Supplementary material: *Human Activity Recognition with Pose-driven Attention to RGB*

Fabien Baradel
fabien.baradel@liris.cnrs.fr
Université Lyon, INSA Lyon, CNRS, LIRIS, F-69621, Villeurbanne, France

Christian Wolf
christian.wolf@liris.cnrs.fr
Université Lyon, INSA Lyon, LIRIS CITI Laboratory, INRIA, CNRS, Villeurbanne, France

Julien Mille
julien.mille@insa-cvl.fr
Laboratoire d'Informatique de l'Univ. de Tours, INSA Centre Val de Loire, 41034 Blois, France

We provide additional information about our work in this document. Visual explanations of our approach as well as visualizations of our attention processes at test time can be found in the attached **video**. For a better reproducibility of our model we provide details of our architecture, training steps and some implementation details in the following.

1 Network architectures and Training

Architectures — The pose network f_{sk} consists of 3 convolutional layers of respective sizes 8×3, 8×3, 5×75. Inputs are of size $20\times300\times3$ and feature maps are, respectively, 10×150, 5×75 and $1\times1\times1024$. Max pooling is employed after each convolutional layer, activations are ReLU. The glimpse sensor f_g is implemented as an Inception V3 network [2]. Each vector $v_{t,:,i}$ corresponds to the last layer before output and is of size 2048. The LSTM network f_h has a single recurrent layer with 1024 units. The spatial attention network f_p is an MLP with a single hidden layer of 256 units and sigmoid activation. The temporal attention network f'_p is an MLP with a single hidden layer of 512 units and sigmoid activation. The feature extractor f_u is a single linear layer with ReLU activation. The output layers of both stream representations are linear layers followed by softmax activation. The full model (without glimpse sensor f_g) has 38 millions trainable parameters.

Training — All classification outputs are softmax activated and trained with cross-entropy loss. The glimpse sensor f_g is trained on the ILSVRC 2012 data [7]. The pose learner is trained discriminatively with an additional linear+softmax layer to predict action classes. The RGB stream model is trained with pose parameters θ_{sk} and glimpse parameters θ_g frozen.

Implementation details — Following [3], we cut videos into sub sequences of 20 frames and sample sub-sequences. During training a single sub-sequence is sampled, during testing we sample 10 sub-sequences and average the logits. We apply a normalization step

on the joint coordinates by translating them to a body centered coordinate system with the "middle of the spine" joint as the origin. If only one subject is present in a frame, we set the coordinates of the second subject to zero. We crop sub images of static size on the positions of the hand joints (50×50 for NTU, 100×100 for SBU and MSR). Cropped images are then resized to 299×299 and fed into the Inception model.

Training is done using the Adam Optimizer [1] with an initial learning rate of 0.0001. We use minibatches of size 64 and dropout with a probability of 0.5. Following [3], we sample 5% of the initial training set as a validation set, which is used for hyper-parameter optimization and for early stopping. All hyper-parameters have been optimized on the validation sets of the respective datasets. When transferring knowledge from NTU to SBU, the target networks were initialized with models pre-trained on NTU. Skeleton definitions are different and were adapted. All layers were finetuned on the smaller datasets with an initial learning rate 10 times smaller then the learning rate for pre-training.

Runtime — For a sub-squence of 20 frames, we get the following runtimes for a single Titan-X (Maxwell) GPU and an i7-5930 CPU: A full prediction from features takes 1.4ms including pose feature extraction. This does not include RGB pre-processing, which takes additional 1sec (loading Full-HD video, cropping sub-windows and extracting Inception features). Classification can thus be done close to real-time. Fully training one model (w/o Inception) takes $\sim$4h on a Titan-X GPU. Hyper-parameters have been optimized on a computing cluster with 12 Titan-X GPUs. The proposed model has been implemented in Tensorflow.

References

[1] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Machine Learning (ICML)*, 2015.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[3] A. Shahroudy, J Liu, T.-T. Ng, and G. Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016.

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.