Key Frame Extraction for Salient Activity Recognition

Sourabh Kulhare*, Shagan Sah†, Suhas Pillai‡, Raymond Ptucha*
*Computer Engineering †Center for Imaging Science ‡Computer Science
Rochester Institute of Technology, Rochester, USA
Email: sk1846@rit.edu

***Abstract*—Surveillance cameras have become big business, with most metropolitan cities spending millions of dollars to watch residents, both from street corners, public transportation hubs, and body cameras on officials. Watching and processing the petabytes of streaming video is a daunting task, making automated and user assisted methods of searching and understanding videos critical to their success. Although numerous techniques have been developed, large scale video classification remains a difficult task due to excessive computational requirements. In this paper, we conduct an in-depth study to investigate effective architectures and semantic features for efficient and accurate solutions to activity recognition. We investigate different color spaces, optical flow, and introduce a novel deep learning fusion architecture for multi-modal inputs. The introduction of key frame extraction, instead of using every frame or a random representation of video data, make our methods computationally tractable. Results further indicate that transforming the image stream into a compressed color space reduces computational requirements with minimal affect on accuracy.**

I. Introduction

Decreasing hardware costs, advanced functionality and prolific use of video in the judicial system has recently caused video surveillance to spread from traditional military, retail, and large scale metropolitan applications to every day activities. For example, most homeowner security systems come with video options, cameras in transportation hubs and highways report congestion, retail surveillance has been expanded for targeted marketing, and even small suburbs, such as the quiet town of Elk Grove, CA utilize cameras to detect and deter petty crimes in parks and pathways.

Detection and recognition of objects and activities in video is critical to expanding the functionality of these systems. Advanced activity understanding can significantly enhance security details in airports, train stations, markets, and sports stadiums, and can provide peace of mind to homeowners, Uber drivers, and officials with body cameras. Security officers can do an excellent job at detecting and annotating relevant information, however they simply cannot keep up with the terabytes of video being uploaded on a daily basis. Automated activity analysis can scrutinize every frame, databasing a plethora of object, activity, and scene based information for later analysis. To achieve this goal, there is a substantial need for the development of effective and efficient automated tools for video understanding.

Conventional methods use hand-crafted features such as motion SIFT [1] or HOG [2] to classify actions of small temporal extent. Recent successes of deep learning [3] [4] [5] in the still image domain have influenced video research. Researchers have introduced varying color spaces [6] , optical flow [7], and implemented clever architectures [8] to fuse disparate inputs. This study analyzes the usefulness of the varying input channels, utilizes key frame extraction for efficacy, and introduces a multi-modal deep learning fusion architecture for state-of-the-art activity recognition.

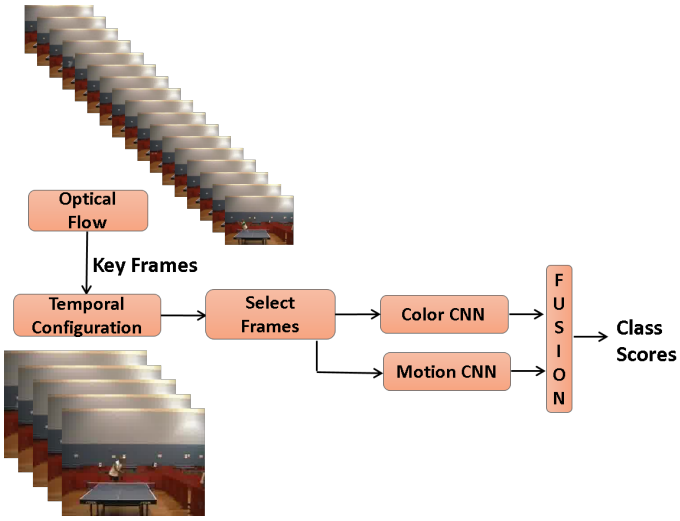

Fig. 1. Illustration of Key Frame extraction work flow using the optical flow. The key frames are inputs to independent color and motion CNN's.

Large scale video classification remains a difficult task due to excessive computational requirements. Karpathy et al. [8] proposed a number of techniques for fusion of temporal information. However, these techniques process sample frames selected randomly from full length video. Such random selection of samples may not take into account all useful motion and spatial information. Simonyan and Zisserman [7] used optical flow to represent the motion information to achieve high accuracy, but with steep computational requirements. For example, they reported that the optical flow data on 13K video snippets was 1.5 TB.

To minimize compute resources, this work first computes key frames of a video, and then analyzes key frames and their neighbors as a surrogate for analyzing all frames in a video. Key frames, or the important frames in a video, can

form a storyboard, in which a subset of frames are used to represent the content of a video. We hypothesize that deep learning networks can learn the context of the video using the neighbors of key frames. Voting on key frame regions then determines the temporal activity of a video snippet. Some events can be represented by fewer key frames whereas complex activities might require significantly more key frames. The main advantage with this approach is the selection of frames which depend on context of the video and hence overcome the requirement to train a network on every frame of a video.

In this work, we also experimented with multi-stream Convolutional Neural Network (CNN) architectures. Our multi-stream CNN architecture is biologically inspired by the human primary cortex. The human mind has always been a prime source of inspiration for various effective architectures such as Neocognitron [9] and HMAX models [10]. These models use pre-defined spatio-temporal filters in the first layer and later combine them to form spatial (ventral-like) and temporal (dorsal-like) recognition systems. Similarly in multi-stream networks, each individual slice of a convolution layer is dedicated to one type of data representation and passed concurrently with all other representations.

From the experimental perspective, we are interested in answering the following questions: 1) Does the fusion of multiple color spaces perform better than a single color space?; 2) How can one process less amount of data while maintaining model performance?; and 3) What is the best combination of color spaces and optical flow for better activity recognition? We address these questions by utilizing deep learning techniques for large scale video classification. Our work does not focus on competing state-of-the-art accuracy rather we are interested in evaluating the architectural performance while combining different color spaces over key-frame based video frame selection. We extended the two stream CNN implementation proposed by Simonyan and Zisserman [7] to a multi-stream architecture. Streams are categorized into color streams and temporal streams, where color streams are further divided based on color spaces. The color streams use RGB and YCbCr color spaces. YCbCr color space has been extremely useful for video/image compression techniques. In the first spatial stream, we process the luma and chroma components of the key frames. Chroma components are optionally downsampled and integrated in the network at a later stage. The architecture is defined such that both luma and chroma components train a layer of convolutional filters together as a concatenated array before the fully connected layers. Apart from color information, optical flow data is used to represent motion. Optical flow has been a widely accepted representation of motion, our multi-stream architecture contains dedicated stream for optical flow data.

The main contributions of this paper are to understand the individual and combinational contributions of different video data representations while efficiently processing only around key frames instead of randomly or sequentially selected frames. We evaluated our methods on various CNN architectures to account for color, object and motion contributions from video data.

The rest of the paper is organized as follows. After the introduction, Section 2 overviews the related work in activity recognition, various deep learning approaches for motion estimation and video analysis. Section 3 introduces our salient activity recognition framework and multi-stream architectures with multiple combinations of different data representations. Section 4 presents the experimental results, Section 5 contains experimental findings and analysis of potential improvement and Section 6 contains concluding remarks.

II. Related Work

Video classification has been a longstanding research topic in the multimedia processing and computer vision fields. Efficient and accurate classification performance relies on the extraction of salient video features. Conventional methods for video classification [11] involve generation of video descriptors that encode both spatial and motion variance information into hand-crafted features such as Histogram of Oriented Gradients (HOG) [2], Histogram of Optical Flow (HOF) [12], and spatio-temporal interest points [13]. These features are then encoded as a global representation through bag of words [13] or fisher vector based encoding [14] and then passed to a classifier [15].

Video classification research has been influenced by recent trends in machine learning which utilize deep architectures for image classification [3], object detection [16] [17], scene labeling [18] and action classification [19] [8]. A common workflow in these works is to use group of frames as the input to the network, whereby the model is expected to learn spatial, color, spatio-temporal and motion characteristics. [20] introduces 3D CNN models for action recognition with temporal data. This extracts features from both spatial and temporal domain by performing 3D convolution. Karpathy [8] compared different fusion combinations for temporal data on very large datasets. They [8] state that stacking of frames over time gives similar results as treating them individually, indicating that spatial and temporal data may need to be treated separately. Unsupervised learning methods such as Convolutional Gated Restricted Boltzmann Machines [21] and Independent Subspace Analysis [22] have also shown promising results. Recent work by Simonyan and Zisserman [7] decomposes video into spatial and temporal components. The spatial component works with scene and object information in each frame. The temporal component signifies motion across frames. Ng et al. [23] evaluated the effect of different color space representations on the classification of gender. Interestingly, they presented that gray scale performed better than RGB and YCbCr space.

III. Methodology

In this section, we describe our learning model for large scale video classification including pre-processing, multi-stream CNN, key frame selection and the training procedure in detail. At test time, only the key frames of a test video are passed through the CNN and classified into one of the

activities. This helps to not only show that key frames are capturing the important parts of the video but also that the testing is faster as compared to passing all frames though the CNN.

A. Color Stream

Video data can be naturally decomposed into spatial and temporal information. The most common spatial representation of video frames is the RGB (3-channel) data. In this study, we compare it with the Luminance and Chrominance color space and their combinations thereof. YCbCr space separates the color into the luminance channel (Y), the blue-difference channel (Cb), and the red-difference channel (Cr).

The spatial representation of the activity contains global scene and object attributes such as shape, color and texture. The CNN filters in the color stream learn the color and edge features from the scene. The human visual system has lower acuity for color differences than luminance detail. Image and video compression techniques take advantage of this phenomenon, where the conversion of RGB primaries to luma and chroma allow for chroma sub-sampling. We use this concept while formulating our multi-stream CNN architectures. We sub-sample the chrominance channels by factors of 4 and 16 to test the contribution of color to the framework.

B. Motion Stream

Motion is an intrinsic property of a video that describes an action by a sequence of frames, where the optical flow could depict the motion of temporal change. We use an OpenCV implementation [24] of optical flow to estimate motion in a video. Similar to [25], we stack the optical flow in the x- and y- directions. We scale these by a factor of 16 and stack the magnitude as the third channel.

C. Key Frame Extraction

We use the optical flow displacement fields between consecutive frames and detect motion stillness to identify key frames. A hierarchical time constraint ensures that fast movement activities are not omitted. The first step in identifying key frames is the calculation of optical flow for the entire video and estimate the magnitude of motion using a motion metric as a function of time [26]. The function is calculated by aggregating the optical flow in the horizontal and vertical direction over all the pixels in each frame. This is represented in (1).

$$M(t) = \sum_i \sum_j \mid OF_x(i,j,t) \mid + \mid OF_y(i,j,t) \mid \qquad (1)$$

where $OF_x(i,j,t)$ is the x component of optical flow at pixel i,j in frame t, and similarly for y component. As optical flow tracks all points over time, the sum is an estimation of the amount of motion between frames. The gradient of this function is the change of motion between consecutive frames and hence the local minimas and maximas would represent stillness or important activities between sequences of actions. An example of this gradient change from a UCF-101 [27] video is shown in Figure 2. For capturing fast moving activities, a temporal constraint between two selected frames is applied during selection [28]. Frames are dynamically selected depending on the content of the video. Hence, complex activities or events would have more key frames, whereas simpler ones may have less.

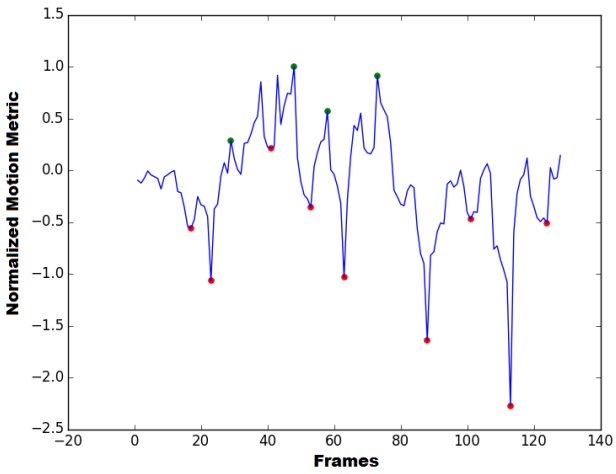

Fig. 2. Example of selected key frames for a boxing video from UCF-101 dataset. The red dots are local minima while the green are local maxima.

Fig. 3. Example of selected key frames for a boxing video from UCF-101 dataset.

D. Early Fusion

We consider a video as a bag of short clips, where each clip is a collection of 10 frames. While preparing the input data for the CNN models, we stacked these 10 frames, the RGB/Y/CbCr to generate 30/10/20 channels, respectively. We select a key frame, then define a clip as the four preceding frames and five proceeding frames relative to the key frame.

The early fusion technique combines the entire 10 frame time window of the filters from the first convolution layer of the CNN. We adapt the time constraint by modifying the dimension of the these filters as $F \times F \times CT$, where F is the filter dimension, T is the time window (10) and C is the number of channels in the input (3 for RGB). This is an

alternate representation from the more common 4-dimensional convolution.

E. Multi-Stream Architecture

We propose a multi-stream architecture which combines the color-spatial and motion channels. Figure 4 illustrates an example of the multi-stream architecture. The individual streams have multi-channel inputs, both in terms of color channels and time windows. We let the first three convolutional layers learn independent features but share the filters for the last two layers.

F. Training

Our baseline architecture is similar to [3], but accepts inputs with multiple stacked frames. Consequently, our CNN models accept data, which has temporal information stored in the third dimension. For example, the luminance stream accepts input as a short clip of dimensions 224×224×10. The architecture can be represented as C(64,11,4)-BN-P-C(192,5,1)-BN-P-C(384,3,1)-BN-C(256,3,1)-BN-P-FC(4096)-FC(4096), where C(d,f,s) indicates a convolution layer with d number of filters of size $f \times f$ with stride of s. P signifies max pooling layer with 3×3 region and stride of 2. BN denotes batch normalization [29] layers. The learning rate was initialized at 0.001 and adaptively gets updated based on the loss per mini batch. The momentum and weight decay were 0.9 and $5e^{-4}$, respectively.

The native resolution of the videos was 320 × 240. Each frame was center cropped to 240 × 240, then resized to 224 × 224. Each sample was normalized by mean subtraction and divided by standard deviation across all channels.

IV. Results

A. Dataset

Experiments were performed on UCF-101 [27], one of the largest annotated video datasets with 101 different human actions. It contains 13K videos, comprising 27 hours of video data. The dataset contains realistic videos with natural variance in camera motion, object appearance, pose and object scale. It is a challenging dataset composed of unconstrained videos downloaded from YouTube which incorporate real world challenges such as poor lighting, cluttered background and severe camera motion. We used UCF-101 split-1 to validate our methodologies. Experiments deal with two classes of data representation; key frame data and sequential data. Key frame data includes clips extracted around key frames where sequential data signifies 12 clips extracted around 12 equally spaced frames across the video. 12 equally spaced frames were chosen as that was the average number of key frames extracted per video. We will use the terms key frame data and sequential data to represent the extraction of frame locations.

B. Evaluation

The model generates a predicted activity at each selected frame location, and voting amongst all locations in a video clip is used for video level accuracy. Although transfer learning boosted RGB and optical flow data performance, no high performing YCbCr transfer learning models were available. To ensure fair comparison among methods, all model results were initialized with random weights.

The first set of experiments quantify the value of using key frames. Table I shows that key frame data consistently outperforms the sequential data representation. Table II, which uses two stream architectures, similarly shows that key frame data is able to understand video content more accurately than sequential data. These experiments validate that there is significant informative motion and spatial information available around key frames.

TABLE I
SINGLE STREAM EXPERIMENT RESULTS.

	Data	Sequential	Key Frames
		Accuracy	
1	Y-only	39.72 %	42.04%
2	CbCr-only	35.04 %	35.04 %
3	RGB-only	38.44 %	46.04 %
4	OF-only	42.90 %	46.54 %

Table I shows that optical flow data is perhaps the single best predictor. Optical flow data contains very rich information for motion estimation, which is important for activity recognition. Parameter training with three channel optical flow representation required less computational resources because it represents information of 10 video frames with only 224×224×3 size of data. The ten frame stacked RGB-only model (10× the 1^{st} layer memory of OF-only) resulted in similar accuracy, but took three more days to train than the optical flow model. The luminance only and chrominance only models gave less promising results.

TABLE II
TWO STREAM EXPERIMENT RESULTS.

	Data	Sequential	Key Frames
		Accuracy	
1	Y+CbCr	45.30 %	47.13 %
2	Y+CbCr/4	-	43.40 %
3	Y+CbCr/16	-	42.77 %
4	Y+OF	41.68 %	44.24 %

TABLE III
MULTI-STREAM EXPERIMENT RESULTS. *EPOCHS = 15

	Data	Sequential	Key Frames
		Accuracy	
1	Y+CbCr+OF	48.13 %	49.23 %
2	Y+OF+RGB	45.33* %	46.46* %

Table II demonstrates multiple channel results. The fusion of luminance data with chrominance data is the best performing

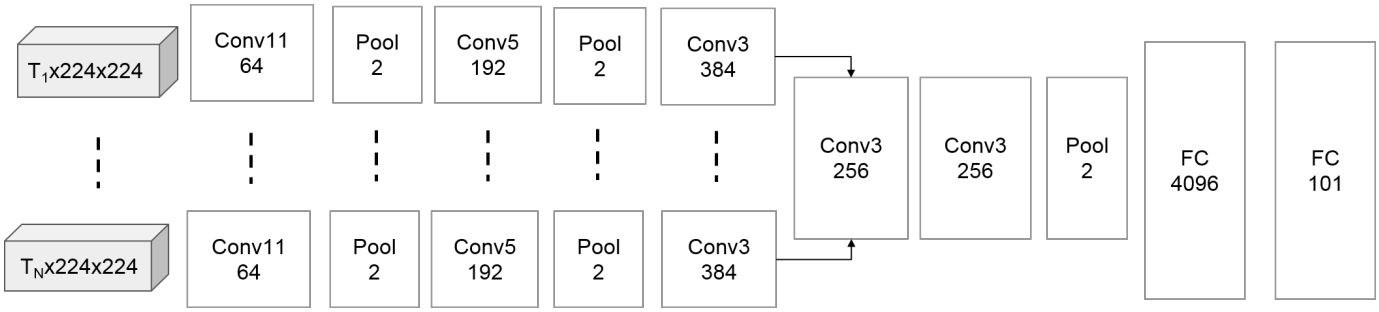

Fig. 4. Multi-stream architecture overview. The inputs are 224 x 224 images with different number of channels. T_i is the temporal input channels.

dual stream model. CNNs can take weeks to learn over large datasets, even when using optimized GPU implementations. One particular factor strongly correlated with training time is pixel resolution. It has long been known that humans see high resolution luminance and low resolution chrominance. To determine if CNNs can learn with low resolution chrominance, the chrominance channels were subsampled by a factor of four and sixteen. As shown in the Table II, lowering chrominance resolution did not have a big impact on accuracy. Despite this small change in accuracy, the training time was reduced dramatically.

To further understand what combination of channel representations will provide best activity understanding, Table III contrasts three stream CNN architectures. Once again, the usage of YCbCr is superior to RGB, with a 47.73% top-1 accuracy on UCF-101.

C. Visualization of Temporal Convolution Filters

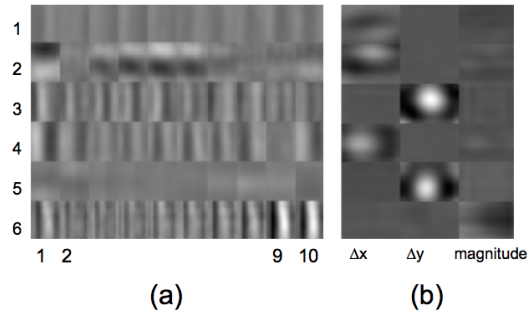

Fig. 5. Visualization of six 11 x 11 convolutional filters from first layer. (a) shows luminance filters and (b) shows filters from optical flow. Rows correspond to separate filters with separate channels (10 for luminance and 3 for optical flow).

Figure 5 illustrates examples of trained (11×11) filters in the first convolutional layer. The luminance filters are 10 channels and the optical flow filters are x-, y- and magnitude. It can be observed that the filters capture the motion change over the x- and y- directions. These filters allow the network to precisely detect local motion direction and velocity.

V. Discussion

Deep learning models contain a large number of parameters, and as a result are prone to overfitting. A dropout [30] ratio of 0.5 was used in all experiments to reduce the impact of overfitting. Trying a higher dropout ratio may help the model to generalize well, as our learning curves indicate the UCF 101 data may be overfitting. We used batch normalization [31], which has shown to train large networks fast with higher accuracy. As the data flows through the deep network, the weights and parameters adjust the data to minimize internal covariance shift between layers. Batch normalization reduces this internal covariance shift by normalizing the data at every mini-batch, giving a boost in training accuracy, especially on large datasets.

For multi-stream experiments, we experimented with transfer learning and fine tuned the last few layers of the network. Unfortunately, there were no pretrained models for YCbCr data. A color conversion of pretrained RGB filters to YCbCr filters yielded low YCbCr accuracy. As a result, we trained all models from scratch for a fair comparison.

We also experimented with Motion History Images (MHI) in place of optical flow. A MHI template collapses motion information into a single gray scale frame, where intensity of a pixel is directly related to recent pixel motion. Single stream MHI resulted 26.7 % accuracy. This lower accuracy might be improved by changing the fixed time parameter during the estimation of motion images; we used ten frames to generate one motion image.

Our main goal was to experiment with different fusion techniques and key frames, so we did not apply any data augmentation. All results in Tables I through III, except for the Y+OF+RGB, trained for 30 epochs so that we can compare performance on the same scale. The Y+OF+RGB model was trained for 15 epochs. We did observe the trend that running with higher number of epochs increased the accuracy significantly. For example, the single stream OF-only with key frames in Table I jumped to 57.8% after 117 epochs.

VI. Conclusion

We propose a novel approach to fuse color spaces and optical flow information in a single convolutional neural network architecture for state-of-the-art activity recognition. This study shows that color and motion cues are necessary and

their combination is preferred for accurate action detection. We studied the performance of key frames over sequentially selected video clips for large scale human activity classification. We experimentally support that smartly selected key frames add valuable data to CNNs and hence perform better than conventional sequential or randomly selected clips. Using key frames not only provides better results but can significantly reduce the amount of data being processed. To further reduce computational resources, multi-stream experiments advocate that lowering down the resolution of chrominance data stream does not harm performance significantly. Our results indicate that passing optical flow and YCbCr data into our multi-stream architecture at key frame locations of videos offer comprehensive feature learning, which may lead to better understanding of human activity.

REFERENCES

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[5] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 427–436.

[6] S. Chaabouni, J. Benois-Pineau, O. Hadar, and C. B. Amar, "Deep learning for saliency prediction in natural video," 2016.

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

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[9] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[10] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[11] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1996–2003.

[12] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1932–1939.

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

[14] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 3551–3558.

[15] H. Wang, A. Klser, C. Schmid, and C. L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 3169–3176.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[18] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1915–1929, 2013.

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

[20] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 221–231, 2013.

[21] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 140–153.

[22] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3361–3368.

[23] C.-B. Ng, Y.-H. Tay, and B.-M. Goi, "Comparing image representations for training a convolutional neural network to classify gender," in *Artificial Intelligence, Modelling and Simulation (AIMS), 2013 1st International Conference on*. IEEE, 2013, pp. 29–33.

[24] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image analysis*. Springer, 2003, pp. 363–370.

[25] G. Gkioxari and J. Malik, "Finding action tubes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 759–768.

[26] W. Wolf, "Key frame selection by motion analysis," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 1228–1231.

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

[28] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," in *Multimedia Computing and Systems, 1999. IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 756–761.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.