Multi-view Action Recognition using Cross-view Video Prediction (Supplementary)

Shruti Vyas, Yogesh S Rawat, and Mubarak Shah

Center for Research in Computer Vision, University of Central Florida, USA
{shruti,yogesh,shah}@crcv.ucf.edu

This document provides additional details supplementary to the main paper. It covers network architecture details, experimental details, and additional qualitative results.

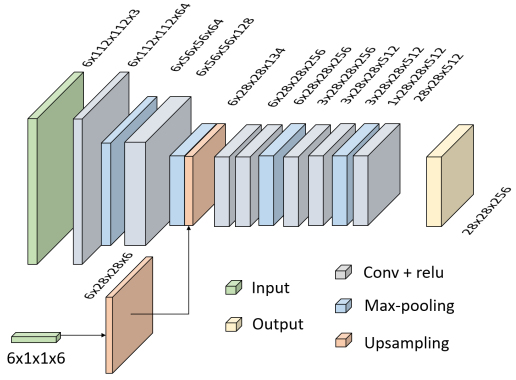
1 Model Details

The proposed approach consists of two main components: Representation Learning network (RL-NET) and Video Rendering Network (VR-NET) (shown in main manuscript - Figure 1). RL-NET is used to learn a representation r given some observations o_i. It consists of a base network (ENC-NET), which is shared among all the observations to encode individual observations, and a Blending Network (BL-NET), which integrates encodings from all the observations to learn a unified representation r. The representation r is then passed to the VR-NET which renders a video from arbitrary view and time. The multi-tasking architecture, which also performs action recognition, consists of another branch called Classification Network (CL-NET). It takes the learned representation r as input and predicts confidence scores for the activity classes. We will discuss the architecture details for all these networks in the following subsections.

1.1 Representation Learning Network (RL-NET)

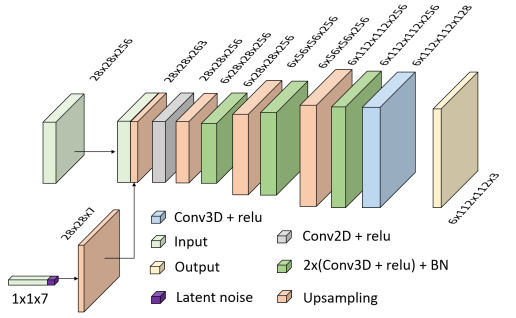
The overview of the RL-NET is shown in Figure 2.C (main manuscript). ENC-NET is the encoding part of RL-NET which is shared among all the observations and the corresponding encodings are passed to the BL-NET for representation learning. A detailed architecture for this part of the network is shown in Figure 1. The ENC-NET for this architecture utilizes 3D convolution in combination with max-pooling to get video level encodings. Each observation is a set of multiple video clips from different viewpoints. We also tested our representation learning network for a single view dataset using a similar architecture. However, for single view each observation was a set of multiple input clips from the same video/viewpoint.

The encodings learned by the base network are passed on to the blending network (BL-NET), which is a recurrent network with bi-directional convolutional LSTM cells. The architecture of BL-NET is shown in Figure 2.E (main manuscript). The number of recurrent steps in the network will be equal to the



Type	Kernel/ shape	Input	Output
Input	6x112x112x3	Video	x
Input	6x1x1x6	View	v
Conv + relu	3x3x3	x	x
Max-pooling	1x2x2	x	x
Conv + relu	3x3x3	x	x
Max-pooling	1x2x2	x	x
Upsampling	1x28x28	v	v
Concat	-	x,v	x
2x(Conv + relu)	3x3x3	x	x
Max-pooling	2x1x1	x	x
2x(Conv + relu)	3x3x3	x	x
Max-pooling	3x1x1	x	x
2x(Conv + relu)	3x3	x	e

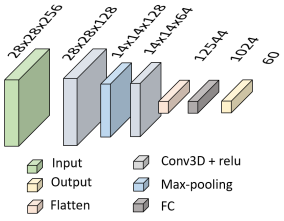
Fig. 1. Network architecture of ENC-NET used in RL-NET for representation learning r for set of video observations o_i. The diagram on the left side shows the change in feature volumes, along with size, as the input to the network flows through the layers of the network. The network takes two input, video clips and corresponding viewpoints, and generates a video. The table on the right shows details of the layers in the network including kernel sizes and layer types. Conv: 3D convolution in all the layers except the last two layers where it will be 2D convolution as the temporal extent has been compressed to a single channel, v: viewpoint, e: embeddings for observations.



Type	Kernel/ shape	Input	Output
Input	28x28x256	r	r
Input	1x1x7	view + z	v-z
Upsampling	28x28	v-z	x
Concat	-	r,x	x
Conv2D + relu	3x3	x	x
Upsampling	6x1x1	x	x
2x(Conv3D + relu) + BN	3x3x3	x	x
Upsampling	1x2x2	v	v
2x(Conv3D + relu) + BN	3x3x3	x	x
Upsampling	1x2x2	x	x
2x(Conv3D + relu) + BN	3x3x3	x	x
Conv2D + relu	3x3x3	x	x
Conv2D + relu	1x1x1	x	i

Fig. 2. Network architecture of VR-NET used for video rendering. VR-NET takes the learned representation r along with query viewpoint v and latent noise z as input and generates a video. The representation r, viewpoint v and latent noise z are first integrated together followed by a convolution operation. A video is generated using 3D convolution along with batch-normalization and upsampling. BN: batch normalization, z: latent noise, i: generated video.

number of observations (o_i) provided to the RL-NET. The first layer of BL-NET comprises of 128 3x3 kernels in each direction with a total of 256 kernels for the convolution operation. The final layer has 256 3x3 kernels which produce a representation of size 28x28x256.



Type	Kernel/ shape	Input	Output
Input	28x28x256	r	x
Conv + relu	3x3	x	x
Maxpooling	2x2	x	x
Conv + relu	3x3	x	x
Flatten	12544	x	x
FC	1024	x	x
FC	60	x	c

Fig. 3. Network architecture of the classification branch (CL-NET) of the multi-task approach. This network takes the learned representation r as input and predicts class probabilities. FC: fully connected layer, c: predicted class probabilities.

1.2 Video Rendering Network (VR-NET)

The video rendering network takes the learned representation r along with query viewpoint v and latent noise z to synthesize a video clip. A detailed architecture of the proposed VR-NET is shown in Figure 2. It first integrates the representation r with viewpoint v and latent noise z followed by a convolution operation. A video is generated using convolutions along with upsampling of features. Similar network architecture is used for the experiments where clips are generated using the learned representation from a single view. It uses 3D convolutions with temporal information and latent noise i.e. no viewpoint information.

1.3 Classification Network (CL-NET)

We propose to use the learned representation for activity classification. The network is trained for both video rendering as well as activity classification simultaneously using multi-task learning. We added a classification branch (CL-NET) on top of representation which predicts class probabilities for activities. A detailed architecture for this branch is shown in Figure 3. It consists of 2D convolutions followed by fully connected layers for class predictions.

2 Experimental Details

An outline of the proposed framework is shown in Figure 4 where we see two training approaches discussed in the main manuscript (Section 3.3); a generalized framework for the first approach for unsupervised representation learning by rendering a query view is shown in Figure 4(a). This involved pretraining without a classifier and classifier was added later on for training and the second approach is shown in Figure 4(b) where we trained our network for action recognition task along with video rendering. We observed similar performance in both approaches and followed the second approach for training all our networks later on.

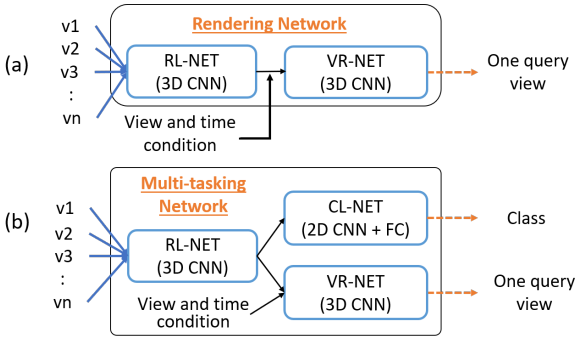


Fig. 4. (a) Generalized network architecture for representation learning using cross-view video rendering using RL-NET and VR-NET. (b) Multitasking network for classification along with video rendering.

2.1 Action Recognition

We perform our action recognition experiments on two different datasets, NTU-RGB+D and N-UCLA. Both datasets have action sequences captured from 3 different views. In the cross-subject training, we randomly choose two of the views as input to learn the representation and render the left out view. In cross-view experiments, there are only two views available for training and therefore we randomly choose one for representation learning and the other for rendering. The proposed network allows us to make inference using a different number of clips and views irrespective of how it was trained. This is useful for cross-subject evaluation when we train our network using two different input views and test on a single view as is reported by other studies.

Hyperparameters We implement the proposed method on Keras with Tensorflow backend. We train all our networks using Adam optimizer [5] with a learning rate of 2e-5, $\beta 1$=0.9, $\beta 2$=0.999, and a decay of 1e-6. The networks were trained until convergence of loss. The batch normalization layers use a momentum of 0.9 with ϵ=1e-3 and no scaling. The CL-NET branch has dropouts after every fully connected layer where we use a dropout rate of 0.5. We use a skip rate of 3 frames for all our action recognition experiments.

Viewpoint and Time The NTU-RGB+D dataset was captured from three camera positions. The action videos were recorded in two conditions: (i) when the actor is facing CAM2, and (ii) when the actor is facing CAM3. The viewpoint involved 5 parameters, including camera height, camera distance, camera orientation, horizontal-pan, and, vertical-pan. Horizontal and vertical pan in our training set was determined based on the cropping of the input frames which was done randomly. We take a random 112x112 crop from the resized video frames before feeding to the network. The camera orientation was encoded based on its location

around the actor and we use angular position in a range $(-\pi/2, \pi/2)$. The other viewpoint parameters were normalized between 0-1. In N-UCLA dataset, the range of camera orientation was between -$2\pi/3, 2\pi/3$ and camera height and distance was set as 1 and 1.7 for all the samples. The horizontal and vertical pan were estimated based on the random frame crop similar to NTU-RGB+D dataset. Time conditioning was encoded as a single dimensional feature based on the frame position within the long action sequence. The position of the frame was normalized between 0-1 using the total length of the action sequence.

3 Additional Results

Quantitative Comparison We compare the performance of our proposed method with the recent works on view-invariant action recognition (Table 1).

Table 1. A comparison of cross-subject (CS) and cross-view (CV) action recognition on NTU-RGB+D dataset.

Method	**Modality**	**CS**	**CV**
Shahroudy et al.[21]	Skeleton	62.9	70.3
Depth+Skeleton[17]	Skeleton	75.2	83.1
VA-LSTM [31]	Skeleton	79.4	87.6
GCA-LSTM [11]	Skeleton	76.1	84.0
IndRNN [10]	Skeleton	81.8	87.9
STA-LSTM [23]	Skeleton	73.4	81.2
Att-LSTM [33]	Skeleton	80.7	88.4
DPRL+GCNN [24]	Skeleton	83.5	89.8
ST-GCN [29]	Skeleton	81.5	88.3
VA-RNN [32]	Skeleton	79.4	87.6
CNN-BiLSTM [8]	Flow	80.9	83.4
STA-Hands [1]	RGB-S	73.5	80.2
Pose Est. [13]	RGB-S	84.6	-
DSSCA - SSLM[22]	RGB-DS	74.9	-
HOG [15]	Depth	32.2	22.3
S-Norm Vector [30]	Depth	31.8	13.6
HON4D [16]	Depth	30.6	7.3
Shuffle & learn [14]	Depth	61.4	53.2
CNN-LSTM [12]	Depth	66.2	53.2
CNN-BiLSTM [8]	Depth	68.1	63.9
Proposed	Depth	**71.8**	**78.7**
Proposed (all-views)	Depth	79.4	-
CNN-LSTM [12]	RGB	56	-
DA-NET[25]	RGB	-	75.3
Att-LSTM [33]	RGB	63.3	70.6
CNN-BiLSTM [8]	RGB	55.5	49.3
Proposed	RGB	**82.3**	**86.3**
Proposed-(all-views)	RGB	88.9	-

The table we show here in supplementary is more comprehensive than the main manuscript and besides RGB shows comparison with different modalities. Our model performs well in both CS and CV evaluation for RGB. We also observe good results when generalized to depth modality. We observe that the proposed method provides significant improvement in CV evaluation for both RGB ($\sim$ 11%) and depth ($\sim$ 15%) modality which demonstrates that the learned representation is robust to viewpoint change. Even though we do not use skeleton data the performance of our method is comparable to the state-of-the-art approaches employing 3D skeleton. It is important to note that 3D skeleton is more robust to view-point changes as compare to RGB and depth. However, additional sensors are required to capture this skeleton data and extracting 3D pose from RGB is a challenging problem.

Table 2. A comparison of cross-subject (CS) and cross-view (CV) action recognition on N-UCLA MultiviewAction3D dataset.

Method	**Modality**	**CS**	**CV**
Actionlet [26]	Skeleton	-	69.9
HBRNN-L [3]	Skeleton	-	83.5
DLVIF [6]	Skeleton	81.1	77.2
R-NKTM [20]	Skeleton	-	78.1
Att-LSTM [33]	Skeleton	-	90.7
VA-Fusion [32]	Skeleton	-	88.7
MST-AOG [27]	Depth	-	53.6
HOPC [18]	Depth	-	71.9
CNN-LSTM [12]	Depth	-	50.7
CNN-BiLSTM [8]	Depth	-	62.5
NKTM [19]	RGB-S	-	75.8
MST-AOG [27]	RGB-S	81.6	73.3
Hanklets [7]	RGB	54.2	45.2
DV-Views[9]	RGB	50.7	58.5
LRCN [2]	RGB	-	64.7
Proposed-scratch	RGB	35.1	43.4
Proposed	RGB	**87.5**	**83.1**

Transfer Learning for Action Recognition We use the representation learned on NTU-RGB+D dataset for N-UCLA dataset which has a different set of scenes and users. We perform both CV as well as CS evaluation with two different variations. The evaluation of the proposed method after transfer learning from NTU-RGB+D dataset on N-UCLA dataset is shown in Table 2. Here we present a more comprehensive comparison with state-of-the-art where besides RGB we show comparison with other modalities as well. We observe that the network performs poorly when train from scratch which is due to the small size of this dataset. However, using the learned representations from NTU-RGB+D significantly increases the performance. Our method outperforms previous RGB/depth

based approaches [12, 8] and achieves performance comparable to most of the skeleton-based methods.

Qualitative Results Here we are showing some qualitative results for video generation on NTU-RGB+D dataset. The ground truth and predicted video frames from the three networks trained on NTU-RGB+D dataset are shown in Figure 5 and 6.

Fig. 5. Video frames generated for arbitrarily selected query time and view on NTU-RGB+D dataset. The network is given video clips from three views at a randomly selected time. This network was trained on input video clips from all the three views and a video was generated from any one of these views at an arbitrary time. Row-1: Stomachache/heart pain, Row-2: Eat meal/snack, Row-3: Wear on glasses, and Row-4: Touch other person's pocket.

To validate the effectiveness of cross-view prediction we compute the Peak Signal to Noise Ratio (PSNR) [4] and the Structural Similarity Index Measure (SSIM) [28] of the synthesized video frames. The evaluation is shown in Table 3 where we compute the scores for first and last predicted frames along with the full video. We observe that the quality of the synthesized frames remains similar from first to the last frame in the generated video. Also, when comparing the three training scenarios the network shows better quality with three input views (Figure 6).

The RL-NET performs representation learning based on a set of input observations. These observations can be from different viewpoints (varying views) and different time in a video (Figure 7). The visual appearance of any activity changes with viewpoint as well as time. This analogy between time and viewpoint for variation in visual appearance of any activity allows us to use the two concepts interchangeably during representation learning. This idea makes the proposed architecture even more powerful as a network trained under some settings can be tested on different set of parameters in terms of number of views and number of clips. The proposed BL-NET supports this further due to its recurrent structure which allows it to learn a representation for different varying

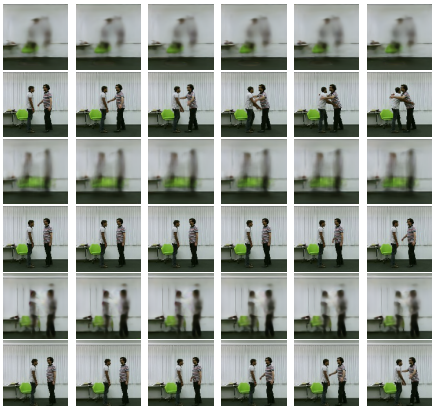


Fig. 6. Generated (row 1, 3, and 5) and corresponding ground truth (row 2, 4, and 6) video frames (action class- 55, hugging other person from NTU-RGB+D dataset) at different time-steps generated using models trained under different configurations. Row-1-2: Given two random views generates third unseen view (M-1), Row-3-4: Given view 2 and 3 generates unseen view 1 (M-2), and Row-5-6: Given all three views generates one given view at an arbitrary query time (M-3).

Table 3. Quantitative evaluation of cross-view video prediction on NTU-RGB+D cross-subject test split. FF: first frame, LF: last frame, and FV: full video.

	FF		LF		FV	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
M-1	19.9	0.68	19.8	0.68	19.8	0.68
M-2	19.9	0.69	19.9	0.69	19.8	0.68
M-3	23.3	0.79	23.2	0.79	23.3	0.79

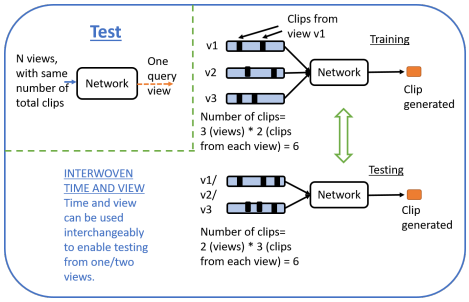


Fig. 7. Interwoven time and view dimensions: time and view dimensions can be used interchangeably while making predictions with our network.

observations. Hence, our proposed approach can be generalized to single view datasets as well.

References

1. Fabien Baradel, Christian Wolf, and Julien Mille. Human action recognition: Pose-based attention draws focus to hands. In *The IEEE ICCV Workshops*, Oct 2017.
2. Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE conference on CVPR*, 2015.
3. Yong Du, Yun Fu, and Liang Wang. Representation learning of temporal dynamics for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 25(7):3010–3022, 2016.
4. Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE, 2010.
5. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
6. Yu Kong, Zhengming Ding, Jun Li, and Yun Fu. Deeply learned view-invariant features for cross-view action recognition. *IEEE Transactions on Image Processing*, 2017.
7. Binlong Li, Octavia I Camps, and Mario Sznaier. Cross-view activity recognition using hankelets. In *IEEE CVPR*, 2012.
8. Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Unsupervised learning of view-invariant action representations. In *Advances in Neural Information Processing Systems*, 2018.
9. Ruonan Li and Todd Zickler. Discriminative virtual views for cross-view action recognition. In *IEEE CVPR*, 2012.
10. Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5457–5466, 2018.
11. Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2018.
12. Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *IEEE Conference on CVPR*, 2017.
13. Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *IEEE Conference on CVPR*, 2018.
14. Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*. Springer, 2016.
15. Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and hog2 for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 465–470, 2013.
16. Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 716–723, 2013.
17. Hossein Rahmani and Mohammed Bennamoun. Learning action recognition model from depth and skeleton videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

18. Hossein Rahmani, Arif Mahmood, Du Huynh, and Ajmal Mian. Histogram of oriented principal components for cross-view action recognition. *IEEE transactions on PAMI*, 2016.
19. Hossein Rahmani and Ajmal Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *Proceedings of the IEEE conference on CVPR*, 2015.
20. Hossein Rahmani, Ajmal Mian, and Mubarak Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE transactions on PAMI*, 2018.
21. Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on CVPR*, 2016.
22. Amir Shahroudy, Tian-Tsong Ng, Yihong Gong, and Gang Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Transactions on PAMI*, 2018.
23. Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI conference on AI*, 2017.
24. Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *IEEE Conference on CVPR*, 2018.
25. Dongang Wang, Wanli Ouyang, Wen Li, and Dong Xu. Dividing and aggregating network for multi-view action recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018.
26. Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):914–927, 2014.
27. Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *IEEE Conference on CVPR*, 2014.
28. Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
29. Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018.
30. Xiaodong Yang and YingLi Tian. Super normal vector for activity recognition using depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 804–811, 2014.
31. Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017.
32. Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE PAMI*, 2019.
33. Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding attentiveness to the neurons in recurrent neural networks. In *European Conference on Computer Vision*, 2018.