Selecting active frames for action recognition with 3D convolutional network

Hoang Tieu Binh[1], Akihiro Sugimoto[2], Ma Thi Chau[3], and Bui The Duy[3]

[1]Hanoi National University of Education
[2]National Institute of Informatics - Japan
[3]University of Engineering and Technology - Hanoi National University

Abstract

Recent applications of Convolutional Neural Networks, especially 3-Dimensional Convoltutional Neural Networks (3DCNNs) for human action recognition (HAR) in videos have widely used. In this paper, we use a multi-stream framework which is a combination from separated networks with different kind of input generated from unique video dataset. To achieve the high results, firstly, we proposed a method to extract the active frames (called Selected Active Frames - SAF) from a videos to build datasets for 3DCNNs in video classifying problem. Second, we deploy a new approach called Vote fusion which considered as an effective fusion method for ensembling multi-stream networks. From the various datasets generated from videos, we extract frames by our method and feed into 3DCNNs for feature extraction, then we carry out training and then fuse the results of softmax layers of these streams. We evaluate the proposed methods on solving action recognition problem. These method are carried on three well-known datasets (HMFB51, UCF101, and KTH). The results are also compared to the state-of-the-art results to illustrate the efficiency and effectiveness in our approach.

Keywords: Action recognition, fusion network, 3D convolutional neural networks, fusion network.

I. Introduction

Human action recognition has received a significant number of research in recent years. Since Alex Krizhevsky *et al.* [1] showed the best result in Imagenet 2012 competition, there is a revolution on action recognition matter. Going with stronger hardware infrastructure, larger action databasse, the recognition results increase day by day. From the first stage of recognition process, data usually was used for extracting the features before applying some traditional recognizing procedures, the methods SIFT (Scale Invariant Feature Transform) [2], SURF (Speeded Up Robust Feature) [3] and HOG (Histogram of Oriented Gradients) [4] are the examples.

Nowadays, deep learning and aritificial intelligence show some huge advantages among other feature representaion methods. Especially Convolutional Neural Network is one of the deep structure which archive state-of-the-art result in various domains. In the domain of images classification, Convolutional Networks (ConvNets) [5] demonstrates the powerful potential of learning visual representations [1] from raw visual data.

However, expansion of CNNs to action recognition in video in recent recent research remain unable to achieve significant improvements over tradition hand-craft features for video-based action recognition. Beside this, 2D-CNN architectures or any other previous methods are not able to exploit the important characteristic of video like cube information which describes motion and action in video.

Over the last five years, 3DCNN and its variations for action recognition often focus on short video intervals with various number of frames from 1 to 16 [6], [7], [8], [9], [10], [11]. Especcially, Varol *et al.* [12] used 100-frames dataset as temporal resolutions for spatio-temporal convolutional networks.

Equipped with large-scale training datasets like Imagenet [13], MS-Celeb-1M [14], THUMOS Dataset [15], and Activitynet [16]..., CNNs are the most chosen method for the still-image recognition tasks such as face, object, scene, human action recognition [17], [18], [19].

In this paper, we have two contributions related to action recognition in videos. Firstly, we propose a method for extracting a dataset from video for training process. Secondly, we present a novel fusion method for ensembling multiple networks. The explanation of these two contributions are described in Section III and Section IV. The result is evaluated and commented in Section VII.

We implemented our approach using Python version 3 with Tensorflow backend and model are based on C3D [8].

II. Related work

Several recent works on how to extract motion information and deploy it in 3DCNNs show the significant improvement results. The nature of using 3DCNN in modeling actions in temporal manner is no argument. From the successful C3D network implemented by Du Tran et al [8], there are many version of C3D which deployed end-to-end networks which trained directly from videos. [6] used two-stream convolutional networks for action recognition in videos which split data into two dataset called spatial stream and temporal stream. The first network use single frame as input data, the second network use multi-frame optical flow as temporal information. Both networks output will be fused by class score ensemble to generate classification at the end. Varol *et al.* in [12] used Long-term tTemporal Convolutions (LTC) to learn video representations to demonstrate LTC-CNN with increased temporal extents will improve the accuracy of action recognition. Hakan Bilen *et al.* [9] introduced dynamic images which are extracted from video with a weighted number applying on each frame. In [20], to encode the information extracted from video, Optical Flow Co-occurrence Matrices (OFCM) which based on the co-occurrence matrices computed over the optical flow field are captured.

Also related to our work is the fusion methods which show some optimistic signal when applying to multiple networks. By using multiple streams neural network with various fusing methods could issue better results [21]. [22] uses two-stream network fusion for video action recognition on both spatial and temporal information. Bosting fusion method is used in [23] demonstrated that the accuracy increase 7.2% and 7.2% when executing on UCF101 and HMDB51.

III. Fusion methods

Fusion method is the combination of the two or multiple layers from two or multiple networks. They are powerful procedures which improve networks' performance. It reduces the variance portion in the bias-variance decomposition of the prediction error. There are number of ways of fusing multiple networks. Our project has experimented with different fusion methods that all tend to contribute to accuracy improvement. In addition, the trade off between number of models and their complexity has been investigated and we show that fusing learning may lead to accuracy gains along with reduction in training time. Concretely, we go futher with three popular fusion methods which mentioned by Christoph Feichtenhofer in [22].

Without loss of generality, assume that we have two feature maps $x^a \in \mathbb{R}^{H \times W \times D}$ and $x^b \in \mathbb{R}^{H \times W \times D}$, we define a function $f : x^a, x^b \to y$ which fuses these map a, b to produce $y \in \mathbb{R}^{H \times W \times D}$ where H, W, D as the height, width, depth of channels of the respective feature maps. When applying to convolutional neural network architectures, consist of convolutional, pooling, fully-connected layers, there are multiple options for applying f at different layers, for example early-fusion, late-fusion or multiple layer fusion [7].

Avarage fusion: $y^{sum} = f^{avg}(x^a, x^b)$ evaluate the average of the two feature maps at the same location i, j and feature channel d:

$$y^{avg}_{i,j,d} = avg(x^a_{i,j,d}, x^b_{i,j,d}), \tag{1}$$

where $1 \leqslant i \leqslant H, 1 \leqslant j \leqslant W, 1 \leqslant d \leqslant D$ and $x^a, x^b, y^{sum} \in \mathbb{R}^{H \times W \times D}$.

Max fusion: Similar to Sum fusion in equation (1), $y^{max} = f^{max}(x^a, y^b)$ takes the maximum of the two feature maps:

$$y^{max}_{i,j,d} = max(x^a_{i,j,d}, x^b_{i,j,d}), \tag{2}$$

where i, j, d are defined as (1), $y^{max} \in \mathbb{R}^{H \times W \times D}$.

Concatenation fusion: $y^{cat} = f^{cat}(x^a, x^b)$ combine two feature maps at them same location i, j across the feature channel d.

$$\begin{aligned} y^{cat}_{i,j,2d} &= x^a_{i,j,d}, \\ y^{cat}_{i,j,2d-1} &= x^b_{i,j,d}, \end{aligned} \tag{3}$$

where $y^{cat} \in \mathbb{R}^{H \times W \times 2D}$.

Vote fusion: Basing on the classification results of all networks, the labels in which most networks recognize are chosen. If all networks have the same result, max/average fusion on these networks are applied to get the final recognition result (noted that, we compare the classification results on each test samples). Comparing to Majority Voting [24], Vote fusion method will be more flexible because our classification not only depends on the voting but also uses fusion scores of all networks. We carry out on multi-model frameworks which include more than two networks. The special situation for this method in the circumstance that we have only one or two networks, in this case, we use other fusion methods as mentioned above.

IV. Extracting data method

In a video, there are a series of frames which contain similar information, our work is looking for the way to select the most informative frames which represent for the whole clip.

Assume that, the difference between two consecutive frames is small, we recognize that the more difference between two frames will represent the more action in these frames. In our work, we measure the Euclidean image distance between two consecutive frames (the distance between their corresponding points in the image space).

The Euclidean distance of two images x, y of fixed size M by N is written by:

$$d_E^2(x,y) = \sum_{i,j=1}^{MN} g_{i,j}(x^i - y^i)(x^j - y^j) = (x-y)^T G(x-y). \quad (4)$$

Where the symmetric matrix $G = (g_{i,j})_{MN \times MN}$ will be referred to as metric matrix [25].

The most informative frame is the frame which has the largest distance between two continuous images. It is called Selected Active Frame (SAF) and is calculated by using the following procedure:

Algorithm 1: Extract frame by finding the largest distance between adjacent images in a segment of a video clip

```
Input  : One segment with fixed number of frames
Output: The most active frame
d_max = 0
for i ← 1 to number of frames do
    F_i ← the i^th frame
    F_{i+1} ← the next frame after F_i
    d_i = Euclidean_distance(F_{i+1}, F_i)
    if d_i > d_max then
        d_max = d_i
        F_max = F_i
    end
end
return F_max
```

The input of the algorithm is a series of consecutive frames which extracted from videos. We further expand the method in case of the segment length equals to the whole clip. We call this Full Selected Active Frames (FSAF) method.

To the SAF method, a video is divided into 16 segments with the same number of frames. In each segment, we apply above algorithm to collect 16 frames for constructing a 16-frame dataset. For the FSAF method, the most 16 active frames from beginning to the end of video are selected.

We have comparison chart of SAF and FSAF as Figure 1 with three random videos from HMDB51 dataset. It is simple to realize that, the SAF line is little more straightforward than FSAF one. In our implementation, the accuracy of these two method is similar with a very slightly difference. Thus, we only show the SAF results for representative method.

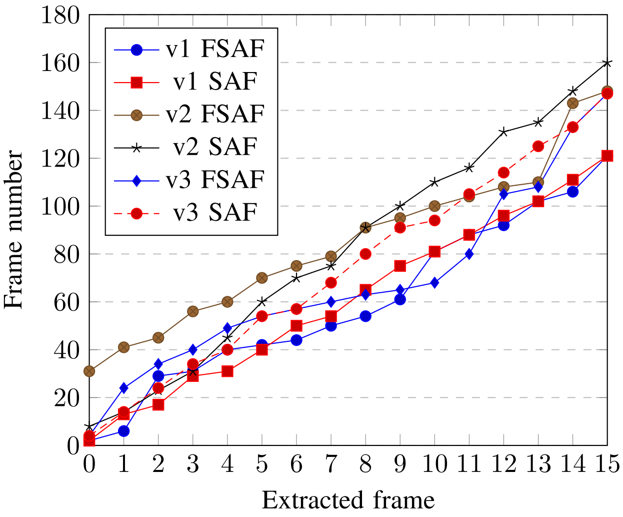

Fig. 1: Comparing between FASF and SAF method, v1, v2, v3 denote for video 1, video 2, video 3, respectively. The SAF lines look nearly straight line than FSAF thus the frames are chosen in a segment instead of stretching all over the clip. Better viewed in color.

V. Multi-stream 3D Convolutional Neural Network

Multimodal networks or multi-layer framework are shown promising performance. In our work, we use three streams as shown in Fig. 2. It includes RGB stream, Optical flow stream and SAF stream.

A. RGB stream

Sequentially, 16 raw RGB images are extracted from one video to form a totaly RGB dataset. The images can be resized to the desired resolution. This approach are used to implement in recent work [11], [12], [23].

B. Optical flow stream

Among many flow estimations such as Lucas-Kanede [26], Brox [27], we employ Farnebáck method [29] develop in [12], [28] to compute optical flow for KTH, HMDB51 and UCF101. The Farnebáck is considered as dense optical flow with fast and low error rate method [12]. All the training results are saved for predicting process.

C. SAF stream

Based on the method proposed IV, video is split into 16 segments without overlap and same number of frames. SAF method is applied on all videos from three datasets and is executed independently with training process. The output of this stream is also saved and considered as input of fusing process.

The multi-stream network architecture is displayed by the Fig. 2. Each stream is similar to the network which describe in Fig. 3. The vote fusion process is also represented in visual way for being well-understood.

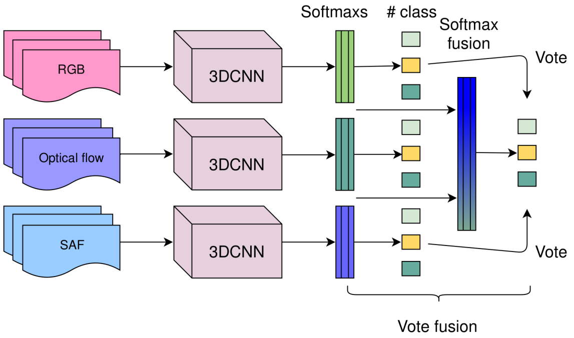

Fig. 2: Multi-stream framework. Three datasets are fed into networks, the outputs as softmax layers are fused to become final softmax. The predict process includes voted classes and score from final softmax layer.

Our network architecture is shown in Fig. 3. The input is a 16-frame block of images which are normalized to 112×112. The first convolution operator has 64 kernels with size of 3×3. Max pooling is then performed, followed by interleaving convolution/maximization layers (C2-M5). The number of kernels are increased from 64 to 256 to extract more features. The size of fully connected layer (dense layer) is $512 \times 4 \times 4 = 8192$ which seems to be sufficient to represent the characteristics of human actions. Due to avoid over-fit, we drop out the dense layer with 50 percent. The output is softmax layer which is the probability distribution related to number of classes possible classifications.

VI. Experiment

A. Datasets

We conduct evaluation on three popular datasets such as UCF101, HMDB51 and KTH.

Firstly, UCF101 [30] is a large dataset which consist of 101 actions in 13320 videos, every action has 25 groups with similar background, it contains 4 to 7 videos for each group (average 131 clips per action). The videos are in .avi format and have dimensions of 320×240 pixels with frame rate is 25 frames per second.

The second dataset is HMDB51 [31], smaller than the first one with 51 actions contained in 6766 clips (average of 132 clips per action) . The videos have spatial resolution of 320×240 and frame rate of 25 fps.

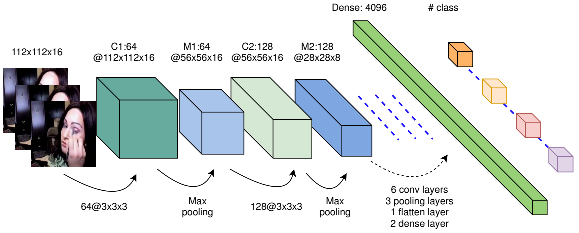

Fig. 3: Network architecture. We use the same structure for all networks, the only difference is the kind of input data including Optical flow images, RGB images and SAF images.

Both two datasets are the same resolution, frame rate and number of clips per action, however the classification results are far different (refer to section VII). It is possible that the HMDB51 clips are shorter, the background is more complicate than the UCF101. beside this, HMDB51 contains several some categories like chewing, talking, laughing, smiling,... and several other categories like drinking, eating. Such similar categories are much difficult in categorizing and easy to be ambiguous.

The last database is KTH [32] which contains six types of human actions including walking, jogging, running, boxing, hand waving and hand clapping. This dataset has 600 videos performed by 25 subjects and 4 scenarios. The dimension of the clip is 160×120 and frame rate of 25 frames per second. The length of each clip is much longer than both two above datasets (approximately 20 seconds per clip).

B. Implement detail

We extract RGB images, Optical flow images and Selected Active Frame images as three datasets. The data constructing process is independent to the whole progress causes saving a lot of time. In our experiment, we use a Core i7 computer with 16 cores, 32Gb RAM, NVIDIA GeForce GTX 1070 with 8Gb memory.

In order to avoid overfitting, the networks are based on the model were pre-trained by Sports-1M dataset which has one million videos in 487 categories [7]. The input data is subtracted by mean value of sport1M before having fed data to the networks. We apply data augmentations (cropping, flipping, RGB jittering) [1], [33] to increase the diversity of videos. We use Adam optimizer [34] with learning rate 10^{-4} as initiation. The training processes stop after 10000 iterations.

The code is forked from Github [35] with our additional modules for SAF and ensemble process. We share our code at https://github.com/binhhoangtieu/C3D-tensorflow.

VII. Evaluation

First, we compare the accuracy between the methods which carry on KTH dataset to illustrate the efficiency of SAF method. All these results are conducted on single stream network. Our network structure applied on this case is described in Fig. 2. The result are shown in Table I. Our network can recognize at a high level accuracy in almost actions. The best recognizable actions comparing to other method are **Jogging** and **Running** with 92% and 89.2%, respectively. Second, we compare our multi-stream framework with others works. Our networks are shown in 3. Two datasets HMDB51 and UCF101 are used in order to make the comparison between other state-of-the-art achievements. The results in Table II show that: Toward single stream, SAF method achieve the best result with 49.4% and 88.1% correspond to HMDB51 and UCF101 dataset. In the manner of using multi-stream, our methods perform significant improvement on triple fusion methods. Among these, three-stream with Vote fusion method gain the best accuracy with 66.2% and 94.9% for HMDB51 and UCF101, respectively.

Method	**Boxing**	**Hand clapping**	**Hand waving**	**Jogging**	**Running**	**Walking**	**Average**
3DCNN [36]	90	94	97	84	79	97	90.2
Schuldt [32]	97.9	59.7	73.6	60.4	54.9	83.8	71.7
Dollár [37]	93	77	85	57	85	90	81.2
Niebles [38]	98	86	93	53	88	82	83.3
Jhuang [39]	92	98	92	85	87	96	91.7
ASF (Ours)	92	90	91.3	**92**	**89.2**	87.2	90.3

TABLE I: Action recognition for accuracy on dataset KTH. The unit is in percentage. All methods use single-stream structure.

Input	**HMDB51**	**UCF101**	**Network structrure**
RGB [8]	50	85.2	3-C3Ds
DI [11]	46.8	78.4	C3D
OF [11]	48.9	78.2	C3D
RGB+DI+OF [11]	57.9	88.6	3-C3Ds
RGB [23]	53.1	85.4	3DCNN
DI+RGB+Trajectory [9]	65.2	89.1	2DCNN
SAF	49.4	88.1	C3D
RGB+OF+SAF	64.4	92.9	3-C3Ds+Max fusion
RGB+OF+SAF	65.9	94.2	3-C3Ds+Avg fusion
RGB+OF+SAF	66.2	94.9	3-C3Ds+Vote fusion

TABLE II: Comparision between some state-of-the-art results. The unit is in percentage

We also implement three fusion methods separately for KTH dataset. The results reported in Table III show that the vote fusion slightly better than Max fusion and Average fusion. Comparing to single-stream networks shown in Table I and Table II, multi-stream structure with vote fusion shows significant improvement in accuracy, increase $3.5\,6.8\%, 6.8\%$ comparing to SAF network on KTH, HMDB51, UCF101 respectively.

Fusion method	**KTH**	**HMDB51**	**UCF101**
Max fusion	90.8	64.4	92.9
Avg fusion	93.8	65.9	94.2
Vote fusion	93.8	66.2	94.9

TABLE III: Fusion Comparison table

Concretely, we show our deeply analysis about the difference between fusion methods. The Table IV makes comparison Max fusion and Average fusion inside Vote fusion method. This happen when all networks recognize absolutely differently. Our framework will apply these fusion for final labels. In our test case, with KTH dataset, only two instances are unalike and Max fusion truly recognized 1 instance, occupied 50%. With the HMDB51, Max fusion predicted 246 instance, occupied 36.66%, Average fusion gained 39.64% in correcting rate. Last, on UCF101 dataset, the truly recognized rate are 69.65% and 75.55% for Max fusion and Average fusion respectively. According to these statistics, the max fusion may get better result in case of applying inside Vote fusion. With this conclusion, in our work, we apply Average fusion for entire Vote fusion progress.

Dataset	Test instances	Different labels	Max Fusion		Average fusion	
			Number	Percent	Number	Percent
KTH	129	2	1	50%	0	0.00%
HMDB51	1722	671	246	36.66%	266	39.64%
UCF101	3318	458	319	69.65%	346	75.55%

TABLE IV: Fusion methods comparison. Different labels column represents for number of labels which are classified differently between streams. The Max Fusion and Avg Fusion columns represent for number of classes which methods are truly recognized. The Percent column is calculated from the ratio between Number and Different labels columns.

VIII. Conclusion

In this paper, we have concretely described an approach on video classification using 3D convolutional network with selected active frame extracting method. Beside this, we also demonstrate the efficiency in recognizing when applying vote fusion method. Our results show significant improvement comparing the state-of-the-art results.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *NIPS*, pp. 1–9, 2012.

[2] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, 2004.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, 2008.

[4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, IEEE, 2005.

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, 1998.

[6] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos,” *NIPS*, 2014.

[7] A. Karpathy and T. Leung, “Large-scale video classification with convolutional neural networks,” *CVPR*, 2014.

[8] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convoutional Networks,” *ICCV*, 2015.

[9] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, “Dynamic Image Networks for Action Recognition,” *CVPR*, 2016.

[10] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, “Real-time Action Recognition with Enhanced Motion Vector CNNs,” *CVPR*, 2016.

[11] L. Jing, Y. Ye, X. Yang, and Y. Tian, “3D Convolutional Neural Network With Multi-Model Framework for Action Recognition,” *ICIP*, 2017.

[12] G. Varol, I. Laptev, and C. Schmid, “Long-term Temporal Convolutions for Action Recognition,” *PAMI*, 2017.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, 2015.

[14] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “MS-Celeb-1M: A dataset and benchmark for large scale face recognition,” in *ECCV*, 2016.

[15] H. Idrees, A. R. Z. nad Yu-Gang Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, “The THUMOS challenge on action recognition for videos ”in the wild”,” *CoRR*, 2016.

[16] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in *CVPR*, IEEE, 2015.

[17] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” in *NIPS*, 2014.

[18] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *CVPR*, 2014.

[19] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *CVPR*, 2014.

[20] W. R. S. Carlos Caetano, Jefersson A. dos Santos, “Optical flow co-occurrence matrices: A novel spatiotemporal feature descriptor,” in *ICPR*, 2016.

[21] Z.-H. Z. J. W. W. Tang, “Ensembling neural networks: Many could be better than all,” *Artificial Intelligence*, vol. 137, 2002.

[22] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” *CVPR*, 2016.

[23] X. Yang, P. Molchanov, and J. Kautz, “Multilayer and Multimodal Fusion of Deep Neural Networks for Video Classification,” *Proceedings of the 2016 ACM on Multimedia Conference - MM ’16*, 2016.

[24] C. Ju, A. Bibaut, and M. J. van der Laan, “The relative performance of ensemble methods with deep convolutional neural networks for image classification,” *CoRR*, vol. abs/1704.01664, 2017.

[25] L. Wang, Y. Zhang, and J. Feng, “On the euclidean distance of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, 2005.

[26] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, pp. 674–679, Morgan Kaufmann Publishers Inc., 1981.

[27] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2004.

[28] I. Laptev, I. Willow, C. Science, and E. N. Suprieure, “Efficient feature extraction , encoding and classification,” *CVPR*, 2014.

[29] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Image Analysis*, Springer Berlin Heidelberg, 2003.

[30] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” *CRCV-TR-12-01*, 2012.

[31] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: a large video database for human motion recognition,” in *ICCV*, 2011.

[32] C. Schüldt, B. Caputo, C. Sch, and L. Barbara, “Recognizing human actions : A local SVM approach,” *ICPR’*, 2004.

[33] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *CoRR*, 2013.

[34] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[35] “C3D-tensorflow.” https://github.com/hx173149/C3D-tensorflow/, 2016. [Online; accessed 07-Mar-2018].

[36] S. Ji, M. Yang, K. Yu, and W. Xu, “3D convolutional neural networks for human action recognition,” *PAMI*, vol. 35, 2013.

[37] R. Dollr, P., V., Cottrell, G., and B. S., “Behavior recognition via sparse spatio-temporal features,” in *ICCV VS-PETS*, 2005.

[38] J. C. Niebles, H. Wang, and L. Fei-Fei, “Unsupervised learning of human action categories using spatial-temporal words,” *International Journal of Computer Vision*, 2008.

[39] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, “A biologically inspired system for action recognition,” in *International Conference on Computer Vision (ICCV)*, 2007.