Discriminative Dictionary Design for Action Classification in Still Images and Videos

Abhinaba Roy · Biplab Banerjee · Amir Hussain · Soujanya Poria

Received: 9 June 2020 / Accepted: 10 February 2021 / Published online: 3 March 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract
In this paper, we address the problem of action recognition from still images and videos. Traditional local features such as SIFT and STIP invariably pose two potential problems: 1) they are not evenly distributed in different entities of a given category and 2) many of such features are not exclusive of the visual concept the entities represent. In order to generate a dictionary taking the aforementioned issues into account, we propose a novel discriminative method for identifying robust and category specific local features which maximize the class separability to a greater extent. Specifically, we pose the selection of potent local descriptors as filtering-based feature selection problem, which ranks the local features per category based on a novel measure of distinctiveness. The underlying visual entities are subsequently represented based on the learned dictionary, and this stage is followed by action classification using the random forest model followed by label propagation refinement. The framework is validated on the action recognition datasets based on still images (Stanford-40) as well as videos (UCF-50). We get 51.2% and 66.7% recognition accuracy for Stanford-40 and UCF-50, respectively. Compared to other representative methods from the literature, our approach exhibits superior performances. This proves the effectiveness of adaptive ranking methodology presented in this work.

Keywords Action recognition · Local features · Feature mining · Random forest

Introduction

Recognition of visual concepts is one of the most active research areas in computer vision. Especially human action recognition from images and videos has been popular amongst researchers in recent times. With growing amount of visual data available from various sources, intelligent analysis of human attributes and activities has gradually attracted the interest of the computer vision community. One of the widely used approaches in action recognition is based on local descriptors that are based on three stages: 1) extraction of local descriptors, 2) codebook (dictionary) generation and feature encoding, and 3) classification based on the encoded features. Efficiency of such a model depends upon a number of factors, and effective codebook generation is undoubtedly the most noteworthy.

Standard codebook generation process is based on vector quantization of local descriptors extracted from the available training data in which the cluster centroids define the codewords, the basic building blocks that are ultimately used to encode the underlying visual entities. Specifically, an entity is represented by a vector where the $i^{th}$ component can be either the number of local descriptors that fall in the $i^{th}$ cluster or a measure of proximity of local descriptors to the $i^{th}$ cluster centroid. Needless to mention, the quality of the extracted local descriptors affects representation power of the codewords which, in turn, has a direct impact on the recognition performance. For instance, the descriptors extracted from background regions or the ones shared by
many visual categories add little to the discriminative capability of the codebook in comparison with the ones specifically extracted from the objects of interest. However, it is impossible to ensure the selection of potentially useful local descriptors in advance since such feature extraction techniques are typically engineered and ad hoc. In other words, there are certain immediate advantages if the most discriminative local descriptors are used for the purpose of a cogent codebook construction, though the process is intrinsically complex in general. Selection of the discriminative local descriptors for effective codebook generation coping with action recognition from images is the very core topic of this paper. We propose a simple algorithm, which gradually filters out unrepresentative descriptors before constructing a compact global codebook. The proposed method is generic in the sense that it can work with different types of local features irrespective of the underlying visual entities they refer to. Specifically, we represent each still image by a large pool of category-independent region proposals [1]. Each region proposal is represented by convolutional neural network (CNN) features (4096 dimensions) obtained from a pre-trained network. We propose a sequential method for codebook construction, which first clusters the local descriptors of each entity using the nonparametric mean-shift (MS) technique [2]. The cluster centroids thus obtained represent the reduced set of non-repetitive local features for the entity from now onwards. Another round of MS clustering on the new set of local descriptors calculated from all the entities of a given category is followed, and the centroids thus obtained are employed to build a temporary codebook specific to each category. Further, we propose an adaptive ranking criteria to highlight potentially discriminative codewords from each category specific codebook, and the global dictionary is built by accumulating these reduced set of codewords from all the categories. Efficient codebook construction is not new to the computer vision literature. However, we argue that our codebook construction technique explicitly incorporates the class support and a novel notion of distinctiveness based on conditional entropy is introduced.

We can summarize the main highlights of this paper as follows:

- The initial two-level MS-based clustering of the local descriptors on the entity and the category level largely reduces the effects of repetitive and uninteresting descriptors, yet selecting representative codewords from each locally dense region in the feature space. We further propose a novel adaptive measure to rank and select a subset of discriminative codewords per visual category under consideration using the concepts of conditional entropy and term frequency-inverse document frequency (tf-idf) score followed by an adaptive ranking technique.

The proposed ranking method ensures that the selected set of codewords are frequent in the entities of the same category while being sporadic in other visual categories.

- We evaluate the codebooks learned in this way for action categorization from still images. We observe that the learned codebooks, when used in conjunction with efficient feature encoding techniques, sharply outperform similar techniques from the literature. Specifically, considering the size of the local descriptors, we consider the locality-constrained linear coding (LLC) [3] for action recognition in images and Fisher vector [4] for videos. Consecutively, classification is performed using random forest classifier.

- As post-processing of classification results, we apply label propagation algorithm to improve the classification of random forest.

The rest of the paper is organized as follows: We discuss a number of related works from the literature in Section §2. The proposed action recognition framework is described in Section 3. Experimental results are reported in Section 4, followed by concluding remarks and ideas of possible future endeavor.

Related Work

In this section, we highlight two aspects of the proposed framework and discuss relevant techniques from the literature. First, action representation from images and videos with a focus on local feature encoding-based methods is addressed, and a discussion on the relevant codebook construction techniques is subsequently to be followed.

Action Recognition From Still Images and Videos—Use of Local Features

Recognition of human actions and attributes [5] has been approached using traditional image classification methods [6]. In the standard dictionary learning-based scenario, a typical framework extracts dense SIFT [7] from the training images, and codebook is constructed by clustering the SIFT descriptors by k-means clustering. Further, efficient encoding techniques including bag of words (BoW), LLC, Fisher vector are used to represent the images before the classification stage is carried out in such a feature space [8]. Since the inherent idea of the BoW-based frameworks is to learn recurring local patches, a different set of approaches directly models such object parts in images. Such techniques either initially define a template and try to fit it to object parts or iteratively learn distinctive parts for a given category. Discriminative part-based models (DPM) [9] are used extensively for this purpose, and they served
as the state of the art for a period. The hierarchical DPM model is used to parse human pose for action recognition in [10]. An efficient action and attributed representation based on sparse bases of local features is introduced in [11]. An expanded part model for human attribute and action recognition is proposed in [12]. The effects of empty cavity, ambiguity and pooling strategies are explored in order to design the optimal feature encoding for the purpose of human action recognition in still images in [13]. Very recently, the part learning paradigm has gained much attention because of its ability to represent mid-level visual features. Given a large pool of region proposals extracted from the images, such techniques iteratively learn part classifiers with high discriminative capabilities. Methods based on partness analysis [14] and deterministic annealing for part learning [15] are some of the representatives in this respect. The notion of a part is further extended to videos by constructing spatio-temporal graphs of the local keypoints over the video frames [16]. Similarly, the BoW framework exhibits impressive performance in recognizing action from video data. Broadly, the video-level features can be categorized into hand-engineered and deep features. The popular handcrafted feature set includes STIP [17], selective STIP [18], dense and improved trajectory [19] and optical flow-based features [20]. Further, several descriptors are used to encode the scene around such detected keypoints (HOG, HOF, etc.). In contrast, the deep CNN structures for videos combine separate models for static frames and the inter-frame motion. While a standard image-based CNN models (AlexNet, GoogleNet, VGGNet, etc.) can be used to extract per frame features, sophisticated optical-flow CNN is modeled for capturing the motion efficiently [21].

**Dictionary Learning**

Initial works in dictionary learning proposes k-means clustering to create a dictionary, followed by a bag-of-words (BoW) encoding [22, 23]. Amongst alternate approaches that focus on Sparse coding, Qiu et al. [24] report a sparse dictionary-based representation for action. Liu et al. [25] propose a Hessian-regularized sparse coding method for action recognition. Lu et al. [26] proposed the idea of slicing frame to patches in different scales and using patches to train dictionaries. Fanello et al. et al. [27] introduces a real-time action recognition method with dictionary learning. Xu et al. [28] propose a two-stream dictionary learning architecture that consists of interest patch (IP) detector and descriptor.

There are not many approaches that focus on dictionary learning for the task of action recognition. Rather, they are more interested in the development of proper feature representations. Contrary to these approaches, in this work we focus on construction of discriminative dictionary and prove the effectiveness using generic feature representations.

**Proposed Algorithm**

We detail the proposed action recognition framework in this section. As already mentioned, the proposed framework consists of four major stages: 1) extraction of local features, 2) discriminative dictionary construction, 3) feature encoding, and 4) action classification.

For notational convenience, let us consider that $\mathcal{TR} = \{X_i, Y_i\}_{i=1}^N$ constitutes $N$ training examples belonging to $L$ action categories where each $X_i$ represents an image or a video and $Y_i$ is the corresponding class label. Entities in $\mathcal{TR}$ are represented by a set of local descriptors $F_i = \{F_i^1, F_i^2, \ldots, F_i^N\}$ where $F_i^k \in \mathbb{R}^d$ and $d = 4096$ or $d = 162$, respectively, depend on whether the underlying $X_i$ is an image or a video. In addition, $a_i$ represents the number of local descriptors extracted from $X_i$. Further, $\{C_1, C_2, \ldots, C_L\}$ represents the set of category-specific codebooks learned by the proposed algorithm by exploiting the local features extracted from $\mathcal{TR}$, whereas $C = \{C_1, C_2, \ldots, C_L\}$ is the global codebook obtained by the concatenation of the local ones.

The framework is elaborated in the following sections.

**Extraction of Local Features**

We consider category-independent region proposals to highlight local regions in still images, whereas the popular STIP features are used for video streams.

Region proposal generation techniques highlight region segments in the image where the likelihood of the presence of an object part is high. This provides a structured way to identify interesting locations in the image and thus reduces the search space for efficient codeword generation. We specifically work with the objectness paradigm for region proposals generation from still images, which is based on modeling several aspects regarding the characteristics of the objects in a Bayesian framework. Each region proposal is further represented by the CNN features. We prefer the ImageNet pre-trained VGG-F [29] model, which has an architecture similar to AlexNet [30] and comprises 5 convolutional layers and 3 fully connected layers. The main difference of VGG-F and AlexNet is that VGG-F contains less convolutional layers and uses a stride of 4 pixels leading to better evaluation speed than the AlexNet architecture. In case of videos, the representation of local variations depends on local STIP keypoints. STIP features are the extension of the Harris corner detectors for images to the spatiotemporal domain. They are detected at locations where the video frame-level intensities have significant local variations in
both space and time. Histogram of oriented gradients (HOG) and histogram of optical flow (HOF) features are extracted around each STIP point.

**Discriminative Dictionary Learning**

We first build category specific codebooks and then concatenate all the local codebooks to generate a global codebook.

**Separate Dictionary Learning for Each Category**

For a given $l \in \{1, 2, \ldots, L\}$, the dictionary learning process is summarized as follows:

1. For each training instance with the category label $l$, we first group the local descriptors using MS clustering and consider the cluster centroids as constituting the reduced set of local descriptors. MS is an iterative, nonparametric clustering method which does not require an estimation of the number of clusters as input. Instead, it relies on the kernel density estimate in the feature space to group samples which form dense clusters. Given $F_i = \{F_i^1, F_i^2, \ldots, F_i^u\}$, the kernel density estimate at a point $F_i^k$ is expressed as

$$f(F_i^k) = \frac{1}{a_i h^2} \sum_{m=1}^{u_i} K\left(\frac{F_i^k - F_i^m}{h}\right)$$

where $K$ is a radially symmetric kernel function and $h$ defines the width of the Parzen window to highlight the neighborhood around $F_i^k$. A cluster is identified as the region where the data density is locally maximum. This can alternatively be interpreted as the local regions where $\forall F_i \approx 0$. $\forall F_i$ can efficiently be calculated by iteratively shifting the centroids of the Parzen windows until the locally dense regions are reached [2]. Since all the descriptors in a dense region in the feature space highlight near similar local features, the mean-shift clustering is able to select one unique representative for all of them. Further, since mean-shift implicitly estimates the number of clusters present in the dataset, hence, the problem of over-merging is greatly reduced. On the other hand, spherical clustering techniques like k-means and fuzzy c-means create suboptimal codebooks as most of the cluster centroids fall near high-density regions, thus under-representing equally discriminant low-to-medium density regions. MS resolves such problem by focusing on locally dense regions in the feature space. Let $\tilde{F}_i = \{\tilde{F}_i^1, \tilde{F}_i^2, \ldots, \tilde{F}_i^u\}$ represents the new set of local descriptors for the $i^{th}$ training instance where each $\tilde{F}_i^k$ represents a cluster centroid.

2. Once $\tilde{F}_i$ are constructed for all the training instances with category label $l$, we vector-quantize all such $\tilde{F}_i$ using MS clustering to build a temporary codebook $C_l = \{C_{l1}^1, C_{l1}^2, \ldots, C_{l1}^{m_l}\}$ for the category with each $C_{l1}^k$ representing a codeword (cluster centroid). Similar to the previous stage, it is guaranteed that $C_l$ is ensured to capture all the potential local features for the $i^{th}$ category.

$\{C_1, C_2, \ldots, C_L\}$ are constructed in the similar fashion for $l \in \{1, 2, \ldots, L\}$. It is to be noted that the labels of the codewords depend upon the action categories they refer to. Further, the sizes of the $C_l$s may differ from each other. The $C_l$s thus obtained are not optimal in the sense that they contain many codewords with low discriminative property. Such codewords need to be eliminated in order to build robust category-specific codebooks. However, we need a measure to rank the descriptors based on their discriminative ability. In this respect, the following observations can be made:

- A potentially discriminative codeword is not frequent over many of the categories constituting the dataset.
- Most of its nearest neighbors in $\{C_1, C_2, \ldots, C_L\}$ share the same class label with the codeword under consideration.

We model the first observation in terms of the idea of conditional entropy, whereas the second observation is replicated by the tf-idf score.

For a given codeword $C_{l}^k$, we find out the labels of its $T$ nearest neighbors over the entire set of codewords in $\{C_1, C_2, \ldots, C_L\}$ and subsequently define the conditional entropy measure as:

$$H(Y|C_{l}^k) = -\sum_{l'=1}^{L} p(l'|C_{l}^k) \log_2 p(l'|C_{l}^k)$$

where $p(l'|C_{l}^k)$ represents the fraction of the retrieved codewords with label $l'$. For discriminative codewords, i.e., the ones which do not span many categories, $H$ is small, whereas the value of $H$ grows with the selection of codewords shared by many categories.

In addition to the $H$ score, we also expect the nearest neighbors to be populated from the same category as of $C_{l}^k$. In order to impose this constraint, we define the tf-idf score for $C_{l}^k$ as follows:

$$TI(C_{l}^k) = \frac{|C_{l}^k| |C_{l}^k| \in knn(C_{l}^k) AND l' = l|}{|C_{l}^k| |C_{l}^k| \in knn(C_{l}^k)|}$$

Both the measures are further combined in a convex fashion to define the ranking measure as follows:

$$Rank(C_{l}^k) = w_1 \frac{1}{H(Y|C_{l}^k)} + (1 - w_1) TI(C_{l}^k)$$

\( \odot \) Springer
We repeat this stage for all the codewords in \( \{ C_1, C_2, \ldots, C_L \} \). As already mentioned, the \( \text{Rank}(C_l) \) has high values for potentially discriminative- and category-specific codewords.

**Number of Codeword Selection**

We rank the codewords on the basis of the \( \text{Rank} \) scores. In order to select the number of optimal codeworks to select, we use an adaptive algorithm. This is in stark contrast to the related work done in [31], where top \( B \) codewords were chosen in a greedy fashion in order to define the final codebook \( C_l \) for category \( l \). For this adaptive algorithm, we make a spanning tree of the codewords \( C_l \). We first create a spanning tree \( G \) with \( C_l \) as nodes and the edge weights as the difference between the features. Note that the nodes are connected sequentially based on the ranked list. On this spanning tree, we carry out a dominant set clustering [32]. More specifically, we carry out a binary clustering on the graph to get two subgraphs. We select the subgraph (set of codewords) with a higher rank. This results in variable numbers of codewords in each of the classes. Algorithm 1 describes the adaptive number of codeword selection process.

**Global Dictionary Construction**

The local codebooks obtained in the previous stage are concatenated in order to obtain a global codebook \( \tilde{C} = [\tilde{C}_1 \tilde{C}_2 \ldots \tilde{C}_L] \).

**Feature Encoding Using \( \tilde{C} \)**

We represent each visual entity with respect to \( \tilde{C} \) for still images and videos separately. We find that LLC-based encoding works best while dealing with the CNN features in case of action recognition in still images, whereas Fisher vector outperforms other BoW-based encoding methods for video-based features. For each entity, we consider all the initially extracted local features for encoding.

**Classification**

The final classification is performed using random forest ensemble classifier [33]. The decision tree learning algorithm used is information gain, and bootstrap aggregation is employed to learn the ensemble model. Thus, the forest reduces classifier variance without increasing bias. Random subspace splitting is used for each tree split, and we consider \( \sqrt{d} \) features for each split given \( d \) original feature dimensions. The generalization is performed by applying majority voting on the outcomes of the learned trees.

**Label Propagation**

In order to refine and further strengthen the classification results from random forests, we apply an additional round of label propagation. In its original form, label propagation is a semi-supervised classification way to propagate labels from labeled sample to unlabeled samples [34]. Based on the idea that samples should have same labels if they are neighbors to each other, label propagation “propagates” labels of labeled samples to unlabeled samples according to the proximity. The more similarly samples are, the more easily propagate labels between them. The similarity of samples is calculated as and exponential of negative distance between them (Equation 5).

\[
    w_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma}\right)
\]  

(5)

where \( d_{ij} \) is the distance between \( i^{th} \) and \( j^{th} \) sample. The degree of difficulty for label propagation is described by probabilistic transition matrix \( T_{ij} \), which is defined as

\[
    T_{ij} = P(x_i \rightarrow x_j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{kj}}
\]  

(6)
where $T_{ij}$ is the probability of switching label (inference) from instance $x_i$ to $x_j$. Larger $w_{ij}$ leads to larger $T_{ij}$ that allows labels to propagate more easily. The process of label propagation continues until labels of all samples tend to be stable. Label propagation defines a label matrix $Y$ with all the label probabilities of data points $x_i$s. In this work, we attempt to propagate the labels of most confident labels onto the other inferences with less confidences. First we detect all the output labels from the random forest classifier which most likely provide true inference for a certain test instance. In the cases where there is ambiguity amongst the tree classifier outcomes, we expect propagation of confident labels will solve this issue and provide better classification performance. In order to measure the ambiguity between individual tree classifiers in the random forest, we device a measurement to determine the confidence of inference for each test instances. We define the confidence of the inferred labels in terms of agreement of trees in majority voting of random forest. Hence, confidence of inferred $y_i$ from a test instance $x_i$ is denoted by

$$conf_i = \frac{N_{y_i}}{N}$$  \hspace{1cm} (7)$$

where $N_{y_i}$ denotes the number of trees which inferred $y_i$ as the label of $x_i$ and $N$ is the total number of trees. We rank the test samples in order of confidence of inferences from them. Consecutively, we use top 20% confident inferences and apply label propagation to improve the outcome of the rest of the inferences. The algorithm related to label propagation is shown below.

### Experimental Details

#### Dataset

We consider the Stanford-40 [11] still image action recognition database and UCF-50 [35] video-based action recognition dataset to evaluate the effectiveness of the proposed framework. Stanford-40 actions is a database of human actions with 40 diverse action types, e.g., brushing teeth, reading books, blowing bubbles, etc. The number of images per category ranges between 180 and 300 with a total of 9352 images. We use the suggested [11] train–test split with 100 images per category as training and remaining for testing. On the other hand, the UCF-50 dataset contains videos representing 50 actions in an unconstrained environments. The dataset contains a total of 6700 videos with about 100 – 150 videos per category. This dataset is a superset of the popular UCF-11 dataset. We randomly select 60% of the videos per category to represent the training set and the remaining 40% is used to evaluate the classification performance of the proposed framework.

### Experimental Setup

The following experimental setup is considered in order to evaluate the performance of the proposed framework for both the datasets.

- MS clustering is used in conjunction with the Gaussian kernel. The adaptive bandwidth parameter ($h$) is fixed empirically as $D^m$ ($1 \leq m \leq 10$), where $D$ is the average pairwise distance of all the local descriptors extracted from all the visual entities of each category. The same setup is repeated for MS clustering in the entity and the category levels (section §3.2.1).

---

**Algorithm 2 : Label propagation for improved classification**

**Input :** Test instances with confident inferences $(x_1, y_1), \ldots, (x_l, y_l)$. Other test instances with not confident inferences $(x_{l+1}, y_{l+1}), \ldots, (x_{l+u}, y_{l+u})$. $y_i$ is the inference from random forest for instance $x_i$. $Y_L = y_1, \ldots, y_l$; $Y_U = y_{l+1}, \ldots, y_{l+u}$;

$Y = [Y_U; Y_L]$

**Output :** Labels of all instances

1. Calculate $w_{ij}$ between samples;
2. Calculate probabilistic transition matrix $T$;
3. Structure matrix $Y$;
4. $TY \rightarrow Y$;
5. Normalize $Y$;
6. Reset $Y_L$;
7. Repeat step 4 until $Y$ does not change/maximum number of iteration is reached;
We extract 500 region proposals per image for the Stanford-40 dataset. Figure 1 depicts the extracted region proposals for a pair of images from the dataset. We further discard proposals, which are largely overlapping to each other (overlap of ≥ 50%) in order to highlight potentially discriminative local patches in the images. The STIP keypoints are extracted from the videos using the publicly available implementation of [17].

The number of final distinctive codewords selected for each class is set adaptively as discussed in section §3.2.2.

As for feature encoding, for LLC, 100 – 200 nearest neighbors per local descriptor are considered to encode the images. We select the optimal hyper-parameters by cross-validation. Each image in the Stanford-40 dataset is optimally represented by a sparse vector of length $8000 \times 1$ (100 neighbors in LLC), whereas each video has a feature length of $32400 \times 1$ ($2 \times$ feature dimension(162) $\times$No. of Gaussian components(100)).

Each component tree in the random forest model is essentially a classification and regression tree (CART) [33]. We conduct experiments with random forest of different sizes ($500 – 2000$) and find that a random forest with 1000 CART trees exhibits superior performance.

For post-processing based on label propagation, we fix the threshold for “confident” inferences to 20%. This value is determined by fivefold cross-validation for both image and video data.

We compare the overall classification performance of the proposed technique with the representative techniques from the literature. All the experiments are

---

**Table 1** Effect of adaptive number of code word selection for Stanford-40

| Method                                                | Classification accuracy |
|-------------------------------------------------------|-------------------------|
| Top $B$ ($B=200$ chosen empirically) codewords and LLC encoding [31] | 49%                     |
| Proposed framework with adaptive number of codewords and LLC encoding | 49.4%                   |
repeated multiple times, and the average performance measures are reported.

### Performance Evaluation

In this section, we evaluate the performance of our framework on action recognition over benchmark datasets with still images (Stanford-40) and videos (UCF-50).

**Evaluation on Stanford-40**

We evaluate the performance of our approach in twofold. First we provide an ablation study to evaluate performance of our adaptive number of codeword selection (§3.2.2). The results are shown in Table 1. In the previous work of [31], a number of codewords are chosen empirically after cross-validation, still adaptive selection improves that performance by 0.4%. This goes on to show that adaptive selection not only eliminates the need to execute cross-validation to select the number of code words, but also improves previous performance by a margin.

Next, we demonstrate the effect of label propagation (§3.4.1) in our pipeline. We compare the effect of addition of label propagation in Table 2. We achieve a performance improvement of 1.8%. A simple assumption that neighboring data points tend to have similar labels clearly improves the performance of random forest. This also shows that label propagation is an effective tool to reduce the tree classifier confusions that arise in a random forest. It also adds helps majority voting, that is applied in random forest alleviate the problem of tree classifier confusions and avoid misclassification.

Finally, we evaluate the performance of our approach against other related action recognition pipelines in the literature. Table 3 mentions the accuracy assessment of different techniques for the Stanford-40 dataset. The performances of the methods based on handcrafted SIFT-like features are comparatively less ($\approx 35.2\%$) [3]. This can be attributed to the fact that differences in human attributes for many of the action classes are subtle. Label-consistent K-SVD provides 32.7% accuracy. Part learning-based strategies obtain better recognition performance in this respect by explicitly modeling category-specific parts. Classification accuracy of 40.7% is obtained with the generic expanded part models (EPM) of [12], which is further enhanced to 42.2%, while the contextual information is incorporated in EPM. The best performance with shallow features obtained for this dataset is 45.7% by [11], which performs action recognition by combining bases of attributes, objects and poses. Further, they derive their bases by using large amount of external information. It is worth noting that the ImageNet pre-trained AlexNet reports a classification accuracy of 46% [30]. With our framework, we observe an improvement of 5.2% over them. It can be argued that our method encapsulates the advantages of deep and shallow models effectively in a single framework. The CNN-based region proposals are capable of encoding high-level abstractions from the local regions. Since the images are captured in unconstrained environments, the backgrounds are uncorrelated in different images of a given category. The per category dictionary learning strategy reduces the effects of such background patches and the proposed ranking measure further boosts the proposals corresponding to the

| Method | Classification accuracy |
|--------|-------------------------|
| Top B ($B=200$ chosen empirically) codewords [31] and Fisher vector encoding | 64% |
| Proposed framework with adaptive number of codewords and Fisher vector encoding | 64.5% |
shared human attributes, human–objects interaction, etc., for a given action category. In contrast to other techniques which are based on SVM classifier, our framework relies on the random forest model which does not explicitly require any cross-validation. We observe that the performance of the random forest model gradually improves with growing number of CART trees within the range $500 - 1000$ and a random forest model with 1000 trees outputs the best performance. Further, the addition of label propagation overcomes the problem of misclassification due to confusion created because of ensemble nature of random forest.

UCF-50 Dataset

For UCF-50, we divide our experiments in similar way we do for Stanford-40. Table 4 shows the effect of adaptive selection of number of codewords. Similar to that of Stanford-40, we observe an increment of 0.5% in result. Table 5 shows the effect of label propagation on random forest. Label propagation improves the performance of random forest by 2.2%. Encoding videos properly is inherently more complex than images due to added difficulty of encapsulating changes along progression of time. This in turn creates confusion in an ensemble setting such as random forest. Label propagation works well in this scenarios which is evident from such a high increment of result.

We compare the performance of our framework with that of three different shallow representations from the literatures with similar train–test split (Table 6). The standard STIP (HOG + HOF) with the BoW encoding and the frame-based GIST descriptors [39] exhibit classification performances of 47.9% and 38.8%, respectively. Since the differences between many of the action classes in UCF-50 are fine-grained and the videos contain substantial camera motion and cluttered backgrounds, models based on global descriptors fail drastically in discriminating the action classes. The ActionBank [40] model based on learned action templates provides improved recognition performance (57.9%), although it requires numerous supervised information to learn the templates. Improved dense trajectory [19] and dense trajectory with additional RootSIFT normalization provided 65.2%. All the aforementioned setups are based on the SVM classifiers.

In contrast, our framework exhibits the best average recognition accuracy of 66.7%. (We use GMM with 100 components.) The enhancement of the performance of the proposed framework is attributed to the robust ranking measure, which selects recurrent and discriminative local features and reduces the effects of background patches by assigning low distinctiveness scores. This is established since the recognition performance of the system sharply decreases (recognition accuracy of $\approx 58\%$ when all the codewords are considered) as more codewords per category are considered to build the dictionary. We do not compare with the methods that use deep features since deep features have greater ability to encapsulate the spatial–temporal complexities involved in a video. In this work, we do not attempt to compare abilities of different feature representations rather the effectiveness of our pipeline.

Conclusion

We introduce a novel supervised discriminative dictionary learning strategy for the purpose of action recognition from still images as well as videos. We take advantage of the available training samples to adaptively rank local features which are both robust and discriminative. Further, we cluster the local features at the entity and category levels to eliminate the effects of features corresponding to non-recurrent or background locations. The adaptive ranking paradigm proposed in this work holds wider applications in areas including feature selection, ranked set generation for retrieval, etc. The effectiveness of this dictionary learning approach is validated on challenging datasets (Stanford-40, UCF-50), on which superior performance measures can be

| Method | Classification accuracy |
|--------|------------------------|
| Proposed framework with adaptive number of codewords and Fisher vector encoding | 64.5% |
| Proposed framework with adaptive number of codewords and Fisher vector encoding + label propagation | 66.7% |

| Method | Classification accuracy |
|--------|------------------------|
| GIST [39] | 38.8% |
| STIP (HOG+HOF) + bag of words | 47.9% |
| ActionBank [40] | 57.9% |
| Improved dense trajectory [19] | 65.3% |
| Proposed framework (with adaptive number of code-words and Fisher vector encoding + label propagation) | 66.7% |
observed in comparison with popular techniques from the literature.

Declarations

**Conflicts of Interest** The authors declare that they have no conflict of interest.

**Ethical Approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**References**

1. Alexe B, Deselaers T, Ferrari V. Measuring the objectness of image windows. IEEE Trans Pattern Anal Mach Intell. 2012;34(11):2189–202.
2. Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell. 2002;24(5):603–19.
3. Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y. Locality-constrained linear coding for image classification. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010. pp. 3360–3367.
4. Perronnin F, Sánchez J, Mensink T. Improving the fisher kernel for large-scale image classification. In European conference on computer vision. Springer, 2010. pp. 143–156.
5. Cheng G, Wan Y, Saudagar AN, Namduri K, Buckles BP. Advances in human action recognition: a survey. arXiv preprint 2015. arXiv:1501.05964
6. Yang W, Wang Y, Mori G. Recognizing human actions from still images with latent poses. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010. pp. 2030–2037.
7. Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. 2004;60(2):91–110.
8. Chatfield K, Lempitsky VS, Vedaldi A, Zisserman A. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC. 2011. vol. 2, p. 8.
9. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell. 2010;32(9):1627–45.
10. Wang Y, Tran D, Liao Z, Forsyth D. Discriminative hierarchical part-based models for human parsing and action recognition. J Mach Learn Res. 2012;13:3075–102.
11. Yao B, Jiang X, Khosla A, Lin AL, Guibas L, Fei-Fei L. Human action recognition by learning bases of action attributes and parts. In 2011 International Conference on Computer Vision. IEEE, 2011. pp. 1331–1338.
12. Sharma G, Jurie F, Schmid C. Expanded parts model for human attribute and action recognition in still images. In Proc IEEE Conf Comput Vis Pattern Recognit. 2013. pp. 652–659.
13. Zhang L, Li C, Peng P, Xiang X, Song J. Towards optimal vlad for human action recognition from still images. Image and Vision Computing. 2016.
14. Juneja M, Vedaldi A, Jawahar C, Zisserman A. Blocks that shout: Distinctive parts for scene classification. In Proc IEEE Conf Comput Vis Pattern Recognit. 2013. pp. 923–930.
15. Sicre R, Jurie F. Discriminative part model for visual recognition. Comput Vis Image Underst. 2015;141:28–37.
16. Zhou Y, Ni B, Hong R, Wang M, Tian Q. Interaction part mining: A mid-level approach for fine-grained action recognition. In Proc IEEE Conf Comput Vis Pattern Recognit. 2015. pp. 3323–3331.
17. Laptev I. On space-time interest points. Int J Comput Vis. 2005;64(2–3):107–23.
18. Chakraborty B, Holte MB, Moeslund TB, Gonzalez J, Roca FX. A selective spatio-temporal interest point detector for human action recognition in complex scenes. In Computer Vision (iccv), 2011 IEEE International Conference on. IEEE, 2011. pp. 1776–1783.
19. Wang H, Schmid C. Action recognition with improved trajectories. In Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013. pp. 3551–3558.
20. Xiong W, Lee JC-M. Efficient scene change detection and camera motion annotation for video classification. Comput Vis Image Underst. 1998;71(2):166–81.
21. Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. In Adv Neural Inf Proces Syst. 2014. pp. 568–576.
22. Shukla P, Biswas KK, Kalra PK. Action recognition using temporal bag-of-words from depth maps. In MVA. 2013. pp. 41–44.
23. Bettadapura V, Schindler G, Ploetz T, Essa I. Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 2619–2626. IEEE, 2013.
24. Qiu Q, Jiang Z, Chellappa R. Sparse dictionary-based representation and recognition of action attributes. In Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011. pp. 707–714.
25. Liu W, Wang Z, Tao D, Yu J. Hessian regularized sparse coding for human action recognition. In International Conference on Multimedia Modeling. Springer, 2015. pp. 502–511.
26. Lu C, Shi J, Jia J. Abnormal event detection at 150 fps in matlab. In Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013. pp. 2720–2727.
27. Fanello SR, Gori I, Metta G, Odone F. Keep it simple and sparse: Real-time action recognition. J Mach Learn Res. 2013;14(1):2617–40.
28. Xu K, Jiang X, Sun T. Two-stream dictionary learning architecture for action recognition. IEEE Trans Circuits Syst Video Technol. 2017;27(3):567–76.
29. Chatfield K, Simonyan K, Vedaldi A, Zisserman A. Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint 2014. arXiv:1405.3531
30. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In Adv Neural Inf Proces Syst. 2012. pp. 1097–1105.
31. Roy A, Banerjee B, Murino V. Discriminative dictionary design for action classification in still images. In International Conference on Image Analysis and Processing. Springer, 2017. pp. 160–170.
32. Pavan M, Pelillo M. Dominant sets and pairwise clustering. IEEE Trans Pattern Anal Mach Intell. 2007;29(1):167–72.
33. Bishop CM. Pattern recognition. Mach Learn. 2006;128.
34. Zhu X, Ghahramani Z. Learning from labeled and unlabeled data with label propagation. 2002.
35. Reddy KK, Shah M. Recognizing 50 human action categories of web videos. Mach Vis Appl. 2013;24(5):971–81.
36. Li LJ, Su H, Fei-Fei L, Xing EP. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In Adv Neural Inf Proces Syst. 2010. pp. 1378–1386.
37. Jiang Z, Lin Z, Davis LS. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011. pp. 1697–1704.
38. Lazebnik S, Schmid C, Ponce J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06). IEEE, 2006. vol. 2, pp. 2169–2178.

39. Oliva A, Torralba A. Building the gist of a scene: The role of global image features in recognition. Prog Brain Res. 2006;155:23–36.

40. Sadanand S, Corso JJ. Action bank: A high-level representation of activity in video. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012. pp. 1234–1241.