CASCADED TEMPORAL SPATIAL FEATURES FOR VIDEO ACTION RECOGNITION

Tingzhao Yu[1,2], *Huxiang Gu*[1], *Lingfeng Wang*[1], *Shiming Xiang*[1], *Chunhong Pan*[1]

1. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
2. School of Computer and Control Engineering, University of Chinese Academy of Sciences
{tingzhao.yu, hxgu, lfwang, smxiang, chpan}@nlpr.ia.ac.cn

ABSTRACT

Extracting spatial-temporal descriptors is a challenging task for video-based human action recognition. We decouple the 3D volume of video frames directly into a cascaded temporal spatial domain via a new convolutional architecture. The motivation behind this design is to achieve deep nonlinear feature representations with reduced network parameters. First, a 1D temporal network with shared parameters is first constructed to map the video sequences along the time axis into feature maps in temporal domain. These feature maps are then organized into channels like those of RGB image (named as *Motion Image* here for abbreviation), which is desired to preserve both temporal and spatial information. Second, the *Motion Image* is regarded as the input of the latter cascaded 2D spatial network. With the combination of the 1D temporal network and the 2D spatial network together, the size of whole network parameters is largely reduced. Benefiting from the Motion Image, our network is an end-to-end system for the task of action recognition, which can be trained with the classical algorithm of back propagation. Quantities of comparative experiments on two benchmark datasets demonstrate the effectiveness of our new architecture.

Index Terms— action recognition, motion image, spatial-temporal decomposition, cascaded architecture

1. INTRODUCTION

Action recognition in real-world application plays a fundamental role in video analysis [1, 2, 3]. It aims to recognize the action being taken place from a video sequence. However, it is challenging primarily due to the long temporal duration,quantities redundant frames, cluttered backgrounds and viewpoint variations.

Existing action recognition methods can be divided into two categories: 1)methods based on hand-crafted features and 2) methods based on deep-convnet features. The first category based upon hand-crafted features extracts the unique spatial-temporal features and consecutively designs an effective classifier for action recognition. These features include Space Time Interest Points (STIP) [6], Dense Trajectories [7], improved Dense Trajectories (iDT) [8] and Dense SIFT [9]. Nevertheless, methods based on deep-convnet features extract the features and design the classifier simultaneously. The emergence of deep-convnet features for action recognition is partly due to their fantastic performance in computer vision areas, such as image classification [10], object recognition [11] and image segmentation [12].

This work was supported by the National Natural Science Foundation of China under Grants 61403376, 61573352, 91646207, and 91438105, the Beijing Natural Science Foundation under Grant (4162064), and the Youth Innovation Promotion Association CAS.

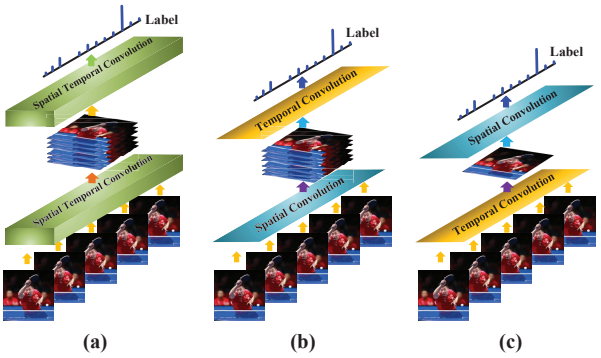

Fig. 1. A Simplified Comparison about C3D [4], F_{ST}CN [5] and The Proposed Architecture. (a) C3D exploits the spatial-temporal features simultaneously, (b) F_{ST}CN factorizes the spatial-temporal features into spatial and temporal domain, while (c) the Proposed Architecture decouples the spatial-temporal features into cascaded temporal and spatial domain.

Note that a video is a sequence of variable length frames, and a natural approach is to treat video frames as still images and apply traditional Convolutional Neural Network (CNN) directly at the individual frame level [13]. Nevertheless, temporal information is a critical component, and the descriptor derived by simply concatenating the frame level features can lose temporal information to some extent. While the main distinction between videos and images is their temporal continuity, one key ingredient in video analysis is the exploration of temporal information. On the one hand, 3D Convolutional Neural Network (3D CNN) [14] extends 2D CNN to temporal domain by performing 3D convolution, and this is later improved by Convolutional 3D (C3D) [4]. C3D can capture motion information among multiple adjacent frames. On the other hand, studies have shown that human visual cortex for action recognition contains two streams [15], a ventral stream for object recognition and a dorsal stream for motion recognition. Inspired by this mechanism, [16] proposes a two-stream ConvNet architecture which incorporates both a spatial and a temporal network. The spatial network tackles the problem of object recognition on frame level and the temporal network handles the issue of motion recognition on optical flow level. Two-stream ConvNet is proved to be effective and further ameliorated by [17], [18] and [19]. Besides, Spatial-temporal Convnet [20] explores multiple strategies extending the connectivity in temporal domain to capitalize the local Spatial-temporal information. These strategies include late fusion, early fusion and slow fusion.

Another approach in dealing with video sequence is temporal feature pooling [21], in which the authors investigate several variations of the basic max-pooling architecture, such as Conv Pooling,

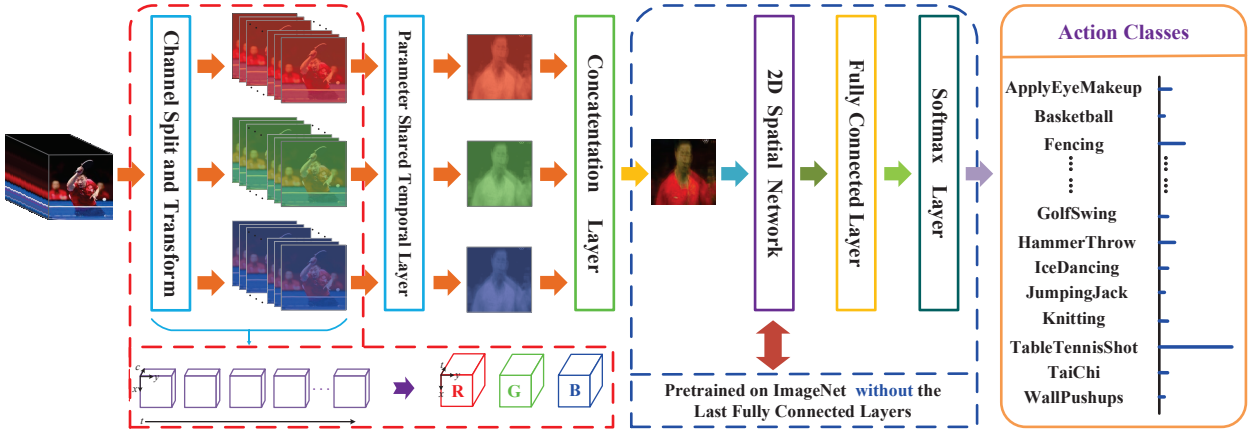

Fig. 2. The schematic architecture of the proposed network. The architecture consist in two cascaded convolution networks. The first network corresponds to three separated temporal network with shared structure and parameters, while the second network is related to the prevalent spatial network such as VGG, Inception and ResNet. Before temporal convolution, the video sequences are splitted through channel and reconstructed in terms of temporal axis (The Red Dashed Box). The spatial network is pretrained on ImageNet without the last fully connected layer and the last layer is set to be 101, corresponding to the desired number of video classes (The Blue Dashed Box).

Late Pooling, Slow Pooling, Local Pooling and Time-Domain Convolution Pooling. Furthermore, [22] proposes a Rank Pooling architecture through an inner-optimization of temporal semantics.

However, training such a network is still challenging either because of the lack of video datasets or because of the large amount of network parameters. We propose a novel end-to-end net architecture for video action recognition. The contributions of this paper are summarized as follows.

1) A channel-independent and parameter-shared shallow temporal network is elaborately established, through which a video sequence can be transformed into three temporal feature maps, named *Motion Image*, to compress the motion information.
2) The spatial-temporal network is dexterously decoupled into a cascaded temporal-spatial unit, and then the scale of network parameters is largely reduced.
3) A new video augmentation craftsmanship is implemented for a balance between dataset size and parameter scale. This technique dramatically boosts the performance from **45.3%** to **72.1%**.
4) The network is an end-to-end system, which permits us to utilize the prevalent spatial network pretrained on ImageNet, and then the cascaded architecture can be fine-tuned based on video datasets subsequently.

Note that our new network is mainly inspired by C3D [4], Dynamic Image [23] and $F_{ST}CN$ (Factorized Spatial-Temporal Convolutional Networks) [5], we differ from them in three aspects:

1) **Portability** - *C3D* convolves the video sequence in spatial-temporal domain simultaneously, while we cascade a 1D temporal convolution and a 2D spatial convolution consecutively.
2) **Velocity** - *Dynamic Image* first derives the dynamic feature maps through RankSVM and then treat the dynamic image as the standard input of a spatial network, while we train our network (both temporal and spatial) end-to-end via back propagation.
3) **Simplicity** - $F_{ST}CN$ factorizes the 3D convolution cube into spatial and temporal domain, while we decompose it into a cascaded temporal-spatial domain. Both of the factorization make sense, while our decomposition consumes much fewer parameters.

A more intuitive explanation about the difference among C3D, $F_{ST}CN$, and our network can be found Fig. 1.

2. THE PROPOSED CASCADED NETWORK

The key ingredient for action recognition lies in the exploration of spatial-temporal features. Video sequences consist of abundant 3D spatial-temporal information over time. Given an input video $\mathcal{I} \in \mathbb{R}^{n_x \times n_y \times n_t}$ and a 3D convolution cube $\mathcal{K} \in \mathbb{R}^{m_x \times m_y \times m_t}$, the corresponding 3D convolution output $\mathcal{O}$ [4] is

$$\begin{aligned} \mathcal{O} &= \mathcal{I} \circledast \mathcal{K} \\ &= \sum_{c} \sum_{p=1}^{m_x} \sum_{q=1}^{m_y} \sum_{r=1}^{m_t} \mathcal{K}^{pqr} \mathcal{I}^{(i_x+p)(i_y+q)(i_z+r)}, \end{aligned} \tag{1}$$

where $\circledast$ denotes 3D convolution, c is the number of feature maps, $i_x = 1, 2, \cdots, n_x, i_y = 1, 2, \cdots, n_y$ and $i_t = 1, 2, \cdots, n_t$ indicates a certain position in spatial temporal domain. m_x, m_y, m_t and n_x, n_y, n_t are the corresponding height, width and temporal length for the convolution cube and input video, respectively.

In our network, we decouple the 3D spatial-temporal network with two cascaded temporal and spatial networks. The temporal network manipulates on several stacked video sequences independently on three channels, while the spatial network performs similar to regular CNNs such as VGG, Inception and ResNet. Furthermore, the network is trained end-to-end. A more detailed configuration can be found in Fig. 2.

2.1. Cascaded Temporal Spatial Network

Distinguishingly, for 3D convolution, we can decompose the 3D convolution cube $\mathcal{K}$ as

$$\mathcal{K} = k_t \otimes K_{xy}, \tag{2}$$

where $\otimes$ denotes the Kronecker product, $k_t \in \mathbb{R}^{m_t}$ is a temporal 1D convolution filter and $K_{xy} \in \mathbb{R}^{m_x \times m_y}$ is a spatial 2D convolution kernel [5]. In most cases, Equation (2) is not strictly hold. However, considering that the size of 3D convolution cube is generally small

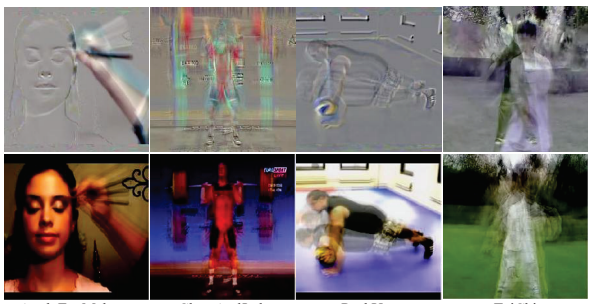

Fig. 3. Comparison of *Dynamic Image* and *Motion Image*. The top row is the *Dynamic Image*, and the bottom row is our *Motion Image*.

enough (usually $3 \times 3 \times 3$), and that the video sequences are usually low rank, thus this factorization is a fair approximation of the original convolution cube.

Then Equation (1) can be rewritten as two cascaded convolutions

$$\begin{aligned} F_t(i_x, i_y, :) &= \mathcal{I}(i_x, i_y, :) * k_t, \qquad i_x = 1, 2, \cdots, m_x, \\ & \qquad\qquad\qquad\quad i_y = 1, 2, \cdots, m_y. \\ F_{ts}(:, :, i_c) &= F_t(:, :, i_c) * K_{xy}, \quad i_c = 1, 2, 3. \end{aligned} \tag{3}$$

Here F_t is the *Motion Image*, F_{ts} is the **cascaded** Temporal Spatial Feature Map and $i_c = 1, 2, 3$ corresponds to R, G, B three channels. Formally, we restrict the number of the last Temporal Convolutional Filters to be 1, thus to a certain extent, F_t can be treated as a standard image. Details can be found in next subsection, and Fig. 3 demonstrates a visualization of these *Motion Images*. The main advantage of this cascaded temporal spatial architecture lies in the decline of convolution parameters (from $m_x m_y m_t$ to $m_t + m_x m_y$), and the acceleration in computation (Fig. 4).

2.2. Channel Split Transformation and Temporal Convolution

Given a video sequence, usually we uniformly select 16 frames, firstly, we resize each frame into 224×224, and split the 16 frames into three parts in terms of image channels, as is illustrated in Fig. 2 the Red Dash Box. Then the 4D input video data ($224 \times 224 \times 16 \times 3$) can be transformed into three 3D data cubes, with each data cube of size $224 \times 224 \times 16$. Afterwards, we regard each data cube as a 16-channel feature map, thus the temporal convolution can be implemented by standard 2D spatial convolution operators 1×1, instead of 3D convolution operators $1 \times 1 \times 16$. Note that the temporal networks for three data cube (corresponding to three channels) are structure and parameter shared. Implementation details can be found in section 3.2. This transformation makes it possible for us to exploit much longer temporal information.

Note that the convolutional filters can be regarded as generalized linear classifiers on the underlying data patches and each convolutional filter corresponds to a latent concept. Thus, the temporal convolution in Equation (3) can be regularized as

$$F_t(i_x, i_y, :) = \sum_t \alpha_t \varphi(\mathcal{I}_t) \tag{4}$$

where $\varphi(\mathcal{I}_t)$ deserves a nonlinear transformation in a small patch of frame t, and α_t is the corresponding weight for time t. While in

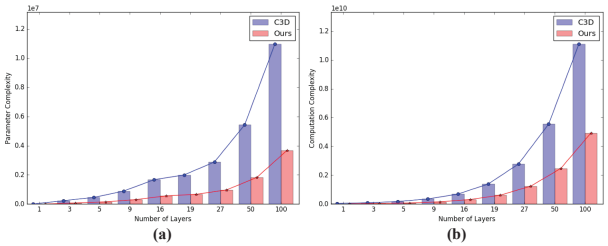

Fig. 4. Comparison of C3D and Ours on Convolutional Layers. *Left*: Parameter Complexity, *Right*: Computation Complexity (*Approx*).

Table 1. Comparison with Dynamic Image on UCF-101.

Method	Split1	Split2	Split3	Average
Mean Image	52.6%	53.4%	51.7%	52.6%
Max Image	48.0%	46.0%	42.3%	45.4%
Dynamic Image	57.2%	58.7%	57.7%	57.9%
Multi Dynamic Image	-	-	-	70.9%
Multi Dynamic Map	-	-	-	67.1%
Ours (without Aug)	44.9%	47.2%	43.7%	45.3%
Ours (with Aug)	**72.1%**	**72.6%**	**71.4%**	**72.1%**

dynamic image [23], this is obtained via

$$\mathbf{d}^* = \sum_t \alpha_t \psi(V_t) \tag{5}$$

using RankSVM, where $\mathbf{d}^*$ is the final dynamic image, $\psi(V_t)$ is the transformed pixel value. Generally speaking, the dynamic image is a special case of our temporal network for that our architecture deserves more nonlinearity.

2.3. Spatial Convolution Network

As mentioned before, the last temporal convolution layer comprises 1 filter, thus the temporal network outputs three channel-independent *Motion Image*s. We concatenate them as the input of spatial network. For simplicity, *ResNet-50* [24] pretrained on ImageNet is utilized as our spatial network. And the last fully connected layer is modified to fit the number of video classes. Besides, this network adopts a large Dropout ratio for sake of overfitting. The networks is implemented through *Keras* [25] and the release code and model can be found at `https://github.com/Tsingzao/motion_image`.

3. EXPERIMENTS

3.1. Datasets

We verify the effectiveness of our cascaded temporal spatial architecture on two state-of-the-art action recognition datasets, namely *UCF-101* [26] and *HMDB-51* [27].

UCF-101. UCF-101 [26] is a challenging dataset due to the large variations in pose, appearance, viewpoint, scale, background, illumination and camera motion. UCF-101 consists in 13,320 videos and comprises of 101 human action categories. UCF-101 provides three splits of training and testing videos. In our experiment results, we report the average classification accuracy over three splits.

Table 2. Results combined with Single Frame on UCF-101.

Method	Split1	Split2	Split3	Average
Single Frame	60.3%	52.8%	65.1%	59.4%
MDI + Frame	-	-	-	76.4%
Ours + Frame	**83.3%**	**82.6%**	**93.7%**	**86.6%**

HMDB-51. HMDB-51 [27] dataset is another challenging human action recognition dataset with 6,849 videos, divided into 51 human action classes. HMDB-51 also offers three splits of training and testing videos, however, for simplicity, we report the classification accuracy on the first split at some researches do.

3.2. Implementation Details

Our architecture mainly consists in two cascaded networks, the Temporal Convolution Net and the Spatial Convolution Net. The Temporal Convolution Net involves three temporal convolution layers, and the Spatial Convolution Net is set to be a 50-layer net *ResNet-50* [24]. For Temporal Convolution Net, the number of convolutional filters are set to be $64, 32$ and 1, where 1 guarantees that the three channel output can be concatenated as a *Motion Image*. The Spatial Convolution Net is pretrained on ImageNet and the last fully connected layer is changed to meet the corresponding video classes.

In training stage, we first *uniformly* select 16 frames each video, data augmentation on video level will be described in subsection 3.3 in detail. Unlike most training strategies, we utilize *Adadelta* to train our network instead of *SGD*, because we found that SGD requires much more iterations to convergent. The learning rate *lr*, decay rate ρ and constant ϵ are set to be 1.0, 0.95 and 10^{-8}.

3.3. Data Augmentation

Data augmentation is critical for deep convolution networks especially when the dataset is scarcity. In our experiment, we firstly split each video into $16-$frame video clips uniformly, for example, a $160-$frame video sequence can be divided into 10 video clips. Through this technique, the UCF-101 dataset can be extend from $13,320$ to $148,932$, and HMDB-51 can be extend from $6,849$ to $24,794$. Then we randomly choose a subset of 100 video clips, and resize the selected video frames into 224×224. 100 video clips guarantee that each subset contains at least or nearly 1 instance for each video class. In each subset, we train our network with 10 epoches, and on the entire dataset we train our network with 3 epoches. This data augmentation technique is vital as illustrated in Table 1 (about 25 percents promotion).

3.4. Comparison with Dynamic Image

We firstly demonstrate the effectiveness of our *Motion Image* compared with *Dynamic Image*. Table 1 presents the results on UCF-101. To be fair, the results about *Mean Image*, *Max Image* and *Dynamic Image* has been reported by [23]. Table 1 shows that training our network end-to-end performs poor, and even much worse than the *Mean Image* and *Max Image*. This is probably due to that our network consumes extra parameters at the *Temporal Convolution Stage*, and the amount of video dataset is relatively smaller than the scale of network parameters. When the video dataset is augmented about 10 times the size of the original dataset, our network performs best among all of the methods being compared.

Table 3. Comparison with state-of-the-art algorithms.

Method	UCF-101	HMDB-51
iDT [8]	85.9%	51.9 %
C3D [4]	82.3%	-
$F_{ST}CN$ [5]	84.5%	49.0%
ConvNet [20]	65.4%	-
Two-Stream [16]	86.9%	52.8%
Dynamic Image [23]	76.9%	42.8%
Proposed	**86.6%**	**52.9%**

3.5. Comparison with State-of-the-art Method

We also exploit a fusion strategy inspired by two-stream network. The strategy involves fusing our net architecture with a single frame (randomly selected) spatial network. Note that, for simplicity, the fusion strategy is adopted at the decision level. The results are reported in Tabel 2. Tabel 2 indicates that the integration with single frame promotes the performance nearly 10 percents. Usually fusing different methods generally improves the performance, and this case holds for their complementary property.

We then compare our architecture with state-of-the-art algorithms and the results are presented in Table 3. As is reported, iDT is probably the most valid handcraft feature, and Two-Stream is the most competitive deep feature. Table 3 illustrates that the proposed architecture is effective and comparable compared with other state-of-the-art algorithms (we don't report the results combined with iDT). The cascaded architecture and video augmentation jointly make it possible for us to perform better on small dataset.

3.6. Visualization

We then consider the visualization of *Motion Image*. As noted before, the *Motion Image* can be obtained via a concatenation of three temporal feature maps. Fig. 3 presents the results of *Dynamic Image* (top row) and *Motion Image* (bottom row). In fact, *Dynamic Image* performs rank pooling in temporal domain while *Motion Image* handles convolve pooling. Both of them depict the action information in a still image, whereas our *Motion Image* behaves much like a standard RGB image. And that's a feasible reason why our *Motion Image* performs better than *Dynamic Image* utilizing a 2D spatial network pretrained on ImageNet (natural images).

4. CONCLUSION

We decouple the 3D spatial-temporal descriptor into the cascaded temporal-spatial domain for human action recognition. We achieve this goal by cascading together a 1D temporal network and a 2D spatial network. We organize the output of the 1D temporal network as *Motion Image* to preserve both temporal and spatial information. The 2D spatial network, which performs as regular CNNs, extracts features from *Motion Image*. We largely reduce the scale of parameters associated to our cascaded architecture, and our network is capable of being trained via tricks of back propagation in an end-to-end manner.

Note that the essence of *Motion Image* is temporal pooling. We adopt temporal pooling through temporal convolution based on the correlation among neighbor frames. Nevertheless, the video data itself is desired to be low rank, and our future work will focus on dig out more valuable information based on that assumption.

5. REFERENCES

[1] Yiğithan Dedeoğlu, B Uğur Töreyin, Uğur Güdükbay, and A Enis Çetin, "Silhouette-based method for object classification and human action recognition in video," in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 64–77.

[2] Muhammet Bastan, Hayati Cam, Ugur Gudukbay, and Ozgur Ulusoy, "Bilvideo-7: an mpeg-7-compatible video indexing and retrieval system," *IEEE MultiMedia*, vol. 17, no. 3, 2010.

[3] Tarkan Sevilmiş, Muhammet Baştan, Uğur Güdükbay, and Özgür Ulusoy, "Automatic detection of salient objects and spatial relations in videos for a video database system," *Image and Vision Computing*, vol. 26, no. 10, pp. 1384–1396, 2008.

[4] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 4489–4497.

[5] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 4597–4605.

[6] Ivan Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[7] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu, "Action recognition by dense trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 3169–3176.

[8] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 3551–3558.

[9] Robin Aly, Relja Arandjelovic, Ken Chatfield, Matthijs Douze, Basura Fernando, Zaid Harchaoui, Kevin Mcguiness, Noël O'Connor, Dan Oneata, Omkar Parkhi, et al., "The axes submissions at trecvid 2013," *arXiv preprint arXiv:1507.02159*, 2013.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 91–99.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3431–3440.

[13] Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano, "Toward automatic phenotyping of developing embryos from videos," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1360–1371, 2005.

[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 495–502.

[15] Melvyn A Goodale and A David Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[16] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 568–576.

[17] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[18] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, "Convolutional two-stream network fusion for video action recognition," *arXiv preprint arXiv:1604.06573*, 2016.

[19] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes, "Spatiotemporal residual networks for video action recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 3468–3476.

[20] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1725–1732.

[21] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 4694–4702.

[22] Basura Fernando, ANU EDU, and Stephen Gould, "Learning end-to-end video classification with rank-pooling," in *Proceedings of the 33th International Conference on Machine Learning (ICML)*, 2016, pp. 236–243.

[23] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[25] François Chollet, "Keras," `https://github.com/fchollet/keras`, 2015.

[26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[27] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2556–2563.