Body Posture and Face Orientation Estimation by Convolutional Network with Heterogeneous Learning

Kaoruko Okuno, Takayoshi Yamashita, Hiroshi Fukui,
Shuzo Noridomi, Koji Arata, Yuji Yamauchi, Hironobu Fujiyoshi
Chubu University
Kasugai, Aichi, Japan
{okuno, fhiro}@vision.cs.chubu.ac.jp, {yamashita,hf}@cs.chubu.ac.jp

***Abstract*—Autonomous driving system switches over to a manual driving mode by human when the system is not able to drive itself. The system has to constantly monitor whether the driver can drive the vehicle by the driver's posture and face orientation. Conventional methods for estimating posture and face orientation perform feature extraction and recognition for each task, and thus require an appreciable amount of processing time. In this paper, we propose a method that performs multiple tasks by Deep Convolutional Neural Network (DCNN) with heterogeneous learning, by sharing the feature extraction process. The body posture and face orientation estimation can be performed simultaneously. In evaluation, we have achieved a high accuracy of 98% in body posture estimation, and 91% in face orientation estimation. The processing time for a single image has been 2.6 ms when the we employ a GPU, and 34.1 ms in CPU. We confirm that proposed method can perform body posture and face orientation estimation in real time.**

I. Introduction

When the autonomous driving system is unable to continue driving itself by any reason, it switches over to a manual driving mode by human. To perform this turn over process, it has to constantly monitor whether the driver can drive the vehicle. The body posture and face orientation estimation is important methods to realize driver monitoring system. Conventional methods for measuring posture and face orientation perform feature extraction and recognition for each task, and it requires an appreciable amount of processing time. In this paper, we employ heterogeneous learning[2] into a training of DCNN. It is possible to perform body posture and face orientation estimation in a single DCNN. It shares the feature extraction process for the body parts and face orientation estimation and it can be performed simultaneously. In the heterogeneous learning of the proposed method, the regression values of coordinates of body posture and face orientation are output from the output layer in DCNN. As a result, the proposed method results in a compact network architecture and can perform processing in real time, even on a CPU.

II. Related Works

Shotton et al. proposed a method that employs the random forests method [6] to detect the body posture from depth images and detect the center of gravity of each part [4]. This method constructs random forests from depth images with learning samples where each part is correctly labelled with a different color, and branches to the left or right according to a threshold value using the differences in distance of two separate pixels as feature quantities. The body posture is then ascertained by finding the center of gravity for each part label. Fanelli et al. proposed a method that uses regression forests[7] to estimate face orientations from depth images [5]. In this method, large quantities of depth images depicting a variety of face orientations and expressions were produced artificially and used as input. Since their method used depth images, it could accommodate differences between individual subjects, and since it used artificially generated images, it was also robust against changes of face orientation and expression. Toshev proposed a method that performs skeleton detection using DCNN from RGB images [3]. Human beings can infer the positions of hidden joints based on their movements and positional relationships to other parts. The convolution and pooling processes in DCNN can obtain an overall grasp of the skeleton's positional relationships. This makes it possible to roughly estimate the positions of hidden joints. DCNNs with the same structure are connected in series to obtain a rough estimate of the entire skeleton, after which each estimated point can be used as input to obtain a more accurate skeleton estimation. Body posture estimation performed using the machine learning methods like [5][6] can be used as a natural user interface in video games. But, it is not robust to self-occlusion. Since face orientation estimation requires a large face image, it can be difficult to estimate from whole-body image. The method based on DCNN like [3] is efficient for these tasks. However, handling multiple DCNNs leads to increased processing costs. Therefore, independent DCNNs for body posture and face orientation estimation are very inefficient process from the viewpoint of real-time processing.

III. Proposed Method

We employ heterogeneous learning[2] into a training of DCNN. It is possible to perform body posture and face orientation estimation in a single DCNN. We introduce the

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

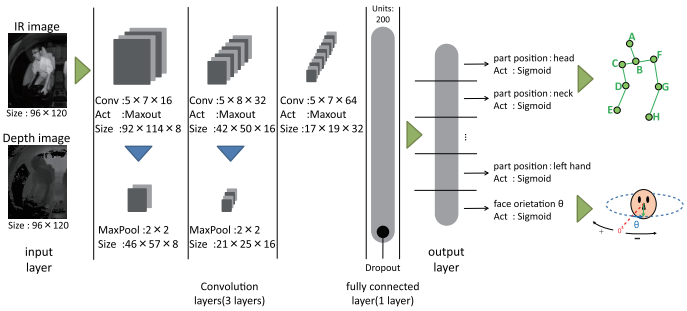


Fig. 1. Structure of DCNN incorporating heterogeneous learning.

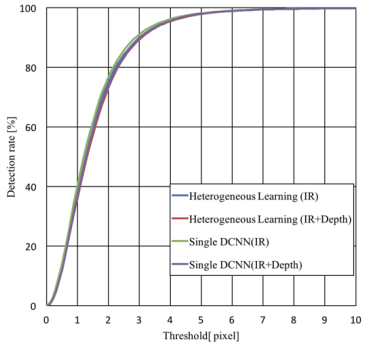


Fig. 2. Evaluation of body posture estimation.

DCNN architecture and heterogeneous learning method in this section.

A. Body posture and face orientation estimation

To obtain robustness against illumination change such as a vehicle is driven into a building or tunnel, or when driving at night, we introduce IR and depth images as input. These images are obtained using a camera mounted near the interior rear view mirror with a full view of the driver's seat and front passenger seat. The camera produces images with a resolution of 640×480 pixels. In the proposed method, the vicinity of the driver's seat is first cropped and resized to a 96×120 pixel image that is input to the DCNN as shown in Figure 1. The DCNN consists 3 convolution layers and 2 fully connected layers. The 1st convolution layer has 16 filters. The filter size is 7×5. The activation function is maxout. After applying this 1st layer, the output feature maps are 8 and the size is 114×92. These feature maps feed to pooling process. Feature maps is down sampling to 57×46 by maxpooling. The 2nd convolution layer receives this feature maps and convolve 32 filters with 8×5. The number of output feature maps in this layer are 16 with 50×42. These feature maps are also applied maxpooling. The size of feature maps is 25×21. The feature maps are flattened to input fully connected layer. The fully connected layer has 200 units. The activation function of this layer is ReLU. There are 17 output units, comprising 8×2 part positions and one face orientation angle. The driver part positions correspond to the x and y coordinates of 8 body parts: head, neck, right shoulder, right elbow, right hand, left shoulder, left elbow and left hand. The face orientation angle corresponds to the orientation of the driver's head in the left-right direction (yaw angle), with an angle of zero indicating that the driver is looking straight ahead. From the driver's point of view, positive angles are to the right, and negative angles are to the left.

B. Heterogeneous Learning

Heterogeneous learning is a learning and recognition method that can handle multiple tasks in a single network. In general, when performing multiple recognition tasks, the number of DCNN that has to be constructed is proportional to the number of tasks. This is inefficient for real applications. In heterogeneous learning, multiple tasks can simultaneously train and run by a single network, and so the computational cost does not increase greatly when the number of tasks increases. In the heterogeneous learning of the proposed method, the regression values of the coordinates of body parts and face orientation estimation are output from the output layer. We define error functions for each task as follows.

1) Error function of the skeleton detection task: Since the body posture estimation outputs the x and y coordinates of 8 body positions, the learning error E_s is given by Equation (1):

$$E_s = \sum_{n=0}^{N} ||L_n - O_n||_2^2 \tag{1}$$

where L_n is a teacher signal, O_n is an output value, and N is the number of body part positions.

2) Error function of the face orientation estimation: Since there is only one output unit corresponding to the face orientation, the learning error E_g is given by Equation (2):

$$E_g = (L - O)^2 \tag{2}$$

Therefore, the total error E is given by Equation (3):

$$E = \alpha E_s + (1 - \alpha) E_g \tag{3}$$

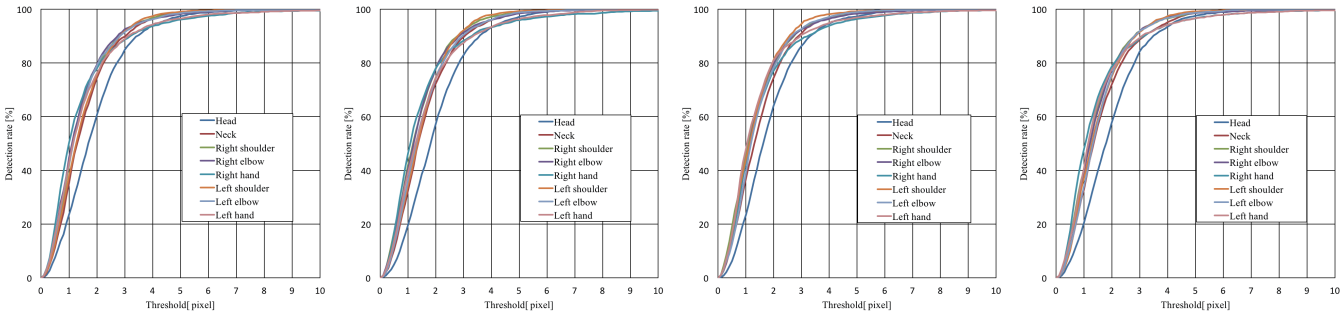
where α is a weighting factor for the body posture and face orientation estimation learning errors that can be used as a parameter to determine which of these tasks is to be prioritized. We set a value of $\alpha = 0.5$, giving the two tasks equal weighting. We created mini-batches by selecting multiple learning samples from the learning data set. In the mini-batch learning method, which is widely used in DCNN learning, M learning samples are selected at random from the learning samples and input to the DCNN. The error E from these samples is determined, and the error back-propagation of Equation (4) is used to update the parameter of DCNN w. Here, η is a learning coefficient, and for this study we set a value of $\eta = 0.001$.

$$w \leftarrow w - \eta \frac{\partial E}{\partial w} \tag{4}$$

IV. Experiments

A. Evaluation metrics

We evaluated the body parts estimation by using a Euclidean distance and the percentage of correct parts (PCP) metric as



(a)Heterogeneous (IR) (b)Heterogeneous (IR+depth) (c)Proposed DCNN (IR) (b)Proposed DCNN(IR+depth)

Fig. 3. Evaluation of each part in body posture estimation.

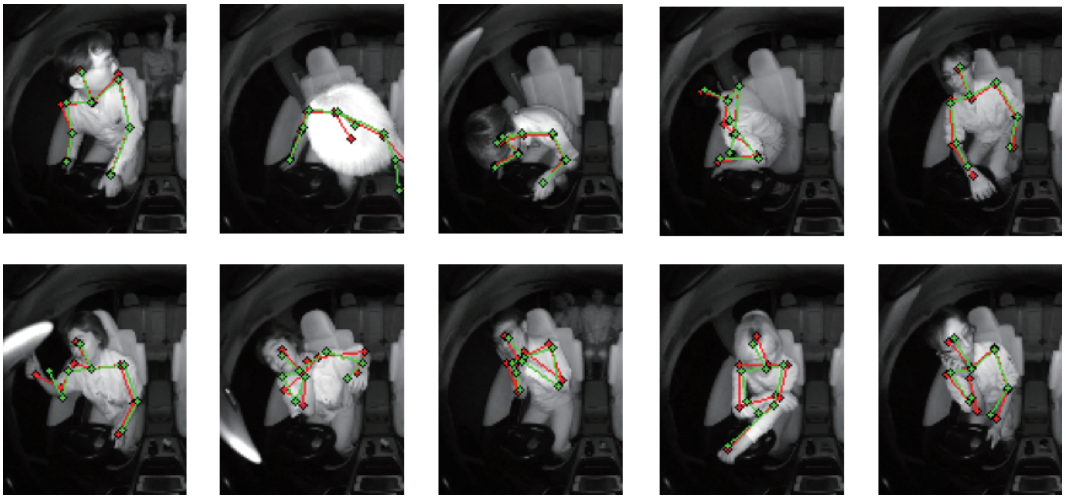


Fig. 4. Example of Body posture estimation.

used by Toshev and Szegedy [3]. First, to evaluate Euclidean distances, we determined the Euclidean distance between the output value and the correct label, recording a successful detection if this distance was less than a threshold value, and a failed detection otherwise. We varied the threshold value from 0 to 10 pixels, allowing us to confirm the change in accuracy. For the PCP evaluation, we focused on the positions of two neighboring body parts, and the detected position was collected as successful when the estimation error for each body part was less than half the Euclidean distance between these parts. In face orientation estimation, we determined the difference between the correct label and the output value, and collected a successful estimation when the difference between these angles was less than a threshold value, or an unsuccessful estimation otherwise. We varied the threshold value from 0 to 20 degree to compare the results obtained with varying levels of accuracy. The data set we used consisted of 32,914 sample images from 12 test subjects, of which 30,000 were used for learning, and 2,914 were used for evaluation. We concentrated on scenes with a variety of posture variations, such as where the driver was looking away from the road or using a smart phone, opening the sun visor, or touching his or her face.

B. Performance of body posture estimation

Figure 2 shows the body posture estimation results. it was possible to achieve the same level of detection accuracy as with a single DCNN. Also, when the input of each method was provided as IR images or IR+depth images, we found that better precision was achieved using IR images. Figure 3 shows the evaluation results for each body part. In all methods, with a threshold value of 2 pixels, the lowest accuracy (about 60%) occurred at the head. This is probably because the correct label for the head is at its center, and there was some variation in the correct labels for this part. Figure 4 shows some examples of skeleton detection results obtained by the proposed method from IR input images. The red points are the correct labels, and the green points are the output values. As this figure shows, The DCNN trained by heterogeneous learning enables estimate the head position, albeit with an offset from the correct label of the head, so there was no major loss of accuracy. It can also be seen that the posture was correctly estimated even the part occurs self-occlusion, such as the shoulders, neck and hands.

Furthermore, better accuracy was achieved for parts on the right side of the body in all methods. This is because the way the camera was positioned meant that the parts of the driver on the left side appeared larger and thus tended to depart from the correct label positions by a larger margin. Table 1 shows

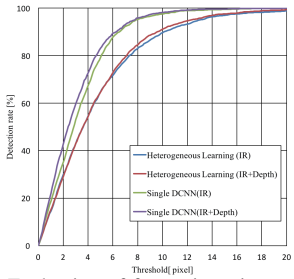


Fig. 5. Evaluation of face orientation estimation.

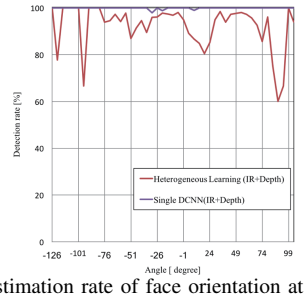


Fig. 6. Estimation rate of face orientation at each angle.

the results of PCP evaluation to assess the skeletal results without the effects of apparent size in the camera image. In all methods, similar detection accuracies were achieved for the left and right sides. Also, in all methods, the accuracies were better for the upper arms than for the lower arms. This is because, as can be seen in Figs. 3 and 4, the hands moved a lot more, making it harder to detect their positions compared with the elbows and shoulders.

C. Performance of face orientation estimation

Figure 5 shows the face orientation estimation evaluation results. With a threshold value of 10 degree, the recognition rate was about 10% lower than when using a single DCNN. Figure 6 shows the recognition rate achieved when evaluating the face angle in 5 degree increments with the threshold value set to 10 degree. The vertical axis shows the recognition rate, and the horizontal axis shows the face angle. This figure shows that the accuracy was lower at face angles greater than 70 degree, between 0 degree and 30 degree, and below -90 degree. Figure 7 shows some images of angles for which low accuracy was achieved. First, when the face angle is greater than 70 degree or less than -90 degree, the driver is performing movements that deviate significantly from the usual posture as shown in Fig. 7(a) and (c), and for which there are few learning samples. Also, when the face angle is between 0degree and 30 degree, the driver performed movements such as leaning forwards while continuing to look to the front as shown in Fig. 7(b), or shifting the body forwards or backwards, resulting in changes of other angles besides the yaw angle that was recognized in this experiment. It is therefore expected that better accuracy could be achieved by considering other angles besides the yaw angle in samples that depart significantly from the usual forward-facing posture.

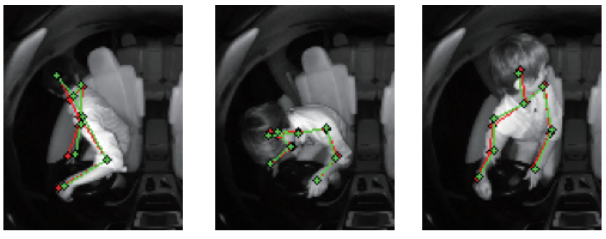


Fig. 7. Example of Body posture estimation.

D. processing time

Table 2 shows the time taken to process a single image. For the GPU, we used a GTX 1080, and for the CPU, we used an Intel Core i7-4790 3.60 GHz. According to this table, the total processing time per image for a single DCNN would be 4.4 seconds on a GPU or 67.0 ms on a CPU. On the other hand, the processing time of proposed method is 2.6 ms on a GPU and 34.1 ms on CPU. With the proposed method, we reduced these times to 1.8 ms on a GPU and 32.9 ms on a CPU. Even when run on a CPU, the proposed method is fast enough to perform real-time processing at 30 fps.

V. Conclusion

We have proposed introducing DCNN into heterogeneous learning to perform skeleton detection and face orientation estimation of vehicle drivers. By introducing heterogeneous learning, the proposed method results in a compact network architecture and can perform processing in real time, even on a CPU. In the future, we will study methods for driver behavior recognition that make use of the results obtained by estimation.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[2] X. Yang, S. Kim, and F. P. Xing, “Heterogeneous Multi-task Learning with Sparsity Constrain”, Advances in Neural Information Processing Systems 22, 2009.
[3] A. Toshev, and C. Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks”, Computer Vision and Pattern Recognition, 2015.
[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images”, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011.
[5] G. Fanelli, J. Gall, and L. Van Gool, “Real Time Head Pose Estimation with Random Regression Forests”, Computer Vision and Pattern Recognition, 2011.
[6] L. Breiman, “Random forests”, Machine learning, 2001.
[7] M. Dantone, J. Gall, G. Fanelli, L. Gool, “Real-time Facial Feature Detection using Conditional Regression Forests”, Computer Vision and Pattern Recognition, 2012.
[8] E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, “Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation”, IEEE Intelligent Transportation Systems Conference, 2007.