Video Event Categorization (TPA 11)

Y S S V Sasi Kiran, *CS13B055*

Abstract

The state of the art deep neural networks and several other feature extractors like Improved Dense Trajectories are a kind of differential operator and hence act as high pass filters. This results in the extracted features getting biased to the higher frequency information. Multi Skip Feature Stacking (MIFS) is the state-of-the-art feature extraction method which takes this into account. The features extracted using MIFS are high-dimensional and hence several dimensionality reduction methods are tried out. Additionally, a deep network is trained on these extracted features to give very good results on the task of action recognition in videos.

I. Introduction

ACTION Recognition in videos has recently become very popular among the research community. It has several application ranging from self driving cars to Virtual Reality. The task of action recognition in videos has been one of the most difficult tasks in the field of computer vision mainly due to the non-availability of large video datasets compared to images. Several approaches have been tried in this field starting from handcrafted features to very deep convolutional neural networks.

Prior to the advent of deep learning, several handcrafted features like Scale invariant feature transform (SIFT) [13], Space Time interest points (STIP) [12], Dense Trajectories [19] have been employed to the task of action recognition in videos. Improved Dense Trajectories [20] has been the state-of-the-art feature extractor for videos prior to the introduction of MIFS [11] .

With the advent of GPUs and large data availability, deep learning has recently been in the limelight. It has become popular in the Computer Vision community after the winning entry of Convolutional Neural Networks for the IMAGENET-2012 challenge [9]. It has been applied to various fields like Object Detection [15] [18], Object Recognition [4] [3] [14] , Human Pose Estimation [21], etc.

Convolutional Neural Networks have been first applied to this task by Karpathy et.al. [8] . With the inspiration from Human Biology, Simoyan and Zisserman introduced two-stream architecture [16] for the task of action recognition in videos. Several improvements have been made to this two-stream architecture [2] and the current state-of-the-art on the action recognition uses residual connections and two-stream architecture [1] .

We approach the task of action recognition with a mixture of both handcrafted features and deep learning extracted features. We use MIFS for handcrafted features and make several experiments on the features extracted. The following sections first describe MIFS and then gives details about each of the further steps used till the final classification

A. Dense Trajectories

Scale invariant feature transform (SIFT) [13], Space Time interest points (STIP) [12] can be used for extracting interest points in the frames and the feature vectors corresponding to these points can be extracted at each frame. But these do not take several factors like camera motion and motion of objects in the scene into account while calculating these interest points. Dense Trajectories [19] overcomes this problem by tracking the interest points across time using trajectories extracted from optical flow information at each frame.

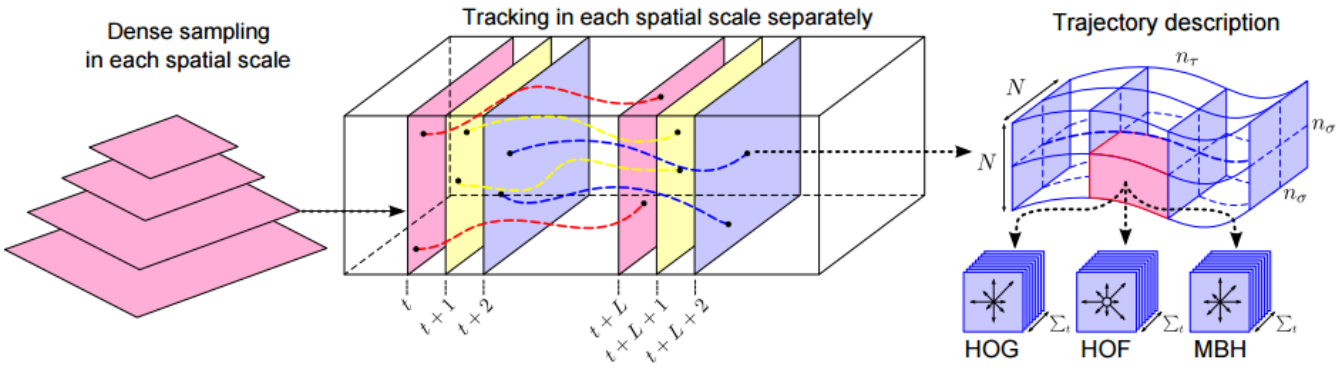

Fig. 1. The above figure shows the feature extraction using dense trajectories. Left Image shows the extraction of features at different spatial scales. Middle Image shows the tracking of points over L frames. Right image shows the extraction of features by dividing the neighborhood around the trajectory into spatio-temporal grid.

From Figure 1 we observe the process of extraction of Dense Trajectory features. Histogram of Gradient (HOG), Histogram of Optical Flow (HOF) are used along with Motion Boundary Histogram (MBH) for feature extraction at each trajectory point. MBH is computed similar to HOG and HOF, but uses the derivative of optical flow in the x and y direction to compute the descriptor.

B. Multi Skip Feature Stacking

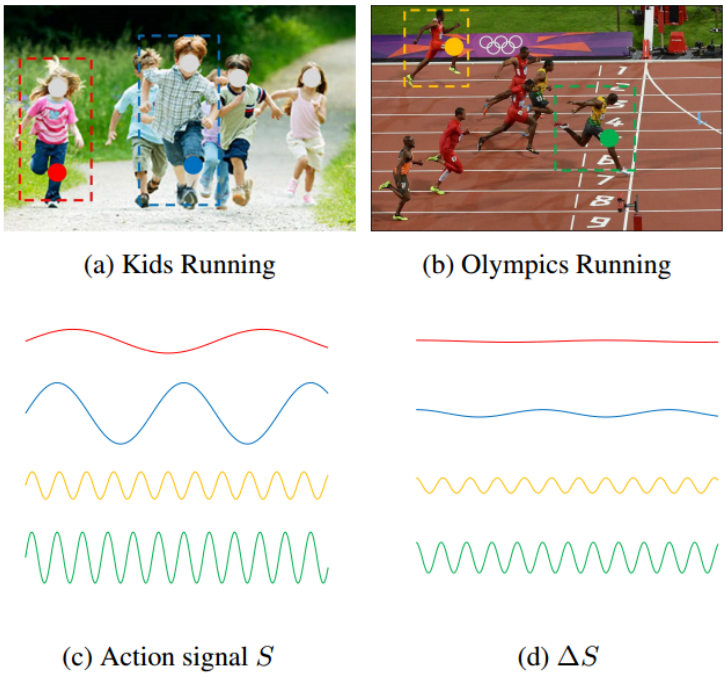

Fig. 2. The figure shows an example of the same action signal having different frequencies in different scenarios. (c) marks the variation of signal across time of a single point in the images (a) and (b). Image (d) shows the result of differentiation of signals in (c). This image is taken from MIFS paper.

We observe from Figure 2 that different videos of the same class have different frequencies of the same action. Conventional feature extractors like IDT do not take this into account and extract features by sampling at fixed frequency. Multi Skip Feature Stacking (MIFS) addresses this problem by extracting features at different frequencies in the video and stacking them to form the descriptor for the video.

MIFS features can be extracted using any feature extractor for videos. Here, we use IDT as feature extractor. The only difference in MIFS from the traditional IDT is that it uses different sampling interval at each spatial scale of the video. These extracted descriptors are L2-normalized and stacked together as the output descriptors of MIFS.

C. Restricted Boltzmann Machines

Restricted Boltzmann Machines is a generative model of neural networks where the model tries to learn the probability representation of the input. Restricted Boltzmann machines can be employed for several tasks like Data Augmentation, Topic Modeling, Dimensionality Reduction, etc. A typical RBM consists of visible and hidden units which are connected in a bipartite fashion similar to fully connected neural network. This is illustrated in Figure 3

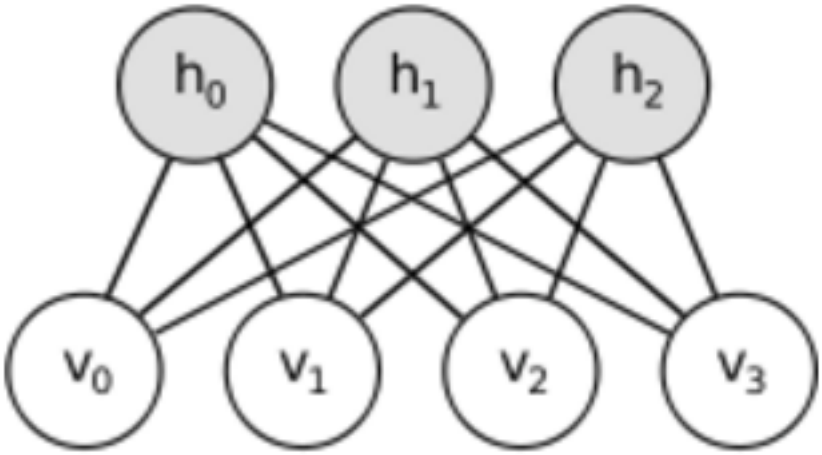

Fig. 3. The above figure shows the structure of a simple RBM where the connections are only present between visible and hidden units.

The energy function of a typical RBM is given by $E(v,h) = -\sum c_i v_i - \sum b_j h_j - \sum W_{i,j} v_i h_j$

RBMs are trained using Contrastive Divergence algorithm. The input to the RBM is the unlabeled training data and at each stage of the training the parameters of the model are updated to maximize the likelihood of data. Then the hidden representation can be computed for the input training data which can be used for classification task as reduced dimensional data.

D. Principal Component Analysis

Principal Component Analysis (PCA) is one of the primary tools used for the dimensionality reduction of data. PCA tries to project the data along new directions such that the co-variance of the new features is minimized and its variance is maximized. The best k-rank approximation of SVD of the data matrix which minimizes the reconstruction error in the data is also given by PCA.

Since the data extracted by MIFS is of huge dimensionality, we can use PCA for dimensionality reduction. But due to very large number of features, PCA by itself can not be applied directly on the data due to the in-feasibility of the computation of scatter matrix. Hence, Randomized PCA [6] is used for dimensionality reduction which tries to approximate PCA.

II. ALGORITHMIC DESCRIPTION

We first use MIFS to extract the features of the given input video. MIFS features are computed using the following method. We first extract IDT using a tracking of 15 frames (L=15) and camera motion stabilization of the video. Each IDT descriptor consists of trajectory information as well as HOG, HOF and MBHx and MBHy descriptors. We additionally concatenate the

three dimensional normalized location information of the interest points.

We stack the descriptors extracted at different scales and consider them as the raw feature descriptors. These descriptors are encoded using Fisher Vector Encoding using a Gaussian mixture model of 256 Gaussians. Then these encoded vectors are normalized using L2-normalization. We then perform several experiments on the features extracted using the above method

We first explore the basic Linear SVM classifier applied directly on the extracted features. We then look at several variants of kernels used while training SVMs like polynomial, rbf kernels. We then consider the influence of dimensionality reduction on the extracted feature vectors. Randomized PCA and RBMs are the dimensionality reduction techniques explored. Finally, we train Deep Neural Networks on the extracted features. This deep network is trained using 3 and 4 hidden layers. Additionally, Batch Normalization [7] is employed to make the model easily trainable.

All the results are reported on two standard action recognition data-sets UCF-101 and HMDB-51. These data-sets consists of three splits of training and test data. Mean Accuracy (MAcc) over the three splits is used as the evaluation criteria. We compare the results at each step with the results obtained using traditional MIFS. We finally compare the best results obtained with best results of traditional results of MIFS as well as current state-of-the-art.

III. OUTPUT

We first run Linear SVM with penalty term (C=100) and get the following results on direct MIFS features as shown in Table I. One-vs-rest classification is used in Linear SVM for the multi-class problem

TABLE I

THE ABOVE TABLE SHOWS THE ACCURACY FOR EACH SPLIT AND MEAN ACCURACY (MACC) FOR UCF-101 AND HMDB-51 DATASETS ON APPLYING LINEAR SVM WITH C=100.

C=100	Split-1	Split-2	Split-3	MAcc(%)
UCF-101	64.24	64.13	64.90	64.423
HMDB-51	87.04	87.36	87.15	87.183

We then explore the influence of cost term or slack term (C) on the SVM. The results are shown in Table II. We observe that $C = 100$ gives the best results in both UCF-101 and HMDB-51 datasets and hence set this value of C for all the further experiments involving SVMs.

TABLE II

THE ABOVE TABLE SHOWS THE VARIATION OF MEAN ACCURACY (MACC) FOR UCF-101 AND HMDB-51 DATASETS ON APPLYING LINEAR SVM WITH DIFFERENT VALUES OF C.

MAcc(%)	C=50	C=100	C=150
HMDB-51	64.06	64.42	62.83
UCF-101	86.67	87.18	84.15

We then explore possible variations of kernels for training SVMs. We explore polynomial kernel and rbf kernel. The best results of polynomial kernel are obtained at $degree(d) = 3$ and $gamma(\gamma) = 0.002$. Similarly, the best results are obtained for rbf kernel at $gamma(\gamma) = 0.002$. These results are shown in Table III. We observe that among all these kernels, the linear kernel gives the best results. The Polynomial and RBF kernels are trained in a one-vs-one fashion for multi-class classification.

TABLE III

THE ABOVE TABLE SHOWS THE VARIATION OF MEAN ACCURACY (MACC) FOR UCF-101 AND HMDB-51 DATASETS WITH DIFFERENT KERNELS. THESE RESULTS ARE SHOWN FOR BEST PARAMETERS OF EACH KERNEL

MAcc(%)	Linear	Polynomial	RBF
HMDB-51	64.42	34.43	58.49
UCF-101	87.18	42.17	81.36

We then experiment with various dimensionality reduction techniques like randomized PCA and RBMs on the MIFS features. We reduce the dimensions of the data to 500 dimensional data. For training RBMs, we first binarize the inputs. We first find the average feature value and then binarize the values by considering all the values higher than average as 1 and the rest as 0. We then train the RBM for 100 epochs using Adam Optimizer and Contrastive Divergence. We then train the linear SVM with best parameters on these reduced features and observe the results in Table IV.

TABLE IV

THE ABOVE TABLE SHOWS THE VARIATION OF MEAN ACCURACY (MACC) FOR UCF-101 AND HMDB-51 DATASETS WHEN LINEAR SVM IS TRAINED ON DIMENSIONALLY REDUCED DATA.

MAcc(%)	PCA(N=500)	RBM(N=500)
HMDB-51	54.68	59.23
UCF-101	77.81	79.13

We then train deep neural network on the MIFS features for classification. We train different configuration of networks. For training we use Adam Optimizer on training data. We use a batch size of 100 on training data. We use Xavier Initialization

[5] to initialize the parameters of the network. The different types of networks are shown in Table VI, VII and the parameters for training are shown in Table V.

TABLE V

THE ABOVE TABLE SHOWS THE PARAMETER USED FOR TRAINING THE NEURAL NETWORK.

Parameters	Values Used
Loss Function	Cross Entropy
Optimizer	Adam Optimizer
Batch Size	100
Initialization	Xavier
Learning Rate	0.002
Max Epochs	100

TABLE VI

THE ABOVE TABLE SHOWS THE CONFIGURATION FOR NETWORK-1.

Layer	Paramters
FC-1	116736×2048
FC-2	2048×1024
Batch Norm	-
FC-3	1024×512
Drop-out	$prob = 0.5$
Softmax	$512 \times C$

TABLE VII

THE ABOVE TABLE SHOWS THE CONFIGURATION FOR NETWORK-2.

Layer	Paramters
FC-1	116736×1024
FC-2	2048×1024
Batch Norm	-
Softmax	$1024 \times C$

We train both the networks and observe the mean accuracy (MAcc) as shown in Table VIII. We observe that the second network performs better compared to first network. This is probably because of the depth of first network making it slightly harder to train on small data.

TABLE VIII

THE ABOVE TABLE SHOWS THE VARIATION IN MEAN ACCURACIES ON UCF-101 AND HMDB-51 DATASETS WHEN DIFFERENT NETWORKS ARE TRAINED ON THE DATA.

MAcc(%)	Network-1	Network-2
HMDB-51	62.58	65.49
UCF-101	86.33	89.85

A. Observation

The overall comparison of the best methods are done with state-of-the-art in Action Recognition. The results are shown in Table IX.

TABLE IX

THE ABOVE TABLE SHOWS THE COMPARISON OF MEAN ACCURACY OF DIFFERENT METHODS WITH THE CURRENT STATE-OF-THE-ART.

MAcc(%)	HMDB-51	UCF-101
MIFS Paper	65.1	89.1
Linear SVM	64.42	87.18
Network-2	65.49	89.85
ResNet on 2-stream	70.3	94.6

We observe that training a neural network on MIFS features achieves very good results. The state-of-the-art is achieved by combining ResNet base two-stream fusion features with IDT features.

IV. RESULTS

Some videos where the model performs accurately are shown below in Figure 4.

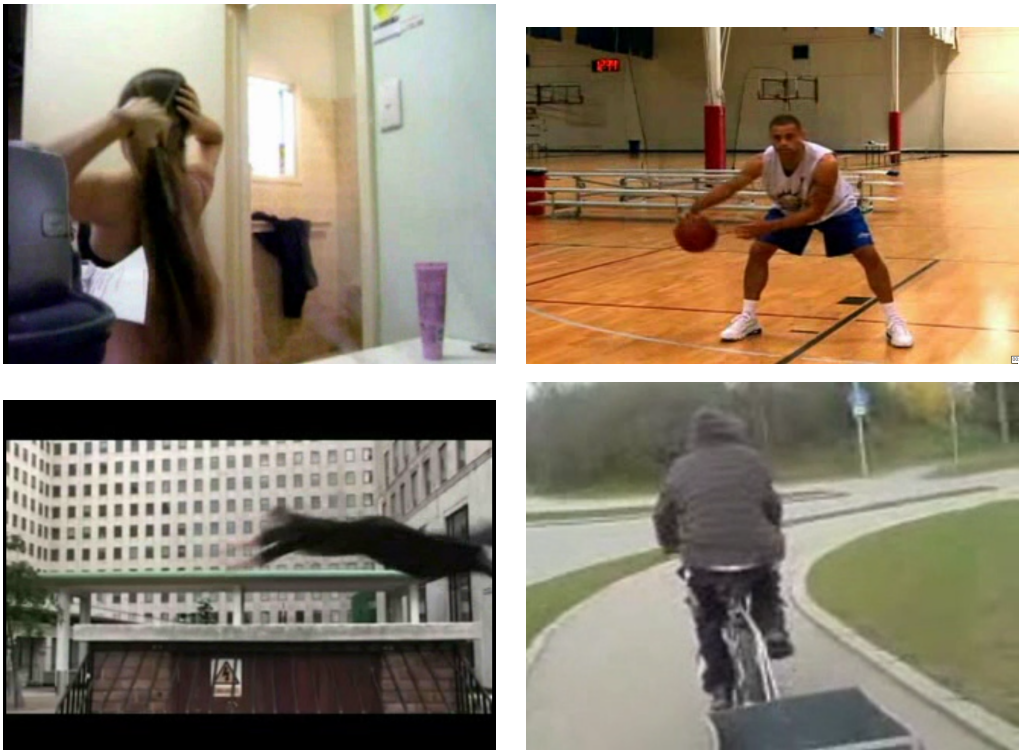

Fig. 4. The above images are some images where the model performs accurately. From top-left to bottom right combing, dribbling, jumping, riding bike respectively. The above results are produced by trained model on HMDB51 images.

V. Conclusion

Hence, we observe that training a neural network on the features extracted from MIFS increases the mean accuracy. The best accuracy obtained on UCF-101 is 89.85% and on HMDB-51 is 65.49%. As done in the state-of-the-art the future scope of this project is to combine these vectors with two-stream extracted vectors.

References

[1] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes, *Spatiotemporal residual networks for video action recognition*, CoRR **abs/1611.02155** (2016).

[2] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, *Convolutional two-stream network fusion for video action recognition*, CoRR **abs/1604.06573** (2016).

[3] Ross Girshick, *Fast R-CNN*, Proceedings of the International Conference on Computer Vision (ICCV), 2015.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (Washington, DC, USA), CVPR '14, IEEE Computer Society, 2014, pp. 580–587.

[5] Xavier Glorot and Yoshua Bengio, *Understanding the difficulty of training deep feedforward neural networks.*, AISTATS (Yee Whye Teh and D. Mike Titterington, eds.), JMLR Proceedings, vol. 9, JMLR.org, 2010, pp. 249–256.

[6] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, ArXiv e-prints (2009).

[7] Sergey Ioffe and Christian Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, CoRR **abs/1502.03167** (2015).

[8] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, *Large-scale video classification with convolutional neural networks*, Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (Washington, DC, USA), CVPR '14, IEEE Computer Society, 2014, pp. 1725–1732.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), Curran Associates, Inc., 2012, pp. 1097–1105.

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, *HMDB: a large video database for human motion recognition*, Proceedings of the International Conference on Computer Vision (ICCV), 2011.

[11] Zhen-Zhong Lan, Ming Lin, Xuanchong Li, Alexander G. Hauptmann, and Bhiksha Raj, *Beyond gaussian pyramid: Multi-skip feature stacking for action recognition.*, CVPR, IEEE Computer Society, 2015, pp. 204–212.

[12] Ivan Laptev, *On space-time interest points*, Int. J. Comput. Vision **64** (2005), no. 2-3, 107–123.

[13] David G. Lowe, *Distinctive image features from scale-invariant keypoints*, Int. J. Comput. Vision **60** (2004), no. 2, 91–110.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, Advances in Neural Information Processing Systems 28 (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), Curran Associates, Inc., 2015, pp. 91–99.

[15] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, CoRR **abs/1409.1556** (2014).

[16] Karen Simonyan and Andrew Zisserman, *Two-stream convolutional networks for action recognition in videos*, Proceedings of the 27th International Conference on Neural Information Processing Systems (Cambridge, MA, USA), NIPS'14, MIT Press, 2014, pp. 568–576.

[17] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, *Ucf101: A dataset of 101 human actions classes from videos in the wild*, CoRR **abs/1212.0402** (2012).

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, *Going deeper with convolutions*, Computer Vision and Pattern Recognition (CVPR), 2015.

[19] Heng Wang, A. Klaser, C. Schmid, and Cheng-Lin Liu, *Action recognition by dense trajectories*, Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (Washington, DC, USA), CVPR '11, IEEE Computer Society, 2011, pp. 3169–3176.

[20] Heng Wang and Cordelia Schmid, *Action recognition with improved trajectories*, Proceedings of the 2013 IEEE International Conference on Computer Vision (Washington, DC, USA), ICCV '13, IEEE Computer Society, 2013, pp. 3551–3558.

[21] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh, *Convolutional pose machines.*, CoRR **abs/1602.00134** (2016).