# EFFICIENT OBJECT FEATURE SELECTION FOR ACTION RECOGNITION

*Tianyi Zhang*[⋆] *Yu Zhang*[†] *Jianfei Cai*[⋆] *Alex C. Kot*[⋆]

[⋆] Nanyang Technological University, Singapore
[†]Bioinformatics Institute, A*STAR, Singapore

## ABSTRACT

Currently most action recognition or video classification tasks highly rely on the motion features such as state-of-the-art Improved Dense Trajectory (IDT) features. Despite the huge success, IDT features lack of rich static object-level information. In this paper, we make use of the object-level features for action recognition tasks. For efficiently and effectively processing large-scale video data, we propose a two-layer feature selection framework including local object feature selection (LS) and global feature selection (GS). Both of the selection methods can improve recognition accuracy while greatly reducing the feature dimension or feature processing complexity. Experimental results show that the selected object-level features contain complimentary information to IDT features and the combination with IDT features can further improve the recognition accuracy significantly.

***Index Terms***— Action Recognition, Feature Selection

## 1. INTRODUCTION

Action recognition is an important task in video analysis or video surveillance applications. In recent years, many large-scale and practical datasets such as UCF101 [1], HMDB51 [2], and THUMOS [3] have been proposed. For large-scale action recognition, one major challenge is how to design representative and discriminative features that can characterize human actions. Many powerful features have been developed such as Improved Dense Trajectory (IDT) [4], Hierarchical Independent Subspace Analysis (Hierarchical ISA) [5], etc.

Among the hand-crafted features, the Improved Dense Trajectory [4] (IDT) feature is the state-of-the-art. Many methods adopt IDT as the basic feature and add variations to achieve better results, such as stacked Fisher Vector [6] and motion words [7]. IDT is generated as follows: it first removes the camera motion and estimates the dense wrapped optical flow. The wrapped optical flow is linked into trajectories and the approximately static trajectories are deleted. Local features are extracted along trajectories and each IDT feature is represented as the concatenation of descriptors, e.g., trajectory shape, Histogram of Gradient (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH), where trajectory shape, HOF and MBH can be interpreted as long-term, short-term zero order and short-term first-order motion information respectively.

Despite the great success, IDT also has some limitations. First, IDT lacks strong static information. Unlike the motion information which is richly expressed by three descriptors (trajectory shape, HOF and MBH) from different perspectives, the only static feature is HOG, which is a relatively weak feature especially compared with the fast developed CNN features for image classification. Second, IDT is a kind of local feature along each point trajectory, which lacks mid-level or object-level information. This motivates us to investigate how static, object-level information could help the action recognition task and how it could be complimentary to the motion information.

There are some recent developments in the deep learning models for action recognition. Karpathy et al. [8] utilized 3-D convolution to incorporate motion information into training CNN models for videos. Currently the most successful CNN model for action recognition is the two-stream CNN model [9]. Although the layers of the two-stream model contain mid-level information, it requires large video datasets for training and the training process is extremely time-consuming. Some video classification tasks such as Event Detection [10] applied pre-trained CNN model on each video frame to improve time efficiency.

In this paper, we also use the publicly available pre-trained image CNN model to extract object-level features for action recognition task. Moreover, for efficiently and effectively processing large-scale video data and high-dimension features, we propose a two-layer feature selection framework including a local object feature selection (LS) and a global feature selection (GS) to remove noise and reduce feature dimensions. Experimental results show that the selected object-level features contain complimentary information to IDT features and the combination with IDT features can further improve the recognition accuracy significantly.

The main contributions of this work are listed as follows: **(a)** We make use of object-level features and show its effectiveness for action recognition task. **(b)** We propose local object feature selection (LS) to efficiently reduce the noisy objects. **(c)** We propose to use global feature selection (GS) to further improve the accuracy and reduce the feature dimensions of video representations.

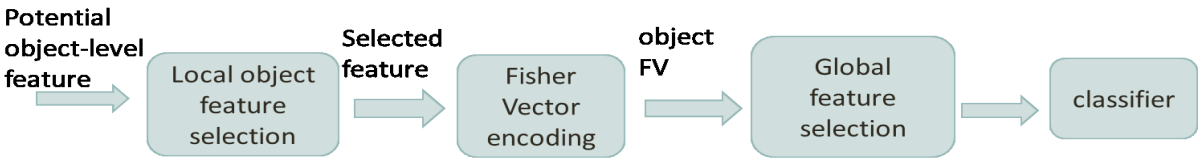



**Fig. 1**. Proposed two-level feature selection framework.

## 2. METHODS

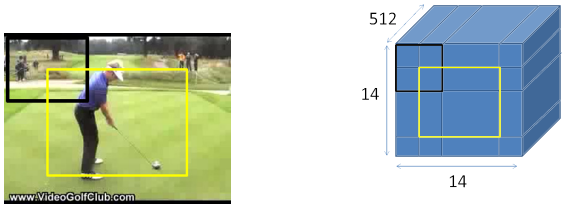
The main part of our work is shown in Fig. 1, where the input is some potential object-level features corresponding to different spatial locations at different frames. We first perform local object feature selection (LS) on them. After performing PCA to reduce the local feature dimensions and training GMM models for the local-selected features, we encode all of them into a Fisher Vector (FV) [11], which essentially converts one video into one global feature representation. Considering FV is of high dimension and some dimensions might introduce noise information, we further perform global feature selection (GS) on the Fisher Vectors and then feed them into a linear SVM classifier. In the following, we describe how to generate potential object-level features locally and how to do local object feature selection (LS) and global feature selection (GS).

### 2.1. Generate Potential Object-level Features

The state-of-the-art technique to generate potential object-level features is to apply object proposal methods such as Bing [12] and Selective Search [13] first and then apply Convolutional Neural Network (CNN) on each proposal, such as in RCNN [14]. However, such approach is too complex to be practical for video analysis, since a video contains thousands of frames and each frame could have thousands of object proposals. Thus, in this paper, we do not use any object proposal method and only apply CNN once on each image. The potential object-level features are obtained by multi-scale pooling directly on the image-level CNN feature domain. In this way, we greatly reduce the complexity.

The reason that multi-scale pooling in CNN domain can provide some potential object-level features comes from the structure of CNN. In particular, before the fully connected layers, the output of each convolution or pooling layer still keeps the original spatial information of the input image. In other words, each spatial cell in the output of the convolution or pooling layer corresponds to a region in the original image. Fig. 2 illustrates such correspondence relationship. If the layer is deep enough, the output feature of such layer represents some long-range or mid-level static features, which we call potential object-level features.

Fig. 3 shows the diagram of extracting potential object-level features. Particularly, we use the pre-trained *imagenet-vgg-verydeep-19* [15] model from MatConvNet [16]. We input each video frame into the pre-trained model and extract the output of $relu5\_4$ layer $(14 \times 14 \times 512)$. Taking multi-scale objects into consideration, we apply dense multi-scale pyramid max-pooling operations in the spatial domain $(14 \times 14)$ and obtain $(6 \times 6 \times 512)$, $(3 \times 3 \times 512)$, $(2 \times 2 \times 512)$ and $(1 \times 1 \times 512)$ feature outputs for a single frame, similar to [10]. Each 512-dimension output is considered as a potential object-level feature $f_t^j$ , which denotes the $j_{th}$ feature of the $t_{th}$ frame, and we extract totally 50 features for a single frame.




(a) input frame (b) output layer of $relu5\_4$

**Fig. 2**. Illustration of correspondence region. (a) is the input frame into the CNN model and (b) is the output feature map of some pooling layer (e.g. $relu5\_4$ layer). The pooling area on the feature map corresponds to a region in the original frame, which is illustrated by the bounding box of the same color.

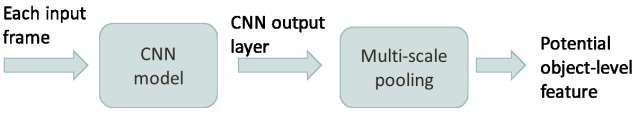



**Fig. 3**. Extraction of potential object-level feature.

### 2.2. Local Object Feature Selection

The potential object-level features extracted in Sec. 2.1 are likely to include a lot of non-object information since it comes from multi-scale dense pooling. Considering the most important static feature for human action recognition is the human body shape or the objects that the human body interacts with, here we propose to use the results of a human detector to filter the potential object-level features so as to select the features around human body.

Particularly, we first apply human detection on frame $t$ and output a set of bounding boxes $H_t = \{h_t^1, h_t^2, ..., h_t^{M_t}\}$. Then we calculate the correspondence region $b_t^j$ of each feature $f_t^j$ in the input frame $t$. We keep feature $f_t^j$ if there exists at least one human detection bounding box $h_t^k$ that satisfies $O(b_t^j, h_t^k) > \theta$, where $O$ is the intersection over union (IoU) score and $\theta$ is a pre-set threshold. In this way, we filter out many non-human related features.

### 2.3. Global Feature Selection

After we encode the selected object-level feature Fisher Vector (FV) [11] for each video, we further perform global feature selection to remove noise features and reduce feature dimensions. In particular, given the FVs and class labels of all the training videos, we calculate the importance score for each dimension of FVs using the feature selection method developed in [17]. The importance score is defined as the mutual information $I(d_i, y) = H(y) + H(d_i) - H(d_i, y)$, where $H$ is the entropy, y denotes the action label, $d_i$ is the $i_{th}$ dimension feature with 1-bit quantization and zero thresholding. Finally, we keep the subset of dimensions of FVs with higher importance scores, and use them as the final global feature representation to learn a linear SVM classifier for recognition.

## 3. EXPERIMENTS

We evaluate our methods on UCF11 [18] dataset. Following [18], we use 25 leave-one-out cross-validation and report the average accuracy over all classes. We utilize the code of [19] for human detection and keep the output boxes with the detection score larger than 0.1. We follow the methods of [20] to calculate the correspondence spatial region for each object-level feature described in Sec. 2.1. We randomly sample about half of the training features to perform PCA to keep 90% energy and compute a GMM model with 256 components. We use linear SVM of Liblinear [21] as the final classifier with its default parameters for recognition.

### 3.1. Experimental Results

**Results of local object feature selection (LS)**: Table 1 shows the effect of local object feature selection (LS). $\theta$ is the IoU threshold introduced in Sec. 2.2. $\theta = 0$ means using all the original potential object-level features without LS. LS ratio (LSR) is the ratio between the number of selected features over the original feature number. We can see that with proper parameter $\theta$, LS can help improve the recognition accuracy with much fewer object-level features (less than 20%). It demonstrates that LS is an effective way to select object-level features.

**Results of global feature selection (GS)**: Table 2 shows the accuracy of the GS with different selection ratios and different $\theta$ for LS. It can be seen that GS is effective for the original features without LS or weakly selected features ($\theta = 0.2$). But the effect of GS is negligible for the strongly selected features ($\theta = 0.35$). Such phenomena can be explained as: if the input feature is already strongly selected, the additional selection would be likely to discard some useful information instead of noise information, which results in the decrease of the accuracy.

| $\theta$ | 0 | 0.2 | 0.25 | 0.3 | 0.35 | 0.4 |
|---|---|---|---|---|---|---|
| LSR | 1.0 | 0.43 | 0.34 | 0.26 | 0.19 | 0.12 |
| accu | 85.08 | 85.41 | 86.35 | 86.84 | **87.45** | 86.10 |

**Table 1**. Local selection performance on UCF11 [18]. $\theta$ is the IoU threshold. Local selection ratio (LSR) is the proportion of selected features.

| GS ratio | LS0 | LS0.2 | LS0.35 |
|---|---|---|---|
| 100% | 85.08 | 85.41 | **87.45** |
| 75% | 85.34 | 85.32 | 87.11 |
| 50% | **85.62** | **85.83** | 87.02 |
| 25% | 83.21 | 85.01 | 85.06 |
| 12.5% | 81.93 | 83.26 | 83.55 |

**Table 2**. Global feature selection (GS) performance on UCF11 with different object-level features: non-selected, local-selected with $\theta = 0.2$ and $\theta = 0.35$.

**Results of combining IDT features**: Since our extracted features do not contain motion information, it is meaningful to compare and combine with state-of-the-art IDT features. For fair comparison, we also encode IDT features with Fisher Vector and perform GS for each video. The default dimension of IDT is 426 and PCA is applied to reduce its dimension to 200. We compute a GMM model with 256 components and use the default parameters of linear SVM of Liblinear. The first column of Table 3 shows the performance of IDT features. Comparing with the results in Table 2, we can see that the performance of our selected object-level features is slightly inferior to that of IDT features.

Next we concatenate the two Fisher Vectors of IDT features and our selected object-level features, and perform GS for the concatenated Fisher Vector. Table 3 shows the performance with different selection ratios and different $\theta$ for LS. It can be seen that such concatenation greatly improves the accuracy over that of using the IDT features alone, up to 4.39% increase in accuracy. It clearly demonstrates that the selected object-level features contain much complimentary information to the IDT features. GS can significantly boost up the performance for such concatenation cases. This suggests that GS is more effective for features with very high dimensions.

### 3.2. Visualization of Discriminative Objects

We use GS to select and visualize the most discriminative objects in videos to further understand the object-level informa-

(a) basketball shooting (b) walking with a dog (c) trampoline jumping (d) soccer juggling (e) diving

**Fig. 4**. Sample results of visualization of global feature selection.

| GS ratio | IDT | LS0 | LS0.2 | LS0.3 | LS0.4 |
|---|---|---|---|---|---|
| 100% | 89.02 | 92.51 | 92.60 | 92.17 | 92.40 |
| 75% | **89.12** | 92.76 | 92.94 | 92.17 | 92.76 |
| 50% | 88.42 | **92.76** | 93.04 | 92.51 | 92.75 |
| 25% | 86.54 | 92.46 | **93.20** | 92.68 | **93.51** |
| 12.5% | 85.20 | 91.75 | 92.56 | **93.10** | 93.26 |

**Table 3**. Accuracy performance on UCF11, with either IDT alone or concatenated with local-selected object-level features under different $\theta$ ($\theta$ = 0, 0.2, 0.3 and 0.4).

tion being selected. In other words, we want to use GS to automatically choose the most discriminative ones from the original potential object-level features without using LS and visualize their correspondence regions. In this way, we want to demonstrate the effectiveness of GS and the meaningfulness of using human detection in LS. In particular, given one class $c$, we set label 1 for this class and set label $-1$ for the rest classes. Then we compute the importance score using the methods described in Sec. 2.3. Fisher Vector is a $2KD$ vector where $K$ is the number of GMM components and $D$ is the dimension of each local feature. The score of a GMM component is the summation of the scores of the $2D$ dimensions belonging to this component. All the object-level features will be assigned to different GMM components based on the nearest neighbour criteria. Fig. 5 illustrates the process of the GMM component score calculation and patch assignment. Then we select the GMM component with the highest score, retrieve the object-level features assigned to it and show their correspondence regions in the original frame.

Fig. 4 gives some sample results of visualization. The identified regions are mostly around human body parts, such as the leg part for the *walking with dog* class or the upper body part for the *diving* class, which suggests the meaningfulness of using human detection in LS. More importantly, the identified regions coincide well with the corresponding action labels, which indicates GS is able to select discriminative object-level information.

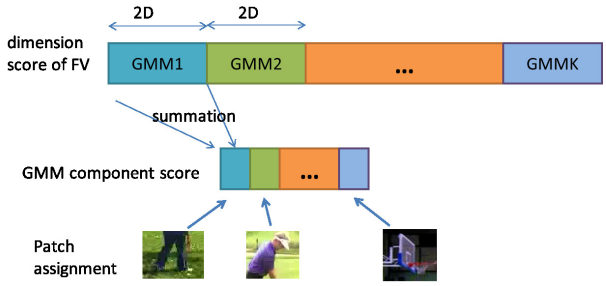



**Fig. 5**. Identifying the most discriminative objects.

## 4. CONCLUSION

In this paper, we have proposed to generate potential object-level features by directly performing multi-scale pooling on the image-level CNN feature domain, select local object-level features by human detection, and then select global FV dimensions via mutual information. Experimental results show that the proposed two-layer feature selection framework helps improve classification accuracy while significantly reducing feature storage cost at different stages. Combining with IDT features, our selected object-level features significantly improve the classification accuracy of using IDT alone.

## 5. ACKNOWLEDGEMENT

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office. This research is also partially supported by Singapore MoE AcRF Tier-1 Grant RG138/14.

## 6. REFERENCES


[1] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," in *CRCV-TR-12-01*, 2012.

[2] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *ICCV*, 2011, pp. 2556–2563.

[3] A. Gorban, H. Idrees, Y.G. Jiang, A. Zamir, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," 2015.

[4] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013, pp. 3551–3558.

[5] 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 *CVPR*, 2011, pp. 3361–3368.

[6] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *ECCV*, 2014, pp. 581–595.

[7] E. H. Taralova, F. De la Torre, and M. Hebert, "Motion words for videos," in *ECCV*, 2014, pp. 725–740.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014, pp. 1725–1732.

[9] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014, pp. 568–576.

[10] Z. Xu, Y. Yang, and A. G Hauptmann, "A discriminative cnn video representation for event detection," in *CVPR*, 2015, pp. 1798–1807.

[11] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010, pp. 143–156.

[12] M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *CVPR*, 2014, pp. 3286–3293.

[13] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, pp. 154–171, 2013.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[16] A. Vedaldi and K. Lenc, "Matconvnet-convolutional neural networks for matlab," *arXiv:1412.4564*, 2014.

[17] Y. Zhang, J. Wu, and J. Cai, "Compact representation for image classification: To choose or to compress?," in *CVPR*, 2014, pp. 907–914.

[18] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *CVPR*, 2009, pp. 1996–2003.

[19] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *ICCV*, 2009, pp. 1365–1372.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014, pp. 346–361.

[21] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *JMLR*, vol. 9, pp. 1871–1874, 2008.