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 \( x_{rgb}^l \in \mathbb{R}^{C_1 \times H_1 \times W_1} \) and \( x_{flow}^l \in \mathbb{R}^{C_1 \times H_1 \times W_1} \) be the feature map of a layer \( l \in \{0, 1, ..., L\} \) in spatial- and temporal-CNN (2D version) respectively, with \( C_1, H_1, W_1 \) being the number of channels, height and width of the feature map. We denote \( x_{rgb}^l_{h,w} \in \mathbb{R}^{C_1 \times H_1 \times W_1} \) and \( x_{flow}^l_{h,w} \in \mathbb{R}^{C_1 \times H_1 \times W_1} \) 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 uses only 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 \( x_{flow}^l \) to \( x_{link}^l \) has the following form (counterpart to Eq. 4 in the main paper):

\[
x_{link}^l = W_{2D_{link}} \odot x_{flow}^l.
\]

where \( \odot \) 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 \( \mu \) and variance \( \sigma \) of all spatial responses in \( x_{link}^l \) (counterpart to Eq. 5 in the main paper):

\[
x_{h,w}^l = \frac{x_{link}^l_{h,w} - \mu}{\sigma}.
\]

2. ConvNet architectures

For the 2D convolutional version, we choose CaffeNet 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 as the main component for our 3D-FCAN network. In 3D architecture, all convolution and pooling layers are made up of 3D operations. C3D network has 8 convolution and 3 fully connected layers. C3D architecture is then given by: \( C(3, 64, 1) \) – \( RL \) – \( P(2, 2) \) – \( C(3, 128, 1) \) – \( RL \) – \( P(2, 2) \) – \( C(3, 256, 1) \) – \( RL \) – \( C(3, 256, 1) \) – \( RL \) – \( P(2, 2) \) – \( C(3, 512, 1) \) – \( RL \) – \( C(3, 512, 1) \) – \( RL \) – \( P(2, 2) \) – \( C(3, 512, 1) \) – \( RL \) – \( C(3, 512, 1) \) – \( RL \) – \( P(2, 2) \) – \( FC(4096) \) – \( D(0.5) \) – \( FC(4096) \) – \( D(0.8) \) – \( FC(101) \).

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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>parameters</td>
<td>57,282,021</td>
<td>134,674,341</td>
<td>78,409,573</td>
</tr>
</tbody>
</table>

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

References


