https://doi.org/10.20965/jaciii.2020.p0676

Paper:

Visualization Method Corresponding to Regression Problems and Its Application to Deep Learning-Based Gaze Estimation Model

Daigo Kanda, Shin Kawai, and Hajime Nobuhara

Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba
1-1-1 Tennoudai, Tsukuba, Ibaraki 305-8573, Japan
E-mail: {kanda@cmu., kawai@cmu., nobuhara@}iit.tsukuba.ac.jp
[Received February 20, 2020; accepted July 2, 2020]

The human gaze contains substantial personal information and can be extensively employed in several applications if its relevant factors can be accurately measured. Further, several fields could be substantially innovated if the gaze could be analyzed using popular and familiar smart devices. Deep learning-based methods are robust, making them crucial for gaze estimation on smart devices. However, because internal functions in deep learning are black boxes, deep learning systems often make estimations for unclear reasons. In this paper, we propose a visualization method corresponding to a regression problem to solve the black box problem of the deep learning-based gaze estimation model. The proposed visualization method can clarify which region of an image contributes to deep learning-based gaze estimation. We visualized the gaze estimation model proposed by a research group at the Massachusetts Institute of Technology. The accuracy of the estimation was low, even when the facial features important for gaze estimation were recognized correctly. The effectiveness of the proposed method was further determined through quantitative evaluation using the area over the MoRF perturbation curve (AOPC).

Keywords: CNN, eye tracking, Grad-CAM, regression problem

1. Introduction

The gaze contains substantial information on human sensory perception, which is often specific individual. Additionally, the gaze includes implicit internal information that cannot be understood from the facial expression alone. Consequently, gaze information would have broad applications, if it could be analyzed accurately.

Smart devices are popular throughout the world [1], and several people have the latest technology devices. Gaze estimation by a smart device can be expected to promote innovation in several fields. Since smart devices are generally portable, this would also allow gaze estimation technology to be used in various places.

Gaze estimation methods can be divided into model-based (estimating gaze using pupil orientation) and appearance-based (estimating gaze from an eye image) [2]. In this study, we focus on a method using a convolutional neural network (CNN) [3], which is an appearance-based method. CNN learns from a large data set consisting of facial images and gaze positions to estimate gaze. Using CNN enables the capture of minute features that are difficult for a person to find or understand. Furthermore, it is possible to robustly estimate a large number of patterns in an image by learning. Unfortunately, the reasoning behind the results is unknown because the CNN inference process is a black box [4]. Therefore, it is not possible to judge whether the model uses features that are useful for gaze estimation, such as eyes and faces. This can lead to incorrect inferences. Thus, in deep learning, model visualization is an essential factor in evaluating model performance.

In this study, we visualize representative deep learning-based gaze estimation [3], clarifying what image features contribute to a prediction. In [5], the gaze estimation model is visualized. In this study, the model's quantitative evaluation is performed and extended. We focus on the Grad-CAM [6] visualization method for CNN. Grad-CAM is a gradient-based method for determining where a CNN has looked and what it has estimated. We chose Grad-CAM because it is widely used and can rapidly implement any convolutional neural network. Grad-CAM visualize the models that output results with the greatest probability because it only visualizes the features using positive signs of the gradients. This means that Grad-CAM can only visualize features that increase the output of the model. We therefore propose to modify Grad-CAM's basic approach, making it suitable for a regression problem such as gaze estimation modeling. The proposed method can visualize features that make the output closer to the true value, not features that increase the output.

2. Related Works

In this section, we discuss previous research on visualization techniques and the gaze estimation methods analyzed in this study. In particular, we describe a deep learning-based gaze estimation model that is intended to be implemented on smart devices. We also describe re-

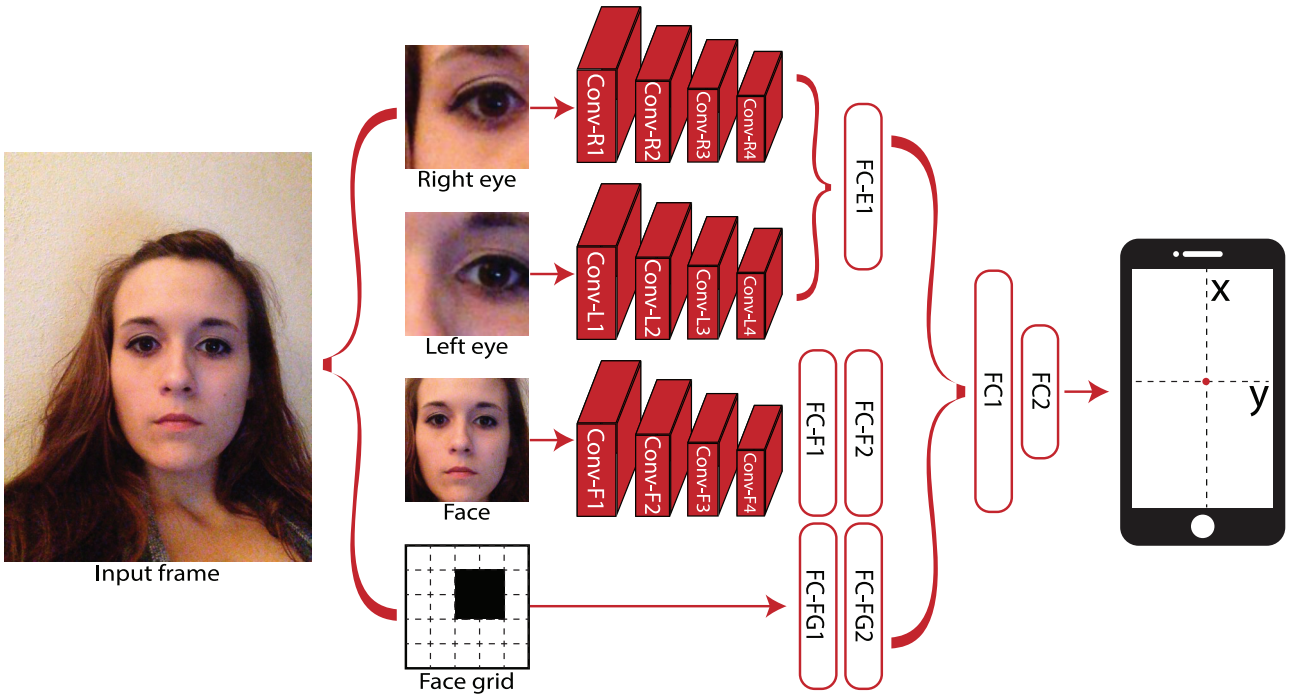


Fig. 1. Overview of iTracker. The input image is divided into four images, and input to the network. The output is distance, in centimeters, from the front-facing camera, which indicates the gaze position on the screen of a smart device.

lated works discussing the visualization method used in this study.

2.1. Gaze Estimation

Gaze estimation methods can be divided into two types: model-based and appearance-based [2]. Model-based approaches use a geometric model of the eye. This approach can be divided into corneal reflection-based and shape-based methods. In corneal reflection-based methods, the cornea reflects light, and eye features are estimated based on the reflected light. This method depends on an external light source, which limits its use. Shape-based methods infer gaze direction from observed eye shapes, such as pupil centers and iris edges. These approaches perform poorly with low image quality and variable lighting conditions because of the simplicity of the eye shape model.

Appearance-based methods use eye images as input and directly infer gaze. These methods can potentially work on low-resolution images, but require large amounts of user-specific training data. However, by using large amount of data, the model can generalize well to novel faces without needing user-specific training data. Another advantage of appearance-based methods is that they can handle unconstrained use of smart devices. They can estimate gaze without assumptions regarding geometric properties of the environment or the camera and user's relative positions. Several appearance-based methods have been proposed to estimate gaze in smart devices. Zhang et al. [7] proposed an algorithm that takes only the face image as input and performs two-dimensional and three-dimensional gaze estimation using a convolutional neural network with spatial weights applied on the feature maps. Huang et al. [8] collected an unconstrained dataset (Rice TabletGaze dataset) consisting of 51 subjects, each with 4 different postures and 35 gaze locations. Using a baseline algorithm based on the multilevel histogram of oriented gradient (HoG) features and the Random Forests regressor, they achieved a mean error of 3.17 cm. These studies develop gaze estimation methods and datasets, but do not implement their estimators on smart devices.

Krafka et al. [3] achieved a mean error of 1.71 cm and 2.53 cm without calibration using iTracker, a CNN for eye tracking, on a mobile device. Besides, they have implemented iTracker on smartphones (iOS), achieving speeds of 10–15 frames/s. **Fig. 1** shows overview of iTracker. The inputs of iTracker are the image of the face, the image of both eyes, and the grid display of the position of the face in the full image. The output is the position (x, y) of the gaze on the screen. iTracker was trained by GazeCapture, which contains almost 2.5 million frames of gaze data from over 1450 people, collected using crowdsourcing. The specific preparation procedure of GazeCapture is as follows. First, a dot appears at a random location on the screen for 2 s and the subject looks at the dot. Second, after 2 s, left and right indicators appear on the screen, and the subject taps 'left' or 'right.' This ensures that the subject's attention is directed at the screen. Image capture begins 0.5 s after the start of point display. The procedure is repeated 60 times, and then the subject is asked to change the screen orientation. GazeCapture considers scalability, reliability, and diversity. Thus, it is an effective data set for gaze estimation.

The internal state of deep learning-based gaze estimation methods is a black-box due to internal complexity.

Therefore the reason for the output cannot be explained, and there is a risk in terms of social implementation. In this paper, we propose a visualization method that can explain the gaze estimation model and attempt to explain the internal state of iTracker.

2.2. Visualization Techniques

Visualization provides human-level understanding of the underlying reasons for a CNN's output. The visualization method highlights the pixels or regions that contributed to the estimation in the input image. Making the black box process explicit based on the results of the visualization increases confidence in predictions and makes it easier to redesign of the CNN's architecture and to cleanse datasets to improve performance. Previous research on visualizing CNN models [9–11] has visualized CNN predictions by visualizing pixels that contributed to the output by back-propagating the output to the input. Such approaches do not distinguish between classes. Gaze estimation models typically have multiple classes or outputs; thus, in this study, we focus on Grad-CAM, which can visualize features related to specific outputs.

Grad-CAM is a visualization method that uses the gradient of the feature map of the convolutional layer. It is widely used because it can be implemented more easily and can be adapted to a more diverse variety of models than back-propagation methods. To use Grad-CAM, we first compute gradients of the classification score y^c of class c for the final convolution layer feature map A^k_{ij}. i and j indicate the location of pixels in the feature map. k indicates the channel. The gradient is calculated using the backpropagation method and averaged using global average pooling (GAP) to obtain the importance weight α^c_k for each channel, which is defined as

$$\alpha^c_k = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^k_{ij}}. \quad \ldots\ldots\ldots\ldots \quad (1)$$

Since gradients change by y^c when A^k_{ij} changes slightly, y^c increases when a pixel with a large gradient changes, that is, when the probability that a model will be classified into class c increases. As the gradients decrease, y^c decreases and can be considered a feature of other classes. When GAP averaging is used, α^c_k indicates how important each feature map is to the target class c.

Next, the weighted sum of the weight and the feature map is calculated, and the activation function Rectified Linear Units (ReLU) is applied to the weighted sum in order to hide the part that has a negative effect on the class decision. The relevant formula is

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha^c_k A^k\right). \quad \ldots\ldots\ldots \quad (2)$$

Using Grad-CAM, we can see where in the image the classification score of class c was determined. Grad-CAM can be visualized easily for problems of probability output such as multi-class classification; however, it cannot be used for problems of infinite output range, such as regression. In this paper, we propose a visualization method appropriate to regression problems such as a gaze estimation model, based on Grad-CAM.

3. Grad-CAM Variant Corresponding to Regression Problems

The proposed visualization technique can be used for regression problems with two-dimensional (x,y) outputs. **Fig. 2** shows an overview of the proposed method. Conventional Grad-CAM calculates the gradient by differentiating the classification score y^c directly from the feature map A^k_{ij}. A positive gradient corresponds to an increase in the probability of occurrence and a larger y^c. Conversely, a negative gradient reduces y^c. Therefore, in Grad-CAM, negative gradients are eliminated by multiplying by ReLU, because inputs that reduce y^c are not the basis of the model's judgment.

A gaze estimation CNN's output can range up to infinity, in contrast to a probability model whose outputs are between 0 and 1. Instead, the output estimation value shows improved accuracy when the difference from the true value is smaller. In conventional Grad-CAM, features that increase the output are visualized in a multi-class classification problem. However, in the regression problem, even if the features that increase the output are visualized, the reason for the model's judgment is not fully clear. The true value may be greater than the estimated value. If so, then the estimated value must decrease, not increase, to approach the true value. Therefore, features that reduce the output must be visualized in this case. Conversely, if the estimate is less than the true value, features that increase the output must be visualized. Moreover, while the outputs x and y can be visualized in Grad-CAM, determining how much each x and y value contributed to the estimation is difficult. Thus, we want to overlay the x and y heat maps to determine which position in the image contributes the most to the estimation. To solve this problem, in place of the classification score y^c, we propose the inverse of the distance between the estimated value (x,y) and the true value (x',y'), defined as

$$d = \frac{1}{\sqrt{(x-x')^2+(y-y')^2}}. \quad \ldots\ldots\ldots \quad (3)$$

In this case, the importance weight α for each channel is

$$\alpha^d_k = \frac{1}{Z}\sum_i\sum_j \frac{\partial d}{\partial A^k_{ij}}. \quad \ldots\ldots\ldots\ldots \quad (4)$$

The final heat map is calculated as follows,

$$L_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha^d_k A^k_{ij}\right). \quad \ldots\ldots \quad (5)$$

To visualize the features activated when the difference between the estimated value and the true value becomes small, we introduce the distance between the estimated value and the true value. Since Grad-CAM visualizes only the features in which the output is increased by multiply-

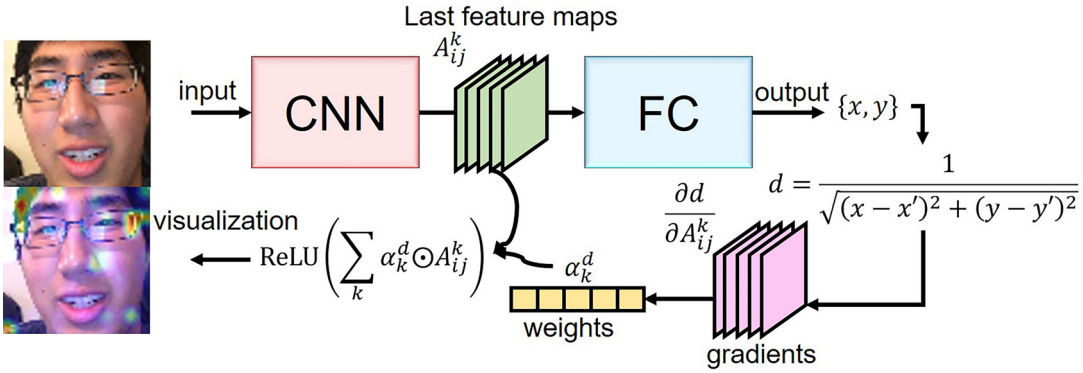


Fig. 2. Overview of the proposed method: Grad-CAM computes gradients of the output of the model with respect to the feature map of the final convolutional layer, but the proposed method computes gradients of the reciprocal of the distance between the estimated value and the true value, with respect to the feature map of the final convolutional layer, for use in regression problems. GAP and ReLU are calculated in the same manner as Grad-CAM.

ing by ReLU, the feature in which d increases shows the feature in which the distance decreases. Gradients for d with respect to A_{ij}^k are

$$\frac{\partial d}{\partial A_{ij}^k} = \frac{\partial d}{\partial x}\frac{\partial x}{\partial A_{ij}^k} + \frac{\partial d}{\partial y}\frac{\partial y}{\partial A_{ij}^k}. \quad \ldots\ldots\ldots \quad (6)$$

$\partial x/\partial A_{ij}^k$ and $\partial y/\partial A_{ij}^k$ are gradients for the heat maps of x and y, respectively; these gradients are the same as Grad-CAM's basic approach. However, depending on the size of $\partial d/\partial x$ and $\partial d/\partial y$, the component of either x or y more strongly affects the heat map. $\partial d/\partial x$, $\partial d/\partial y$ is

$$\frac{\partial d}{\partial x} = -\frac{x - x'}{\left((x-x')^2 + (y-y')^2\right)^{\frac{3}{2}}}, \quad \ldots\ldots \quad (7)$$

$$\frac{\partial d}{\partial y} = -\frac{y - y'}{\left((x-x')^2 + (y-y')^2\right)^{\frac{3}{2}}}. \quad \ldots\ldots \quad (8)$$

When $x > x'$, the gradients are negative, and the heat map for x shows where to look to reduce x because the features that reduce x are activated. Moreover, since the gradients are positive for $x < x'$, we can see where to look to increase x. The same is true for $\partial d/\partial y$. When the difference between the estimated value and the true value is large, the associated feature appears more prominently on the heat map. The proposed method thus enables visualization of where the estimated value is calculated when the difference between the true value and the value estimated by the model is large.

4. Experiments

In this section, we first compare the proposed method with Grad-CAM and show the effectiveness of the proposed method. Next, we visualize the iTracker model with the proposed method and clarify the features that contributed to its estimation.

4.1. Setup

We implemented iTracker in the Keras environment and set it to learn using small GazeCapture, consisting 48,000 training data and 5,000 test data. Each data image is 128×128, and the batch size is 32. The optimizer Adam (lr $= 0.003$, beta_1 $= 0.9$, beta_2 $= 0.999$) was used. In the original iTracker, the weights of the convolution layer of the right eye and the left eye are shared. However, in this study, weights are not shared, in order to visualize input images of each eye separately. The trained model's distance error is 2.28 cm. In Section 4, we discuss the use of four types of gradients to compare the proposed method with Grad-CAM. The gradients of the output x with respect to the last feature maps and the gradients of the output y with respect to the last feature maps are typical approaches of Grad-CAM. We consider these approaches ineffective because they can not visualize features that contribute to the reduction of the outputs. The sum of the gradients of x and y is also computed for comparison with the proposed method, and is defined as

$$\frac{\partial o_{xy}}{\partial A_{ij}^k} = \frac{\partial x}{\partial A_{ij}^k} + \frac{\partial y}{\partial A_{ij}^k}. \quad \ldots\ldots\ldots\ldots \quad (9)$$

Grad-CAM using this gradient can visualize the features of x and y at the same time; however, when the one gradient is large, the feature of the other gradient appears relatively small. The proposed method of changing the computing gradients with respect to last feature maps was described in Section 3.

4.2. Quantitative Evaluation

In this section, we evaluate the proposed method using AOPC [12]. AOPC is a greedy iterative procedure that measures how the output changes as it progressively removes important features from the input image. The AOPC formula is

$$AOPC = \frac{1}{L+1}\left\langle \sum_{k=0}^{L} f\left(x_{MoRF}^{(0)}\right) - f\left(x_{MoRF}^{(k)}\right)\right\rangle_{p(x)}, \quad (10)$$

where $\langle\cdot\rangle_{p(x)}$ denotes the average over all the images in the data set. $f(x)$ indicates model function. $x^{(k)}_{MoRF}$ represents perturbed images and is defined as

$$x^{(0)}_{MoRF} = x, \quad \ldots\ldots\ldots \quad (11)$$

$$\forall\, 1 \leq k \leq L : x^{(k)}_{MoRF} = g\left(x^{(k-1)}_{MoRF}, r_k\right). \quad \ldots \quad (12)$$

Where g is the function that removes information at a specified region r (i.e., a single pixel or local neighborhood) within the image $x^{(k-1)}_{MoRF}$. As k increases, the region is progressively replaced by randomly sampled values from a uniform distribution according to an ordered heat map. A Heat map ordering where the most sensitive regions are ranked first implies a steep variation of the MoRF, and thus, a larger AOPC. AOPC was computed using 5,080 images of GazeCapture. To reduce random effects, AOPC was repeated 10 times. For each AOPC, we perturbed the first 40 regions, replacing for the 10×10 neighborhood.

Figure 3 show the results of replacing regions from images randomly. Since iTracker is given three images as input, we compared cases where only the face image was perturbed and where one eye was perturbed. AOPC was also computed for each output (x, y) of the model. Replacing regions of the face image input most significantly affects the output of the model. By contrast, the left eye and right eye images do not significantly affect the output. Thus, the model seems to estimate gaze mainly using face image. Similar results have been reported elsewhere [3]. By removing input components, a model without eye input shows good results. Furthermore, and perhaps surprisingly, the AOPC values of the left eye were smaller than that of the right eye. The AOPC values of output y were likewise found to be less than x. It is believed that this is because the variance in the y-axis direction is small due to the constraints of the data set, and the learning is therefore biased.

Figure 4 show results of AOPC relative to random selection. Each graph shows the results of computing AOPC by each Grad-CAM method and the proposed method when input images are perturbed correspondingly. The upper part of the figure shows the AOPC for output x, and the lower part of the figure shows the AOPC for output y. Overall, from the viewpoint of the frequency of the highest score, compared with other methods, the proposed method exhibits higher or comparable AOPC values. The proposed method shows the highest result (three times at (a), (b), and (c)). In particular, the proposed method exhibits higher AOPC values with reference to output x compared with the other methods, which perturbs the face input image (a). We believe that the proposed method is effective in visualizing the model, because the face image is considered to contribute most to the output. In contrast, the AOPC for the output y when the face is perturbed is lower in the proposed method than in the other methods. The proposed method uses the reciprocal of the distance. If the effect of output x is large, features related to output y may have not been visualized well. AOPC values that perturbed right eye image were reduced for all methods. This shows that visualization of right eye images does not indicate the features that most contributed to the estimation.

Table 1 summarizes the results of difference of heat map image size by visualization method. Good heat maps should highlight the relevant regions with low noise. In terms of complexity, the file size of the visualized heat maps should be smaller than noisy ones. Heat map sizes visualized by each method average a size of 34,241 images. Heat maps visualized by proposed method are smaller than other methods in terms of the face images.

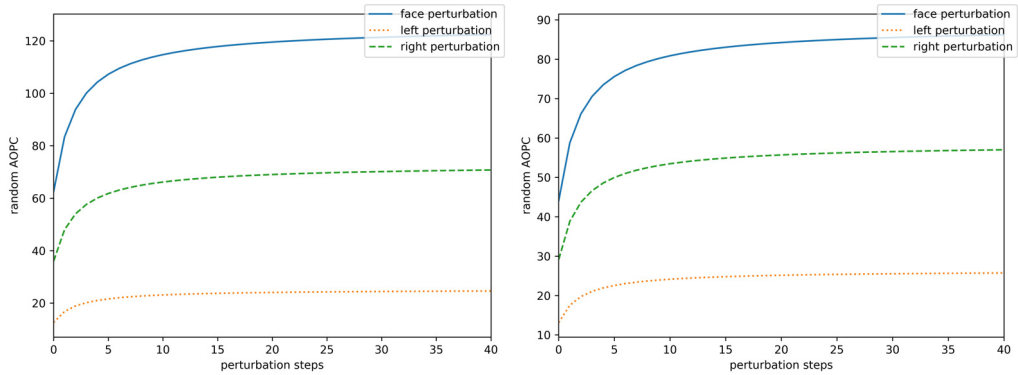
4.3. Visualization Experiments and Discussion

4.3.1. Results of Proposed Method

Examples of the visualization results are shown in **Fig. 5**. **Figs. 5(a)** and **(f)** are the original images used for the input. The images (b) through (e) and (g) through (j) show the heat map of the region by each of the four visualization methods. (a)–(e) show the results when the error between estimated value and true value is small and (f)–(j) show the results when the error is large. From the results of the proposed method, the edges of hair, glasses, and the nose are taken as features, and it is shown that the estimated value of the model approaches the true value by changes in that feature. (f)–(j) show the results when the error between the estimated value and the true value is large. The proposed method yields active features under the eyes and around the nose. It is considered that the feature around the eye can be caught because a change in the eye can estimate the gaze. However, the most active features appear on the nasolabial fold. The model judges the nasolabial fold as one of the edges representing the face and it appears in the heat map.

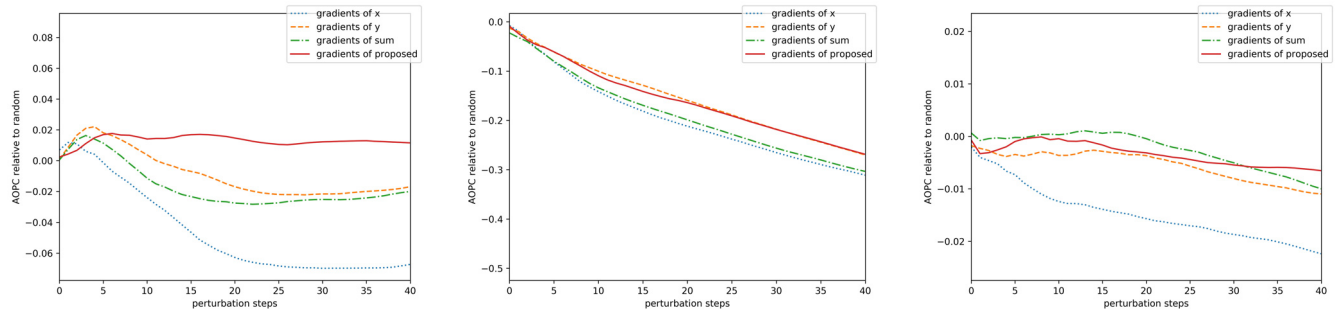
The results of the visualization of the eye image are shown in **Fig. 6**. As before, **Figs. 6(a)–(e)** show the results of the visualization of the right eye image. **Figs. 6(f)–(j)** show the results of the visualization of the left eye image. In Grad-CAM for X (**Fig. 6(b)**) and for Y (**Fig. 6(c)**), the edge of the eye and the conjunctiva are highlighted as features. These seem to be effective features for gaze estimation. However, the proposed method (**Fig. 6(e)**) does not show the eye as a feature, but instead shows the eyebrow as contributing to the estimation. Even if the model learns the characteristics of the eye, the estimated value is not close to the true value. Similarly, in the case of the left eye, the proposed method does not capture the eye and shows the feature of bringing the edge of the nose close to the true value.

Figure 7 shows the difference in visualization results with and without glasses. Some GazeCapture participants wore and removed glasses during the experiment. Visible features change with and without glasses. In **Fig. 6(a)**, the conjunctiva of the eye is caught as a feature. However, the edge of the glasses is shown as a feature when the glasses are worn. Since it is difficult to estimate the gaze using the edge of glasses, the error of the estimation value appears to have become large. In GazeCapture, people of various

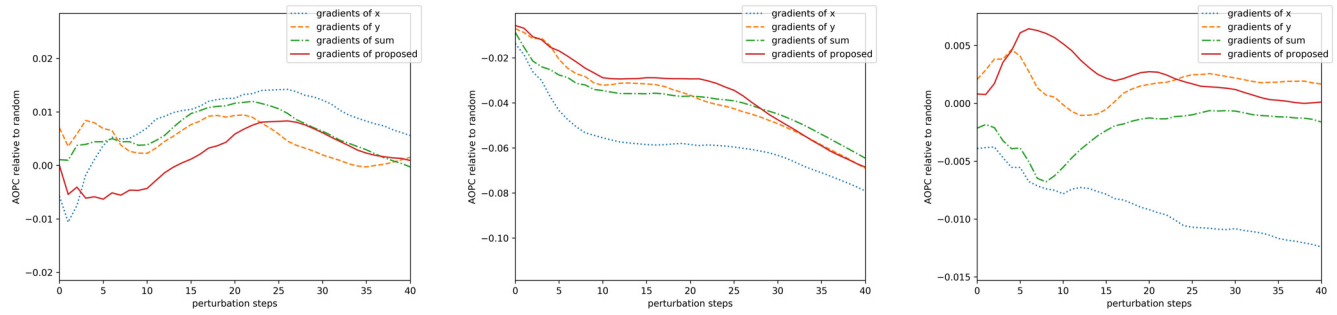


(a) Random AOPC with respect to output x (b) Random AOPC withe respect to output y

Fig. 3. Results of random AOPC: (a) indicates AOPC values with reference to output x, which perturb face and eye images randomly, (b) indicates AOPC values with reference to output y.



(a) AOPC with respect to output x: face perturbation (b) AOPC with respect to output x: right eye perturbation (c) AOPC with respect to output x: left eye perturbation



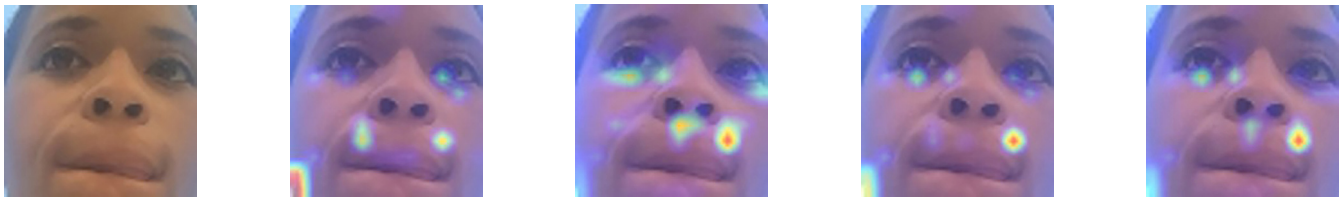
(d) AOPC with respect to output y: face perturbation (e) AOPC with respect to output y: right eye perturbation (f) AOPC with respect to output y: left eye perturbation

Fig. 4. Results of AOPC relative to random: (a)–(c) indicate AOPC values with respect to output x when perturbing the face, right eye, and left eye, respectively, according to the ordered heat map. (d)–(f) indicate AOPC values with respect to output y when perturbing the face, right eye, and left eye, respectively, according to the ordered heat map.

Table 1. Comparison of heat map complexity, measured in terms of file size (bytes).

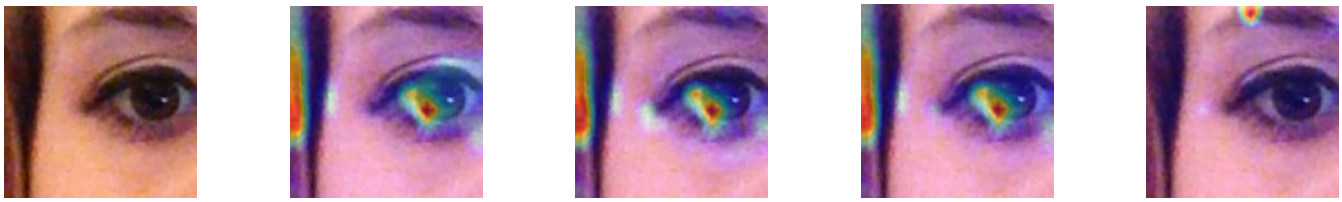
File size (bytes)	Right eye	Left eye	Face
x	7763.46	7256.73	9153.83
y	9184.52	8388.06	10274.86
Sum	8561.32	7628.39	9422.10
Proposed	7892.26	8088.58	9086.95

(a) Original image (b) Grad-CAM: X (c) Grad-CAM: Y (d) Sum gradients (e) Proposed method

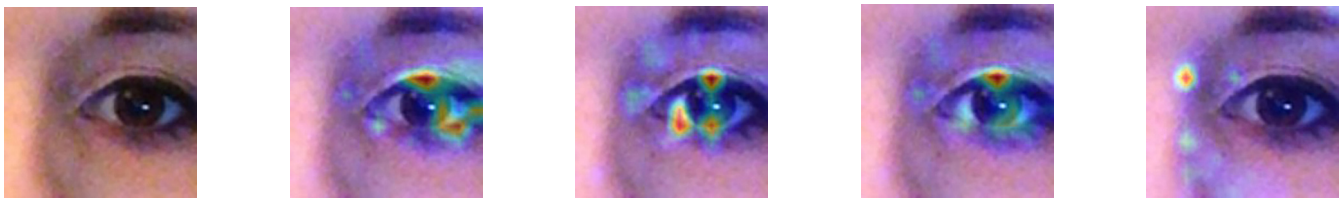


(f) Original image (g) Grad-CAM: X (h) Grad-CAM: Y (i) Sum gradients (j) Proposed method

Fig. 5. Visualization of iTracker input image (face) features using the proposed method: (a)–(e) heat map when the error between the estimated value and the true value is small, (a) original image used for input, (b) heat map for output x, (c) heat map for output y, (d) heat map with the sum of x and y gradients, and (e) heat map using the proposed method. (f)–(j) Heat map when the error between the estimated value and the true value is large, (f) original image used as input, (g) heat map on output x, (h) heat map for output y, (i) heat map with the sum of x and y gradients, and (j) heat map using the proposed method.



(a) Original image (b) Grad-CAM: X (c) Grad-CAM: Y (d) Sum gradients (e) Proposed method



(f) Original image (g) Grad-CAM: X (h) Grad-CAM: Y (i) Sum gradients (j) Proposed method

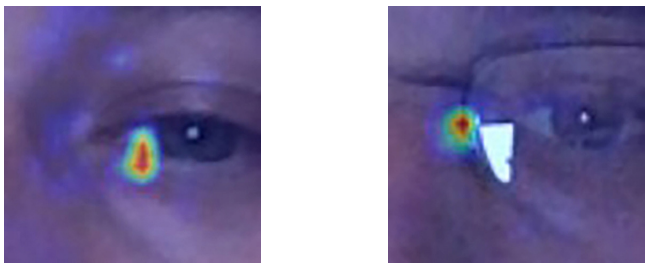
Fig. 6. Visualization of iTracker input image (eye) features using the proposed method: (a)–(e) visualization of features of the right eye, (b) heat map for output x, (c) heat map for output y, (d) heat map with the sum of x and y gradients, and (e) heat map using the proposed method. (f)–(j) Visualization of features of the left eye, (f) original image used as input, (g) heat map on output x, (h) heat map for output y, (i) heat map with the sum of x and y gradients, and (j) heat map using the proposed method.

appearances and conditions participated in the experiment to enhance the robustness of the data set. No conditions were set for glasses, and most of the data were obtained from people who did not wear glasses. Therefore, it is considered that the accuracy decreased because the model caught the features of the glasses when glasses were worn.

4.3.2. Discussion

From the results of several visualizations using the proposed method, it has become clear that iTracker makes estimates using the edges of facial features and parts, such as the glasses, eyes, nose, jaw, and eyebrow. The edge of the facial feature predicts the angle of the face, and it is considered that this might contribute to the gaze estimation. However, there was no correlation between the error of the estimate and the position of the model's gaze, and the accuracy greatly differed according to the image, even if a different image captured the same features. This is considered to be caused by the weak relationship between the features and the estimated value. The large error is considered to be included in the change of the estimated value, even though the estimated value changes when the features change. Since the model captures the features of the face, the correlation between the features and the estimated value is strengthened by expanding the data set, and an estimated value with a small error can thus be obtained.

The data set should be cleansed to improve the accuracy



(a) Without glasses (b) With glasses

Fig. 7. Differences in the visualization of features with glasses: (a) visualization without eyeglasses and (b) visualization with eyeglasses.

of the model. GazeCapture has various experimental environments because of the large number of participants. For example, it is necessary to adjust the data because even disregarding glasses, the data are biased. It is considered that technical improvement of the cropping of faces and eyes is necessary.

iTracker uses Apple's SDK to crop the face and eye images from input frames. In the present method, parts such as eyebrows are not considered, and some images do not include eyebrows. Differences in the features appearing in the input image can adversely affect the estimation. Therefore, it is necessary to crop the image while considering the features of the face.

iTracker shares the weights of the convolutional layers of both eyes. However, both eyes often catch the same features, and the images of left eye are not important to estimation, as clarified in Section 4.2. Therefore, it is thought that inputting only one eye image causes only small deterioration of estimation accuracy. Furthermore, the weights of the model cannot be reduced by removing the input image of one eye, but can reduce the cropping process of the input frame, leading to improved processing speed.

In the future, it will be necessary to examine the contribution of each input image to the model's output. We could rank the features for each input image using the proposed method. Furthermore, the features can be ranked in all the input images by considering the contribution of each input image.

5. Conclusions

In this paper, we proposed a method to extend the Grad-CAM visualization technique to visualize a gaze estimation model corresponding to a regression problem. We demonstrated that the proposed method is better than Grad-CAM by using AOPC and quantitative evaluation. Using the proposed method, the iTracker trained by GazeCapture was visualized and the features contributing to the output were investigated. The proposed method showed important features for gaze estimation such as eyes and nose; however it was clarified that there is no correlation between accuracy and features. We also suggested that cleansing the data, improving the cropping technique of the image, and reducing the input image of the eye tends to improve the accuracy and reduce the weight of the model.

Acknowledgements

This work was supported by the HAYAO NAKAYAMA Foundation for Science & Technology and Culture.

References:

[1] Pew Research Center, "Smartphone Ownership Is Growing Rapidly Around the World, but Not Always Equally," https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/ [accessed July 12, 2019]

[2] D. W. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.32, No.3, pp. 478-500, 2010.

[3] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye Tracking for Everyone," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2176-2184, 2016.

[4] Integrated Innovation Strategy Promotion Council Decision, "AI Strategy 2019 – AI for Everyone: People, Industries, Regions and Governments," https://www8.cao.go.jp/cstp/english/humancentricai.pdf [accessed July 15, 2019]

[5] D. Kanda, B. Wang, K. Tomono, S. Kawai, and H. Nobuhara, "Visualization technique for improving gaze estimation models based on deep learning," 6th Int. Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII 2019), 2019.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," arXive preprint, arXiv: 1610.02391, 2016.

[7] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's Written All over Your Face: Full-Face Appearance-Based Gaze Estimation," IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2299-2308, 2017.

[8] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "TabletGaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets," Machine Vision and Applications, Vol.28, No.5-6, pp. 445-461, 2017.

[9] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," Lecture Notes in Computer Science, Vol.8689, European Conference on Computer Vision, pp. 818-833, 2014.

[10] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," Proc. of 3rd Int. Conf. on Learning Representations (ICLR 2015), pp. 1-14, 2015.

[11] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," Digital Signal Processing, Vol.73, pp. 1-15, 2018.

[12] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the Visualization of What a Deep Neural Network Has Learned," IEEE Trans. on Neural Networks and Learning Systems, Vol.28, No.11, pp. 2660-2673, 2017.

Name:
Daigo Kanda

Affiliation:
Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba

Address:
1-1-1 Tennoudai, Tsukuba, Ibaraki 305-8573, Japan

Brief Biographical History:
2019 Received B.E. degree, National Institute of Technology, Tsuruoka College

Name:
Shin Kawai

Affiliation:
Department of Intelligent Interaction Technologies, University of Tsukuba

Address:
1-1-1 Tennoudai, Tsukuba, Ibaraki 305-8573, Japan

Brief Biographical History:
2017 Received the Ph.D. degree from Department of Intelligent Interaction Technologies, University of Tsukuba
2017-2018 Postdoctoral Fellow, University of Tsukuba
2018- Assistant Professor, University of Tsukuba

Main Works:
• "Generalized Discretisation of Continuous-Time Distributions," The J. of Engineering, Vol.2020, No.7, pp. 259-267, 2020.
• "Interpretation of Kitamori's Partial-Model-Matching Method in a Descriptor-Form Expression," 4th IEEE Conf. on Control Technology and Applications (CCTA 2020), 2020.
• "General Mapping Discrete-Time Models of a Descriptor System with an Arbitrary Initial Condition," Automatica, Vol.87, pp. 428-431, 2018.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Hajime Nobuhara

Affiliation:
Department of Intelligent Interaction Technologies, University of Tsukuba

Address:
1-1-1 Tennoudai, Tsukuba, Ibaraki 305-8573, Japan

Brief Biographical History:
2002- Post Doctoral Fellow, University of Alberta
2002-2006 Assistant Professor, Tokyo Institute of Technology
2006- Assistant Professor, University of Tsukuba
2013- Associate Professor, University of Tsukuba

Main Works:
• Computational intelligence, multi-media processing

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)