Teaching Where to Look: Attention Similarity Knowledge Distillation for Low Resolution Face Recognition

Sungho Shin[1], Joosoon Lee[1], Junseok Lee[1], Yeonguk Yu[1], and Kyoobin Lee[1] *

School of Integrated Technology (SIT), Gwangju Institute of Science and Technology (GIST), Cheomdan-gwagiro 123, Buk-gu, Gwangju 61005, Republic of Korea.
{hogili89,joosoon1111,junseoklee,yeon_guk,kyoobinlee}
@gist.ac.kr

1 Implementation Details of Previous Methods

Various knowledge distillation approaches have been proposed to transfer the teacher network's knowledge to student network. Usually, the loss term of the knowledge distillation methods can be defined as the summation of the target task's loss and the distillation loss. $L_{total} = L_{target} + L_{distill}$. To compare the A-SKD with the previous distillation methods (HORKD [1], F-KD [2], and AT [5]), we re-implemented those methods using the official implementation code.

The distillation loss function of the AT [5] is defined as the L2 distance between the teacher and student network's self-attention maps. For the ResNet50, the AT distills the self-attention maps on the four blocks. We performed the experiments using the official implementation of AT (`https://github.com/szagoruyko/attention-transfer`).

The distillation loss function of the F-KD [2] is defined as the L2 distance between the teacher and student network's feature maps. The penultimate layer's features (output of the backbone network before input to the margin) are utilized for the distillation. We performed the experiments using the official implementation of F-KD (`https://github.com/fvmassoli/cross-resolution-face-recognition`).

The loss function of the previous SOTA method (HORKD [1]) is defined as

$$\mathcal{L}_{distill} = L_1 + \alpha L_2 + \beta L_3 + \gamma L_C \tag{1}$$

where L_1, L_2, L_3, and L_C indicate the individual-level, pair-level, triplet-level, and group-level knowledge distillation loss, respectively; the different order relational distillation losses are tuned by the factor of α, β, γ in order. In our implementation, $\alpha = 0.1$, $\beta = 0.2$, and $\gamma = 0.1$ after the hyperparameter search using the same protocol with our method. The penultimate layer's features (output of the backbone network before input to the margin) are utilized for the distillation. Because HORKD has no official implementation code, we referenced the official implementation of RKD [3] (`https://github.com/lenscloth/RKD`).

* Corresponding author.

2 Extension to Other Tasks

2.1 Object Classification.

We utilized the ResNet50-CBAM backbone for the ImageNet training. The model was trained by SGD optimizer for 90 epochs with learning rate = 0.4, batch size = 1024, momentum = 0.9, and weight decay = 0.0001. Mixed precision was applied for the efficient training. We referenced the code of `https://github.com/rwightman/pytorch-image-models` for the training and validation on ImageNet benchmark.

2.2 Face Detection

The WIDER FACE [4] was utilized for training and validation to extend A-SKD to face detection. The WIDER FACE contained 32,203 images from 61 event classes, totaling 393,703 faces. The WIDER FACE categorized images into three face detection difficulties: easy, medium, and hard, considering scale, occlusion, and pose. Face detection performance on WIDER FACE was evaluated by overall mean average precision (mAP), and easy, medium, and hard images were evaluated separately. We trained TinaFace [6] on WIDER FACE by combining it with the CBAM attention module. LR images with 16× and 32× down-sampling ratio were trained to validate A-SKD effectiveness. TinaFace used ResNet-50 backbone network; and the loss function combined face detection and attention distillation losses. For the detection task, we distilled the attention maps of the four ResNet blocks, not every convolution layer. The model was trained by SGD optimizer for 150 epochs with learning rate = 0.001, momentum = 0.9, and weight decay = 0.0005 on two Titan RTX (24GB) GPUs with batch size = 8. We used non-maximum suppression threshold = 0.4 and confidence level = 0.02. We referenced the code of `https://github.com/Media-Smart/vedadet` for the implementation of TinaFace.

References

1. Ge, S., Zhang, K., Liu, H., Hua, Y., Zhao, S., Jin, X., Wen, H.: Look One and More: Distilling Hybrid Order Relational Knowledge for Cross-Resolution Image Recognition. Proceedings of the AAAI Conference on Artificial Intelligence **34**(07), 10845–10852 (2020). https://doi.org/10.1609/aaai.v34i07.6715, `https://ojs.aaai.org/index.php/AAAI/article/view/6715`
2. Massoli, F.V., Amato, G., Falchi, F.: Cross-resolution learning for Face Recognition. Image and Vision Computing **99**, 103927 (jul 2020). https://doi.org/10.1016/j.imavis.2020.103927
3. Park, W., Kim, D., Lu, Y., Cho, M.: Relational Knowledge Distillation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2019-June**, 3962–3971 (apr 2019), `http://arxiv.org/abs/1904.05068`
4. Yang, S., Luo, P., Loy, C.C., Tang, X.: WIDER FACE: A Face Detection Benchmark. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5525–5533 (2016). https://doi.org/10.1109/CVPR.2016.596
5. Zagoruyko, S., Komodakis, N.: Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings (dec 2016)
6. Zhu, Y., Cai, H., Zhang, S., Wang, C., Xiong, Y.: Tinaface: Strong but simple baseline for face detection. ArXiv **abs/2011.13183** (2020)