Supplementary Material for 'Two-stream Flow-guided Convolutional Attention Networks for Action Recognition '

An Tran Loong-Fah Cheong
Department of Electrical & Computer Engineering, National University of Singapore
an.tran@u.nus.edu eleclf@nus.edu.sg

1. 2D version of flow-guided convolutional attention networks

In this section, we briefly describe the differences between the 2D-FCAN and 3D-FCAN network. Let $x\_rgb^l \in \mathbb{R}^{C_l \times H_l \times W_l}$, $x\_flow^l \in \mathbb{R}^{C_l \times H_l \times W_l}$ be the feature map of a layer $l \in \{0, 1, ..., L\}$ in *spatial-* and *temporal-CNN* (2D version) respectively, with C_l, H_l, W_l being the number of channels, height and width of the feature map. We denote $x\_rgb^l_{h,w} \in \mathbb{R}^{C_l \times H_l \times W_l}$, $x\_flow^l_{h,w} \in \mathbb{R}^{C_l \times H_l \times W_l}$ as a feature at location h, w of the feature map. Both the 3D-FCAN and 2D-FCAN have similar structures, except for the operations in the convolution and mean-variance normalization layer. Two-stream 2D-CNN [3] uses 1 frame of RGB and 10 consecutive frames of flow-x, flow-y. In contrast, our 2D-FCAN uses only 1 frame of RGB and the corresponding frames of flow-x and flow-y since those flow frames at the same time instance will provide a more accurate attention map to the RGB data. Accordingly, our 2d-convolution layer to reduce a flow feature tensor $x\_flow^l \in \mathbb{R}^{C_l \times H_l \times W_l}$ to $x\_link^l \in \mathbb{R}^{1 \times H_l \times W_l}$ has the following form (counterpart to Equ. 4 in the main paper):

$$x\_link^l = W_{2D\_link} \circledast x\_flow^l. \tag{1}$$

where $\circledast$ denotes a 2d-convolution operation $1 \times 1 \times C_l$ along the channel C_l. In the training phase, we also initialize the filter weights W_{2D_link} to $\frac{1}{C_l}$.

Then, we normalize the feature tensor x_link^l by mean μ and variance σ of all spatial responses in x_link^l (counterpart to Equ. 5 in the main paper):

$$\hat{x}^l_{h,w} = \frac{x\_link^l_{h,w} - \mu}{\sigma}. \tag{2}$$

2. ConvNet architectures

For the 2D convolutional version, we choose CaffeNet [2, 1] architecture as the building blocks for our 2D-FCAN network. CaffeNet has 5 convolution and 3 fully connected layers. Let us denote $C(k, n, s)$ as a convolutional layer with kernel size $k \times k$, n filters and a stride of s, $P(k, s)$ a max pooling layer of kernel size $k \times k$ and stride s, N a normalization layer, RL a rectified linear unit, $FC(n)$ a fully connected layer with n filters and $D(r)$ a dropout layer with dropout ratio r. CaffeNet architecture is then given by: $C(11, 96, 4) - RL - P(3, 2) - N - C(5, 256, 1) - RL - P(3, 2) - N - C(3, 384, 1) - RL - C(3, 384, 1) - RL - C(3, 256, 1) - RL - P(3, 2) - FC(4096) - D(0.5) - FC(4096) - D(0.5) - FC(101)$. For the 3D version, we utilize the C3D [5] as the main component for our 3D-FCAN network. In C3D architecture, all convolution and pooling layers are made up of 3D operations [5]. C3D network has 8 convolution and 3 fully connected layers. C3D architecture is then given by: $C(3, 64, 1) - RL - P(2, 2) - C(3, 128, 1) - RL - P(2, 2) - C(3, 256, 1) - RL - C(3, 256, 1) - RL - P(2, 2) - C(3, 512, 1) - RL - C(3, 512, 1) - RL - P(2, 2) - C(3, 512, 1) - RL - C(3, 512, 1) - RL - P(2, 2) - FC(4096) - D(0.9) - FC(4096) - D(0.8) - FC(101)$.

	CaffeNet [1]	VGG16 [4]	C3D [5]
parameters	57,282,021	134,674,341	78,409,573

Table 1: Number of parameters of different convolution networks: CaffeNet, VGG16, and C3D.

Table 1 compares the complexity of the different CNN models. We observe that VGG16 has much more parameters than both C3D and CaffeNet. C3D has moderate complexities and good performance due to the explicit modeling of the temporal dimension.

References

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093*, jun 2014.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances*

in neural information processing systems, pages 1097–1105, 2012.

[3] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. *arXiv preprint arXiv:1406.2199*, pages 1–11, 2014.

[4] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recoginition. In *Intl. Conf. on Learning Representations (ICLR)*, pages 1–14, 2015.

[5] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *The IEEE International Conference on Computer Vision (ICCV)*, dec 2015.