Chapter 1

Related work

1.1 Action Recognition

First approaches for action classification consisted of two steps a) compute complex handcrafted features from raw video frames such as SIFT, HOG, ORB features and b) train a classifier based on those features. These approaches made the choise of features a signifact factor for network's performance. That's because different action classes may appear dramatically different in terms of their appearance and motion patterns. Another problem was that most of those approaches make assumptions about the circumstances under which the video was taken due to problems such as cluttered background, camera viewpoint variations etc. A review of the techniques, used until 2011, is presented in Aggarwal and Ryoo 2011.

Recent results in deep architectures and especially in image classification motivated researchers to train CNN networks for the task of action recognition. The first significant attempt was made by Karpathy et al. 2014. They design their architecture based on the best-scoring CNN in the ImageNet competition. they explore several methods for fusion of spatio-temporal features using 2D operations mostly and 3D convolution only in slow fusion. Simonyan and Zisserman 2014 used a 2 CNNs, one for spatial information and one for optical flow and combined them using late fusion. They show that extacting spatial context from videos and motion context from optical flow can improve significantly action recognition accuracy. Feichtenhofer, Pinz, and Zisserman 2016 extend this approach by using early fusion at the end of convolutional layers, instead of late fusion which takes places at the last layer of the network. On top that, they used a second network for temporal context which they fuse with the other network using late fusion. Futhermore, Wang et al. 2016 based their method on Simonyan and Zisserman 2014, too. They deal with the problem of capturing long-range temporal context and training their network given limited training samples. Their approach, which they named Temporal Segment Network (TSN), seperates the input video in K segments and a short snippet from each segment

is chosen for analysis. Then they fuse the extracted spatio-temporal context, making, eventually, their prediction.

Some other methods included a RNN or LSTM network for classification like Donahue et al. 2014, Ng et al. 2015 and Ma et al. 2017. Donahue et al. 2014 address the challenge of variable lenghts of input and output sequences, exploiting convolutional layers and long-range temporal recursions. They propose a Long-term Recurrent Convolutional Network (LRCN) which is capable of dealing with the tasks of action Recognition, image caption and video description. In order to classify a given sequence of frames, LRCN firstly gets as input a frame, and in particular its RGB channels and optical flow and predicts a class label. After that, it extracts video class by averaging label probabilities, choosing the most probable class. Ng et al. 2015 firstly explore several approaches for temporal feature pooling. These techniques include handling video frames individually by 2 CNN architectures: either AlexNet or GoogleNet, and consisted of early fusion, late fusion of a combination between them. Futhermore, they propose a recurrnet neural Network architecture in order to consider video clips as a sequences of CNN activations. Proposed LSTM takes an input the output of the final CNN layer at each consequentive video frame and after five stacked LSTM layers using a Softmax classifier, it proposes a class label. For video classification, they return a label after last time step, max-pool the predictions over time, sum predictions over time and return the max or linearly weight the predictions over time by a factor g, sum them and return the max. They showed that all approaches are 1% different with a bias for using weighting predictions for supporting the idea that LSTM becomes progressively more informed. Last but not least, Ma et al. 2017 use a two-stream ConvNet for feature extraction and either a LSTM or convolutional layers over temporally-constructed feature matrices, for fusing spatial and temporal information. They use a ResNet-101 for extracting feature maps for both spatial and temporal streams. They divide video frames in several segments like Wang et al. 2016 did, and use a temporal pooling layer to extract distinguished features. Taken these features, LSTM extracts embedded features from all segments.

Additionally, Tran et al. 2014 explored 3D Convolutional Networks (Ji, Yang, and Yu 2013) and introduced C3D network which has 3D convolutional layers with kernels $3 \times 3 \times 3$. This network is able to model appearence and motion context simultaneously using 3D convolutions and it can be used as a feature extractor, too. Combining Two-stream architecture and 3D Convolutions, Carreira and Zisserman 2017 proposed I3D network. On top of that, the authors emphasize in the advantages of transfer learning for the task of action recognition by repeating 2D pre-trained weights in the 3rd dimension. Hara, Kataoka, and Satoh 2017 proposed a 3D ResNet Network for action recognition based on Residual Networks (ResNet) (He et al. 2015) and explore the effectiveness of ResNet with 3D Convolutional kernels. On the other hand,Diba et al. 2017 based their approach on DenseNets(Huang, Liu, and Weinberger 2016) and extend DenseNet architecture by using 3D filters and pooling kernels instead of 2D, naming this approach as DenseNet3D. Futhermore, they introduce Temporal Transition Layer (TTL), which concatenates temporal feature-maps

extracted at different temporal depth ranges and replaces DenseNet's transition layer. Last but not least, Tran et al. 2017 experiment with several residual network architectures using combinations of 2D and 3D convolutional layer. Their purpose is to show that a 2D spatial convolution followed by a 1D temporal convolution achieves state of the art classification performance, naming this type of convolution layer as R(2+1)D. A more detailed presentation for Action Recognition techniques used until 2018 is included in Kong and Fu 2018.

1.2 Action Localization

As mentioned before, Action Localization can be seen as an extention of the object detection problem. Instead of outputing 2D bounding boxes in a single image, the goal of action localization systems is to output action tubes which are sequences of bounding boxes that contain an performed action. So, there are several approaches including an object-detector network for single frame action proposal and a classifier.

The introduction of R-CNN (Girshick et al. 2013) achieve significant improvement in the performance of Object Detection Networks. This architecture, firstly, proposes regions in the image which are likely to contain an object and then it classifies them using a SVM classifier. Inspired by this architecture, Gkioxari and Malik 2014 design a 2-stream RCNN network in order to generate action proposals for each frame, one stream for frame level and one for optical flow. Then they connect them using the viterbi connection algorithm. Weinzaepfel, Harchaoui, and Schmid 2015 extend this approach, by performing frame-level proposals and using a tracker for connecting those proposals using both spatial and optical flow features. Also their method performs temporal localization using a sliding window over the tracked tubes.

Peng and Schmid 2016 and Saha et al. 2016 use Faster R-CNN (Ren et al. 2015) instead of RCNN for frame-level proposals, using RPN for both RGB and optical flow images. After getting spatial and motion proposals,Peng and Schmid 2016 fuse them exploring and from each proposed ROI, generate 4 ROIs in order to focus in specific body parts of the actor. After that, they connect the proposal using Viterbi algorithm for each class and perform temporal localization by using a sliding window, with multiple temporal scales and stride using a maximum subarray method. From the other hand, Saha et al. 2016 perform, too, frame-level classification. After that, their method performs fusion based on a combination between the actioness scores of the appearence and motion based proposals and their overlap score. Finally, temporal localization takes place using dynamic programming.

On top of that, Singh et al. 2017 and Kalogeiton et al. 2017 design their networks based on the Single Shot Multibox Detector Liu et al. 2015). Singh et al. 2017 created an online real-time spatio-temporal network. In order their network to execute real-time, Singh et al. 2017 propose a novel and efficient algorithm by adding boxes in tubes in every frame if they overlap more than a threshold, or alternatively, terminate the action tube if for k-frames no box was

added. Kalogeiton et al. 2017 designed a two-stream network, which they called ACT-detector, and introduced anchor cuboids. For K frames, for both networks, Kalogeiton et al. 2017 extract spatial features in frame-level, then they stack these features. Finally, using cuboid anchors, the network extracts tubelets, that is a sequence of boxes, with their corresponding classification scores and regression targets. For linking the tubelets, Kalogeiton et al. 2017 follow about the same steps as Singh et al. 2017 did. For temporal localization, they use a temporal smoothing approach.

Most recently, YOLO Network (Redmon et al. 2015) became the inspiration for Hu et al. 2019 and El-Nouby and Taylor 2018. In Hu et al. 2019, concepts of progression and progress rate were introduced. Except from proposing bounding boxes in frame level, they use YOLO together with a RNN classifier for extracting temporal information for the proposals. Based on this information, they create action tubes, seperated into classes. Some other approaches include pose estimation like Luvizon, Picard, and Tabia 2018 do.They proposed a method for calcualatin 2D and 3D poses and then they performed action classification. They use the diffferentiable Soft-argamax function for estimating 2D and 3D joints, because argmax function is not differentiable. Then, for T adjacent poses, they create an image representation with time and N_j joins as $x - y$ dimensions and having 2 channes for 2D poses and 3 channels for 3D poses. They use Convolutional Layers in order to produce action heats and then using max plus min pooling and a Softmax activation they perform action classification. Most of aforementioned networks use per-frame spatial proposals and extract their temporal infomation by calculating optical flow. On the other hand, Hou, Chen, and Shah 2017 design an architecture which icludes proposal in video segment level, which they called Tube CNN (T-CNN). Video segment level means that the whole video is seperated into equal length video clips, and using a C3D for extracting features, it returns spatio-temporal proposals. After getting proposals, Hou, Chen, and Shah 2017 link the tube proposals by an algorithm based on tubes' actioness score and overlap. Finally, classification operation is performed for the linked video proposals.

1.3 Our implementation

We propose a network similar to Hou, Chen, and Shah 2017. Our architecture is consisted by the following basic elements:

- One 3D Convolutional Network, which is used for feature extraction. In our implementaion we use a 3D Resnet network which is taken from Hara, Kataoka, and Satoh 2018 and it is based on ResNet CNNs for Image Classification He et al. 2015.
- Tube Proposal Network for proposing action tubes (based on the idea presented in Hou, Chen, and Shah 2017).
- A classifier for classifying video tubes.

Pending ... more commentary and a figure

Bibliography

[1] J.K. Aggarwal and M.S. Ryoo. "Human Activity Analysis: A Review". In: *ACM Comput. Surv.* 43.3 (Apr. 2011), 16:1–16:43. ISSN: 0360-0300. DOI: 10.1145/1922649.1922653. URL: http://doi.acm.org/10.1145/1922649.1922653.

[2] Ross B. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CoRR* abs/1311.2524 (2013). arXiv: 1311.2524. URL: http://arxiv.org/abs/1311.2524.

[3] Shuiwang Ji, Ming Yang, and Kai Yu. "3D convolutional neural networks for human action recognition." In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2013), pp. 221–31.

[4] Jeff Donahue et al. "Long-term Recurrent Convolutional Networks for Visual Recognition and Description". In: *CoRR* abs/1411.4389 (2014). arXiv: 1411.4389. URL: http://arxiv.org/abs/1411.4389.

[5] Georgia Gkioxari and Jitendra Malik. "Finding Action Tubes". In: *CoRR* abs/1411.6031 (2014). arXiv: 1411.6031. URL: http://arxiv.org/abs/1411.6031.

[6] A. Karpathy et al. "Large-Scale Video Classification with Convolutional Neural Networks". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

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

[8] Du Tran et al. "Learning Spatiotemporal Features with 3D Convolutional Networks". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2014), pp. 4489–4497.

[9] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[10] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *CoRR* abs/1512.02325 (2015). arXiv: 1512.02325. URL: http://arxiv.org/abs/1512.02325.

[11] Joe Yue-Hei Ng et al. “Beyond Short Snippets: Deep Networks for Video Classification”. In: *CoRR* abs/1503.08909 (2015). arXiv: 1503.08909. URL: http://arxiv.org/abs/1503.08909.

[12] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

[13] Shaoqing Ren et al. “Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’15. Montreal, Canada: MIT Press, 2015, pp. 91–99. URL: http://dl.acm.org/citation.cfm?id=2969239.2969250.

[14] Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. “Learning to track for spatio-temporal action localization”. In: *CoRR* abs/1506.01929 (2015). arXiv: 1506.01929. URL: http://arxiv.org/abs/1506.01929.

[15] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. “Convolutional Two-Stream Network Fusion for Video Action Recognition”. In: *CoRR* abs/1604.06573 (2016). arXiv: 1604.06573. URL: http://arxiv.org/abs/1604.06573.

[16] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. “Densely Connected Convolutional Networks”. In: *CoRR* abs/1608.06993 (2016). arXiv: 1608.06993. URL: http://arxiv.org/abs/1608.06993.

[17] Xiaojiang Peng and Cordelia Schmid. “Multi-region two-stream R-CNN for action detection”. In: *ECCV - European Conference on Computer Vision*. Vol. 9908. Lecture Notes in Computer Science. Amsterdam, Netherlands: Springer, Oct. 2016, pp. 744–759. DOI: 10.1007/978-3-319-46493-0_45. URL: https://hal.inria.fr/hal-01349107.

[18] Suman Saha et al. “Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos”. In: *CoRR* abs/1608.01529 (2016). arXiv: 1608.01529. URL: http://arxiv.org/abs/1608.01529.

[19] Limin Wang et al. “Temporal Segment Networks: Towards Good Practices for Deep Action Recognition”. In: *CoRR* abs/1608.00859 (2016). arXiv: 1608.00859. URL: http://arxiv.org/abs/1608.00859.

[20] João Carreira and Andrew Zisserman. “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”. In: *CoRR* abs/1705.07750 (2017). arXiv: 1705.07750. URL: http://arxiv.org/abs/1705.07750.

[21] Ali Diba et al. “Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification”. In: *CoRR* abs/1711.08200 (2017). arXiv: 1711.08200. URL: http://arxiv.org/abs/1711.08200.

[22] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. “Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition”. In: *CoRR* abs/1708.07632 (2017). arXiv: 1708.07632. URL: http://arxiv.org/abs/1708.07632.

[23] Rui Hou, Chen Chen, and Mubarak Shah. "Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos". In: *CoRR* abs/1703.10664 (2017). arXiv: 1703.10664. URL: http://arxiv.org/abs/1703.10664.

[24] Vicky Kalogeiton et al. "Action Tubelet Detector for Spatio-Temporal Action Localization". In: *ICCV 2017 - IEEE International Conference on Computer Vision*. Venice, Italy, Oct. 2017.

[25] Chih-Yao Ma et al. "TS-LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition". In: *CoRR* abs/1703.10667 (2017). arXiv: 1703.10667. URL: http://arxiv.org/abs/1703.10667.

[26] Gurkirt Singh et al. "Online Real time Multiple Spatiotemporal Action Localisation and Prediction". In: 2017.

[27] Du Tran et al. "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *CoRR* abs/1711.11248 (2017). arXiv: 1711.11248. URL: http://arxiv.org/abs/1711.11248.

[28] Alaaeldin El-Nouby and Graham W. Taylor. "Real-Time End-to-End Action Detection with Two-Stream Networks". In: *CoRR* abs/1802.08362 (2018). arXiv: 1802.08362. URL: http://arxiv.org/abs/1802.08362.

[29] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6546–6555.

[30] Yu Kong and Yun Fu. "Human Action Recognition and Prediction: A Survey". In: *CoRR* abs/1806.11230 (2018). arXiv: 1806.11230. URL: http://arxiv.org/abs/1806.11230.

[31] Diogo C. Luvizon, David Picard, and Hedi Tabia. "2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning". In: *CoRR* abs/1802.09232 (2018). arXiv: 1802.09232. URL: http://arxiv.org/abs/1802.09232.

[32] Bo Hu et al. "Progress Regression RNN for Online Spatial-Temporal Action Localization in Unconstrained Videos". In: *CoRR* abs/1903.00304 (2019). arXiv: 1903.00304. URL: http://arxiv.org/abs/1903.00304.