Supplementary Material: Attentive Action and Context Factorization

Yang Wang[1]
wang33@cs.stonybrook.edu
Vinh Tran[14]
tquangvinh@cs.stonybrook.edu
Gedas Bertasius[2]
gberta@seas.upenn.edu
Lorenzo Torresani[3]
lt@dartmouth.edu
Minh Hoai[14]
minhhoai@cs.stonybrook.edu

[1] Stony Brook University
Stony Brook, NY 11794, USA
[2] University of Pennsylvania
Philadelphia, PA 19104, USA
[3] Dartmouth College
Hanover, NH 03755, USA
[4] VinAI Research
Hanoi, Vietnam

1 Spatial action and context attention maps

We have proposed a method to produce attention maps for spatiotemporal data. However, our method can also be used for spatial data. In this section, we perform experiments on still images, demonstrating the ability of our method for identifying action and context regions of an image. Specifically, we used the Pascal VOC2012 action dataset [2]. This dataset contains still images of 10 actions: Jumping, Phoning, Playing Instrument, Reading, Riding Bike, Riding Horse, Running, Taking Photo, Using Computer, and Walking. Some images in this dataset contain multiple people, and different people may perform different actions. Furthermore, a person may perform more than one action simultaneously, such as Walking and Phoning.

For this dataset, we consider an action sample to be the entire image without any bounding box information. We generate a conjugate sample for each image using the following steps. First, for the action being considered, we identify all the people performing this action in the image. This step will return a set of human bounding boxes if there are multiple people performing the same action. Second, we remove the pixels in identified bounding boxes of the previous step. Third, a pre-trained image completion network [3, 8] is applied to fill in the missing regions with alternative content. This network is composed of one generator for image completion and two discriminators for the local and global context respectively in order to determine that the generated image be completed consistently. Finally, we perform a post-processing step by blending [5, 7] the filled regions with the surrounding pixels. Some action samples and the corresponding conjugate samples generated by the network are shown in Fig. 1.

We use a pre-trained DenseNet-161 model for feature extraction. Each image is represented as a 3D features map $F \in \Re^{H \times W \times D}$. The sizes of images are different, so we resize the smaller dimension to 256 before feeding to the network. During training, we extract

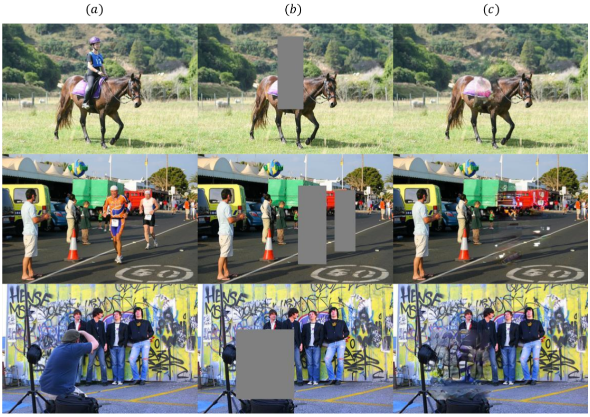


Figure 1: **Action and conjugate samples for Pascal VOC2012 dataset.** From top to bottom: *Riding Horse*, *Running* and *Taking Photo*. (a): action samples; (b) the corresponding bounding boxes for people performing the action in consideration; (c) conjugate samples obtained with image completion.

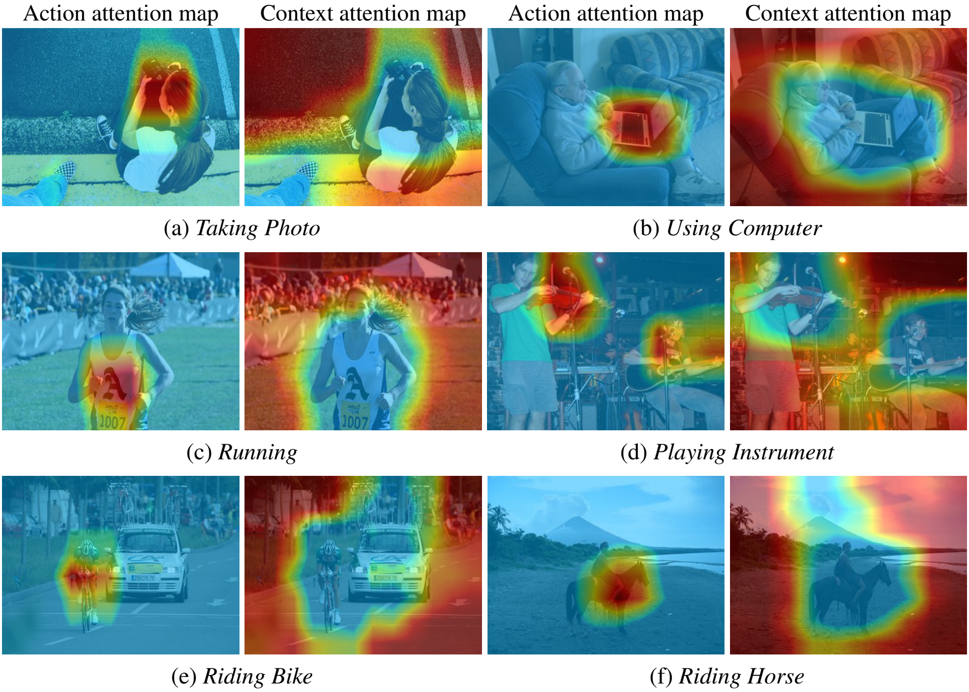
random crops of size 8 × 8 on the feature map and train the network with the mini-batch size of 32. At test time, the entire feature maps are fed into the network for attentive factorization and recognition.

The baseline model using DenseNet-161 features has a mean AP of 78.2% on the validation data. With conjugate samples, our attentive action and context factorization method achieves a mean AP of 80.2%, higher than the state-of-art performance of 75.2% mean AP [9]. Fig. 2 shows action and context attention maps on some action samples in the validation set.

2 Implementation Details

2.1 I3D Backbone

We use the I3D model [1] that was trained on both ImageNet [6] and Kinetics [4] datasets. We choose I3D ConvNet for feature extraction because it is the current state-of-the-art method for human action recognition. Specifically, we use the output of “Mixed_5c”, the last convolutional layer before global average pooling, as our convolutional feature map F. The effective accumulated convolutional stride at this layer are 32, 32, 8 for vertical, horizontal, temporal dimensions, respectively. We always resize the input video frames to have height $\mathcal{H} = 256$ and width $\mathcal{W} = 352$, thus the output feature map has height $H = \mathcal{H}/32 = 8$ and width $W = \mathcal{W}/32 = 11$. For an input video clip that spans $\mathcal{T} = 128$ frames, the output feature map would have the temporal length $T = \mathcal{T}/8 = 16$. We extract features using two-stream I3D Convnets that are trained on the RGB image sequences and the optical flow map sequences respectively.



(a) *Taking Photo*
(b) *Using Computer*
(c) *Running*
(d) *Playing Instrument*
(e) *Riding Bike*
(f) *Riding Horse*

Figure 2: **Examples of attention maps for action and context on Pascal VOC2012 dataset.** Similar to video dataset, the action maps put more attention on the human object interaction, and the context maps focus more on the the surrounding area.

2.2 Training Details on ActionThread

We use an adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) to train the proposed attentional factorization framework on ActionThread. The training procedure starts with a learning rate of 0.01 and stops after 60 epochs. The learning rate is reduced by a factor of 10 at epoch 20 and epoch 40. The weight decay is set to 0.0001.

2.3 Training Details on Hollywood2

We use an adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) to train the proposed attentional factorization framework on Hollywood2. The training procedure starts with a learning rate of 0.001 and stops after 30 epochs. The learning rate is reduced by a factor of 10 at epoch 10 and epoch 20. The weight decay is set to 0.0001.

2.4 Training Details on HACS-30

Most of the video clips provided in the HACS dataset are only two seconds long. To represent a video clip, we extract a sequence of 32 frames (i.e., 16 fps). These frames are resized such that the short side of them is 128 pixels and the aspect ratio is kept the same.

We use a mini-batch SGD with momentum (0.9) to train the backbone network (3D-Res34-RGB) and our proposed framework. The batch size is set to 64. During training, we freeze all the parameters within the BatchNorm layers of the backbone network. We also use a weighted cross entropy loss to combat class imbalance. Empirically we use weight 0.01 for negative clips and 1.0 for positive clips.

First, we need to finetune the backbone network on the HACS-30 dataset. In order to achieve this, we first only train the last fully-connected layer, and then optimize the full network end-to-end. For training only the last fully-connected layer, we use a learning rate of 0.001, and the training stops after 20 epochs. For optimizing the full network end-to-end, we use a learning rate of 0.0001, and the training stops after 15 epochs.

Subsequently, we train different attention frameworks, including our method, on top of the fine-tuned backbone network. The training procedure starts with a learning rate of 0.0001 and stops after 5 epochs. The learning rate is reduced by a factor of 10 at epoch 3.

3 More Visualization

We visualize more examples of action and context attention maps in Figure 3. The action attention maps have higher weights on humans, but not all humans receive the same attention, and not all parts of a human subject receive attention. The weights of the context maps are lower on the human subjects. The context maps have nonuniform distribution over background pixels.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017.

[2] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. www.pascal-network.org/challenges/VOC/voc2012/workshop/, 2012.

[3] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. volume 36, pages 1–14, 2017.

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. arXiv:1705.06950, 2017.

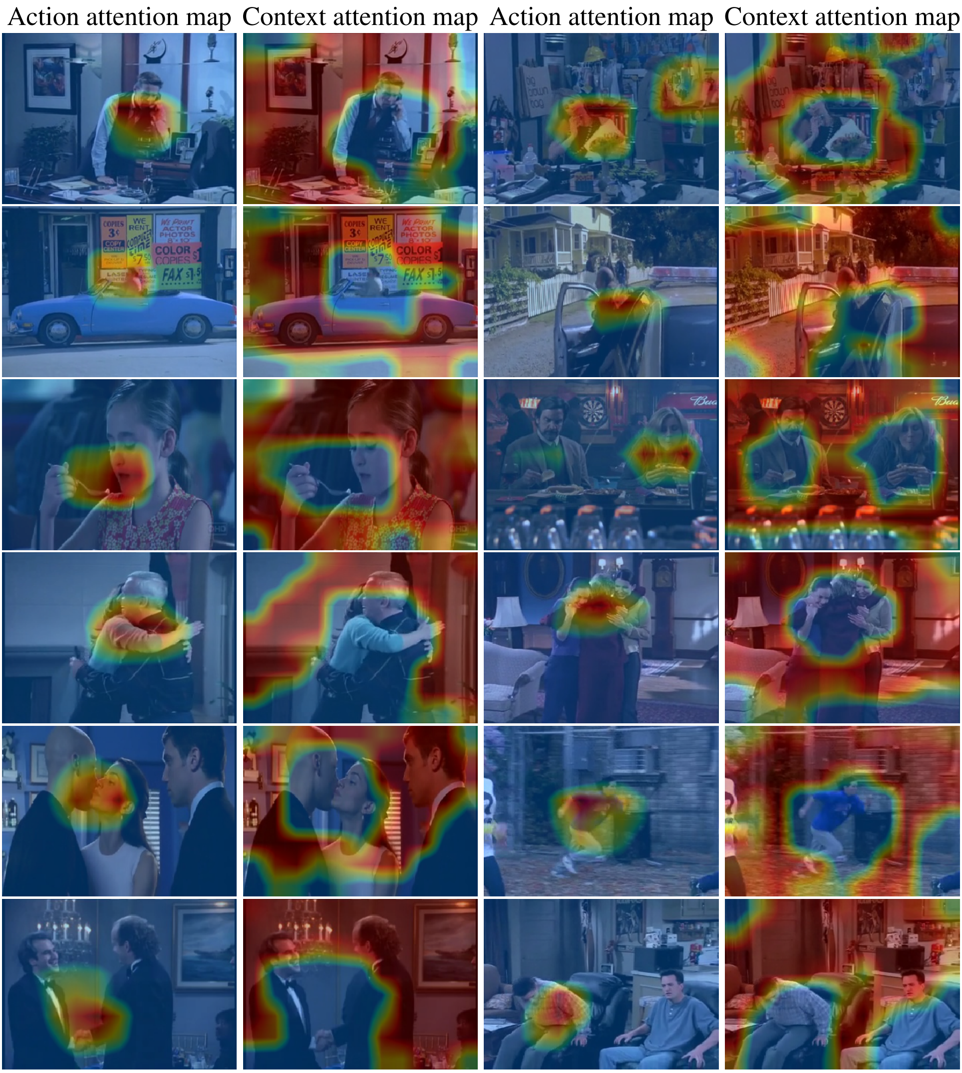
[5] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. volume 22, pages 313–318. ACM, 2003.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[7] Alexandru Telea. An image inpainting technique based on the fast marching method. volume 9, pages 23–34. Taylor & Francis, 2004.

[8] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *Proc. AAAI*, 2019.

[9] Yang Wang and Minh Hoai. Pulling actions out of context: Explicit separation for effective combination. In *Proc. CVPR*, 2018.



Action attention map Context attention map Action attention map Context attention map

Figure 3: **Examples of attention maps for action and context.** The action attention maps have higher weights on humans, but not all humans receive the same attention, and not all parts of a human subject receive attention. The weights of the context maps are lower on the human subjects. The context maps have nonuniform distribution over background pixels.