ACTION TUBELET DETECTION FOR THE ATOMIC VISUAL ACTION DATASET

Carlos Roig *Ramon Morros* *Verónica Vilaplana*

Universitat Politècnica de Catalunya

ABSTRACT

In this paper we have implemented the Action Tubelet detector (ACT) presented at ICCV2017 for the Atomic Visual Actions dataset (AVA). There has been a lot of advance in the research field of video action recognition in the past years and many datasets have been almost solved. The AVA set introduces a new challenge in the field by introducing more complex kinds of actions. After commenting the different methods and datasets used up until now, we review more in detail the ACT-detector and the Atomic Visual Actions dataset. Then we explain our modifications to use the ACT with AVA and finally we make a series of comments and remarks of where the ACT does not perform well and where it could be improved, in order to perform better with more complex action sets.

1. INTRODUCTION

Video action recognition is a field of research in computer vision that tries to find and classify actions, both in space and time, from video data. The first developments in this area using deep learning focused exclusively on the classification aspect of the problem [1, 2]. As the models started improving and architectures started being proposed [3, 4], apart from classification, spatio-temporal detection of the actions has become the new focus [5]. State-of-the-art methods are concentrating on finding the human pose in the video [6] and building bounding boxes that span many different frames, called tubes or *tubelets* [7, 8].

Despite the rapid growth in the field, there are not many publicly available datasets. To have a dataset for video action recognition two things are required: firstly a list of videos, and secondly the bounding boxes at a frame level for every label available. Building a new dataset is a difficult task because of the labeling process, which is very tedious and also the necessity of having a large amount of quality video content without copyright.

There are some very well-known datasets that have become the benchmark in the field, namely UCF-101 [9], J-HMDB [10], Sports-1M [5] and the more recent ActivityNet [11] and Kinetics [12]. But all these sets have different flaws, which will undoubtedly be eradicated in newer sets. The problems of theses sets include the following: an excess of uncommon activities, like Olympic sports, scarcity of examples, being mostly recorded from a first-person point of view and only having one label for each video. This is the current stage in which new datasets have started to appear, trying to address the previous problems, like DALY [13], which stands for Daily Action Localization in YouTube videos or the AVA Dataset [14].

In the following sections, we review the AVA Dataset (Section 2) and comment on why we think it has some interesting features in order to qualify becoming a new benchmark. We then comment on the state-of-the-art method tested with the old datasets (Section 3), followed by implementation details, changes performed and results (Section 4), comments and discussion on why the methods are not working as well as with previous datasets (Section 5). We conclude with some ideas on how the system might be improved in future research (Section 6).

2. AVA DATASET

The Atomic Visual Actions dataset was released by Google Research team trying to address the current problems of the previous benchmark sets.

The set consists in 3-second clips extracted from 15-minute sequences of a pool of movies that depicts common and casual human actions, which can be split in three categories: Person-Person, Person-Object and Pose actions. The categories proposed in AVA have a more complex structure than the ones from other sets, from different reasons like the fact that the actions are more complex, since there can be more than one action being performed at the same time. For example, a person can be talking to another person while standing, which is a common combination of actions. Other reasons are a combination of factors that may decide if a person is doing one action or another, if a person is talking to the phone is not the same as if is talking to someone else, and the unique clue may be the pose of the person.

In terms of numbers, the dataset has 192 15-minute videos, from which 57.6k 3-second clips are extracted and finally we can find more than 210k actions is those clips. The number of different classes in the set is 80, with a very big divergence in terms of samples per class (the class distribution can be seen in [14], Figure 5. Also notice that the y-axis is represented in a logarithmic scale).

3. ACTION TUBELET DETECTOR

The Action Tubelet Detector was presented in the ICCV2017 [7] alongside with [6, 8] in the field of spatio-temporal action detection. The results of the three methods are quite similar and stand at the top of the field.

The ACT method is based on the Single Shot MultiBox Detector (SSD) [15], which is a method for object detection in images using a discretized feature space. The SSD is able to output bounding boxes for each class present in the image with different aspect ratios and sizes. The underlying architecture of the SSD is based on the VGG-16 [16] without the last fully connected layers that are replaces by a series of convolutional layers that decrement the size of the feature layer generating the discretized feature map of SSD.

Building on top of the SSD, the ACT has two neural networks, the first one called the appearance detector focuses on detecting objects and is based on RGB frames while the second one, called the motion detector, uses optical flow images obtained following the same process as in [17]. The input of both networks is modified so it can take multiple consecutive images as input, and output, what the authors call, an anchor cuboid, which is a bounding box that spans over time and has a confidence score for each class in the set plus a background class, the cuboids are called *tubelets*.

In order to train the ACT-detector network, two losses are proposed, one based on the confidence score of each class and the regression loss extracted from the difference between the predicted and the ground truth bounding boxes.
The *tubelets* have a fixed spatial extent over time, in order to solve this limitation, the algorithm proposed in [18] is implemented to link the different *tubelets* into action tubes, adding a temporal smoothing for the overlapping bounding boxes at a frame level, in order to build a progressive evolution of the boxes without any sharp transactions for the same tube.

The tubes no longer have a fixed temporal extent and can follow actions across the whole sequence of frames.
The results are reported on the sets: UCF-101, J-HMDB and UCF-Sports [19, 20] in the categories: classification accuracy, mean average best overlap (MABO) and frame and video mean average precision (mAP) using different thresholds to decide if the detected boxes correspond with the ground truth ([7], Sections 4.4 to 4.6).

4. IMPLEMENTATION AND RESULTS

In this project we have implemented and tested the ACT-detector explained in the previous Section to the AVA dataset.

As we commented in Section 2, the class distribution of AVA is very unbalanced, so the first pre-processing executed on the set, was to trim it to 30 labels, which are the one with the highest number of examples. Then, the number of samples per class was balanced, in order to have 100 examples of each class for training and also 10 for testing, extracted from the test subset.

The final modification of the AVA dataset is that, for each 3-second clip, there are bounding boxes only at the central frame of the clip. In order to use the ACT-detector, we need to have bounding boxes at a frame level, for all the frames of the video. The authors of the AVA dataset have stated that they are already working on the full set bounding boxes, but since they were not available during the realization of this project, the approach that we have taken is to expand the same bounding box for the full 3-second clip.

To train the motion detector, computing the optical flow online was not an option, so we precomputed and saved the flow files and then trained the motion detector with the pre-generated flow images. In section 5.2 we go into more detail about the algorithm selected and the time needed to extract all the flow files for the AVA dataset.

With the processed set, we started training the appearance detector for about one week with one Titan X GPU initialized with the weights of the ILSVRC-2016 [21]. After that the optical flow detector was trained for another week. We save the model every 10000 iterations for both networks and then have tested the classification accuracy at different points of the training.

The results of the AVA dataset paper are reported for the frame average precision with an IoU threshold of 0.5, which are compared to our best frame mAP, corresponding to the appearance network with 110K+ iterations and motion network with 30K+ iterations, in Table 2.

Finally we show the actions that are being best represented by the AC-detector (Table 3). We see that many actions have no representation at all, and a very few have some good performance, the ones obtaining better accuracies are mostly have no representation at all, and a very few have some good

Appearance iter.	Motion iter.	Classif. accuracy
80K	20K	8.71%
100K	30K	10.56%
110K+	30K+	12.38%
110K+	90K+	12.66%

Table 1. Results changing the number of iterations in both networks.

Method	Frame mAP
AVA paper	16.2%
ACT-detector	3.53%

Table 2. Frame mAP comparison.

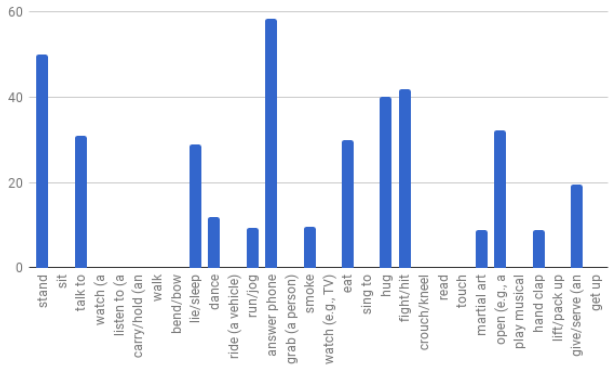

Table 3. Accuracy for each class with the best classifier of Table 1.

performance, the ones obtaining better accuracies are mostly related to Human-Object or Pose actions, while the Human-Human actions are having almost no representation, this may be caused by the fact of having initialized the weights with the ones from the ILSVRC-2016, which are for a different task and may give more information about which kind of objects are being displayed in the frame than the interaction between two people.

In the following section we will comment the results previously shown, why they are worse than the AVA results and some comments about the implementation and performance of the ACT.

5. DISCUSSION

5.1. Why are the results worse than the AVA paper?

If we compare the results of the ACT-detector with the system used in the AVA dataset for the J-HMDB (Table 4) we can see that the ACT-detector works better than the one used in the AVA paper, so why is the ACT working worse on AVA?

Set	J-HMDB	AVA
ACT-detector	65.7%	3.53%
AVA paper	62%	16.6%

Table 4. Results frame mAP comparing both datasets and systems.

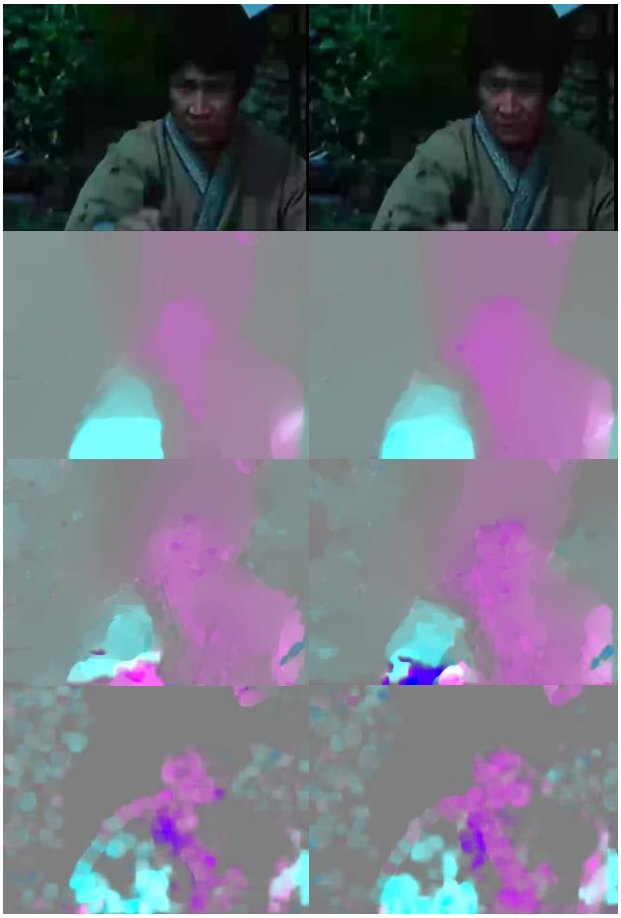

Figure 1. Optical flow comparison, first we have the original frames, then the corresponding flows for Brox, TVL1 and Farneback respectively.

The first idea that comes to mind are the different decisions that we took during the implementation. First of all the networks have probably not converged, considering the results in Table 1. Then the decision of expanding the central bounding box to the full frame was done due to the lack of other options, giving the nature of the 3-second clips, where we can see changes of plane or multiple bounding boxes for the same class, we could not make any assumption to make the bounding boxes more accurate without modifying them manually. Another motive on why the system works way worse is because of the optical flow selected to train the motion network, reviewed more in detail in the following sub-section.

5.2. Optical flow computation comparison

There are many methods used in video activity recognition to compute the optical flow, the most common is called TVL1 Flow (Denseflow) [22]. For the ACT results the authors used the Brox Flow [23]. Both methods give very accurate flow vectors, but the time that it take to compute the results is very high.

The time to compute the optical flow for one 3-second clip with the Brox method is 55 minutes using CPU. For the TVL1 we need 25 minutes per clip. So we will last months for extracting the flow of all the clips of the set.
In order to train the motion detector, we used a simple method called Farneback Flow [24] pre-built in OpenCV [25]. That takes about 10 seconds to extract the flow of one clip. We can see a comparison of the flows in Figure 1.

As it is seen in the figure, the results of the optical flow are inversely proportional to the tame taken to compute them, but given our time limitations we could only use the Farneback approach.

5.3. Performance of the ACT-detector

Even though the SSD is intended to work in real time, the training of the modified SSD architectures (appearance and motion) take a lot of time to be trained.

Due to the lack of time for this project and the limited access to only one GPU, the time that took to train the networks was more than two weeks and without having full convergence of the network weights.
To test the model, with the reduced test set, we required almost one day, around 15 hours to extract the *tubelets*, 3 hours to build the tubes from the *tubelets* and about 15 minutes to obtain the different test results.

5.4. ACT-detector as an atomic action detector

The results of the ACT for the different benchmark datasets are really good, but as we have seen, for the AVA dataset the results are quite bad.

The first and most clear motive why the ACT is not working is because the ground truth bounding boxes are not properly defined. So the appearance and motion detectors are being trained with data that is not clean which may induce wrong classifications.

Another possible cause it the fact that the ACT-detector has been thought-out as a method to follow specific actions, which are always recorded from the same point of view, and do not disappear from the camera view. In the contrary, AVA is a dataset that takes into account previous events, due to the nature of the actions that are present in the set, we can see a conversation that has many different changes of speaker and therefore many camera changes despite being performed during the same action.

The last reason that affects the performance on AVA is that the ACT-detector has been devised to work with clips that only have one action for the whole sequence, AVA has many actions at the same time, and the same person may be performing more than one action, for example, standing and talking to another person. The current *tubelet* extractor can handle more than one action at a time, but the tube builder only takes the most relevant action if there are overlapped *tubelets*. That is why many different labels in Table 3 do not have any kind of representation in the final results.

6. CONCLUSIONS AND FUTURE WORK

The results show on Section 4 and discussed in Section 5, show that the ACT-detector need to be modified in order to work with the AVA dataset. Due to time and computing power limitations, we could not achieve the best results possible with the ACT for the AVA dataset that probably can be obtained, but what we obtained shows the first flaws on the method when trying to find atomic visual actions.

The AVA dataset has a lot of potential, it is a large-scale dataset with complex actions that can help the field evolve towards systems that can understand very complex actions. Also the current benchmarks have already very high accuracy and many new approaches seem to improve the state-of-the-art but only are improving a very small percentage of the previous best, having a new dataset where there are not any good results yet, is promising.

In the last update of the AVA paper, they introduced a novel method for video action recognition that is supposed to beat the state-of-the-art in other benchmark sets ([14], Section 5). They use a neural network approach to extract the optical flow, FlowNet v2 [26] with Faster-RCNN [27].

The best results on benchmark sets, only rely on visual data, since the audio information is not that important for those sets. In the case of AVA, the audio information may be more important than in previous sets. Audio can help giving information of a conversation, give temporal continuity clues, detecting multiple people in the same temporal space and contextual information may be some improvements that could be obtained using audio features.

7. ACKNOWLEDGMENTS

We would like to thank both Albert Gil and Josep Pujal for giving us access to the UPC computing cluster, where we had access to a GPU, and helped us with some troubles during the installation of the system.

8. REFERENCES

[1] S. Ji, et al. *3d convolutional neural networks for human action recognition.* IEEE Trans. Pattern Analysis and Machine Intelligence, 35(1):221–231, Jan 2013. ISSN 0162-8828

[2] J. Ng, et al. *Beyond short snippets: Deep networks for video classification*. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4694–4702, 2015.

[3] K. Simonyan and A. Zisserman. *Two-stream convolutional networks for action recognition in videos*. In Proc. Advances in Neural Information Processing Systems (NIPS), pages 568–576, 2014.

[4] C. Feichtenhofer, A.Pinz, and A. Zisserman. *Convolutional two-stream network fusion for video action recognition*. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1933–1941, 6 2016.

[5] A. Karpathy, et al. *Large-scale video classification with convolutional neural networks*. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1725–1732, 2014.

[6] M. Zolfaghari, et al. *Chained Multi-stream Networks Exploiting Pose, Motion, and Appearance for Action Classification and Detection.* In arXiv preprint arXiv:1704.00616v2, 2017.

[7] V. Kalogeiton, et al. *Action Tubelet Detector for Spatio-Temporal Action Localization.* In arXiv preprint arXiv:1705.01861v3, 2017.

[8] R. Hou, et al. *Tube Convolutional Neural Nerworks (T-CNN) for Action Detection in Videos.* In arXiv preprint arXiv:1703.10664v3. 2017.

[9] K. Soomro, et al. *UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild.* In arXiv preprint arXiv:1212.0402. 2012.

[10] H. Jhuang, et al. *Towards understanding action recognition*. In ICCV, 2013.

[11] F. Caba, et al. *ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding.* Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[12] W. Kay, et al. *The Kinetics Human Action Video Dataset*. In arXiv preprint arXiv:1705.06950, 2017.

[13] P. Weinzaepfel, et al. *Human Action Localization with Sparse Spatial Supervision*. In arXiv preprint arXiv:1605.05197. 2017.

[14] C. Gu, et al. *AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions*. In arXiv preprint. arXiv:1705.08421, 2017.

[15] W. Liu, et al. *SSD: Single Shot MutliBox Detector*. In arXiv preprint arXiv:1512.02325 2016.

[16] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* In arXiv preprint arXiv:1409.1556, 2015.

[17] G. Gkioxari and J. Malik. *Finding action tubes*. In CVPR, 2015.

[18] G. Singh, et al. *Online real time multiple spatiotemporal action localization and prediction on a single platform*. In arXiv preprint arXiv:1611.08563, 2017.

[19] M. Rodriguez, et al. *Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition*, Computer Vision and Pattern Recognition, 2008.

[20] K. Soomro and A. Zamir, *Action Recognition in Realistic Sports Videos, Computer Vision in Sports*. Springer International Publishing, 2014.

[21] O. Russakovsky, J. Deng, et al. *ImageNet Large Scale Visual Recognition Challenge*. IJCV, 2015.

[22] J. Sanchez, et al. *TV-L1 Optical Flow Estimation.* In Image Processing On Line, July 2013.

[23] T. Brox, et al. *High Accuracy Optical Flow Estimation Based on a Theory for Warping.* In Proceeding of the 8th European Conference on Computer Vision, May 2004.

[24] G. Farnebäck. *Two-Frame Motion Estimation Based on Polynomial Expansion.* In Scandinavian Conference on Image Analysis, 2003.

[25] OpenCV, https://opencv.org/

[26] E. Ilg, et al. *FlowNet 2.0: Evolution of optical flow estimation with deep networks*. In CVPR, 2017.

[27] S. Ren, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In arXiv preprint arXiv:1506.01497, 2015