GazeCaps: Gaze Estimation with Self-Attention-Routed Capsules

Hengfei Wang[1*] Jun O Oh[2,3*] Hyung Jin Chang[1] Jin Hee Na[3] Minwoo Tae[2]
Zhongqun Zhang[1] Sang-Il Choi[2]

[1]University of Birmingham [2]Dankook University [3]VTouch, Inc

Abstract

Gaze estimation is the task of estimating eye gaze from facial features. People tend to infer gaze by considering different facial properties from the whole image and their relations. However, existing methods rarely consider these various properties. In this paper, we propose a novel GazeCaps framework that represents various facial properties as different capsules. The capsules respond sensitively to transforms of facial properties by vectorial expression, which is effective for gaze estimation in which many facial components are nonlinearly transformed according to the direction of the head in addition to the perspective. Furthermore, we propose a ***Self-Attention Routing (SAR)*** *module which can dynamically allocate attention to different capsules that contain important information and can be optimized as a single process without iterations. Through rigorous experiments, we confirm that the proposed method achieves state-of-the-art performance on various benchmarks. We also detail the generalization performance of the proposed model through a cross-dataset evaluation.*

1. Introduction

Gaze refers to the direction a person is looking at. It is a typical nonverbal human expression method used to understand human intention, attention, and interaction among people in a group. Gaze estimation can be employed in various fields such as human-computer interactions (HCI) [18], augmented reality/virtual reality (AR/VR) [12], and autonomous driving [14]. This topic has been actively studied in the field of computer vision recently.

Appearance-based gaze estimation becomes more and more popular with the rapid development of deep learning. However, the appearance of the face non-linearly changes according to the rotation of the head. Furthermore, the appearance of the eyes and the area around the eyes, which are most important to gaze estimation, also change accordingly.

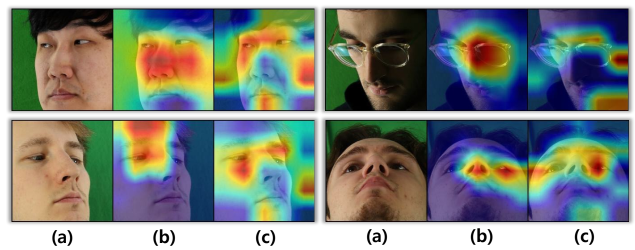

Figure 1. **Grad-CAM [17] visualization of attention maps.** (a) Input images from ETH-XGaze dataset, (b) Attention maps from CNN-based method, (c) Attention maps from GazeCaps with self-attention routing.

The gaze can be inferred differently depending on gender, race, object occlusion, and lighting.

To deal with the changes in intrinsic and extrinsic elements of faces, several gaze estimation methods using deep learning have been proposed. Appearance-based gaze estimation can be broadly classified into two categories according to the type of image used as input for the network. The methods in the first category directly focus on the gaze itself; for this purpose, the gaze is inferred using the pupil and the area around the eyes [2, 3]. Although these methods have achieved a good performance on gaze estimation, the following problems exist: 1) along with eye labels, additional labeling of eye position and head orientation for separating eyes [6, 11, 22] is required; 2) under some scenarios (e.g., occlusion caused by extreme head orientation, dark areas in eye regions) where the module that separates eyes cannot work, the subsequent gaze estimation module will not work either [2, 6]; 3) because the eye segmentation module and the gaze estimation module learn independently and are sequentially combined to form a system, the final gaze estimation result does not guarantee a globally optimal solution [21].

Methods of the other category use an entire face image as the input of gaze estimation networks without segmenting eyes from faces. Early gaze estimation models extract features from an image using a convolution filter. Then, the

*Equal contribution

gaze angle is estimated by regression analysis using a multi-layer perceptron (MLP) [23]. However, CNN has two important issues. (1) It is hard for CNN to capture fine-grained features because of pooling and stride convolution operators. As shown in Figure 1, CNN tends to capture a large continuous area. But even a small change in eye region is quite important for gaze estimation since the area occupied by the eyes in facial images is relatively small. (2) CNN can not reflect the spatial contextual information between partial areas of images (for example, correlation between eye and eyebrows). Randomly arranging the positions of different facial parts (like eyes, nose, mouth) does not form a recognizable face. Same to gaze estimation, knowing the relations between eyes and other facial elements is important for predicting gaze. As shown in Figure 1, the CNN-based method only watches a specific region while the proposed method watches several regions at the same time.

To utilize the contextual information from facial images, [4] attempts to do gaze estimation using a Transformer [5] which can effectively learn facial data with large variations while considering the context of entire faces. These methods achieve better gaze estimation performance compared to methods only using a convolutional neural network (CNN). However, although Transformer can effectively reflect the contextual facial information related to gaze estimation, applying it to gaze estimation still remains challenging. First, because facial images are high-dimensional, a large computation load is incurred when the self-attention operation of Transformers is directly used to grasp the entire context between pixels to images. To solve this problem, a method for embedding images to a lower dimension has been proposed. However, in gaze estimation, because the area occupied by eyes in entire images is very small, the information which is important to gaze estimation may be lost in the low-dimensional embedding process.

For gaze estimation, not only is the spatial contextual information included in face images important but also is the image-capturing conditions. For example, when looking from the front, the appearance of the eyes may vary depending on camera angles, expressions, the shadow caused by lighting, or whether glasses are worn. Therefore, to accurately estimate gaze directions, the correlation between eyes and gazes, and the correlation between various external characteristics, including the direction of heads, must be simultaneously learned. These characteristics can be defined as properties for gaze estimation. In fact, a human infers the gaze in an image by comprehensively considering these various properties. However, existing approaches for gaze estimation, including the methods using Transformers, do not consider these various properties.

The capsule network (CapsNet) [16] effectively utilizes the various properties included in an image. CNN's internal data representation fails to consider the key "spatial hierarchy" between simple and complex entities while CapsNet emphasizes the hierarchical pose relationship between the object's components for recognition and classification. The capsule is implicitly expressed by disentangling objects in images. Subsequently, a more complex aspect is represented by assembling these disentangled objects. Nonlinear changes in the appearance of objects in images can be expressed by adjusting the capsules. The hierarchical structure between simple and complex capsules is learned in the process of establishing a relationship between capsules.

To this end, we propose a method for training capsules (GazeCaps) from face images and estimating the gaze accurately by effectively combining the learned capsules. Furthermore, to solve the high computational burden of vanilla CapsNet and improve performance, we adopt a self-attention mechanism [15] and redesign it for GazeCaps.

Our main contributions are as follows:

- We propose a novel framework that utilizes the capsule concept to solve the problem of gaze estimation. The capsules show a better representational ability compared with CNN-based and Transformer-based methods by encapsulating different facial properties which widely shift with the changes in various viewpoints, movements, and environmental conditions.

- We propose a new SAR module (self-attention routing) for gaze estimation, which does not require iterations to update the coupling coefficients.

- Our proposed GazeCaps achieves state-of-the-art performance in different benchmarks. We demonstrate the advantage of GazeCaps for generalization in gaze estimation through experiments.

2. Related Studies

Gaze estimation from facial images. Several deep-learning-based methods for automatic gaze estimation from facial images have been proposed. The most conventional approach is to extract an image's regional spatial information using a CNN and perform gaze estimation using an MLP-based regression model from the extracted features. In [6], the gaze was estimated by independently training the preceding network for extracting eye patches and head directions, and the trailing network for estimating the gaze from the results of the preceding network. However, this method requires a module [8] for eye segmentation which increases the computational cost of the entire system and causes latency in the data transfer process. Gaze estimation networks capable of end-to-end learning using the whole face as input have also been proposed. [10] utilizes temporal information, in order to improve gaze inference, through the use of RNN which takes video input as opposed to individual images. The method proposed by [21] consists of a region proposal network that segments the area around the eyes in a face image, which acts as input for a separate gaze inference network that independently estimates the gaze. This method combines two networks in one framework and performs end-to-end learning. However, because this method uses only the segmented patches around the eyes in the gaze estimation process, the global context of the entire face cannot be utilized for gaze estimation.

Studies using attention. Studies have also been performed using attention and self-attention [19], which can effectively use the contextual information of data for computer vision tasks. To overcome the computational burden of applying the self-attention module to images, ViT [5] divides the image into several patches and calculates contextual information per patch. In [4], two methods of applying ViT to gaze estimation from face images are proposed. The first involves dividing the face into multiple grids and applying ViT to the image patches corresponding to each grid. The second involves estimating the gaze by applying a multi-layer Transformer encoder to the feature map extracted from the CNN backbone. This study shows that a Transformer which handles the global context by refining the features extracted from a CNN through attention could improve the accuracy of gaze estimation.

Capsule networks. CapsNet [16] was proposed to solve the limitations of max-pooling by dynamic routing and to learn spatial hierarchical relationships between low-level and complex entities. Several attempts have been made to introduce the capsule concept to the gaze estimation problem. In [13], the authors change the gaze estimation problem into a classification problem in order to apply CapsNet, a method designed for classification problems. However, this structure cannot deal with the ambiguity of gazes located at the boundary of each direction because the continuous gaze direction is defined as discrete classes. In [1], the authors design two types of gaze estimation networks in which the capsule concept is applied: one estimates the gaze by putting images restored from capsules into DenseNet and the other estimates the gaze directly from the capsules. This method however does not overcome the disadvantages of the eye separation methods either, because only eye patches are used as input. In contrast, our method receives the entire face as input. In addition, problem transformation is not required since we change the network structure to be suitable for gaze regression. To the best of our knowledge, our method is the first attempt to estimate gaze using capsule structures on entire face images.

3. Methodology

3.1. Preliminaries

CapsNet [16] parses an image as a combination of entities with various properties, which can include numerous types of instantiation parameters such as pose (position, size, orientation), deformation, velocity, albedo, hue, and texture. As shown in Figure 3 (a), they propose a novel connection mechanism called dynamic routing which takes advantage of the vectorial representation of capsules. Dynamic routing between the two capsule layers performs iterations to update the weights of the capsules from the last layer by calculating similarities between the two capsule layers. It ensures that the low-level capsules are connected to the appropriate high-level capsules. In this process, high-level capsules represent more complex entities with more degrees of freedom, which increases the number of properties in the capsule.

The capsule network requires a new activation function that operates on a vector to ensure nonlinearity instead of the functions designed for a perceptron. Furthermore, the output of the function is normalized between 0 and 1 to consider the length of the vector constituting the capsule as the existence probability of the entity represented by the capsule. To satisfy these requirements, they proposed the following “squashing” function:

$$v_j = squash(s_j) = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|} \tag{1}$$

$$s_j = \sum_{i=1}^{n} c_{ij}u_{j|i}, \quad u_{j|i} = W_{ij}v_i \tag{2}$$

where v_j is the vector representing capsule j, and s_j is its total input. n is the number of capsules and c_{ij} is the coupling coefficient that is determined by the iterative dynamic routing process. For all but the first layer of capsules, the total input s_j to a capsule j is a weighted sum over all “prediction vectors” $u_{j|i}$ from the capsules in the layer below. The prediction vectors are produced by multiplying the vector v_i of capsule i in the layer below by a weight matrix W_{ij}. Coupling coefficients are used to indicate how ”strongly” the low-level capsules are coupled to a particular high-level capsule. Coupling coefficients are critical learnable parameters that affect the performance of capsule networks. However, to determine their values, the network must go through iterative updates in one batch, which leads to a large computational workload.

3.2. Architecture of GazeCaps

The overall framework of GazeCaps is described in Figure 2. The framework consists of three parts: *Feature Extraction* to obtain feature maps from an input image; *Capsule Formation* to rearrange the feature maps into primary capsules and route them to a capsule layer through a SAR module for gaze estimation; *Gaze Regression* to conduct gaze regression through a SAR module. We redesign the architecture of the capsule network [16] which is proposed

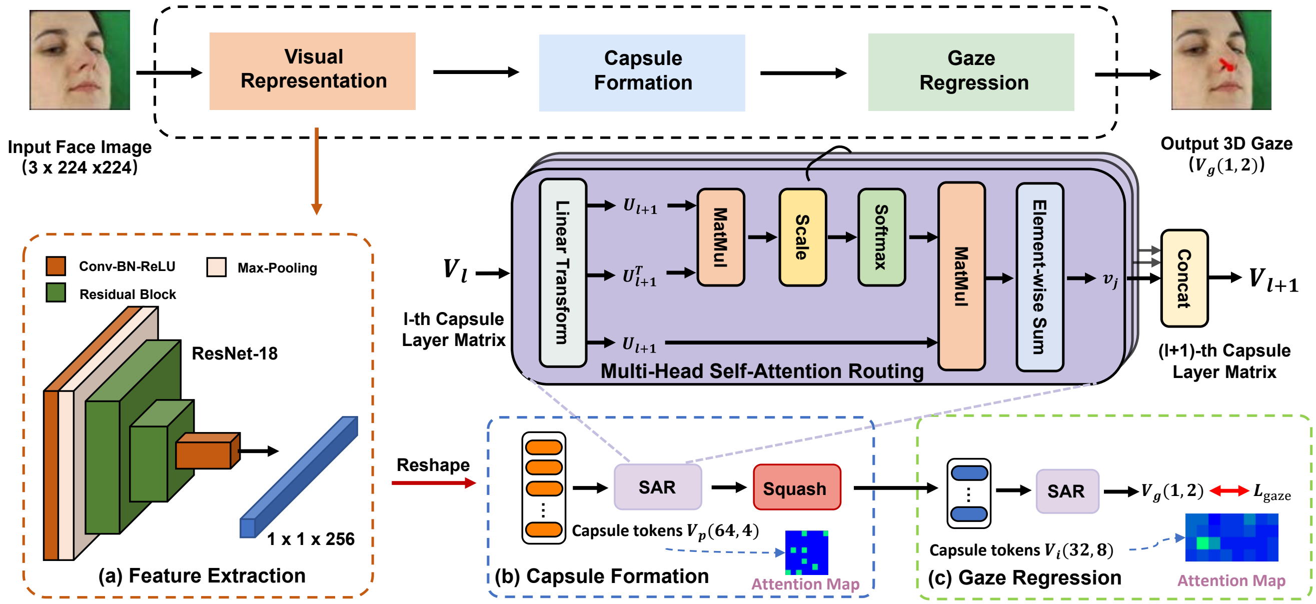

Figure 2. **Overall Structure.** The framework of GazeCaps is divided into three parts: **Feature Extraction**, **Capsule Formation**, and **Gaze Regression**. (a) We first extract image features from the input image through an image encoder which is based on CNN. (b) Then the output features are reshaped to get primary capsules. The SAR module takes the primary capsules to do self-attention for generating the intermediate capsule layer. Then the squash activation is applied to activate the capsules. (c) The gaze direction is estimated by a SAR module.

for the image classification problem to solve gaze estimation, which is a regression problem. The proposed model is trained using angular gaze loss.

Given an input image, GazeCaps first use an image decoder to extract image features. CapsNet [16] contains a single convolution layer that transforms each pixel intensity into activations of local features, which can be used as input to create primary capsules. However, because only low-level features that correspond to very local information are extracted in a single convolution layer, we use a convolution-based feature extractor with deeper layers to extract the structural features of facial components as candidates for the construction of primary capsules (in the proposed model, we used ResNet18 [9], which is effective for image analysis). Given an image I consisting of three channels of size $H \times W$, a feature map (f) of the shape of $1 \times 1 \times 256$ is generated by the feature extractor F ($f = F(I)$).

In *Capsule Formation*, the 256-dimensional vector is reshaped to create n_p capsules which are d_p-dimensional vectors as follows:

$$V_p = reshape(relu(conv(f))), \quad V_p \in \mathbb{R}^{n_p \times d_p} \tag{3}$$

where $V_p = \{v_i | i = 1, ..., n_p\}$ (here, $n_p = 64, d_p = 4$), v_i is the i-th capsule with dimension d_p. V_p can be represented as an $n_p \times d_p$ matrix, where each row is a capsule, and each column corresponds to an attribute at the same position. In the process of creating the primary capsules, information from an input image is no longer "place-coded" in the spatial feature domain. Instead, it is "rate-coded" in the capsule's properties.

The capsules from the primary capsule layer are combined through a SAR module. The output is activated by a squash activation to create the intermediate capsule layer with higher-level capsules. Through this process, low-level capsules evolve into high-level capsules that can represent more complex entities with more degrees of freedom; we can obtain the intermediate capsules using the following routing function:

$$v_i = Routing(V_p) \tag{4}$$

where intermediate capsule layer $V_i = \{v_j | j = 1, ..., n_i\}$ (here, $n_i = 32, d_i = 8$) and $V_i \in \mathbb{R}^{n_i \times d_i}$, v_j is the j-th capsule with dimension d_i. To impart nonlinearity to the capsule layer, we used the squash function as the activation function.

After *Capsule Formation*, *Gaze Regression* adopts a SAR module to estimate a gaze capsule v_g directly. We do not apply the activation function to v_g because errors may occur when using the activation function if the ground truth is located at a distance greater than one from the origin.

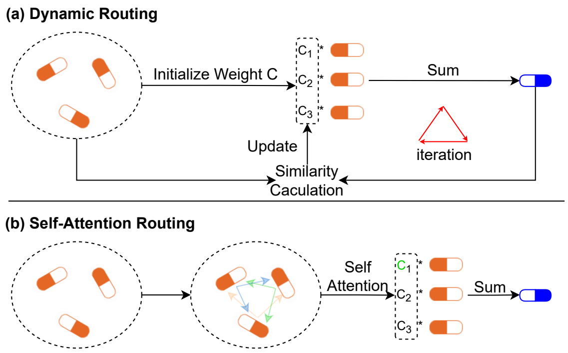

Figure 3. **Illustration of Dynamic Routing and Self-attention Routing.** (a) illustration of dynamic routing which conducts several iterations to update capsule weights C by calculating the similarities between the capsules from the former and latter layers. (b) illustration of self-attention routing which gets capsule weights C directly by adopting a self-attention mechanism.

3.3. Capsule-routing using self-attention

In the proposed model, a novel capsule routing method (SAR) using self-attention [15] is redesigned to effectively construct high-level capsules from low-level capsules. As shown in Figure 3 (b), by applying self-attention to low-level capsules, we use an inter-capsule routing to represent the interrelations among entities of a face image based on the spatial contextual information of the capsules. Furthermore, unlike dynamic routing [16] (shown in Figure 3 (a)), the proposed routing method does not require computationally expensive iteration and is designed to be compatible with the gradient-based optimization process. We used the attention matrix for low-level capsules as the coupling coefficient for high-level capsules.

SAR module in Figure 2 shows a schematic diagram of the proposed routing using self-attention. A prediction matrix U_{l+1} for creating a high-level capsule (v_{l+1}) from low-level capsules (v_l) is obtained as follows.

$$U_{l+1} = WV_l, \quad W \in \mathbb{R}^{d_l \times d_{l+1}} \tag{5}$$

U_{l+1} is composed of the prediction vectors (u_i^l) generated by multiplying the capsule layer matrix V_l by a weight matrix W, where d_{l+1} represents the dimensions of prediction vectors that equals with the higher-level capsule. In general, to express more complex properties, d^{l+1} is larger than d_l.

We generate the attention matrix for capsule routing between the l-th capsule layer and $(l+1)$-th capsule layer by matrix multiplication between prediction vectors as follows:

$$A = softmax(\frac{U_{l+1}{U_{l+1}}^T}{\sqrt{d_{l+1}}}), \quad A \in \mathbb{R}^{n_l \times n_l}. \tag{6}$$

The attention matrix is used as coupling coefficients for capsules in the l-th layer. Using the attention matrix, we can find agreements between the capsules that can effectively represent the interrelations among entities by self-attention; in other words, a capsule with more agreement with other capsules receives higher attention. The attention matrix gives weights to the prediction vectors as follows:

$$X_{l+1} = AU_{l+1}, \quad X_{l+1} \in \mathbb{R}^{n_l \times d_{l+1}} \tag{7}$$

where $X_{l+1} = \{x_i | i = 1, ..., n_l\}$ contains the weighted prediction vectors. Then, v_j in the $(l+1)$-th layer is obtained using the weighted sum of x_is from the l-th layers as $v_j = \sum_{i=1}^{n_l} x_i$. Consequently, using self-attention, we conduct capsule routing much more economically and effectively than dynamic routing.

The number of capsules constituting the higher-level capsule layer may be one or more. The self-attention routing needs as many as the number of capsules in the higher-level layer for multiple capsules. Therefore, the self-attention routing process described above is extended to the multi-subspace (multi-head self-attention), and the number of heads is equal to the number of capsules in the next capsule layer.

3.4. Loss function for GazeCaps training

We use gaze error between estimated gaze and ground truth to train the proposed model. The gaze loss is calculated by the following equation:

$$L_{GazeCaps} = MSE(g_t, g_p) \tag{8}$$

where g_t is the ground truth of gaze, g_p is the predicted gaze direction, $L_{GazeCaps}$ is the loss used in training. The loss is calculated as a mean squared error (MSE).

4. Experiments

4.1. Dataset for Evaluation

We use the ETH-XGaze dataset [20] for network pre-training. ETH-XGaze consists of 1.1M images collected from 110 subjects. We use a training set containing 765K images of 80 subjects to pre-train the model.

A total of four datasets, EYEDIAP [7], Gaze360 [10], MPIIFaceGaze [23], and RT-GENE [6], are selected from well-known public datasets to evaluate the gaze estimation performance. All datasets are labeled for 3D gaze estimation. The EYEDIAP dataset contains 94 videos with 237 minutes obtained from 16 subjects. We divide 16 people into four clusters to evaluate their performance with this dataset and performed 4-fold cross-validation for evaluation. The Gaze360 dataset includes 172K images from 238 subjects, with the widest head poses and gaze distribution. They pre-divide the dataset into 129K images for training, 17K images for validation, and 26K images for evaluation. We use the experimental setting in [10]. The MPIIFaceGaze

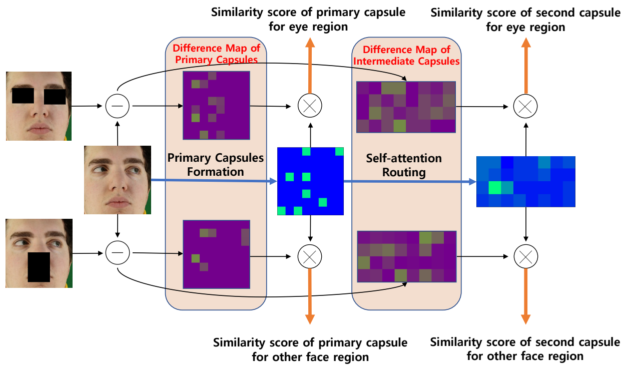

Figure 4. **Capsule and Attention visualization**. The images with blue background are the visualization of **capsule attention weights** and the brightness of the block represents the active intensity, while the images with purple background are the visualization of the **capsule weights' differences** between masked images and original images and the brightness of the block represents the degree of the differences. The square images and the rectangle images correspond to 64 primary capsules and 32 intermediate capsules, respectively. The similarity score is to evaluate the similarity between difference maps and the attention maps from the original image.

Capsule type \ Masked region	eyes	face
Primary Caps	0.83	0.77
Intermediate Caps	0.90	0.77

Table 1. **Activation score**. The activation score is averaged over 10 subjects from ETH-XGaze.

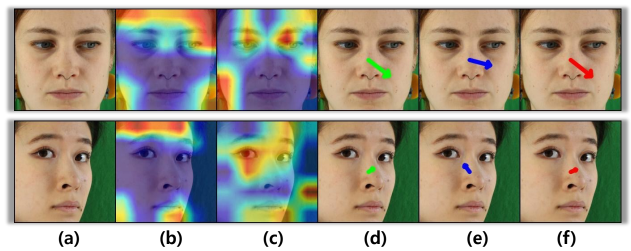

Figure 5. **Grad-CAM [17] and predicted gaze visualization of GazeCaps and CNN-based method**. (a) Input images, (b) Grad-CAM visualization of CNN-based method, (c) Grad-CAM visualization of GazeCaps, (d) Ground truth, (e) Prediction from CNN-based method, (f) Prediction from GazeCaps.

dataset contains 45K images obtained from 15 subjects. We use the leave-one-person-out evaluation to evaluate the performance of this dataset. The RT-GENE dataset contains 123K images from 15 subjects and specifies 13 subjects for training and two subjects for validation. To evaluate the performance of the dataset, we divide 15 people into three clusters and use 3-fold cross-validation.

4.2. Capsule and Attention Visualization

As shown in Figure 4, we visualize the primary capsules and the intermediate capsules in GazeCaps to show the proposed method can capture gaze features accurately. Since the positional information is lost in the formation of the primary capsules, we check which capsule is related to the eye region by calculating the difference map of capsules from the same layer between the eye-masked image and the original image. And we rescale the elements in difference maps and capsule attention maps to $(0, 1)$. In this case, the capsules with a large difference in difference maps are related to the eye region. And we do the same for intermediate capsules. To show that GazeCaps can capture eye features accurately, we calculate the activation score by evaluating the similarity between the difference maps and the capsule attention maps with the following equation:

$$S = 1 - \frac{\sum_i^{n_{caps}} \left(c_i - c_i^d\right)^2}{n_{caps}} \tag{9}$$

where S is the activation score whose range is $(0, 1)$, c_i is the attention weight from attention maps, c_i^d is the difference weight from difference maps, i is the index of capsules, n_{caps} is the total number of the capsules in maps. To make things clear, we also conduct the same visualization of other face regions.

As shown in Table 1, the activation score of eyes is higher than that of other face regions both in the primary capsules and the intermediate capsules, which means GazeCaps can capture eye features and give them more attention. And the activation score of eyes gets larger when the model goes deeper from primary capsules to intermediate capsules. That shows the proposed model focuses on eye region better with the network proceeding. The activation score of other face regions does not change much because GazeCaps also considers the influence from other regions except eyes.

4.3. Ablation Study

Method	EYEDIAP	Gaze360	MPIIFaceGaze	RT-GENE
CNN	6.18°	15.41°	5.21°	13.28°
Transformer	5.84°	11.79°	4.51°	8.00°
Capsules(w/o SA-Routing)	5.44°	11.32°	4.88°	7.65°
Capsules(w/ SA-Routing)	**5.10°**	**10.04°**	**4.06°**	**6.92°**

Table 2. **Ablation Studies**. We compare the performance by permuting the subsequent architecture while maintaining the structure of the feature extractor the same.

To validate the design of our proposed model, we perform an ablation study on capsule representation and self-

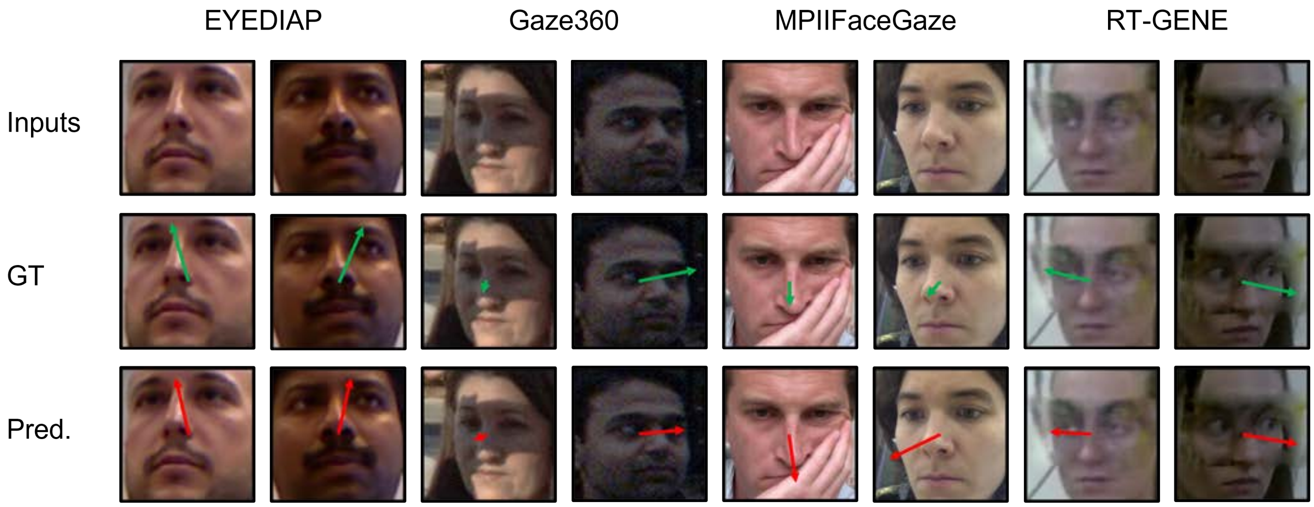

Figure 6. **Qualitative results.** The first row images are input images, and the second and third rows are the ground truth and the estimated result from the proposed method, respectively.

attention routing. To be specific, we compare our proposed model with a CNN baseline and a Transformer baseline to verify capsule representation, and compare with the vanilla CapsNet with dynamic routing to verify self-attention routing. The CNN baseline inludes a ResNet-18 and two layers of MLP. The Transformer baseline contains a ResNet-18 and six Transformer encoder layers followed by a single MLP layer. The feature extractor in all models has the same ResNet-18 [9] structure. For a fair comparison, we control the amounts of parameters of all models to be the same.

As shown in Table 2, we conduct experiments on four datasets to verify the reliability of the results. CNN and Transformer methods are based on scalar representation while the last two models (including our proposed model) are based on vectorial representation (known as capsule). The results show that both of capsule models outperform scalar models on all selected datasets, which demonstrates the big advantage of capsule models on gaze estimation. And we also visualize the attention maps of the proposed GazeCaps and CNN-based method from Grad-CAM [17] as well as the predicted gaze direction from both models (see Figure 5). It shows that the proposed method captures eye region more accurately and has a better performance in gaze estimation.

Besides, our proposed model also shows a better performance than the default CapsNet with dynamic routing. That illustrates that the proposed self-attention routing works better in gaze estimation.

4.4. Cross-dataset Evaluation

Cross-dataset evaluation is well-known to analyze the generalization performance of models. Because our key components are based on capsules instead of neurons, we design the experiments to highlight the effectiveness of capsule representation for generalization performance. We select a CNN baseline which include a ResNet-18 and an MLP layer to represent the methods using neurons. We still control that the two models have a similar number of parameters. Table 3 shows the cross-dataset evaluation results of the two models on the benchmark datasets. Our GazeCaps shows better generalization performance in all cases, as shown in the table. These results show that the features extracted from the proposed capsule-based model are more robust on different datasets than the CNN-based model since it keeps more detailed information and takes the relations among different properties into consideration.

Train \ Test	EYEDIAP	Gaze360	MPIIFaceGaze	RT-GENE
EYEDIAP [7]	-	34.1°/**20.8°**	20.0°/**11.4°**	17.9°/**13.5°**
Gaze360 [10]	13.8°/**9.2°**	-	17.1°/**9.2°**	28.0°/**17.2°**
MPIIFaceGaze [23]	16.4°/**11.4°**	36.1°/**22.4°**	-	18.5°/**12.0°**
RT-GENE [6]	29.1°/**16.9°**	35.9°/**24.0°**	13.7°/**8.8°**	-

Table 3. **Cross-dataset evaluation**. We perform a cross-dataset validation of the CNN-based model and the proposed capsule-based model. In this experiment, the feature extractor is the same for both methods. For direct comparisons, we put the results of CNN-based and Capsule-based methods side by side: (CNN/GazeCaps)

Figure 6 shows the qualitative results on various face images from different datasets. The stable performance on different datasets shows a good generalization ability of the proposed model.

Method	# of Params.	# of FLOPs	Backbone
FullFace [23]	196.6M	2.99G	CNN
RT-GENE [6]	82.0M	30.81G	CNN
Dilated-Net [2]	**3.9M**	3.14G	CNN
Gaze360 [10]	14.6M	12.78G	RNN
CA-Net [3]	34.1M	15.6G	CNN
GazeTR-Pure [4]	227.3M	58.32G	Transformer
GazeTR-Hybrid [4]	11.4M	1.82G	CNN+Transformer
Vanilla CapsNet [4]	11.8M	2.5G	Caps
GazeCaps(Proposed)	11.7M	**1.82G**	Caps

Table 4. **Specifications of gaze estimation methods**. Our proposed method is based on capsule networks, which is different from other existing gaze estimation methods.

Method \ Dataset	EYEDIAP	Gaze360	MPIIFaceGaze	RT-GENE
FullFace [23]	6.53°	14.99°	4.93°	10.00°
RT-GENE [6]	6.02°	12.26°	4.66°	8.00°
Dilated-Net [2]	6.19°	13.73°	4.42°	8.38°
Gaze360 [10]	5.36°	11.04°	**4.06°**	7.06°
CA-Net [3]	5.27°	11.20°	4.27°	8.27°
GazeTR [4]	5.33°	11.00°	4.18°	7.12°
GazeCaps(Proposed)	**5.10°**	**10.04°**	**4.06°**	**6.92°**

Table 5. **Comparison with the SOTA methods**. We compare GazeCaps with other SOTA gaze estimation methods on EYEDIAP, Gaze360, MPIIFaceGaze, and RT-GENE datasets. The reported metric is mean angular errors (in degrees).

4.5. Comparison with State-of-the-art Methods

Table 4 shows the number of parameters and the number of flops of each network used in the experiment to compare the efficiencies of the selected SOTA methods. Our proposed method has fewer parameters than most existing gaze estimation models and the lowest flops among all selected models due to the adoption of capsule representation and self-attention routing, which are $11.7M$ parameters and $1.82G$ flops. It shows that GazeCaps is a lightweight model compared with other models and lowers the computational requirement for gaze estimation model training.

Table 5 shows the gaze estimation results of our method and those of other methods on the EYEDIAP, Gaze360, MPIIFaceGaze, and RT-GENE datasets. FullFace [23], RT-GENE [6], Dilated-Net [2], and CA-Net [3] are CNN-based methods, whereas Gaze360 [10] combines CNN with RNN. GazeTR [4] is a hybrid method that uses CNN features as input for the Transformer[1]. This table indicates that GazeCaps exhibits the best results among all methods on all datasets. Notably, Gaze360 and GazeTR-Hybrid show better results than other previous approaches. Therefore, we can conclude that combining CNN with different types of networks can improve the accuracy of gaze estimation. GazeCaps is also a hybrid approach that uses CNN features as the input for the capsule network. Compared with other hybrid methods, we adopt capsules instead of neurons to design the layers, which give the best performance. In addition, our method can effectively mitigate the computational burden by applying the attention mechanism to capsule routing. The results in Table 4 and Table 5 show that our method is a better hybrid approach than previous methods in terms of accuracy and efficiency.

[1]We retrained the GazeTR model, so the results reported here are different from [4].

5. Conclusions

In this paper, we analyze problems in CNN-based and Transformer-based gaze estimation methods and propose a novel capsule-based network for accurate gaze estimation. We introduce a novel SAR (Self-Attention Routing) module which combines with capsule representation to form a new gaze estimation framework. The final framework GazeCaps achieves state-of-the-art results with a small model size and low computation load. We believe that the capsule representation has great potential to be further explored for gaze estimation. For future work, we will study the interpretation of hierarchical capsules by visualizing capsules and their entities.

Acknowledgment

This work was supported in part by the National Research Foundation of Korea Grant by the Korean Government through the Ministry of Science and Information and Communication Technology (MSIT) under Grant 2021R1A2B5B01001412 and in part by the ICT Challenge and Advanced Network of Human Resource Development (HRD) (ICAN) Program, supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP), under Grant No.2020-0-01824 and No.2022-0-00608 (Artificial intelligence research about multi-modal interactions for empathetic conversations with humans). Also, the research utilised the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/) funded by the Engineering and Physical Sciences Research Council (EPSRC) and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) operated by Advanced Research Computing at the University of Birmingham. Hengfei Wang and Zhongqun Zhang were supported by China Scholarship Council (CSC) Grant No. 202006210057 and No. 202208060266, respectively.

References

[1] Vivien Bernard, Hazem Wannous, and Jean-Philippe Vandeborre. Eye-gaze estimation using a deep capsule-based regression network. In *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2021. 3

[2] Zhaokang Chen and Bertram E Shi. Appearance-based gaze estimation using dilated-convolutions. In *Asian Conference on Computer Vision*, pages 309–324. Springer, 2018. 1, 8

[3] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):10623–10630, 2020. 1, 8

[4] Yihua Cheng and Feng Lu. Gaze estimation using transformer. *2022 International Conference on Pattern Recognition (ICPR)*, 2022. 2, 3, 8

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 3

[6] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. In *European Conference on Computer Vision*, pages 339–357, September 2018. 1, 2, 5, 7, 8

[7] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap database: Data description and gaze tracking evaluation benchmarks. Idiap-RR Idiap-RR-08-2014, Idiap, 5 2014. 5, 7

[8] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 7

[10] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 5, 7, 8

[11] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[12] Joseph Lemley, Anuradha Kar, and Peter Corcoran. Eye tracking in augmented spaces: A deep learning approach. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–6, 2018. 1

[13] Bhanuka Mahanama, Yasith Jayawardana, and Sampath Jayarathna. Gaze-net: appearance-based gaze estimation using capsule networks. *Proceedings of the 11th Augmented Human International Conference*, 2020. 3

[14] Sujitha Martin, Sourabh Vora, Kevan Yuen, and Mohan Manubhai Trivedi. Dynamics of driver's gaze: Explorations in behavior modeling and maneuver prediction. *IEEE Transactions on Intelligent Vehicles*, 3(2):141–150, 2018. 1

[15] Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: capsule network with self-attention routing. *Scientific reports*, 11, 2021. 2, 5

[16] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, 2017. 2, 3, 4, 5

[17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 6, 7

[18] Julian Steil, Michael Xuelin Huang, and A. Bulling. Fixation detection for head-mounted eye tracking based on visual similarity of gaze targets. *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, 2018. 1

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[20] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision (ECCV)*, 2020. 5

[21] Xucong Zhang, Yusuke Sugano, A. Bulling, and Otmar Hilliges. Learning-based region selection for end-to-end gaze estimation. In *BMVC*, 2020. 1, 2

[22] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, June 2015. 1

[23] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2299–2308, 2017. 2, 5, 7, 8