widely used evaluation metric obtained by the area under the precision-recall (PR) curve. To calculate the PR curve, we first calculate the IoU between the predicted head bounding box and its ground truth head bounding box. Subsequently, if the IoU is greater than 0.5, then the predicted head bounding box is classified as true positive detection. In the opposite case, it is classified as false positive detection. Moreover, to compare the efficiency of the proposed network, we used the metrics of the frame per second (FPS) and billion floating point operations per second (BFLOPS). All tests were performed on an Intel Core CPU i9-7960X (2.80 GHz) without a GPU. We compared the detection accuracy and efficiency of the proposed network to the YOLOv3 and Tiny-YOLOv3 networks.

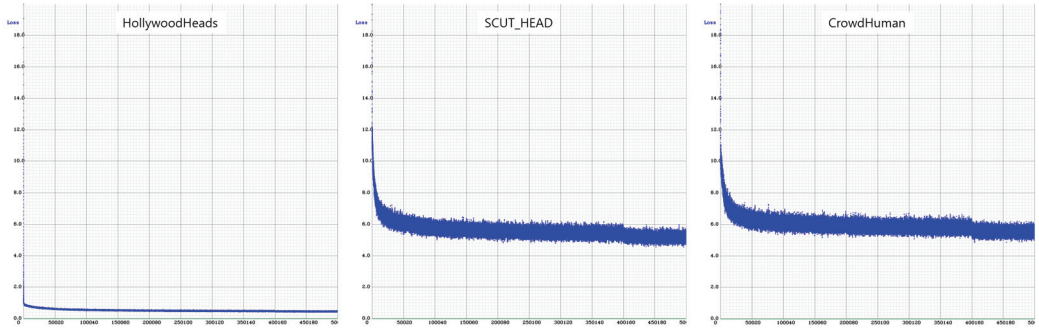

Fig. 7. Validation set loss value graphs of proposed network on three public head datasets.

Table 1 and Fig. 8 show comparisons between the proposed network and the YOLOv3 and Tiny-Yolov3 networks on the three public head datasets. This comparison result reflects that the proposed head detection network outperformed than the Tiny-YOLOv3 network on the three public head datasets. On the other hand, because the SCUT-HEAD and CrowdHuman datasets contain smaller head size and the presence of occlusion in crowded scene, it was more difficult to predict bounding box of heads in these datasets than in the HollywoodHeads dataset. For these reasons, the YOLOv3 network achieved best performance in terms of detection accuracy on the SCUT-HEAD and CrowdHuman datasets. Nevertheless, the proposed network achieved accuracy similar to that of the YOLOv3 network on the HollywoodHeads dataset relative to other datasets. However, the YOLOv3 network had values of 0.39 for FPS and 18.93 for BFLOPS. Hence, it was scarcely operable in real-time without GPU. Furthermore, although the Tiny-YOLOv3 network achieved values of 4.14 for FPS and 1.58 for BFLOPS, it achieved the worst performance in terms of detection accuracy on all public head datasets. In conclusion, the proposed network was 50× and 6× faster than the YOLOv3 and Tiny-YOLOv3 networks, respectively. Hence, the proposed network is sufficiently effective for an APC system on embedded devices. Fig. 9–11 show the result of the proposed network on three public head datasets.

Table 1. Comparison of various networks on three public head datasets

Model	BFLOPS	FPS	AP_{50}		
			HollywoodHeads	SCUT-HEAD	CrowdHuman
YOLOv3	18.93	0.39	78.66	64.45	40.59
Tiny-YOLOv3	1.58	4.14	54.03	12.55	9.47
Proposed	0.14	26.34	72.90	27.69	14.23

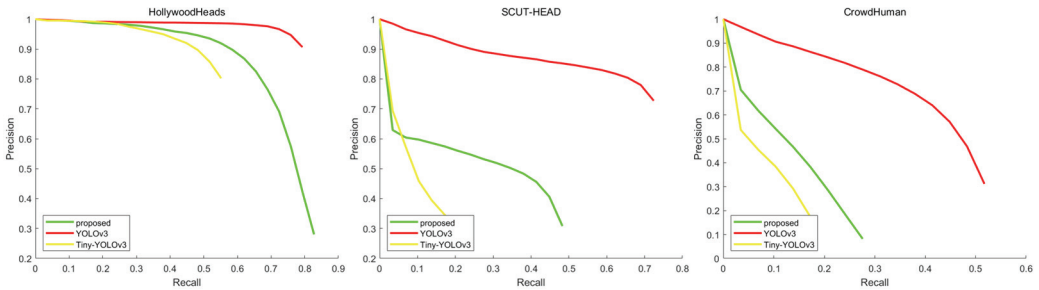

Fig. 8. Precision-Recall (PR) curve of various networks on three public head datasets.

(a) (b) (c)

Fig. 9. Results on the HollywoodHeads dataset: (a) YOLOv3, (b) Tiny-YOLOv3, and (c) proposed images.

(a) (b) (c)

Fig. 10. Results on the SCUT-HEAD dataset: (a) YOLOv3, (b) Tiny-YOLOv3, and (c) proposed.

Moreover, we tested the proposed APC system on a Samsung ARTIK 710 board. For head detection, we use the OpenCV DNN module to inference the proposed head detection network. Because the collected DGIST-HEAD dataset is small, we use a transfer learning technique to train the proposed head detection network. First, we train the network using three public head datasets, after which we retrain the network using the collected DGIST-Head dataset. For head tracking, we simply use the OpenCV Tracking API, which contains the KCF tracker. We installed a camera on the ceiling of our laboratory and measured the accuracy and recognition speed while repeating 50 entering and 50 exiting instances. The experimental results showed 99% accuracy (100% entering, 98% exiting) and a recognition speed of 0.041 seconds despite the fact that only the CPU was used. Fig. 12 shows the installed H/W and the result of the proposed APC system.

5. Conclusion

In this study, we proposed a lightweight head detection network using deep learning applicable to an embedded system. To detect a head box, we proposed a head detection algorithm using deep learning. To

(a) (b) (c)

Fig. 11. Results on the CrowdHuman dataset: (a) YOLOv3, (b) Tiny-YOLOv3, and (c) proposed.

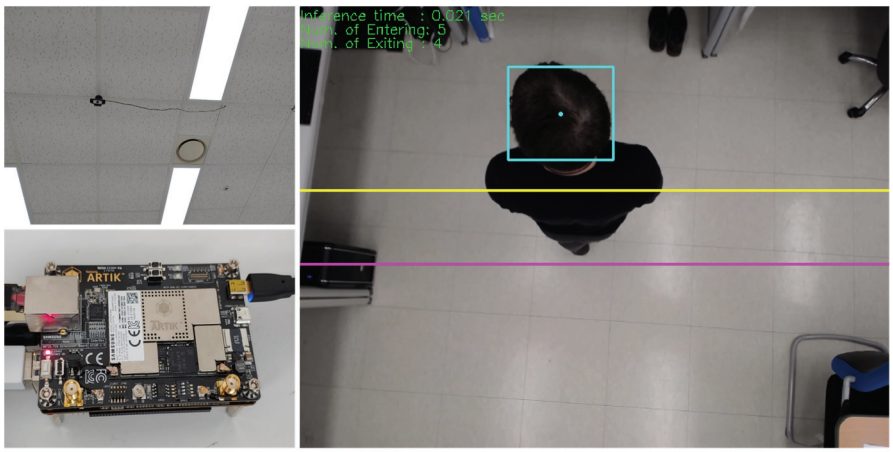

Fig. 12. Installed H/W and result of the proposed APC system.

speed up the network and to make it more accurate in a non-GPU environment, we designed a shallow network with additional branch, applied a depthwise separable convolution technique, reduced the size of the input images, and removed batch normalization. To count the number of passengers, we determined whether each passenger was entering or exiting by means of a passenger direction recognition technique.

For passenger direction recognition, we set two specific lines and a free zone and determined whether each passenger was boarding or alighting depending on the direction by which the passenger passes these lines. To test the performance of the proposed head detection network, we compared the detection accuracy and efficiency of the proposed network to the YOLOv3 and Tiny-YOLOv3 networks. The experimental results demonstrate that the proposed head detection network is sufficiently efficient for application to an APC system. We also implemented and tested the proposed APC system on a Samsung ARTIK 710 board. The experimental results reflect that the proposed APC system is highly accurate and performable in real-time on non-GPU devices.

In the future work, we will develop and apply a passenger recognition algorithm to the APC system. Generally, many public transportation systems use boarding passes or QR codes to identify passengers. However, these methods are inconvenient. Hence, with passenger recognition to verify passengers' identity and destination automatically, transport companies can more reasonably use their resources, enhance their service quality, and reduce transport costs.

Acknowledgement

This work was supported by the DGIST R&D Program of Ministry of Science and ICT (22-IT-02).

References

[1] F. Zhang, T. Y. Wu, J. S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, article no. 40, 2019. https://doi.org/10.1186/s13673-019-0203-8

[2] H. F. Nweke, Y. W. The, G. Mujtaba, U. R. Alo, and M. A. Al-garadi, "Multi-sensor fusion based on multiple classifier systems for human activity identification," *Human-centric Computing and Information Sciences*, vol. 9, article no. 34, 2019. https://doi.org/10.1186/s13673-019-0194-5

[3] X. X. Wang an Y. Shen, "A video traffic flow detection system based on machine vision," *Journal of Information Processing Systems*, vol. 15, no. 5, pp. 1218-1230, 2019.

[4] P. Lengvenis, R. Simutis, V. Vaitkus, and R. Maskeliunas, "Application of computer vision systems for passenger counting in public transport," *Elektronika ir Elektrotechnika*, vol. 19, no. 3, pp. 69-72, 2013.

[5] A. S. A. Nasir, N. K. A. Gharib, and H. Jaafar, "Automatic passenger counting system using image processing based on skin colour detection approach," in *Proceedings of 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, Kuching, Malaysia, 2018, pp. 1-8.

[6] N. Bernini, L. Bombini, M. Buzzoni, P. Cerri, and P. Grisleri, "An embedded system for counting passengers in public transportation vehicles," in *Proceedings of 2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, Senigallia, Italy, 2014, pp. 1-6.

[7] D. Lefloch, F. A. Cheikh, J. Y. Hardeberg, P. Gouton, and R. Picot-Clemente, "Real-time people counting system using a single video camera," in *Proceedings of SPIE 6811: Real-Time Image Processing 2008*. Bellingham, WA: International Society for Optics and Photonics, 2008, pp. 71-82.

[8] C. H. Chen, T. Y. Chen, D. J. Wang, and T. J. Chen, "A cost-effective people-counter for a crowd of moving people based on two-stage segmentation," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 3, no. 1, pp. 12-23, 2012.

[9] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein, "Person head detection in multiple scales using deep convolutional neural networks," in *Proceedings of 2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, 2018, pp. 1-7.

[10] D. Cao, Z. Chen, and L. Gao, "An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks," *Human-centric Computing and Information Sciences*, vol. 10, article no. 14, 2020. https://doi.org/10.1186/s13673-020-00219-9

[11] M. T. N. Truong and S. Kim, "A tracking-by-detection system for pedestrian tracking using deep learning technique and color information," *Journal of Information Processing Systems*, vol. 15, no. 4, pp. 1017-1028, 2019.

[12] G. Chen, X. Cai, H. Han, S. Shan, and X. Chen, "HeadNet: pedestrian head detection utilizing body in context," in *Proceedings of 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, Xi'an, China, 2018, pp. 556-563.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 580-587.

[14] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1440-1448.

[15] 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, 2017.

[16] 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.* Cham, Switzerland: Springer, 2016, pp. 21-37.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 779-788.

[18] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, pp. 6517-6525.

[19] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018 [Online]. Available: https://arxiv.org/abs/1804.02767.

[20] 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," 2017 [Online], Available: https://arxiv.org/abs/1704.04861.

[21] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers," in *Proceedings of 2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, 2018, pp. 2503-2510.

[22] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583-596, 2014.

[23] T. H. Vu, A. Osokin, and I. Laptev, "Context-aware CNNs for person head detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 2893-2901.

[24] D. Peng, Z. Sun, Z. Chen, Z. Cai, L. Xie, and L. Jin, "Detecting heads using feature refine net and cascaded multi-scale architecture," in *Proceedings of 2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, 2018, pp. 2528-2533.

[25] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, “Crowdhuman: a benchmark for detecting human in a crowd,” 2018 [Online]. Available: https://arxiv.org/abs/1805.00123.

[26] J. Redmon, “DarkNet: open source neural networks in C,” 2013 [Online]. Available: https://pjreddie.com/darknet/.

Hyunduk Kim **https://orcid.org/0000-0002-2402-2250**

He received B.S. and M.S. degrees in Mathematics from Kyungpook National University in 2009 and 2012, respectively. He is currently an associate researcher in Division of Automotive Technology from Daegu Gyeongbuk Institute of Science and Technology, Daegu, Korea, joined from 2012. His research interests include artificial intelligence and face analysis.

Myoung-Kyu Sohn **https://orcid.org/0000-0002-1393-7818**

He received B.S. degree in Electrical Engineering from Kyungpook National University in 1997, M.S. degree in Electrical Engineering and Computer Science from Seoul National University in 1999, and Ph.D. degree in Electrical Engineering from Kyungpook National University in 2016. He is currently a principal researcher in Division of Automotive Technology from Daegu Gyeongbuk Institute of Science and Technology, joined from 2005. His research interests include artificial intelligence and human computer interaction.

Sang-Heon Lee **https://orcid.org/0000-0003-1813-1044**

He received B.S., M.S., and Ph.D. degrees in Electrical Engineering from Kyungpook National University in 1993, 1996 and 2013, respectively. He is now a project leader of a principal researcher in Division of Automotive Technology from Daegu Gyeongbuk Institute of Science and Technology Daegu, Korea, joined from 2005. His research interests include artificial intelligence and machine learning.