Temporal Reasoning in Videos using Convolutional Gated Recurrent Units

Debidatta Dwibedi* Pierre Sermanet Jonathan Tompson
Google Brain
{debidatta, sermanet, tompson}@google.com

Abstract

Recently, deep learning based models have pushed state-of-the-art performance for the task of action recognition in videos. Yet, for many action recognition datasets like Kinetics and UCF101, the correct temporal order of frames doesn't seem to be essential to solving the task. We find that the temporal order matters more for the recently introduced 20BN Something-Something dataset where the task of fine-grained action recognition necessitates the model to do temporal reasoning. We show that when temporal order matters, recurrent models can provide a significant boost in performance. Using qualitative methods, we show that when the task of action recognition requires temporal reasoning, the hidden states of the recurrent units encode meaningful state transitions.

1. Introduction

Understanding videos remains one of the biggest challenges in computer vision. While Convolutional Neural Networks (CNN) seem to be the standard building block for processing static images, it is still not clear what the architectural counterpart for processing videos should be. Researchers have come up with many ideas to incorporate temporal information in their models. Today we have a plethora of networks that incorporate Recurrent Neural Networks[5] (RNN), ConvolutionalRNNs[19], 3D/Temporal Convolutions[3, 32] and attention[9, 25, 34] to aggregate temporal information across frames in a video. Even with these architectural advances, the simple baseline of temporal averaging of features obtained by passing single frames through pre-trained networks is still a formidable baseline for action recognition across many datasets[8, 27, 34, 35].

The other big question in video understanding is what is the right data to be working with? For models to *truly* understand what is going on in a video, they need to reason about foreground, background, camera motion, human pose, and context among other concepts. However, it is easier for models to get good performance on the task of action recognition by identifying discriminative features in certain frames of a full video rather than performing all the tasks that are crucial for video understanding. For example, it is easier for models to recognize tennis courts rather track the player's motions to figure out they are playing tennis in a video. Recent work has shown that models can give good performance even when the frames of the video are shuffled or reversed[35, 37]. However, time plays a crucial role in human understanding of everyday actions. One attempt to incorporate more temporal aspects in action recognition tasks is the recently released *20BN Something Something*[10] dataset. It introduces the task of recognizing fine-grained object-agnostic human-object interactions. Models need to perform temporal reasoning to differentiate between closely related actions like opening and closing. As we begin to look at videos and tasks where models need to reason temporally to solve the task, more interesting challenges will arise.

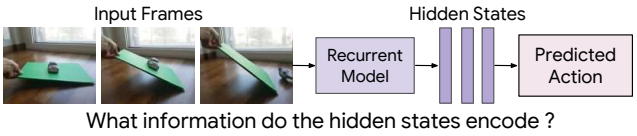

Figure 1: Recurrent models are designed to handle sequential data making them suitable to consume videos as input. In this work, we investigate the information encoded in the hidden states of recurrent units for the task of action recognition in videos.

Videos of humans interacting with objects are of special interest to the robotics community as this opens up the possibility of skill transfer to robots by visual observation. If we expect robots to assist humans in their homes with their daily chores, they need to understand the rich stream of information coming through the sensors mounted on them. One vital component that still needs more work is the ability to understand human-object interactions from videos. In current datasets of action recognition, when humans interact with objects there is typically a single action associated strongly with an object category. For example, if there is a bicycle in the scene there is a strong correlation with the *biking* class. However, in a household setting it is not difficult to imagine scenarios where many actions are performed with the same object. For example, a person might *open*,

*Google AI Resident (http://g.co/airesidency)

close, *lift* or *throw* a common household object like a bag. This makes the task of fine-grained human-object interaction with household objects interesting and challenging.

Our contribution is three-fold: (i) we experiment with recurrent architectures for video understanding on multiple datasets (ii) we achieve competitive performance in fine-grained action recognition on the 20BN Something-Something dataset[10] (iii) we qualitatively analyze the hidden states of the recurrent units and find that when the task of action recognition requires temporal reasoning, the hidden states of the recurrent unit tend to capture meaningful state transitions.

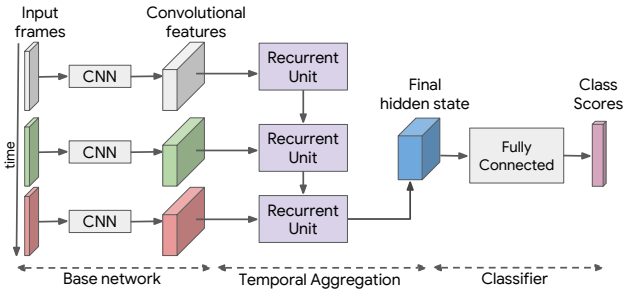

Figure 2: Recurrent architecture used for action recognition.

2. Related Work

Understanding videos is an active research topic with a lot of open questions. Recognizing actions[11, 13, 15, 28] has been a core problem in this area for many years[7, 12, 16–18, 23]. Prior work has investigated a wide variety of both model architectures and train-data source. Researchers have trained models that process videos on raw images, dynamic images[2] and optical flow maps[3, 27] as input. Presently, the best state-of-the art models for action recognition on both UCF1-101, HMDB51 and Charades datasets[26] are built on top of the I3D[3] model pre-trained on the Kinetics[13] dataset. The I3D model uses layers of 3D convolutions which greatly increase the number of free parameters, increasing the need for data to prevent overfitting. Mitigating this limitation, the Kinetics dataset has been shown to be a good dataset for this purpose and the best models for both UCF-101 and HMDB-51 have been pre-trained on the Kinetics.

The use of recurrent architectures for video-based tasks has been a much explored topic in recent years[3, 6, 19–22]. In spite of RNNs being specifically designed to handle sequential data, the authors of [3] found non-recurrent models to be better at action recognition than their recurrent counterparts. This oddity might be attributed to the fact that the task of recognizing actions in most large-scale datasets is not one that is sequential in nature and other non-recurrent models have been better at finding discriminative features over many frames. On the other hand, RNNs have had success in tasks like video object segmentation[31] and video object tracking[36]. We take this opportunity to study the use of recurrent architectures on a video dataset where understanding the sequential nature of the data is essential to solving the task.

3. Architecture for Action Recognition

We present our architecture in Figure 2. We use an Image-Net[24] pre-trained convolutional network to extract low-dimension image features from raw video frames, which we use as the "base" input to the subsequent recurrent module. The base network's outputs from each frame are stacked in a new dimension for time. This is provided as input to a sequential model. At this point, we can use 3D Convnets[3], RNNs, ConvoultionalRNNs or simple spatio-temporal averaging as the choice for sequential model. We choose Gated Recurrent Units [4] as our choice of a recurrent module. We test with both the standard GRU and its convolutional version which is similar to the ConvolutionalLSTM presented in [19]. We use dropout[29] on the outputs of both variants of the GRU at each time-step. For the standard GRU model, we perform spatial averaging before passing it to the GRU. For the ConvGRU model, we perform spatial averaging of the hidden state of the ConvGRU. We take the final hidden state of the recurrent module and pass it through a fully connected layer to predict the scores for each class. The models are trained with the standard cross-entropy loss.

4. Experiments

4.1. Datasets

We evaluate the performance of the proposed architecture on the 20BN-Something-Something dataset[10]. Currently, it is the largest video dataset focused on human-object interactions. The dataset has $108,499$ videos with 174 categories of fine-grained human-object interactions. The training, validation and test sets consists of $86,017$, $11,522$ and $10,960$ videos respectively. The average duration of the videos is 4.0 seconds. The short-duration of the input video should result in actions that can be considered "atomic"; they are actions over a small time scale, such as lifting or pushing an object, rather than long term tasks such as cooking, playing an instrument, etc. In addition, the dataset includes difficult to infer fine-grained task definitions, the semantic contents of which are similar despite having different classification labels. Examples of this include: *Pouring something into something*, *Pouring something into something until it overflows* and *Trying to pour something into something, but missing so it spills next to it*. Another interesting aspect of the dataset is the existence of pretend categories that act as hard examples for some classes. This property might prevent models from simply picking up on some secondary cues to recognize a class. We also experiment the effectiveness of our architecture on

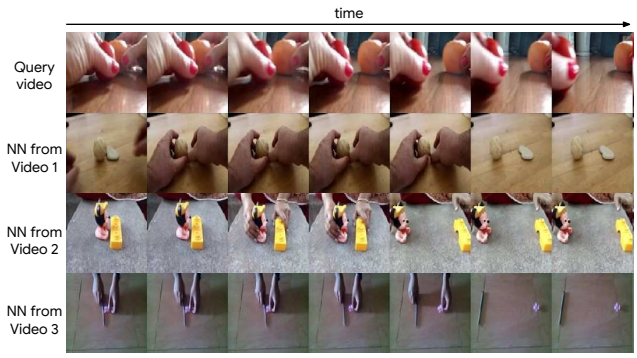

Figure 3: We show examples of retrieved nearest neighbors (NN) for each frame in a query video from the class *"Moving something and something away from each other"*. We observe that the nearest neighbors are temporally coherent and indicative of the transitions expected in a given task. More details in Section 4.5.

the Kinetics [13] dataset which is a large-scale dataset with clips collected from YouTube. There are more than 200K and 10K videos in its train and validation set. There are 400 action classes in Kinetics.

4.2. Evaluation Metrics

To enable effective evaluation, we follow the methodology suggested in [10] and report top-k accuracy for $k = 1, 2, 5$ on all the classes in each dataset. For Kinetics, we report top-k accuracy for $k = 1, 5$ as is common practice[3, 33, 35].

4.3. Training Details

We implemented our model in TensorFlow[1]. We use ImageNet pre-trained InceptionV3[30] as the base network and pool features from *Mixed_7c* layer. While InceptionV3 has been trained to perform classification on static images, it never-the-less encodes useful visual representations. In order to train with a larger batch size, we sample every second for the Something-Something and every fifth frame for Kinetics dataset. For Something-Something we fine-tune only the last convolutional block while for Kinetics we fine-tune the last two convolutional blocks. We train the models using Adam[14] with learning rate of 1.0×10^{-4}. We use a dropout of 0.5 on the GRU outputs at each time-step.

4.4. Architectural Experiments

We implement a simple baseline of spatio-temporal averaging of features and observe it is able to achieve 20.5% accuracy. We find the ConvGRU model performs the best on the validation set on the 20BN Something-Something dataset. We also find our single-scale recurrent models (both GRU and ConvGRU) outperform multi-scale Temporal Relation Network (TRN)[37]. Our model has an accuracy of 39.6% on the test set outperforming multi-scale TRN[37] model. At the time of submission, the best accuracy on the leaderboard is 45.0% but it is unclear what is the model or data used for that entry.

We use the same models on Kinetics. However, the GRU models are slightly worse than the spatio-temporal averaging baseline. This is probably due to the fact that the task of action recognition on Kinetics videos, although difficult in its own right, is not one that involves temporal reasoning as models can get competitive scores with shuffled and reversed frames[35]. We still use GRUs on Kinetics to compare the nature of hidden states of the GRU across these two datasets.

Table 1: We test the impact of various recurrent modules on validation set performance on the 20BN something Something dataset

Model	Accuracy@1	Accuracy@2	Accuracy@5
Single-scale TRN[37]	31.0	-	59.2
Multi-scale TRN[37]	33.0	-	61.3
Spatio-temporal averaging	20.5	32.4	48.2
GRU	35.4	48.1	63.3
ConvGRU	**43.7**	**57.0**	**71.4**

Table 2: Results on the 20BN Something-Something test set

Model	Accuracy@1
3D CNN[32] + Temporal Averaging[10]	11.5
MultiScale TRN[37]	33.6
ConvGRU	**39.6**

Table 3: We test the impact of various recurrent modules on validation set performance on the Kinetics dataset

Model	Pre-trainig Dataset	Accuracy@1	Accuracy@5
I3D-RGB[3]	ImageNet	72.1	90.3
R(2+1)D-RGB[33]	Sports-1M	74.3	91.4
S3D-G[35]	ImageNet	**74.8**	**91.9**
Spatio-temporal averaging	ImageNet	**71.5**	**89.5**
GRU	ImageNet	70.6	88.4
ConvGRU	ImageNet	70.0	88.1

4.5. Qualitative Analysis

In order to visualize the information encoded by the hidden layer of the GRU, we perform the following experiment on the *validation* set. We consider all the videos in a chosen predicted class. For a selected video, there are many frames. For each of these frames we find the nearest neighbor frame in other videos of the same class. The nearest neighbor is found in terms of the hidden state of the GRU (*not* in the pixel space). In Figure 3, we show an example of our nearest neighbor matching process. For each frame in a given query video (row 1), we retrieve the nearest neighbor frame from other videos of the same class. Even though the frames are visually different and have different objects, they are close to each other in the hidden state of the GRU.

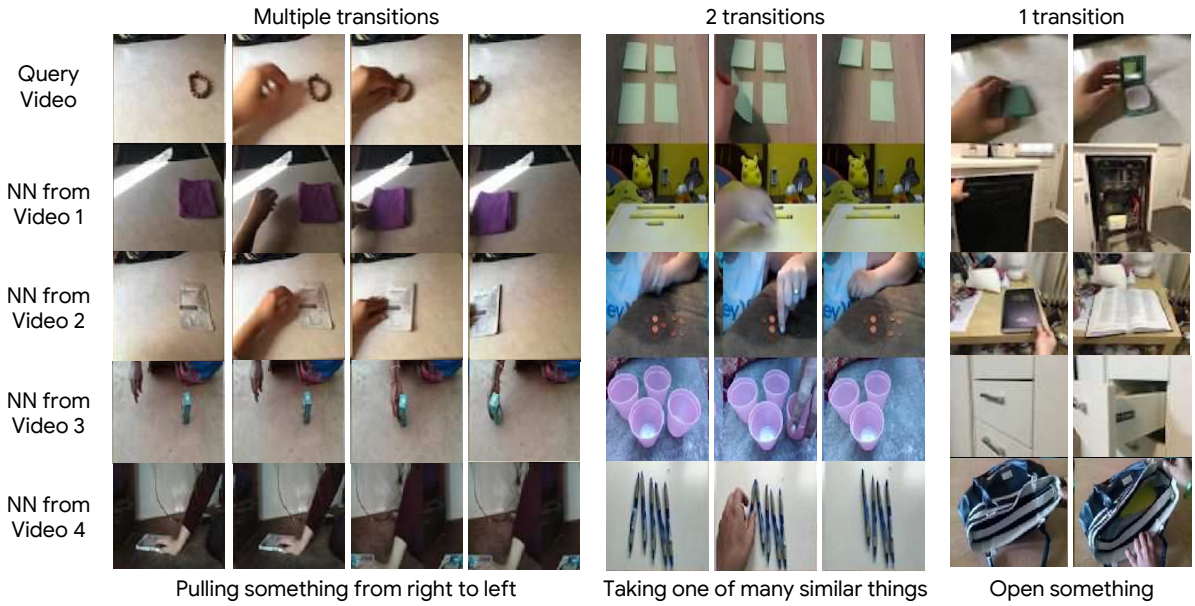

Figure 4: We highlight the transitions for various action classes by looking at some nearest neighbors (NN) in the hidden state of the GRU. It is interesting to note the richness in the representation of the hidden state of the recurrent module. The hidden states can encode the spatial position of objects, abstract concepts related to the group like the count of objects and the state of individual objects changing (objects changing from *closed* to *opened* state).

Such transitions are not observed when we perform nearest neighbor matching on the convolutional features extracted from the base network.

4.5.1 Classes with 0 transitions

Models can get 20.5% accuracy (row 3 in Table 2) without respecting the temporal order of the frames. Hence, there exist classes for which the GRU doesn't have to keep track of transitions. One reason might be that there exist action categories with strong correlation between some objects present and the action. For example, for the class *"Plugging something into something"* the network is able to predict the class correctly on the basis of wires and sockets present in the scene. This is a more common occurrence in the Kinetics dataset where often there is only one frame that is retrieved for all the frames in a video. This is expected as it is possible to get 71.3% accuracy by performing spatio-temporal averaging. This also suggests that in the hidden state of the GRU, the action is represented as a near-constant embedding that does not change much as the video progresses, which is fine for action recognition models.

4.5.2 Classes with one transition

For many classes in the *Something-Something* dataset, there is only one transition, specifically between the beginning and the end of the tasks. It is fascinating that in the hidden state of the GRU, same state of different objects closer. For example, the second column in Figure 4 shows the model clusters together the opened and closed states of a variety of objects like box, dishwasher, book, drawer and bag. Even though we could have chosen similar frames by hand for visualization but these states have emerged as a part of the training process. Furthermore, we have a common representation for the *opened* state of a number of objects on unseen videos from the validation set.

4.5.3 Classes with multiple transitions

There are other classes where we find that the hidden states are transitioning multiple times. In column 1 of Figure 4 we can see the object being *roughly* tracked as the action proceeds. We find these multiple transitions more often in action classes where the location of objects was important like "*Move something up/down*" and "*Pull/Push something to the left/right*".

5. Conclusion

We presented experiments using architectures that combine convolutional and recurrent units to process videos on two large scale datasets: Kinetics and 20BN Something-Something. We observe that in tasks that require us to preform temporal reasoning to correctly classify videos, the hidden state of the recurrent unit encodes rich representations of time-varying object states. While we have presented qualitative results of state transitions, we plan to use these representations for downstream tasks, which will also enable us to evaluate them quantitatively.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 3

[2] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042, 2016. 2

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *arXiv preprint arXiv:1705.07750*, 2017. 1, 2, 3

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 2

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[6] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015. 2

[7] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *null*, page 726. IEEE, 2003. 2

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. 2016. 1

[9] R. Girdhar and D. Ramanan. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems*, pages 33–44, 2017. 1

[10] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2, 3

[11] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*, 2017. 2

[12] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. Ieee, 2007. 2

[13] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 3

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011. 2

[16] I. Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005. 2

[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[18] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–14. IEEE, 2010. 2

[19] Z. Li, E. Gavves, M. Jain, and C. G. Snoek. Videolstm convolves, attends and flows for action recognition. *arXiv preprint arXiv:1607.01794*, 2016. 1, 2

[20] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016.

[21] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4694–4702. IEEE, 2015.

[22] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, 126(2-4):430–439, 2018. 2

[23] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 2

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

[25] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015. 1

[26] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016. 2

[27] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1, 2

[28] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 2

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 3

[31] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. *arXiv preprint arXiv:1704.05737*, 2017. 2

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*, 2014. 1, 3

[33] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. *arXiv preprint arXiv:1711.11248*, 2017. 3

[34] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *arXiv preprint arXiv:1711.07971*, 2017. 1

[35] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning for video understanding. *CoRR*, abs/1712.04851, 2017. 1, 3

[36] T. Yang and A. B. Chan. Recurrent filter learning for visual tracking. *arXiv preprint arXiv:1708.03874*, 2017. 2

[37] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. *arXiv:1711.08496*, 2017. 1, 3