Deep Learning-based Pupil Center Detection for Fast and Accurate Eye Tracking System

Anonymous ECCV submission

Paper ID 3137

1 Implementation details

This section describes the hyper-parameters and optimizer of each network in the proposed PCT system.

1.1 Face Detection Network

We trained the face detection network for 300 epochs in total, and used SGD optimizer with learning rate decay by 10^{-1} at 200 and 250 epoch. The momentum and weight decay of SGD were 0.9 and 5×10^{-4}, respectively. The batch size was 64.

1.2 Glasses Removal Network

We trained the glasses removal network for 200 epochs. We adopted Adam optimizer. The learning rate was 2×10^{-4} and β_1 and β_2 were set to 0.5 and 0.999, respectively. The batch size was 1.

1.3 Segmentation Network

We trained the segmentation network for 600 epochs and the batch size was 128. We used SGD as an optimizer and the initial learning rate was 10^{-2}. We adopted multi-step learning rate decay by 10^{-1} at milestone epochs of 80, 200, 360 and 480. The momentum and weight decay were 0.9 and 5×10^{-4}, respectively. In addition, gradient clipping was used to stabilize training.

1.4 Representation Learning Network

We used Adam optimizer for two discriminators and vectorization network. The learning rate was 10^{-4}. The β_1 and β_2 of Adam optimizer were set to 0.9 and 0.999, respectively. And the weight decay was 5×10^{-4}. Since these networks were trained with the segmentation network at same time, the total number of epochs and batch size were same as the segmentation network.

1.5 Glasses Classifier

We trained the glasses classifier for 200 epochs. We used SGD as an optimizer. The momentum and weight decay of SGD were 0.9 and 5×10^{-4}, respectively. The learning rate starts at 10^{-2} and decreases by 10^{-1} after each 50 epochs.

2 Architectures

This section describes the network architectures we used. In the following figures the first column of each figure refers to the type of operation to be performed in each step, and the second column refers to the type of tensor before and after the operation. Column 3 refers to the stride size and padding size of the convolution or pooling operations. Finally, column 4 represents the size of the convolution filter.

Fig. 1 shows the architecture of face detection network. Furthermore, non-local block in Fig. 1 refers to the non-local block proposed in [1]. CReLU means the activation method proposed in [2]. Fig. 2 shows the architecture of glasses removal network. In Fig. 2, residual block and self-attention block refer to residual block used in [3] and self-attention block proposed in [4] for generator, respectively. Fig. 3 shows the architecture of segmentation network for pupil center detection. Convolution 9(SC) in Fig. 3 means the concatenation and convolution of the output of Convolution 2(★) of the segmentation network with the output of Convolution 8. Figures 4 and 5 show discriminators for measuring mutual information between low level features, latent features and representation, respectively. Fig. 6 shows the architecture of vectorization network. Fig. 7 shows the architecture of glasses classifier network.

The padding type of the other networks except for the glasses removal network is zero padding. On the other hand, in the glasses removal network, only transposed convolution 1 and transposed convolution 2 use zero padding. All other types of operations except the self-attention block employ reflection padding. Self-attention block does not use any padding due to the operation nature. Non-local blocks in face detection networks do not use padding for the same reason.

Type	Input → Output shape (Channel, Height, Width)	(Stride, Padding size)	Filter Size
Convolution 1	3 X 1024 X 1024 → 24 X 256 X 256	(4, 3)	7 X 7
2D Batch Normalization & CReLU			
Max pooling	48 X 256 X 256 → 48 X 128 X 128	(2, 1)	3 X 3
Convolution 2	48 X 128 X 128 → 64 X 64 X 64	(2, 2)	5 X 5
2D Batch Normalization & CReLU			
Max pooling	128 X 64 X 64 → 128 X 32 X 32	(2, 1)	3 X 3
Convolution 3	128 X 32 X 32 → 128 X 32 X 32	(1, 1)	3 X 3
2D Batch Normalization & ReLU			
Convolution 4	128 X 32 X 32 → 256 X 32 X 32	(1, 1)	3 X 3
2D Batch Normalization & ReLU			
Non-Local Block	256 X 32 X 32 → 256 X 32 X 32	-	-
Convolution 5	256 X 32 X 32 → 128 X 32 X 32	(1, 0)	1 X 1
2D Batch Normalization & ReLU			
Convolution 6	128 X 32 X 32 → 128 X 32 X 32	(1, 1)	3 X 3
2D Batch Normalization & ReLU			
Convolution 7	128 X 32 X 32 → 256 X 16 X 16	(2, 1)	3 X 3
2D Batch Normalization & ReLU			
Convolution 8	256 X 16 X 16 → 128 X 16 X 16	(1, 0)	1 X 1
2D Batch Normalization & ReLU			
Convolution 9	128 X 16 X 16 → 256 X 8 X 8	(2, 1)	3 X 3
2D Batch Normalization & ReLU			

Fig. 1. Face Detection Network

Type	Input → Output shape (Channel, Height, Width)	(Stride, Padding size)	Filter Size
Convolution 1	1 X 128 X 128 → 1 X 128 X 128	(1, 3)	7 X 7
Convolution 2	1 X 128 X 128 → 64 X 64 X 64	(2, 2)	6 X 6
2D Instance Normalization & ReLU			
Convolution 3	64 X 64 X 64 → 128 X 32 X 32	(2, 1)	3 X 3
2D Instance Normalization & ReLU			
Convolution 4	128 X 32 X 32 → 256 X 16 X 16	(2, 1)	3 X 3
2D Instance Normalization & ReLU			
Residual block	256 X 16 X 16 → 256 X 16 X 16	-	-
Self-attention block	256 X 16 X 16 → 256 X 16 X 16	-	-
Residual block	256 X 16 X 16 → 256 X 16 X 16	-	-
Transposed convolution 1	256 X 16 X 16 → 128 X 32 X 32	(2, 1)	3 X 3
2D Instance Normalization & ReLU			
Transposed convolution 2	128 X 32 X 32 → 64 X 64 X 64	(2, 1)	3 X 3
Transposed convolution 3	64 X 64 X 64 → 1 X 128 X 128	(2, 2)	6 X 6
Convolution 5	1 X 128 X 128 → 1 X 128 X 128	(1, 3)	7 X 7
Tanh			

Fig. 2. Glasses Removal Network

Type	Input → Output shape (Channel, Height, Width)	(Stride, Padding size)	Filter Size
Convolution 1	1 X 48 X 48 → 64 X 48 X 48	(1, 3)	7 X 7
2D Batch Normalization & ReLU			
Convolution 2 (★)	64 X 48 X 48 → 64 X 48 X 48	(1, 3)	7 X 7
2D Batch Normalization & ReLU			
Max pooling	64 X 48 X 48 → 64 X 24 X 24	(2, 0)	2 X 2
Convolution 3	64 X 24 X 24 → 128 X 24 X 24	(1, 2)	5 X 5
2D Batch Normalization & ReLU			
Convolution 4	128 X 24 X 24 → 128 X 24 X 24	(1, 2)	5 X 5
2D Batch Normalization & ReLU			
Max pooling	128 X 24 X 24 → 128 X 12 X 12	(2, 0)	2 X 2
Convolution 5	128 X 12 X 12 → 256 X 12 X 12	(1, 1)	3 X 3
2D Batch Normalization & ReLU			
Convolution 6	256 X 12 X 12 → 256 X 12 X 12	(1, 1)	3 X 3
2D Batch Normalization & ReLU			
Up sampling	256 X 12 X 12 → 256 X 48 X 48	-	-
Convolution 7	256 X 48 X 48 → 128 X 48 X 48	(1, 1)	3 X 3
Convolution 8	128 X 48 X 48 → 64 X 48 X 48	(1, 1)	3 X 3
Convolution 9(SC)	128 X 48 X 48 → 64 X 48 X 48	(1, 2)	5 X 5
Convolution 10	64 X 48 X 48 → 1 X 48 X 48	(1, 0)	1 X 1

Fig. 3. Segmentation Network

Type	Input → Output shape (Channel, Height, Width)	(Stride, Padding size)	Filter Size
Convolution 1	128 X 48 X 48 → 256 X 48 X 48	(1, 0)	1 X 1
ReLU			
Convolution 2	256 X 48 X 48 → 256 X 48 X 48	(1, 0)	1 X 1
ReLU			
Convolution 3	256 X 48 X 48 → 1 X 48 X 48	(1, 0)	1 X 1

Fig. 4. Discriminator for measure between low level feature and representation

Type	Input → Output shape (Channel, Height, Width)	(Stride, Padding size)	Filter Size
Convolution 1	320 X 12 X 12 → 1024 X 12 X 12	(1, 0)	1 X 1
ReLU			
Convolution 2	1024 X 12 X 12 → 1024 X 12 X 12	(1, 0)	1 X 1
ReLU			
Convolution 3	1024 X 12 X 12 → 1 X 12 X 12	(1, 0)	1 X 1

Fig. 5. Discriminator for measure between latent feature and representation

Type	Input → Output shape (Channel, Height, Width)	(Stride, Padding size)	Filter Size
Convolution 1	64 X 48 X 48 → 64 X 24 X 24	(2, 1)	4 X 4
Convolution 2	64 X 24 X 24 → 128 X 12 X 12	(2, 1)	4 X 4
2D Batch Normalization & ReLU			
Convolution 3	128 X 12 X 12 → 256 X 6 X 6	(2, 1)	4 X 4
2D Batch Normalization & ReLU			
Convolution 4	256 X 6 X 6 → 512 X 3 X 3	(2, 1)	4 X 4
2D Batch Normalization & ReLU			
Convolution 5	512 X 3 X 3 → 1024 X 1 X 1	(1, 1)	5 X 5
FC	1024 X 1 X 1 → 64 X 1 X 1	-	-

Fig. 6. Vectorization Network

Type	Input → Output shape (Channel, Height, Width)	(Stride, Padding size)	Filter Size
Convolution 1	1 X 128 X 128 → 64 X 64 X 64	(2, 2)	5 X 5
LeakyReLU(0.2)			
Convolution 2	64 X 64 X 64 → 128 X 32 X 32	(2, 2)	5 X 5
2D Batch Normalization & LeakyReLU(0.2)			
Convolution 3	128 X 32 X 32 → 256 X 16 X 16	(2, 2)	5 X 5
2D Batch Normalization & LeakyReLU(0.2)			
Convolution 4	256 X 16 X 16 → 512 X 16 X 16	(1, 1)	3 X 3
2D Batch Normalization & LeakyReLU(0.2)			
Convolution 5	512 X 16 X 16 → 1 X 16 X 16	(1, 1)	3 X 3
Global average pooling	1 X 16 X 16 → 1 X 1 X 1	-	-

Fig. 7. Glasses Classifier

References

1. Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.
2. Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. Faceboxes: A cpu real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2017.
3. Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
4. Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.