2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Driver Identification System Using Convolutional Neural Network with Background Removal-based Infrared Data Augmentation

Sanghyuk Kim, Yunsoo Lee, Namhyun Ahn and Suk-Ju Kang

***Abstract*— As the interest of the autonomous driving increases, techniques related to the advanced driver assistance system are evolving together. In this paper, we propose a novel driver identification system using convolutional neural network (CNN) with the background removal-based infrared image data augmentation. It helps to identify who a driver is, and provides the customized driving environment. The process for the proposed identification system is as follows. First, we acquire customized individual infrared images in a driving simulation environment. Second, we augment the large amount of data by using the background removal-based method and several image processing techniques. Third, the augmented data is trained by the low-complexity-based CNN method. Finally, we load all trained weights to the forward network for real-time processing. In the experimental results, the proposed system had the memory resource of 4,795 KB, which are up to 49.0822 times smaller than benchmark algorithms, and the average F_1 score of 0.9418 for the driver identification accuracy.**

I. Introduction

According to the development of autonomous vehicle systems, the human identification [1], an essential module for the advanced driver assistance system (ADAS), has been studied extensively. The main purpose of ADAS is the driver safety, and Working Party 29 (WP29) establishes the internal and external safety standards for autonomous driving levels [2]. In order to improve the driver safety, it is necessary to change the environment of the vehicle to reflect the preference of the driver. It means that characteristics of physical modules including a seat, rear-view mirror and side-view mirrors vary from person to person. Therefore, in order to provide a personalized environment, ADAS must first identify the driver.

A typical method for human identification is to use computer vision techniques. Deep learning algorithms based on convolutional neural network (CNN) have been widely used in the computer vision field [3]-[5]. However, early models like AlexNet [3] are not suitable for automotive applications, which have a constraint for the hardware resource usage such as the memory resource and bandwidth. Therefore, it is difficult to use the typical CNN-based systems with the high computational complexity in the practical case. Recently, many researches consider the reduction of the hardware resource and computational complexity. MobileNet [4] and ShuffleNet [5] can be candidates that are utilized in practical embedded systems of the automotive application. Generally, the existing CNN models are usually based on a database of RGB images. However, these images are very sensitive to illumination changes. This critical problem can occur in an outdoor environment with a large illuminance variation, which is a common case that can occur in a vehicle application. In order to solve this problem, the infrared camera can be used [6], and it operates irrespective of the illumination change according to day and night. Therefore, we propose a novel system based on infrared images, which are robust for the illumination change. In addition, a driving simulator is installed to establish an environment similar to the actual driving situation as shown in Fig. 2. For training CNN models, the amount of data for the database construction is much larger than those of existing machine learning-based models [7]. There is a limitation to directly collect the huge data for training due to time-consuming and costly issues. For the solution, data augmentation can be utilized, and it increases the amount of data using only the limited collected data.

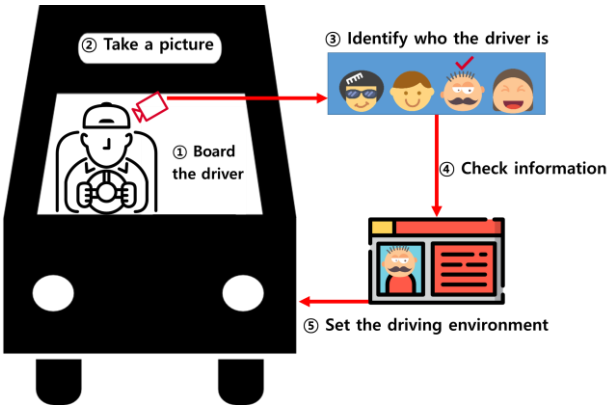
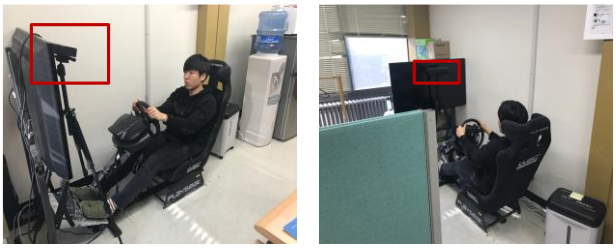


Fig. 1. Overall concept of the proposed driver identification system.



(a) (b)

Fig. 2. Environment of the infrared image database construction using the driving simulator (the red block shows the camera position): (a) front side and (b) rear side.

In this paper, we propose a novel method for driver identification system using CNN with background removal-based infrared image data augmentation as shown in Fig. 3. The proposed method uses an Kinect for window v2 (512 × 424 maximum infrared pixel resolution) [8] made by

Sanghyuk Kim, Yunsoo Lee, and Namhyun Ahn are with the Electronic Engineering Department, Sogang University, Seoul, Korea (e-mail: {hitboy91, profitshore, neition503}@gmail.com).

Suk-Ju Kang(corresponding author) is with the Electronic Engineering Department, Sogang University, Seoul, Korea (e-mail: sjkang@sogang.ac.kr).

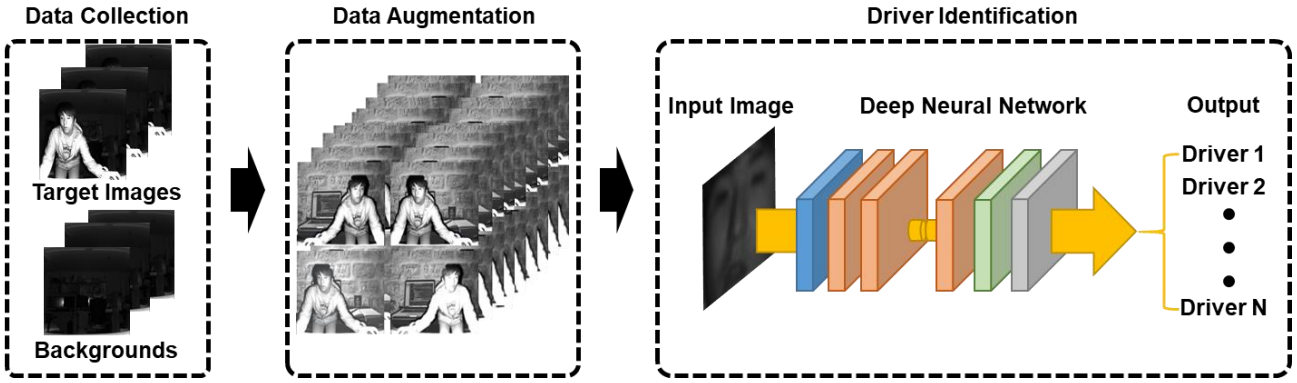


Fig. 3. Overall flow chart of the proposed driver identification system using CNN with background removal-based infrared data augmentation.

Microsoft, infrared camera, which is robust on illuminance change, to identify the driver. In order to train infrared data for the deep neural network, we should augment data. Thus, we propose a background removal-based data augmentation. First, each collected infrared data is divided into foreground and background regions by the proposed background removal based on the pixel-wise relation. In the background removal process, the foreground of an input image is first extracted by Gaussian mixture models [9]–[11], and then, the proposed hole filling algorithm compensates the incorrect foreground region. Next, we augment the extracted foreground region by applying various image processing [12] algorithms such as image flipping and brightness change. Using the augmented data, we develop the driver identification system based on CNN. In this case, we mainly consider two factors: memory resource and network size. Thus, the proposed method is based on MobileNet [4] with the approximate quarter number of the parameters for the reference model. We train only the facial region captured by using Haar-like features [13] to be robust on changing appearance. After training infrared data, we develop the forward network with the optimal weights, and it operates in real time.

This paper is organized as follows: Section II explains the proposed driver identification system using CNN with background removal-based infrared data augmentation. In section III, we represent the experimental results. Section IV concludes this paper.

II. Proposed System

Fig. 3 shows the overall flow chart of the proposed driver identification system using CNN with background removal-based infrared data augmentation in real time. The operation process of the system is mainly divided into the background removal-based data augmentation and the CNN-based driver identification. The detailed process is as follows.

A. Background Removal-based Data Augmentation

Fig. 4 shows the proposed background removal-based data augmentation. The proposed method performs segmentation between the foreground and background region based on Gaussian mixture model (GMM) [9]-[11]. In this case, if the distinction between image characteristics is not clear, holes occur. In order to solve this problem, we use the proposed hole filling algorithm considering the pixel-wise relation. The actual object region is finally selected through a morphological processing. Various image processing algorithms such as image flipping and brightness are applied to the foreground region to change it. The detail processes are explained as follows.

1) Mixed Gaussian Mixture Model

The background removal algorithm based on GMM uses the difference between the target image (foreground image) and the background image. Specifically, the foreground region (or background region) in a given image can be clustered into a group centering on the average intensity level. However, even in the part where the object is cognitively determined as the background region, it can be judged as the foreground region due to the slight difference for the intensity level of a given pixel. In order to overcome this problem, we simultaneously use different conventional GMMs in order to have complementary effects by optimally adjusting parameters.

One GMM only adapts controllable parameters [9]. Between the target and background image, there are values, the means and variances, to define foreground regions. These values describe the components of the Gaussian estimate, $\hat{p}(\vec{x}|X_T, F+B)$. The Gaussian estimate is as follows [9]:

$$\hat{p}(\vec{x}|X_T, F+B) = \sum_{m=1}^{M} \hat{\pi} N(\vec{x}; \hat{\vec{x}}_m, \hat{\sigma}_m^2 I) \qquad (1)$$

where $\vec{x}$ denotes the pixel value, X_T denotes training data set, F and B denote foreground and background regions. M denotes components of GMM, and $\hat{\pi}$ denotes non-negative mixing weights. In addition, $\hat{\vec{x}}_1, \ldots, \hat{\vec{x}}_m$ denote the estimates of the means, $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_m^2$ denote the estimates of variances, and I denotes a proper dimension. The other GMM shown in Fig. 4 adapts parameters and the number of components of the mixture for each pixel, m [9]. After the complementary GMM-based method is operated, foreground and background regions are separated.

2) Noise Removal

In this process, we remove a candidate noise region. First of all, noise is removed by Gaussian filtering [14] for the foreground region generated by the mixed-GMM. Then, we repeat morphological processing such as erosion and dilation to remove the overall noise again.

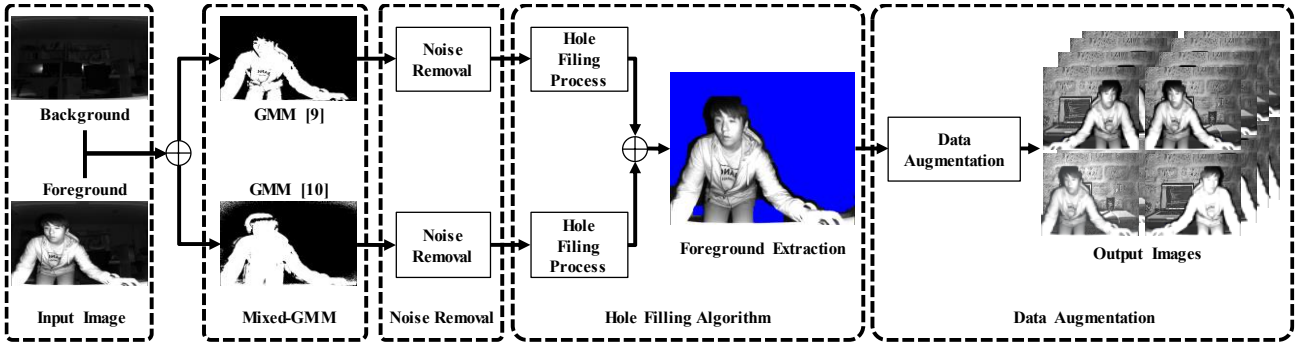


Fig. 4. Flow chart of the proposed background removal-based data augmentation with mixed GMM [9] – [10].

```
find_max_pixel ( ):
for (i = 0 ; i <n_width; i++)
    for (j = 0 ; j <n_height; j++)
        if P_(i,j) = 0
            F_width[i]++
            F_height[j]++

for (i = 0 ; i <n_width; i++)
        if X_max < F_width[i]
            X_max = i #MAX X-axis

for (i = 0 ; j <n_heighth; j ++)
        if Y_max < F_height[j]
            Y_max = j #MAX Y-axis
```

Fig. 5. Pseudo code of maximum foreground position detection.

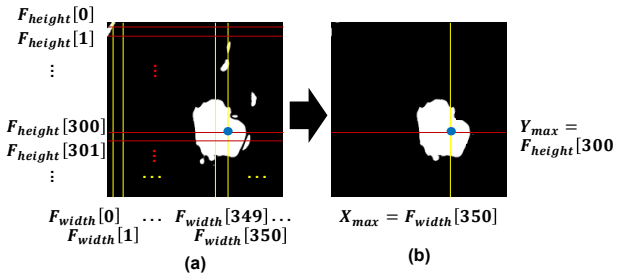


Fig. 6. Example of a flood-fill algorithm with the maximum foreground coordinate: (a) and (b) denote the input and output image. The white and black regions denote foreground and background regions.

Next, we use a flood-fill algorithm [15]. In order to do it, we should find the exact coordinate to apply flood-fill algorithm. In this case, the horizontal and vertical maximum indexes (X_{max}, Y_{max}) are calculated in the foreground region as shown in Fig. 5. The pseudo-code in Fig. 5 is as follows:

- $F_{width}[i]$ and $F_{height}[j]$, where i and j denote x and y-axes of a given pixel, accumulate the number of all foreground pixels in the range of the image width, n_{width}, and height, n_{height}. The maximum foreground coordinate (X_{max}, Y_{max}) is created by selecting the largest $F_{width}[i]$ and $F_{height}[j]$.

In Fig. 6, the flood-fill algorithm is applied in the region including the obtained maximum foreground coordinate, the blue point. If a pixel belongs to the coordinate (X_{max}, Y_{max}), it defines the foreground region, and otherwise, it defines the background region.

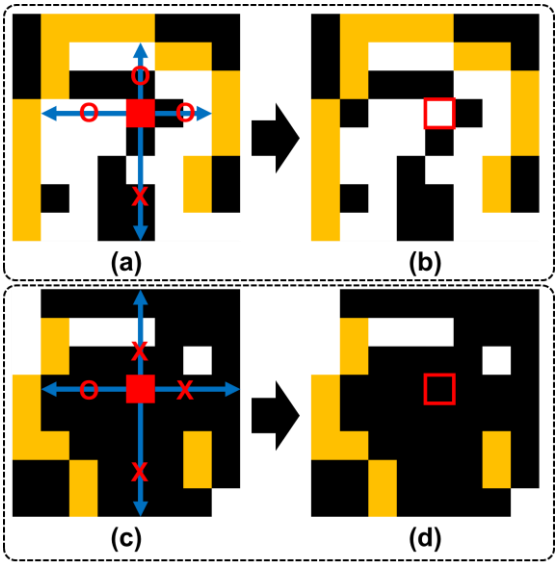


Fig. 7. Proposed hole filling algorithm: (a) and (c) denote each input image, and (b) and (d) denote the output image of (a) and (c) processed by the proposed algorithm. The white and black pixels denote foreground and background pixels, orange pixels denote the boundary from Canny edge detection [16], and red pixels denote a processed pixel.

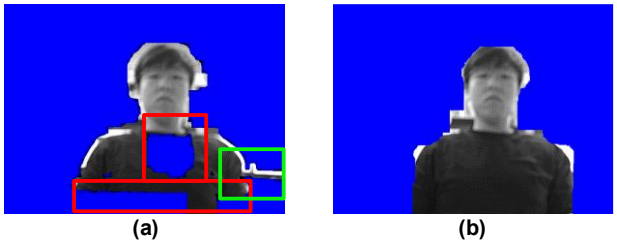


Fig. 8. Results of blurring images applied by each background removal; the inputs are same between (a) and (b). (a) is applied by the single GMM background removal [9], and (b) is applied by the proposed algorithm. The blue area is a background region, the red blocks are the non-detected region of a target image's real foreground, and the green block is the incorrectly detected region of a target foreground.

3) Hole Filling Process

After reducing noise pixels, we apply the proposed hole filling process which is a compensation algorithm for the foreground region which is incorrectly generated in the mixed-GMM, as shown in Fig. 7. A given pixel, a background pixel, changes through the relation with neighboring pixels. First, we extract the average intensity level (P_{avg_fore} and P_{avg_back}) of

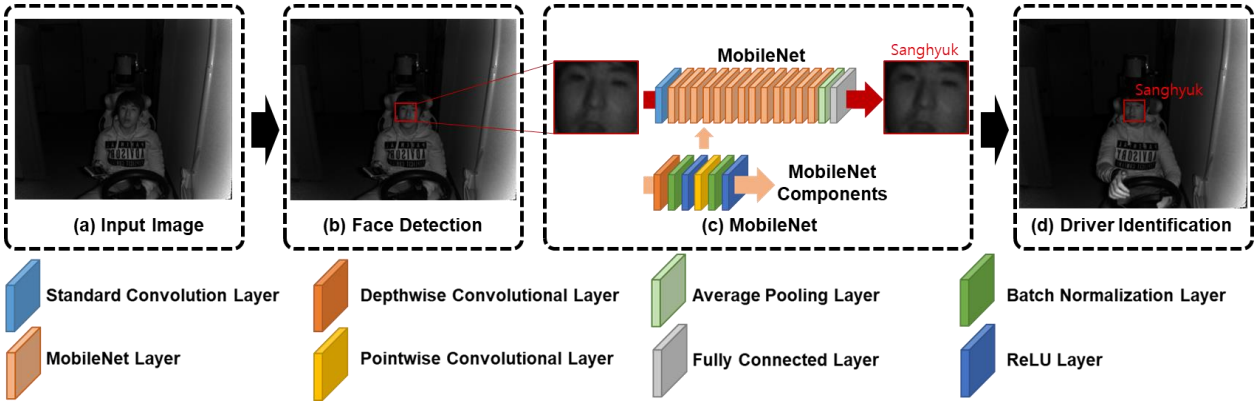


Fig. 9. Flow chart of the proposed driver identification system in real time: (a) input data, (b) face detection, (c) MobileNet and (d) driver identification.

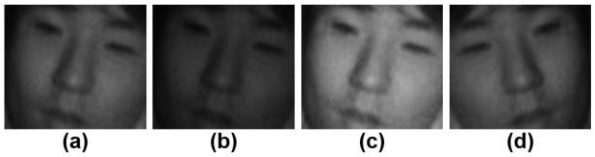


Fig. 10. Examples of augmented database: (a) input image, (b) darkened image, (c) brightened image, and (d) flipped image.

the foreground pixels (P_{fore}) and background pixels (P_{back}) as follows:

$$P_{avg\_fore} = \frac{1}{N}\sum_{n=1}^{N} P_{fore}(x_n, y_n) \quad (2)$$

$$P_{avg\_back} = \frac{1}{M}\sum_{m=1}^{M} P_{back}(x_m, y_m) \quad (3)$$

Next, if a given pixel is closer to foreground average intensity level, we check the boundary created by Canny edge detection [16] surrounding the given pixel. If the number of detected boundaries is more than 3, like the red pixel in Fig. 7 (a), it is redefined as a foreground pixel like the white pixel surrounding the red line in Fig. 7 (b). Otherwise, the given pixel like the red pixel in Fig. 7 (c) does not change like the black pixel surrounding the red line in Fig. 7 (d).

The red blocks of the object region of Fig. 8 (a) do not have unimodal property as a foreground region. Therefore, the proposed algorithm detects the boundary region by using Canny edge detection [16] even though it is blurring in the target image like Fig. 8. When satisfying the boundary condition, the missing foreground region is compensated as shown in Fig. 8 (b). In the case of the green block in Fig. 8 (a), the unnecessary region is removed through erosion in the noise removing step. The final result image is shown in Fig. 8 (b).

4) Data Augmentation

After performing the background removal process, we have to augment data due to the limited number of infrared images. Infrared images are dependent on the camera device, and we can acquire images with different characteristics depending on the camera hardware specification. Due to this critical point, we have to augment the captured infrared images to train CNN models. In order to solve the insufficiency of image figures, we augment infrared database by using various low-level image processing algorithms [12], which apply to foreground regions, such as image flipping and brightness change. Considering the symmetry of an image, the foreground region is flipped. We turn the region right and left, not upside down. In addition, we brighten and darken the input foreground region because the illumination of a foreground region changed through various factors related to the infrared camera as follows:

$$P_{current}(x, y) = \gamma \times P_{previous}(x, y) + \beta \quad (4)$$

where $P_{previous}(x, y)$ and $P_{current}(x, y)$ denote the previous and current pixel intensities in an input pixel, respectively, β denotes an offset intensity, and γ denotes a scale factor. We define $\beta \in \{0\}$ and $\gamma \in \{-30, -20, -10, 1, 10, 20, 30\}$.

B. CNN-based Driver Identification

The proposed system should be operated in the vehicle, and hence, the CNN model should be selected as a model that could be implemented in the embedded environment. Specifically, the hardware resource should be lower than conventional CNN models, and the network model in this paper is constructed considering this point. The detail processes in Fig. 9 are explained as follows.

1) Face Detection

In case of training, detection problems occur when using the entire image captured by the infrared camera. This is that the unique component detection is impossible because the appearance characteristics of the driver can be changed all the time. Thus, we select the facial region as shown in Fig. 9 (b), and train and deduce based on the region. By using Viola-Jones face detection based on Haar-like features [13], a facial region like the red block in Fig. 9 (b), is detected. To extract facial ROI, we rearrange ROI only for the face contour [17] to consider the changeable appearance like the hair style change.

2) Training Process

Among the conventional CNN models, the proposed method is based on a tiny network model with handling the hardware resource issue and the real-time processing, which is MobileNet [4]. MobileNet reduces the number of parameters and computation costs of each conventional convolutional layer by using two separate layers, the depthwise and pointwise convolutional layers. This model adopts a global

TABLE I. VARIOUS ALEXNET VS. MOBILENET

Model (α Name-ρ)	Memory Resource (KB)	Memory Resource (relativeness)	Our DB Accuracy (%)
1.0 AlexNet-244	235,349	49.0822	83.2012
1.0 MobileNet-224	82,829	17.2740	95.3922
0.75 MobileNet-224	57,827	12.0599	96.694
0.5 MobileNet-224	35,721	7.4496	98.0095
0.25 MobileNet-224	16,511	3.4434	97.6503
1.0 MobileNet-100	36,436	7.5987	97.8235
0.75 MobileNet-100	22,994	4.7954	93.6756
0.5 MobileNet-100	12,447	2.5958	91.0807
0.25 MobileNet-100	4,795	1.0000	99.7302

TABLE II. F_1 SCORE OF DRIVER IDENTIFICATION FOR THE PROPOSED SYSTEM

Driver	Precision	Recall	F_1 score
Driver 1	1.0000	0.9858	0.9928
Driver 2	0.9951	0.9903	0.9927
Driver 3	0.9964	0.9175	0.9553
Driver 4	0.9966	0.9932	0.9949
Driver 5	0.8273	1.0000	0.9055
Driver 6	0.6388	0.9958	0.7783
Total	0.9973	0.8922	0.9418

average pooling layer instead of a max pooling layer. It directly outputs the spatial average of feature maps from the last multi-convolutional layers [18]. Global average pooling can regularize structures, which prevent overfitting for universal structures. The augmented data as shown in Fig. 10 is used as follows. We assume that each driver spends short time around within 2 minutes in the registration of the system. We only train facial regions, like the input of Fig. 9 (c), taken in the driving simulator. Since features about the facial region contain limited information compared to other public data sets, like ImageNet dataset [3] and COCO dataset [19] taken in various environments which affect a wide variety of intensity levels, we use a small number of augmented images. Hence, we collect and use a small number of augmented images, which means around 100,000 images for the validation and training dataset, and around 9,000 images are for the test dataset. The deep learning library uses Keras with TensorFlow in Anaconda 3.0, and the laptop has Intel core i7-6700HQ (2.60Hz) and 16 GB of RAM.

3) Inference Process

After the off-line training is performed to identify drivers, we can get the optimal weights for the network. Thus, the CNN model to load the optimal weights is the same as the training model, and the forward network is implemented as shown in Fig. 9. The network in Fig. 9 is based on MobileNet, and each MobileNet layer consists of 6 layers, the depthwise and pointwise convolutional layers, two batch normalization layers and two ReLU layers.

In order to implement the real-time system in Fig. 9 (d), we should consider the computational cost. MobileNet reduces the computational cost by separating the standard convolution to the depthwise and pointwise convolution, and it uses width multiplier α to control the computational complexity. Specifically, the reduced computational cost C_{re} of each layer using above method is controlled as follows [4]:

$$C_{re} = \alpha \cdot I_{ch} \cdot S_F \cdot \{(\alpha \cdot O_{ch} - 1) \cdot (S_k - 1) - 1\} \quad (5)$$

where I_{ch} and O_{ch} denote the number of input and output channels, and S_K and S_F denote the kernel sizes of an input image and the feature maps, respectively. In addition, image resolution affects the computational cost. Thus, we implement the small interference network with a narrow width and low-resolution like the training process.

III. EXPERIMENTAL RESULTS

The proposed system was analyzed in terms of two viewpoints: the training process and the inference process.

A. Analysis of Training Process

Table I shows results comparing standard models of AlexNet [3] and MobileNet [4] to the proposed network models with width multiplier $\alpha \in \{1, 0.75, 0.5, 0.25\}$ and resolution multiplier $\rho \in \{224, 100\}$. In the existing paper [3]-[5], there was a trade-off between the memory resource and the accuracy for each same model. However, the accuracy did not drop off smoothly as the width size and input resolution change. This was because the training environment and real inference environment were quite similar, and hence, it did not cause significant performance degradation.

Table II represents the precision, recall and F_1 score [20] of the proposed system. We used 6 as the number of drivers who need identification. Generally, private cars (in family cars, 4 to 5 people are sufficient) did not require very large number of registered drivers, and hence, 6 was a sufficient number. The proposed system had the average F_1 score of 0.9418 as shown in Table II. The driver 5 and 6 had lower F_1 score compared to the others because the amount of data was reduced by about half as the glasses were worn. Also the proposed system cropped the only facial region without most parts of forehead and chin, the appearance of the glasses had a big effect on the appearance. Therefore, the F_1 scores of driver 5 and 6 were relatively lower than the others.

B. Analysis of Inference Process

In Table I, it was related to the memory resource and accuracy. By reducing the memory usage, it affected the reduction in hardware resources, which decreases the embedded-dependency and the computational complexity. The memory resource in the proposed system was calculated as follows:

$$N_{memory} = 4 \times (n_{out} + n_{train} + n_{non-train}) \quad (6)$$

where N_{memory} denoted the memory resource in bytes, n_{out} denoted the number of all output pixels, n_{train} denoted the number of trainable parameters such as weights of filters, and $n_{non-train}$ denoted the number of non-trainable parameters such as mean and variance in the batch normalization layer.

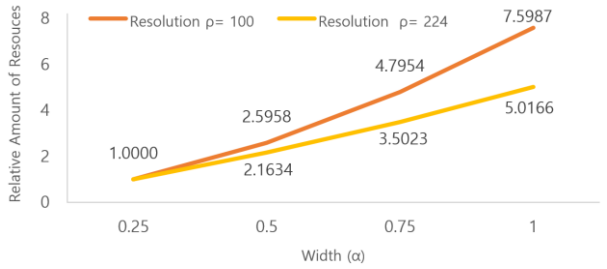


Fig. 11. Memory resource variation of MobileNet: the standard model is α=0.25 of each resolution model.

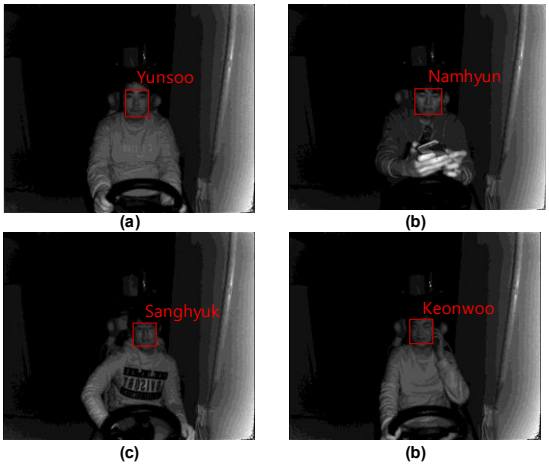


Fig. 12. Driver identification results: (a) driver 1 (Yunsoo) and (b) driver 2 (Namhyun) (c) driver 5 (Sanghyuk), and driver 6 (Keonwoo).

In order to reduce the hardware resource, we selected the optimized network of CNN, which has the smallest memory resource for the real-time system. Therefore, in Table I, we selected the smallest version, α=0.25 and ρ=100. The model had the memory resource of 4,795 KB; it was up to 49.0822 times smaller than the popular benchmark algorithm, 1.0 AlexNet-224.

Fig. 11 shows the memory resource variation of MobileNet. As the width increased, the variation of both model increased, but the model with small resolution, ρ=100, had larger variation. In this situation where n_{train} and $n_{non-train}$ were fixed, n_{out} only became relatively large to ρ=224. Thus, the variation rate of ρ=224 was relatively smaller. As shown in Fig. 12, the system identified the driver in real time.

IV. Conclusion

This paper proposed a novel method for driver identification system using the low-complexity-based CNN with background removal-based infrared data augmentation. First, we used a background removal-based infrared data augmentation with the pixel-wise relation. Second, we developed the driver identification system based on the tiny model of CNN for real-time processing by adjusting width multiplier and resolution. From the experimental results, the proposed system had the memory resource of 4,795 KB, which were up to 49.0822 times smaller than benchmark algorithms, and the average F_1 score of 0.9418 for the driver identification. Based on this study, it could be applied to the ADAS to provide a personalized environment.

Acknowledgment

This work is supported by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 17CTAP-C114672-02), and the Korea Institute of Energy Technology Evaluation and Planning(KETEP) and the Ministry of Trade, Industry & Energy(MOTIE) of the Republic of Korea (No. 20161210200560).

References

[1] N. C. Fung, B. Wallace, A. D. C. Chan, R. Goubran, M. M. Porter, S. Marshall, and F. Knoefel, "Driver Identification Using Vehicle Acceleration and Deceleration Events from Naturalistic Driving of Older Drivers," *In Medical Measurements and Applications*, 2017.

[2] European Commission, "Opinion 03/2017 on Processing Personal Data in the Context of Cooperative Intelligent Transport Systems (C-ITS)," *Working Party*, vol. 29, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *In Neural Information Processing Systems,* pp. 1106-1114, 2012.

[4] 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*, 2017.

[5] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," *arXiv*, 2017.

[6] N. Paramanandham, and K. Rajendiran, "Infrared and visible image fusion using discrete cosine transform and swarm intelligence for surveillance applications," *Infrared Physics & Technology*, vol. 88, pp. 13-22, 2018.

[7] C. Cortes, V. Vapnik "Support-vector Networks," *Machine Learning*, vol. 20, no. 3, pp 273-297, 1995.

[8] Microsoft Kinect for window SDK 2.0, Available:https://www.microsoft.com/en-us/download/details.aspx?id=44561

[9] P. KaewTraKulPong, and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," *Video-based Surveillance Systems,* vol. 1, pp. 135-144, 2002.

[10] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction," *In International Conference on Pattern Recognition*, 2004.

[11] A. B. Godbehere, A, Matsukawa, and K. Goldberg "Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive," *In American Control Conference*, 2012.

[12] R. C. Gonzalez, and R. E. Woods, "Digital Image Processing," *Prentice Hall*, 1977.

[13] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features*," In Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.

[14] T. F. Chan, C. K. Wong, "Total Variation Blind Deconvolution *Image Processing*, vol. 7, no. 3, pp. 370-375, 1998.

[15] A. Asundi, and Z. Wensen, "Fast phase-unwrapping algorithm based on a gray-scale mask and flood fill," *Applied Optics*, vol. 37, no. 23. pp. 5416-5420, 1998.

[16] J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.

[17] S. Kim, G. H. An and S. J. Kang, "Facial Expression Recognition System Using Machine Learning", *In International SoC Design Conference*, 2017.

[18] M. Lin, Q. Chen, and S. Yan, "Network in Network," *arXiv*, 2013.

[19] T. Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," *In European conference on computer vision*, pp. 740-755, 2014.

[20] Y. Yang and X. Liu, "A re-examination of text categorization methods," *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 42-49, 1999.