Improved Capsule Network for Gaze estimation in Wireless Sensor Networks

Mingyuan Luo[1], Xi Liu[2], Wei Wang[3], and Wei Huang*[4]
{406130917323@email.ncu.edu.cn[1], 406130917318@email.ncu.edu.cn[2], wwang@chd.edu.cn[3], n060101@e.ntu.edu.sg[4]}
* indicates the corresponding author

School of Information Engineering, Nanchang University, Nanchang, China[1,2,4]
School of Economics and Management, Chang'an University, Xi'an, China[3]

Abstract. In this study, aiming at the problem of gaze estimation in the wireless sensor network in the car, we use image-based method to estimate gaze based on the single camera sensor. We use the deep learning model and propose the improved model from three aspects based on the original capsule network. The first is to increase the convolution layer, the second is to increase the capsule layer, and the third is to widen the capsule layer in the network. Through many contrast experiments, it is proved that the appropriate use of the first or second improved method can achieve performance over other comparison models, and the prediction results of gaze estimation are almost no different from the real gaze direction.

Keywords: Gaze estimation, Capsule Network, Multi-layer Capsule Network.

1 Introduction

With the continuous development of wireless sensor techniques, wireless sensor network (WSN) has been widely applied in diverse fields, and it has become a frontier hot-spot research domain of interdisciplinary studies. Wireless sensor network integrates a variety of technologies. Through the cooperation of many integrated sensors, it conducts environmental perception, monitors object information collection, and transmits collected information to the user interface through wireless transmission, in order to achieve convenient interconnection without data line restrictions. It is widely acknowledged that, wireless sensor networks are of great importance in many research fields, including environmental monitoring, traffic management, auxiliary driving, etc. Specifically, in the field of auxiliary driving, multiple wireless sensors can be used outside the vehicle to identify vehicles and pedestrians as well as the state of the road. Moreover, within the car, a single wireless sensor or multiple wireless sensors can be utilized to monitor the driver's status in real-time. Therefore, essential driving safety issues, including whether the driver is tired or she / he has abnormal driving behaviors, can be evaluated. It can be easily perceived that, multiple wireless sensors inside and outside the vehicle actually form a wireless sensor network, improving driving experiences for auxiliary driving. In terms of fatigue driving detection, the driver's face can be captured by visual sensors, including visible or infrared cameras in front of the driver. Therefore, the fixation direction of

MOBIMEDIA 2019, June 29-30, Weihai, People's Republic of China

DOI 10.4108/eai.29-6-2019.2282839

eyes of the driver can be automatically analyzed, and whether the driver is in the state of fatigue driving can be identified. The above process is often called as gaze estimation.

Gaze estimation can use various detection methods to obtain the current gaze direction of the detected object. Detection methods are mainly divided into image-based and non-image-based methods. The image-based method obtains the gaze direction by image processing or deep learning technology based on the visible shape and texture information, while the non-image-based method calculates the gaze direction by means of light reflection and eye structure model. Gaze estimation plays an important role in fatigue driving detection. According to the gaze direction by the gaze estimation, if the gaze direction is not on the road in front of the car for a long time (i.e., a few seconds) and moves down to the steering wheel area, the driver can be considered to be in a dangerous state of fatigue driving, and the wireless sensor network should be given an early warning to alert the driver.

In this study, based on the image of the eye area obtained by the single camera in the car, we combined the deep learning technology to gaze estimation. The use of deep learning for gaze estimation saves the process of manually extracting image features. At the same time, deep learning has strong recognition and classification ability and can obtain extremely high gaze estimation accuracy. We used a variety of classic and excellent deep learning models, including traditional convolutional neural network (CNN), deep residual network (ResNet), squeeze network (SqueezeNet), capsule network (CapsNet), and multi-layer capsule network that we innovatively proposed based on the original capsule network. We evaluated and analyzed these models on appropriate data sets, and verified through experiments that one of our improved models could achieve the best performance. The estimation of gaze direction can achieve a result with almost no error with the real direction, which indicates that our work has better practical significance. The structure of this paper is as follows: Section II introduces the concepts of gaze estimation and deep learning; Section III introduces the details of our improved models. In Section IV, the improved models are compared with other deep models to evaluate the performance of different models and analyze the results in the data set. Section V summarizes the conclusion of this research and looks forward to the future.

2 Related work

2.1 Gaze estimation

Combining with the research results of other disciplines, the existing gaze estimation methods are mainly divided into two categories. One is feature-based, which includes pupil center cornea reflection method [1], cross-ratio invariant method [2] and three-dimensional geometric model method [3]. The other is the appearance-based approach, such as the shallow neural network model method [4].

Pupil center cornea reflection method is one of the most commonly used gaze estimation techniques. It mainly uses image processing technology to extract the central region of the pupil and the reflection point of the infrared light source on the cornea for gaze estimation. Among them, the computation about pupil center, use a kind of method of bright and dark pupil normally. When the light from the light source is aligned with the light path of the human eye, the bright pupil effect will be produced when the light passes through the pupil into the retina and then reflects back from the retina. If the light source deviates from the path the eye follows, most of the light entering the pupil will no longer be reflected back from the pupil, and the pupil

will turn black. In this way, by controlling the angle of incident light, two pupil images with different degrees of light and shade can be obtained. By subtracting these two images, the position of the pupil can be quickly located. When the position of the central area of the pupil and the reflection point of the light source on the cornea is obtained, the vector formed by them will change with the change of the gaze direction, and the gaze direction will be calculated based on this point.

Based on the method of cross-ratio invariance, two cameras, one tracking human face and one tracking human eyes were used. In addition, five near-infrared auxiliary light sources were used, four of which were placed around the screen and one on the optical axis of the camera. This method introduces the invariant property of intersection ratio in projective geometry into the mapping space model. Only by obtaining the information such as the center of the pupil and the spot reflected by the light source. The intersection point between the human eye line of sight and the screen can be calculated according to the same intersection ratio, and the gaze direction can be estimated by spatial mapping. In practice, this method is less robust to different eyes, and different characteristic parameters need to be extracted. In addition, this method ignores the angle between the axis of vision and the axis of light and the incongruity between the center of the pupil and the spot reflected by the light source, so there is a certain error.

Three-dimensional eyeball geometric model method to modeling of the physical structure of the human eye, according to the light source, the three-dimensional position of the known parameters and positions, both inside and outside, video camera, screen size and position, and the parameters of the eyes (such as radius of eyeball, radius of cornea, corneal refractive index, etc.), in combination with cameras for eyes image information, calculate the center of the cornea and pupil center of three-dimensional coordinates, and the analytic equation of the optical axis is calculated gaze direction. This method requires high precision and is very complex to calibrate different human eye parameters. However, due to the real-time calculation of three-dimensional coordinates, it is less affected by head movement.

The shallow neural network method does not extract the features of the pupil of the eye image and the corneal reflex point of the light source, but processes the overall image of the eye and estimates the direction of vision. After a large number of eye images are acquired by the camera, shallow neural networks are used to learn the features of these eye images (such as texture details, light and dark changes, etc.) and predict the gaze direction of eyes in a given eye image that is not learned. The advantage of this method is that it does not need to set many complex parameters according to experience, and a large number of eye images are easy to obtain, the disadvantage is that the shallow neural network capacity is small, the generalization ability is not strong.

2.2 Deep Learning

Deep learning is a hot branch of machine learning because it has achieved unprecedented good results in image segmentation, semantic recognition, time series prediction, and other challenging problems. The concept of deep learning originates from the research on artificial neural networks, which was proposed by *Hinton et al.* in 2006. Hinton et al. proposed a layer by layer training algorithm for unsupervised greed based on the deep belief network (DBN) [5], which brought hope to solve the optimization problems related to the deep structure, and then proposed the deep structure of multi-layer automatic encoder. In addition, the convolution neural network [6] proposed by *Lecun et al.* is the first real multi-layer structure learning algorithm, which uses spatial relative relations to reduce the number of parameters to improve training performance.

Deep learning is relative to shallow learning. At present, most classification, regression, and other learning algorithms are shallow learning. Their limitations lie in their limited ability to express complex functions in the case of limited samples and computing units. Deep learning realizes complex function approximation by learning a deep non-linear network structure, represents the distributed representation of input data, and shows the powerful ability to learn the essential features of data sets from a few sample sets. Multi-layer perceptron with multiple hidden layers is a kind of deep learning structure. Deep learning simulates the more neural activity of neural layers, and forms more abstract high-level representation attribute categories or features by combining low-level features, to discover the distributed representation of data features. The schematic diagram of deep learning is shown in **Figure 1**.

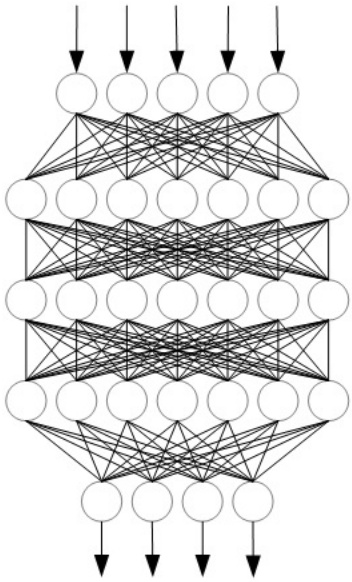

Fig. 1. The schematic diagram of deep learning.

The core idea of deep learning is to find another way to represent data. Specifically, suppose there is a n-layer data processing system S, and the n-layer is S_i $(i = 1,2,\ldots,N)$, the input is I and the output is O. If the input O is equal to the input I, that is, the input I changes through the system without any loss of information and remains unchanged, this means that the input I passes through each layer of S_i without any loss of information, that is, at any level of S_i, it is another representation of the original information (i.e., input I). By adjusting the parameters in the system, deep learning achieves that input O equals input I in the above mentioned, so that we can obtain the features of different levels of input I (i.e. S_i, $i = 1,2,\ldots,N$). For a specific example, input I can be an image, text, language, or other information medium.

Since AlexNet [7], proposed by *Hinton* research group in 2012, won the champion of ImageNet large-scale visual recognition challenge competition [8] that year, people have recognized its ability of deep learning. After that, a large number of excellent models (such as VGG [9], GoogleNet [10] and ResNet [11], etc.) emerged in deep learning and achieved increasingly good results in image recognition, segmentation, and other fields. The rapid

development of deep learning makes it widely concerned. In recent years, there are many new methods and models of deep learning, which can achieve better results in certain fields. For example, in 2014, *Srivastava* proposed a simple method (Dropout) [12] to prevent overfitting of the neural network model, and *Goodfellow* proposed a generated confrontation network (GAN) [13] to evaluate the generated model through confrontation process. In 2015, *Long* proposed a new convolution structure, full convolution network (FCN) [14], and *Loffe* proposed a batch standardization method (BN) [15] to accelerate deep network training by reducing internal covariate transformation. In 2017, *Hinton* proposed a new neuron structure, called capsule [16], which has many advantages over traditional neurons. These methods and models have been widely studied and applied. In this study, we propose a multi-layer capsule network based on the original capsule network and apply it to the gaze estimation of the wireless sensor network.

3 Methodology

In this study, the data we get is eye images, and the data we need to predict is gaze direction vectors. Images and vectors are two completely different data types, which is a challenge. So we want to design models that have the ability to transform high-level semantic information. Aiming at the challenge of gaze estimation, we propose new models from three aspects based on the original capsule network. The first is to increase the number of convolution layers in the capsule network so that the features of input capsule layer are high-level semantic features. The second is to increase the number of capsule layers in the capsule network so that the whole network can learn more complex object characteristics. The third is to increase the number of capsules in the capsule layer, which is equivalent to increasing the width of the network so that the network has a larger capacity. According to the above three points, we proposed CNN6+CapsNet2, CNN9+CapsNet2, CNN1+CapsNet3, CNN1+CapsNet4, CNN1+CapsNet5, CNN1+Wide CapsNet2 and CNN1+Wide CapsNet3. Note that the original capsule network is CNN1+CapsNet2.

3.1 Original capsule network

Capsule network based on capsule neuron structure is a new concept proposed by *Hinton* in 2017, which is inspired by the columnar structure formed by a group of neurons in the cerebral cortex. Cortical minicolumn is common among most mammals, especially primates. It has hundreds of neurons inside, and it has layers inside. The capsule is the structure that Hinton uses to correspond to this columnar structure, which in the neural network is a subnet structure. In terms of cognition, the brain has some prior knowledge, such as looking at the face in the forward direction, which is easy to recognize, while the reverse is much worse. For example, the illusion is the preexisting knowledge in the human brain that affects visual recognition. According to *Hinton*, this knowledge corresponds to some frames, such as coordinate frames, that can be trained and can act as specialized structures (capsules) in identification.

Capsule network aims to solve the inherent problems of traditional convolutional neural network and provide a new optimization algorithm different from the backpropagation algorithm to further solve the problems of dynamic vision, 3D, unsupervised learning and other deep learning problems. The traditional convolutional neural network is not sensitive to the position relationship between objects in the image so that the eye and mouth positions in the

face image are exchanged, and the convolutional neural network is still predicted to be a face. Unlike traditional convolution neurons, whose input and output are scalars, the input and output of capsule neurons are vectors, representing the instantiation parameters of specific entity types in the image. Specifically, the length of the vector represents the probability of the existence of the entity, and the direction of the vector represents the instantiation parameters of the entity (e.g., position, size, direction, and even the degree of deformation, etc.).

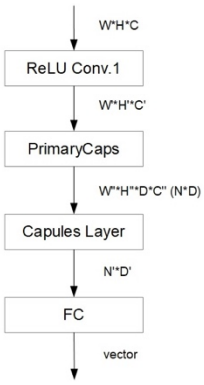

Fig. 2. The network structure of gaze estimation based on the original capsule network. W, H, and C represent the width, height, and channel of the input image, respectively. $W'(W'')$, $H'(H'')$, and $C'(C'')$ represent the width, height, and channel of the feature maps, respectively. $N(N')$ represents the number of capsules. $D(D')$ represents the dimensions of the capsule.

The calculation process of the capsule is as follows. Let the entity vector generated by the capsule in the lower layer be u_i, and the predicted vector u_{ij} obtained by the affine transformation of u_i is expressed as equation 1.

$$u_{ij} = W_{ij}u_i \tag{1}$$

Where W_{ij} is the coefficient of affine variation. The meaning of the prediction vector is to predict the probability of occurrence of high-level entities based on the location of low-level entities. (e.g., if you want to identify a cart, then a layer of low-level entity is the horse and car, then according to the horse can determine the overall position of the carriage, can also according to car to judge the position of the carriage, if these two lower-level entities to judge the position of the carriage are identical, can think this is the cart appears the probability is high, the opposite is low.) And then you need to pass each of the prediction vectors u_{ij} to the higher level capsules. For each of the predicted vectors u_{ij}, the transfer is not going to be equal, but it's going to be multiplied by a coupling coefficient c_{ij} and then sum over all of the vectors, as shown in equation 2.

$$s_j = \sum_i c_{ij}u_{ij} \tag{2}$$

The coupling coefficient c_{ij} can be considered as the prediction degree of low-level entities to high-level entities, which is determined by the dynamic routing algorithm. Finally, a non-linear activation function is used to compress the length of the summed vector to 0 to 1 as the output of the high-level capsule. The calculation process of the non-linear activation function is shown in equation 3.

$$v_i = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{3}$$

The network structure of gaze estimation based on the original capsule network is shown in **Figure 2**, which consists of four substructures, the first three of which are the original capsule network, which can be called CNN1+CapsNet2 according to the structure. The first substructure is a traditional convolution layer used to extract the primary features from the original input image and input them into the later substructure. The second substructure is a primary capsule layer used to detect some low-level entities. The third substructure is an advanced capsule layer used to detect some high-level entities. The dynamic routing algorithm is used to determine the coupling coefficient matrix between the primary capsule layer and the advanced capsule layer. The fourth substructure is the full connection layer. Since the output of the third substructure is some eigenvectors representing high-level entities, and what we need is the gaze direction vector ultimately, we learn the gaze direction vector by passing these eigenvectors through the full connection layer.

3.2 Improved capsule network

Add convolutional layer. Figure 3 shows two network structures based on capsules. It can be seen that these two network structures change the number of convolution layers in the first substructure of the original capsule network. The number of convolution layers in the original network is 1, and these two improved networks are increased to 6 and 9. After adding convolution layer, the network is called CNN6+CapsNet2 and CNN9+CapsNet2 respectively. The purpose of adding the convolution layer can be expressed as follows. First, the feature map of the input capsule layer in the original capsule network is generated by a convolutional layer These features can be considered as low-level image features (for example, texture changes, brightness, etc.). The capsule layer finds specific entities from these low-level image features and learns the features to predict the gaze direction vector. But as we know, from the eye image to the gaze direction vector, these are two completely different kinds of data, this transformation can be thought of as requiring the acquisition of high-level semantic features. Therefore, we increase the number of convolution layers before the capsule layers. More convolutional layers are used to obtain higher-level semantic features, which are provided for the capsule layer to learn gaze direction. It is worth mentioning that the pooling layer is not used between convolution, because the pooling layer will lose a lot of information, which is very unfavorable for the subsequent capsule layers. At the same time, batch normalization layer was used before and after convolution layer to avoid overfitting problem.

Add capsule layer. Figure 4 shows the three capsule networks improved by increasing the number of capsule layers. The second and third sub-structures of the original capsule network contain one capsule layer respectively, so the number of capsule layers in the original capsule network is 2. The improved three capsule networks in **Figure 4** increase the number of capsule layers to 3, 4 and 5, respectively. The improved network is called CNN1+CapsNet3,

CNN1+CapsNet4, and CNN1+CapsNet5 respectively. The purpose of increasing the capsule layer is similar to that of increasing the convolution layer, both of which are to learn complex high-level features and predict the gaze direction better. The difference is that the addition of convolution layer makes the input feature of capsule layer become a high-level semantic feature, while the input feature of capsule layer is still a low-level image feature by adding the capsule layer. The increase in the number of capsule layers enables each capsule to learn a more complex entity object attribute than the previous capsule layer, so that the entire capsule network can learn complex object information. Then, when the information learned in the capsule layer is input into the full connection layer as features, these complex object information can be used as high-level semantic features, which can improve the performance of predicting gaze direction compared with the original capsule network.

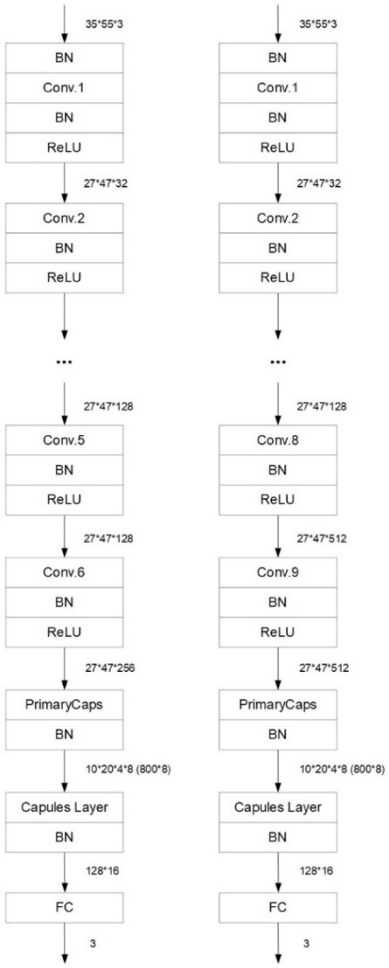

Fig. 3. The structure of improved capsule networks by adding convolutional layer.

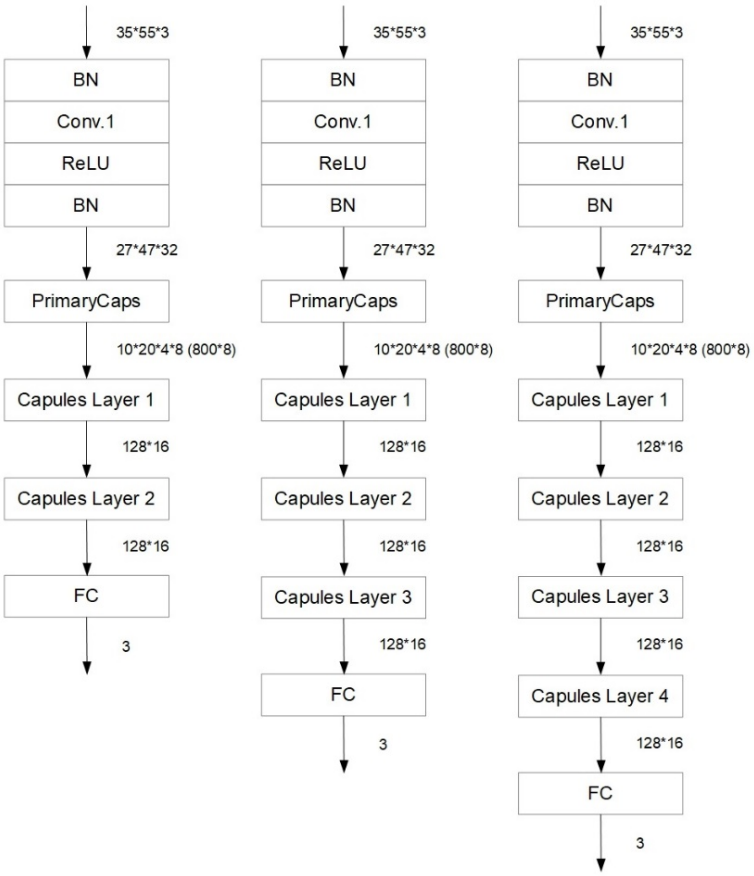

Fig. 4. The structure of improved capsule networks by adding capsule layer.

Increase the width of the capsule layer. Figure 5 shows two kinds of capsule networks that increase the width of capsule layer. The two improved networks are called CNN1+Wide CapsNet2 and CNN1+Wide CapsNet3, respectively. Unlike the previous two methods of increasing the number of layers, the width of the capsule layer is increased. It is important to note that the width of the capsule layer refers to the number of capsules in the capsule layer. Different capsules in the same capsule layer can be considered to represent different entity types in this image, so the number of capsules in the same capsule layer can represent the width of the capsule layer, similar to the filter in the convolution layer representing different features. Increasing the width of the capsule layer also increases the number of parameters of the capsule layer. Specifically, for the last capsule layer of CNN1+Wide CapsNet3, the number of input capsules is 128 and the output 512 capsules. All the input and output capsules are 16-dimensional vectors, so the number of parameters of this layer is 128*512*16*16=16,777,216 (16.78M). For the unwidened CNN1+ CapsNet3, the number of input and output capsules of the last capsule layer is 128, and the number of input and output capsules are 16-dimensional vectors, so the number of parameters is 128*512*16*16=4,194,304 (4.19M). It can be seen that

increasing the width of the capsule layer greatly increases the number of parameters of the capsule layer, and the number of parameters represents the capacity.

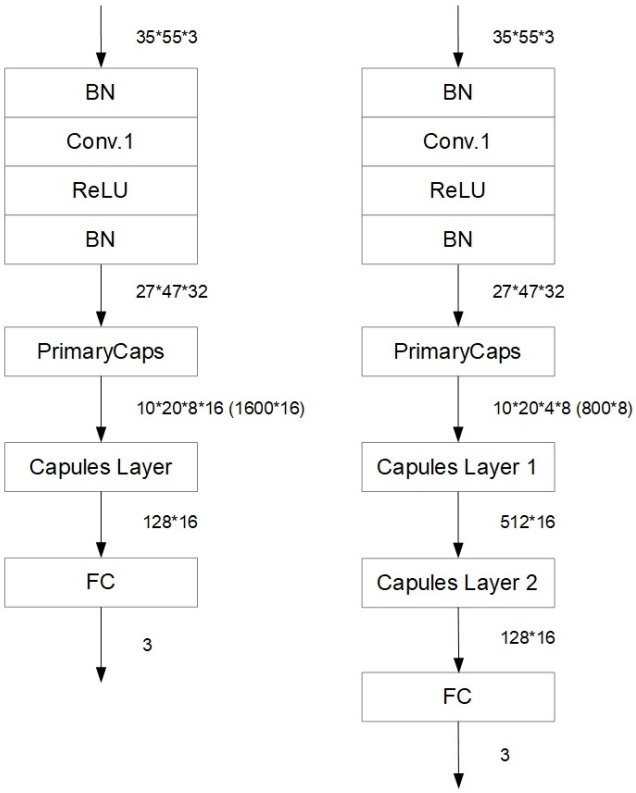

Fig. 5. The structure of improved capsule networks by widening capsule layer.

3.3 Loss function

Since the prediction of gaze direction by all models and the real gaze direction are real vector, we choose the mean square loss (MSE) function to calculate the difference between the prediction direction vector and the real direction vector, and then evaluate the performance of different models. In particular, defined τ as the eye image data set, $x_i \in \tau$ as the i-th eye image, and y_i is the gaze direction vector corresponding to x_i. Given that the prediction function of the model is $P(\cdot)$, $y_i' = P(x_i)$ is defined as the gaze direction vector of input eye image x_i predicted by the model. Therefore, the mean square loss function can be expressed as equation 4.

$$L = \frac{1}{m}\sum_{i=1}^{m}\|y_i' - y_i\|_2^2 = \frac{1}{m}\sum_{i=1}^{m}\|P(x_i) - y_i\|_2^2 \tag{4}$$

Where m represents the number of images in the eye image data set, and $\|\cdot\|_2$ represents the L2 normalization. Obviously, it can be seen that the optimization objective of this loss function is

to minimize the distance between the predicted gaze direction vector y_i' and the real gaze direction vector y_i, so that the predicted gaze direction can be as close as possible to the real gaze direction ($y_i' = y_i$). In the experiment, Adam optimization algorithm was used to optimize the mean square loss.

4 Experiments

4.1 Database

In order to evaluate the performance of the improved capsule network and other comparison networks, a suitable experiment was conducted. At the same time, since image-based gaze estimation is greatly affected by light transformation and other factors, we need a large number of eye image data sets containing gaze direction vectors. and we hope that the range of gaze directions in the data set is wide, rather than concentrated in a small area (e.g., this is a problem with the MPIIGaze dataset which is widely used, the images in the MPIIGaze dataset [17] are collected when people use laptops, so the direction of gaze is limited to the computer screen with a small Angle of view). At the same time, considering the time and money it takes to obtain real data with the camera, we prefer to use software to generate a large number of eye images and gaze direction data that meet the requirements.

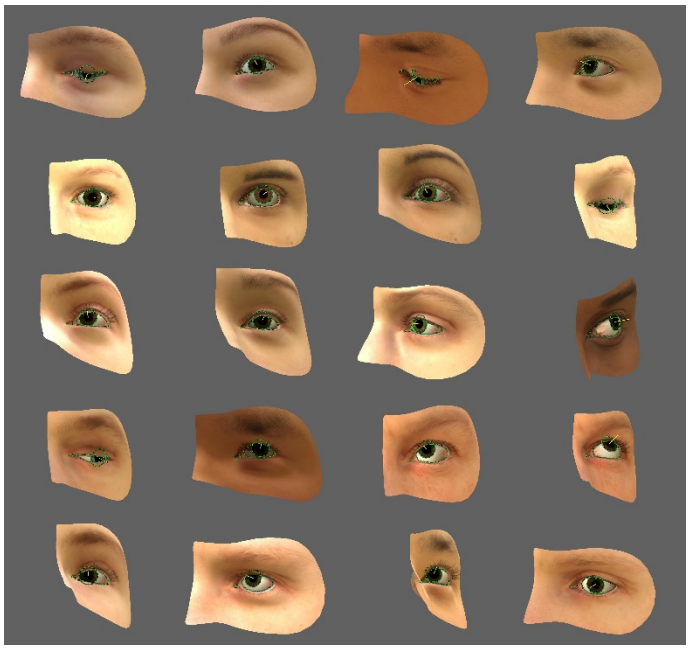

Fig. 6. Generated eye images, which are 800*600 RGB images in a total of 50,000.

We used the program UnityEyes provided in [18] to generate datasets. UnityEyes uses an approximate eyeball model to model real eyeballs, and the most important one is the modeling

of iris refraction. In addition, UnityEyes used methods such as head scanning registration and retopology, and Morphable eye region model to build the region around the eyes, and made detailed modeling of eye movement and details around the eyes. **Figure 6** shows the generated eye images, which are 800*600 RGB images in a total of 50,000. As can be seen, these generated images are quite real, with fine texture and natural lighting levels. Need to point out that green point in the image of **Figure 6** is the characteristic points of the eye, two circles surrounded by green point, one is the pupil and the other is the iris, yellow line is gaze direction vector, these points, lines, and the gray background is added to the image in order to better display the image, the image itself does not contain these points, lines, and the gray background. Each generated image has a corresponding gaze direction vector. Map all the gaze direction vectors to a plane, as shown in **Figure 7**. It's easy to see that the horizontal distribution of the gaze direction is -70 degrees to 70 degrees, and the vertical distribution is -45 degrees to 45 degrees. The range between them goes far beyond the existing set of real gaze data.

After generating eye images, we need to preprocess these images, and the specific operations are as follows. First, Gaussian white noise is added to simulate the noise generated by camera shooting in real situations. Second, since the generated images do not rotate the face, and considering that the camera may rotate relative to the face in the real situation, we randomly select an angle to rotate the image for each image. Third, change the image size to 55*35, and normalize the RGB values to the range of 0 to 1, so that every network training faster. After image preprocessing, all generated images are randomly divided into training set and test set, so that the training set and test set contain the same number of images. After that, we trained all the models on the training set and evaluated them on the test set.

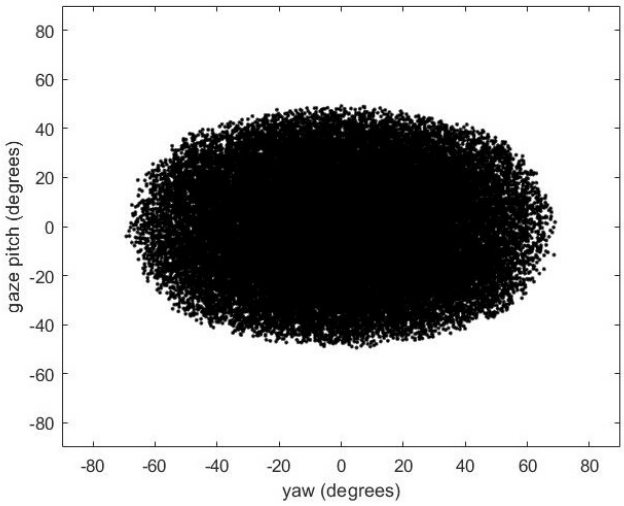

Fig. 7. The gaze direction distribution of the generated dataset.

4.2 Experimental Settings

The improved model based on the original capsule network was compared with some excellent deep learning models proposed in recent years. Our improved models included CNN6+CapsNet2 and CNN9+CapsNet2 by adding convolutional layers, CNN1+CapsNet3, CNN1+CapsNet4, and CNN1+CapsNet5 by adding capsule layers, CNN1+Wide CapsNet2 and CNN1+Wide CapsNet3 by widening capsule layers. The deep models used for comparison included the original capsule network (i.e., CNN1+CapsNet2), convolutional network (i.e., CNN-7 and CNN-12), residual network (i.e., ResNet-18) and squeeze network (i.e., SqueezeNet-v10 and SqueezeNet-v11). The structure of CNN-7, CNN-12, and ResNet-18 is shown in **Figure 8**.

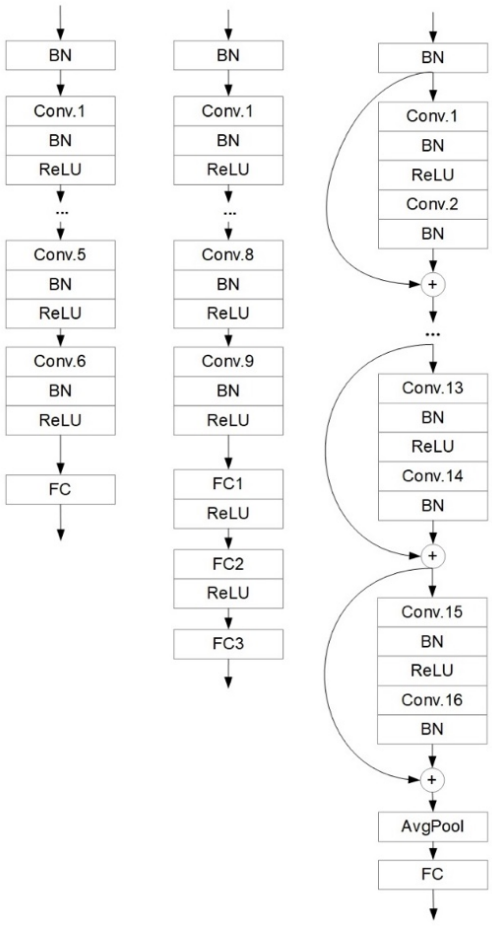

Fig. 8. The structure of CNN-7, CNN-12, and ResNet-18.

All models were tested multiple times on the data set to reduce the non-significance caused by randomness. On this basis, all experimental results are the average of each experimental result. The setting of each super-parameter in the experiment has been tested many times on the experimental platform to compare whether the comprehensive performance of all models can

reach the optimal under different settings of super-parameter. The specific super parameters are set as follows: the training batch size is set to 8, the training epoch is set to 10, and the learning rate is set to 0.01. (In particular, the parameters of CNN6+CapsNet2, CNN9+CapsNet2, CNN1+CapsNet3, CNN1+CapsNet4, CNN1+CapsNet5, CNN1+Wide CapsNet2 and CNN1+Wide CapsNet3 are about 14.35M, 19.15M, 17.39M, 21.59M, 25.78M, 52.77M and 69.30M, respectively. CNN1+CapsNet2, CNN-7, CNN-12, ResNet-18, SqueezeNet-v10 and SqueezeNet-v11 have parameters of about 13.20M, 0.021M, 0.19M, 0.18M, 0.73M and 0.73M respectively). Therefore, the experiment was carried out on a CentOS 7 based high-performance computer equipped with Intel Xeon Silver 4110 CPU, 128G RAM, and Nvidia Titan V GPU, and based on PyTorch 1.0.0 deep learning platform.

4.3 Qualitative Analysis

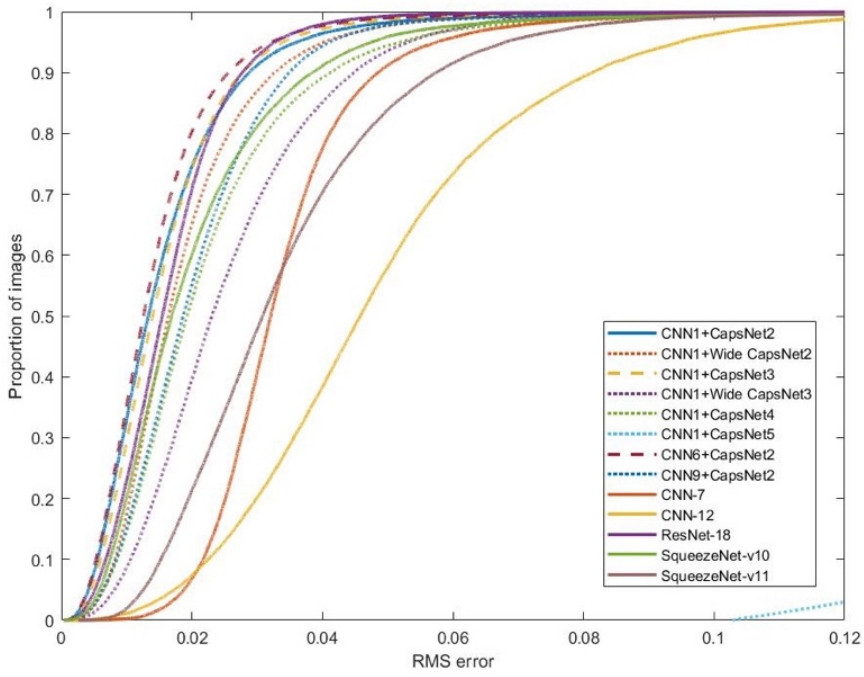

Fig. 9. The proportion of images with RMS error of all models.

We use image proportion and root mean square (RMS) error curves to qualitatively analyze the performance of different models. The independent variable of the curve is the root mean square, and the dependent variable is the image proportion. The specific numerical calculation process is as follows. For each model, each eye image is predicted on the test set, and the predicted gaze direction vector was subtracted from the real gaze direction vector of the eye image to calculate the root mean square error. The root mean square error corresponding to each eye image was formed into a set. Given a threshold, calculate the number of root mean square errors less than the threshold in the set (corresponding to the number of images with the error

of the prediction and real gaze direction vector less than a certain value), divide by the total number of root mean square errors in the set (corresponding to the total number of images), and get the image proportion. The threshold value is increased from zero to obtain the image proportion and root mean square error curve of the model. An obvious corollary is that the larger the area under each image proportion and root mean square error curve is, the better the prediction performance of the corresponding curve model will be. **Figure 9** shows the image proportion and root mean square error curve of all models. As can be seen from the figure, the curve of our improved model CNN6+CapsNet2 is the highest, that is, the performance is the best. After that, the three models with very similar performance are CNN1+CapsNet3, CNN1+CapsNet2 (i.e., the original capsule network) and ResNet-18. After that, it is followed by CNN1+Wide CapsNet2, CNN9+CapsNet2, SqueezeNet-v10, CNN1+CapsNet4, CNN1+Wide CapsNet3, SqueezeNet-v11, CNN-7, CNN-12, and finally, CNN1+CapsNet5 without convergence.

For CNN6+CapsNet2 and CNN9+CapsNet2 improved by adding convolution layer, CNN6+CapsNet2 obtained the best performance, but the performance of CNN9+CapsNet2 was significantly lower than that of the original capsule network. The reason for this result is that proper addition of convolution layer can indeed provide high-level semantic features and then improve the performance of capsule network. However, if too many convolution layers are added, it is very easy for the input image to lose information after the multi-layer convolution transformation including the down-sampling operations (typically involves down-sampling operations, such as stride greater than one or pooling layer), which greatly reduces the information content of the input capsule layer and ultimately reduces the performance of the entire network. For CNN1+CapsNet3, CNN1+CapsNet4 and CNN1+CapsNet5 improved by adding capsule layer, the performance of CNN1+CapsNet3 was slightly higher than that of the original capsule network, while the performance of CNN1+CapsNet4 was far lower than that of the original capsule network, while the performance of CNN1+CapsNet5 did not converge. The reason for this is that adding a layer of capsules allows to learn complex object properties as features and thus slightly improve performance. When the multi-layer capsule layer is added, the final capsule layer learns too complex entities or does not learn any entities (because the most complex entities are corresponding to the previous capsule layer). In this case, the network degenerates, leading to a sharp decline in performance or even non-convergence. For CNN1+Wide CapsNet2 and CNN1+Wide CapsNet3 improved by increasing the width of capsule layer, their performance was lower than that of the original capsule network. The reason for this result is that they are used for several times the parameters of the original capsule network, which makes them enter the overfitting state and thus have low performance in the test set. CNN6+CapsNet2 and CNN1+CapsNet3 that exceed the performance of the original capsule network indicate that appropriate addition of convolution layer or capsule layer can improve the overall performance of the network. The performance of ResNet-18 is close to that of the original capsule network. This is because ResNet-18 has many convolutional layers, which is consistent with our knowledge that even though each convolutional layer is not so wide (that is, the number of feature maps is not large), the performance can be improved by deepening the number of network layers. CNN-7 and CNN-12 ranked last in performance, which was due to their low learning parameters and simple network structure.

4.4 Quantitative Analysis

We use the box plot of root mean square error between the predicted gaze direction vector and the real gaze direction vector to conduct quantitative analysis on all models, as shown in **Figure 10** (the box plot of CNN1+CapsNet5 is too high, so it is not shown in the figure, that is, only the box plot of the convergent model is shown in the figure). Each box in the box plot includes lower limit, lower quartile, median, upper quartile and an upper limit from bottom to top. Where the range of upper and lower limits is 1.5 times the quartile, the data exceeding the upper and lower limits are marked in red and are called outliers. It should be noted that each box in the box plot represents the distribution of root mean square error predicted by the corresponding model, so the lower the box is, the better the network performance will be. As can be seen from the figure, among all convergent networks, CNN6+CapsNet2 has the best performance and CNN-12 has the worst performance. This conclusion is exactly the same as that obtained by the image proportion and root mean square error curves. In addition, the number of outliers in red represents only about 3% of all data.

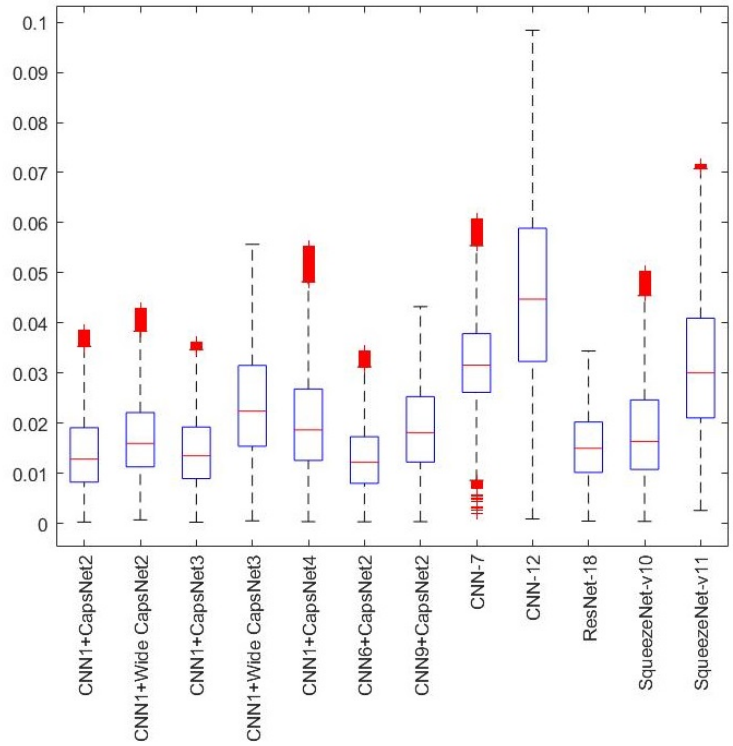

Fig. 10. The box plot of RMS error between the predicted gaze direction vector of different models and the real gaze direction vector.

In addition, Table 1 quantitatively describes the details of the comparison between our improved capsule model and other comparison models through two general parameter estimation methods (i.e., point estimation and interval estimation). Point estimation is estimated by subtracting the median root mean square error of the two comparison models (Model 1-

Model 2). Interval estimation is estimated by subtracting the upper and lower limits of root mean square error (Model 1-Model 2) of the two comparison models respectively. Therefore, if the point estimation value is negative and the interval estimation range is negative, the performance of Model 1 is better than Model 2. As can be seen from Table 1, when CNN6+CapsNet2 and CNN1+CapsNet3 are compared with other comparison models, both point estimation and interval estimation are negative. This shows that from the perspective of point estimation and interval estimation, the root mean square error of CNN6+CapsNet2 and CNN1+CapsNet3 is smaller than that of other comparable models, that is, the performance is better. When CNN1+CapsNet5 is compared with other comparison models, both point estimation and interval estimation are relatively large positive numbers. This shows that from the perspective of point estimation and interval estimation, the root mean square error of CNN1+CapsNet5 is much higher than that of other comparable models, that is, the performance is far worse than that of other comparable models. This is consistent with the conclusions of the image proportion and root mean square error curves and box plot.

Table 1. Point estimation and interval estimation between improved capsule networks and other models.

Model 1	Model 2	Point estimation	Interval estimation
CNN6+CapsNet2	CNN1+CapsNet2	-0.000657	[-0.0041, -0.0000542]
	ResNet-18	-0.0028	[-0.000014, -0.00012125]
	SqueezeNet-v10	-0.0042	[-0.0158, -0.00007336]
	SqueezeNet-v11	-0.0178	[-0.0373, -0.0023]
	CNN-7	-0.0194	[-0.0264, -0.0016]
	CNN-12	-0.0326	[-0.0639, -0.00057647]
CNN9+CapsNet2	CNN1+CapsNet2	0.0053	[0.0000904, 0.0047]
	ResNet-18	0.0031	[-0.00008505, 0.0088]
	SqueezeNet-v10	0.0017	[-0.007, -0.00003716]
	SqueezeNet-v11	-0.0119	[-0.0285, -0.0022]
	CNN-7	-0.0135	[-0.0176, -0.0016]
	CNN-12	-0.0266	[-0.0551, -0.00054027]
CNN1+CapsNet3	CNN1+CapsNet2	-0.000052	[-0.0024, -0.00001901]
	ResNet-18	-0.0015	[-0.00019446, 0.0017]
	SqueezeNet-v10	-0.0029	[-0.0141, -0.00014657]
	SqueezeNet-v11	-0.0165	[-0.0356, -0.0023]
	CNN-7	-0.0181	[-0.0247, -0.0017]
	CNN-12	-0.0312	[-0.0622, -0.00064968]
CNN1+CapsNet4	CNN1+CapsNet2	0.0058	[-0.00007055, 0.0168]
	ResNet-18	0.0037	[-0.0001049, 0.0209]
	SqueezeNet-v10	0.0023	[-0.0005701, 0.0051]
	SqueezeNet-v11	-0.0114	[-0.0164, -0.0022]
	CNN-7	-0.0129	[-0.0055, -0.0016]
	CNN-12	-0.0261	[-0.0430, -0.00056012]
CNN1+CapsNet5	CNN1+CapsNet2	0.2786	[0.1025, 0.5016]
	ResNet-18	0.2764	[0.1023, 0.5057]
	SqueezeNet-v10	0.2750	[0.1024, 0.4899]
	SqueezeNet-v11	0.2614	[0.1002, 0.4684]
	CNN-7	0.2596	[0.1008, 0.4793]
	CNN-12	0.2467	[0.1019, 0.4418]
CNN1+Wide CapsNet2	CNN1+CapsNet2	0.0031	[0.00042176, 0.0044]
	ResNet-18	0.000969	[0.00024631, 0.0085]
	SqueezeNet-v10	-0.00041	[-0.0073, 0.0002942]
	SqueezeNet-v11	-0.0141	[-0.0288, -0.0019]

	CNN-7	-0.0156	[-0.0179, -0.0012]
	CNN-12	-0.0288	[-0.0554, -0.00020891]
CNN1+Wide CapsNet3	CNN1+CapsNet2	0.0096	[-0.00024396, 0.0171]
	ResNet-18	0.0074	[-0.00006851, 0.0213]
	SqueezeNet-v10	0.0060	[-0.0001164, 0.0055]
	SqueezeNet-v11	-0.0076	[-0.016, -0.0021]
	CNN-7	-0.0092	[-0.0051, -0.0014]
	CNN-12	-0.0223	[-0.0427, -0.00038671]

4.5 Discussions

In this section, more details on the use of CNN6+CapsNet2 and CNN1+CapsNet3 models in the gaze estimation problem are presented. In the generated eye dataset, CNN6+CapsNet2 and CNN1+CapsNet3 achieved higher performance than other models. **Figure 11** and **Figure 12** respectively show the results of gaze direction prediction of CNN6+CapsNet2 and CNN1+CapsNet3. As can be seen from the figure, the difference between the predicted and the real gaze direction vector is small, usually within a few degrees. These examples also show that our improved two models can overcome the extremely challenging difficulties in gaze estimation (such as obvious light changes, face rotation or occlusion, etc.), and predict the prediction results with almost no error with the real gaze direction.

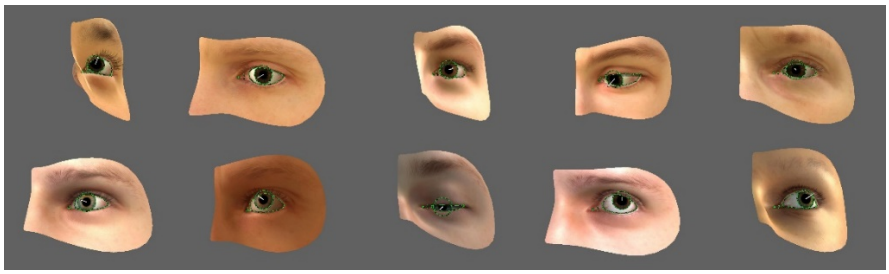

Fig. 11. The examples of gaze direction prediction of CNN6+CapsNet2. The gray segment is the predicted gaze direction vector, and the yellow segment is the real gaze direction vector.

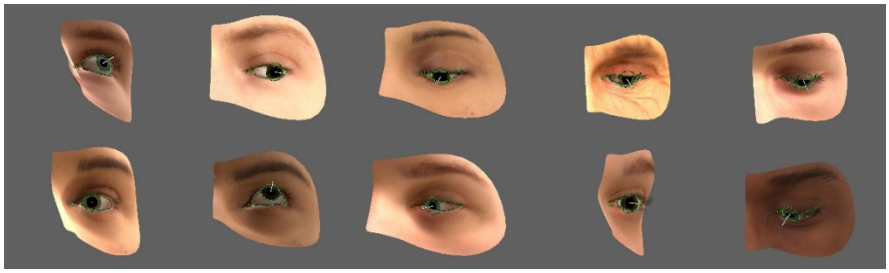

Fig. 12. The examples of gaze direction prediction of CNN1+CapsNet3. The gray segment is the predicted gaze direction vector, and the yellow segment is the real gaze direction vector.

5 Conclusion

In this study, on the basis of the original capsule network, we innovatively consider three important factors to improve the model and apply them to the gaze estimation in wireless sensor network. First, the number of convolutional layers in the original capsule network is increased, so that the convolutional layer can provide high-level semantic features for subsequent capsule layers. Secondly, the number of capsule layers in the original capsule network is increased so that the capsule layer can learn complex object features based on the low-level image features provided by the convolution layer. Third, the width of the capsule layer in the original capsule network is increased to increase the capacity of the whole network. Through experiments, we can draw the following conclusions. Appropriate improvement of the original capsule network from the first and second aspects can improve the performance of the model (e.g., the improved CNN6+CapsNet2 and CNN1+CapsNet3), but if the improvement range is too large, the model performance will decline or even not converge (e.g., CNN9+CapsNet2, CNN1+CapsNet4 and CNN1+CapsNet5). In addition, improvement in the third aspect is likely to result in overfitting of the model and decrease the network performance (e.g., CNN1+Wide CapsNet2 and CNN1+Wide CapsNet3). The reasons for these conclusions are discussed in detail in the experimental analysis section. In the future, we will use more detailed improvement schemes to achieve higher performance improvement.

Acknowledgements. The authors would like to acknowledge the grant 61862043 approved by National Natural Science Foundation of China, key grants 20181ACB20006 and 20171ACB21017 approved by Natural Science Foundation of Jiangxi Province for supporting this study.

References

[1] Gale, Alastair G. "A note on the remote oculometer technique for recording eye movements." Vision research (1982).
[2] Yoo, Dong Hyun, et al. "Non-contact eye gaze tracking system by mapping of corneal reflections." fgr. IEEE, 2002.
[3] Guestrin, Elias Daniel, and Moshe Eizenman. "General theory of remote gaze estimation using the pupil center and corneal reflections." IEEE Transactions on biomedical engineering 53.6 (2006): 1124-1133.
[4] Baluja, Shumeet, and Dean Pomerleau. "Non-intrusive gaze tracking using artificial neural networks." Advances in Neural Information Processing Systems. 1994.
[5] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.
[6] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[8] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115.3 (2015): 211-252.

[9] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
[10] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[11] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[12] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.
[13] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.
[14] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[15] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).
[16] Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." Advances in Neural Information Processing Systems. 2017.
[17] Zhang, Xucong, et al. "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation." IEEE Transactions on Pattern Analysis and Machine Intelligence 41.1 (2019): 162-175.
[18] Wood, Erroll, et al. "Learning an appearance-based gaze estimator from one million synthesised images." Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications. ACM, 2016.