Learning Discriminative Key Poses for Action Recognition

Li Liu, Ling Shao, *Senior Member, IEEE*, Xiantong Zhen, and Xuelong Li, *Fellow, IEEE*

***Abstract*—In this paper, we present a new approach for human action recognition based on key-pose selection and representation. Poses in video frames are described by the proposed extensive pyramidal features (EPFs), which include the Gabor, Gaussian, and wavelet pyramids. These features are able to encode the orientation, intensity, and contour information and therefore provide an informative representation of human poses. Due to the fact that not all poses in a sequence are discriminative and representative, we further utilize the AdaBoost algorithm to learn a subset of discriminative poses. Given the boosted poses for each video sequence, a new classifier named weighted local naive Bayes nearest neighbor is proposed for the final action classification, which is demonstrated to be more accurate and robust than other classifiers, e.g., support vector machine (SVM) and naive Bayes nearest neighbor. The proposed method is systematically evaluated on the KTH data set, the Weizmann data set, the multiview IXMAS data set, and the challenging HMDB51 data set. Experimental results manifest that our method outperforms the state-of-the-art techniques in terms of recognition rate.**

***Index Terms*—AdaBoost, computer vision, extensive pyramidal features (EPFs), human action recognition, pose selection, weighted local naive Bayes nearest neighbor (WLNBNN) classifier.**

I. Introduction

HUMAN ACTION recognition, nowadays, plays a significant role in various applications, e.g., human–computer interaction [1], human activities analysis [2]–[5], and real-time surveillance systems [6], [7]. The goal of human action recognition is to identify the actions being performed in a video sequence under different complications such as cluttering, occlusion, and change of lighting conditions.

Manuscript received June 1, 2012; revised October 2, 2012; accepted November 29, 2012. Date of publication January 11, 2013; date of current version November 18, 2013. This work was supported in part by the University of Sheffield, by the Chinese Scholarship Council, by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, and by the National Natural Science Foundation of China under Grant 61125106, Grant 91120302, and Grant 61072093. This paper was recommended by Associate Editor W. Hu. *(Corresponding author: L. Shao.)*

L. Liu and X. Zhen are with the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: elp11ll@sheffield.ac.uk; elr10xz@sheffield.ac.uk).

L. Shao is with the College of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China, and also with the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: ling.shao@sheffield.ac.uk).

X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2012.2231959

Generally, the approaches to action recognition involve two main stages: 1) feature extraction and representation and 2) action classification.

In recent years, some popular spatiotemporal features extracted from video sequences have been commonly extended from their counterparts in the 2-D image domain and have been demonstrated to achieve relatively good results for action recognition. Those methods include histogram of optical flow [8], 3-D scale invariant feature transform [9], 3-D histogram of oriented gradients (HOG) [10], and 3-D speeded-up robust features [11]. These spatiotemporal features can be used either globally or locally for human action representation.

Local methods [2], [12] [13] represent human actions as a set of spatiotemporal interest points detected from video sequences. Local methods based on the bag-of-features model have achieved impressive results in various action recognition tasks. They follow a typical procedure: unsupervised techniques are used to detect interest points around which spatiotemporal features are extracted; then clustering methods, e.g., K-means clustering [14], are employed to construct a codebook on which all the features from a video sequence are mapped to form a histogram representation; the final representation is then fed to a classifier, e.g., support vector machine (SVM) for action classification. The bag-of-features [4] model tends to be more robust in challenging scenarios, but this kind of sparse representation is often not precise and informative because of the quantization error during codebook construction and the loss of structural relationships among local features.

On the contrary, global methods [15] represent corresponding actions by treating the entire video sequence as a whole, which contains the complete motion and appearance information. Such a representation has attracted much attention due to its abilities to extract more informative and accurate motion features from both spatial and temporal dimensions. However, global methods are quite sensitive to shift, scaling, occlusion, and cluttering, which commonly exist in action sequences because the required background subtraction and segmentation [16] tend not to be very accurate.

Both local and global methods have achieved remarkable results, but action recognition by human suggests that they might be using more information than required [17]. Given the available actions, the human brain can easily recognize what a person is doing by just looking at a few poses without examining the whole sequence [18]. Inspired by this, recently, pose representations for human action recognition have drawn more attention and achieved promising results [18]–[21]. Pose-based representations can capture sufficient appearance information of actions and spatial layout of human bodies, and therefore, it can, to a large extent, overcome the information

loss induced in local representations. If key poses are selected well [18], [19], [21], pose-based representations are able to avoid the limitations such as background variations, occlusions, and shifts in global representations.

A. Proposed Method

In this paper, we aim to model human actions based on key-pose representation. Each frame in an action sequence is treated as a pose. Inspired by the multiresolution analysis in image processing, we describe poses using the extensive pyramidal features (EPFs), which are composed of the Gabor, Gaussian, and wavelet Laplacian pyramids. These features capture the orientation, intensity, and contour information and thus provide an informative representation of poses.

Directly representing the video sequence by all the poses (frames), which contain redundant and indiscriminate information, would confuse the classifiers in action recognition [17]. We therefore propose to select a subset of key poses for the representation of each action by a supervised machine learning algorithm, i.e., AdaBoost [22]. The selected key poses for each action type are not only more compact but also constitute the most discriminative body poses of an action, because the common poses that can exist in different action types have been eliminated.

In our proposed action representation, each frame contains the whole human body with the full spatial structural information, which shares the advantages with the global representation methods, whereas each video sequence is sparsely represented by a subset of key poses, which enjoys advantages of local representations. Therefore, the proposed method can be regarded as a semiholistic representation of human actions and inherits the advantages of global features in the spatial dimensions and meanwhile has the superiority of local features in the temporal axis.

For local representations, the bag-of-features model and its variants plus SVM is the standard method for action recognition. Recently, a classifier called naive Bayes nearest neighbor (NBNN) [23] has been proposed for image classification tasks based on sets of local features. The NBNN classifier avoids the quantization error in the bag-of-features model by computing "image-to-class" distance rather than the "image-to-image" distance. McCann and Lowe [24] later developed the local NBNN (LNBNN) classifier with a remarkable increase in accuracy and decrease in running time compared with NBNN.

Inspired by LNBNN, in this paper, we propose to further improve the NBNN model and introduce an enhanced version of NBNN, named weighted LNBNN (WLNBNN), for the final action classification. We will experimentally show that WLNBNN is a more efficient and accurate classifier for action recognition.

B. Contributions

The contributions of this paper lie in three aspects.

1) We propose to describe action poses by EPFs, which can effectively capture the orientation, intensity, and contour properties in a pose and are tolerant to shifts and scaling.
2) The AdaBoost learning algorithm is employed for selecting discriminative poses, which can significantly improve the performance of action recognition.
3) A new classifier named WLNBNN is proposed for final action classification, which outperforms other classifiers, e.g., NBNN.

The rest of this paper is organized as follows: In Section II, we review the related work. The details of our method are described in Section III. Section IV reports the experiments and results. Finally, we conclude this paper in Section V.

II. Related Work

Action recognition based on pose representation has been applied in a large number of previous works. Thurau and Hlavác [25] presented a method to recognize actions in videos or images based on primitive pose representations, which are described by HOG [26]. Niebels and Fei-Fei [27] employed a pose-based method using a hierarchical model of shape and appearance for action recognition. Yang *et al.* [28] proposed an approach that treats the pose of the person in an image as latent variables and recognizes human actions from still images. Shao *et al.* [20] combined the pose silhouettes with correlograms [29] to achieve action recognition by adopting the k-nearest-neighbor (kNN) classifier.

In a video sequence, however, not all of the poses are informative and discriminative for action recognition. Some poses carry neither complete nor accurate information and would even contain common patterns shared by various action types. Since these poses in a video sequence cannot well represent the action and would cause confusion during the classification phase, a great deal of work has been carried out to select the most representative and discriminative poses, i.e., the key poses. To obtain visually distinct representations, Cooper and Foote [30] presented methods for key frame selection based on capturing the similarity to the represented segment and preserving the differences from other segments' key frames, so that different segments will have visually distinct representations. Zhao and Elgammal [31] developed an effective approach for action classification, in which they first described all the poses with the distribution of local motion features and their spatiotemporal arrangements and then selected a small set of most discriminative poses by comparing their discriminative power for each independent action. Zhuang *et al.* [32] applied an unsupervised clustering method for key-pose selection. Baysal *et al.* [18] selected the most representative and discriminative poses from a set of candidates by ranking the potentiality of each candidate pose in distinguishing an action from others. Cao *et al.* [19] developed a PageRank-based centrality measure to select key poses according to the geometric structure recovered by a manifold learning technique.

However, the aforementioned methods all use unsupervised techniques to measure pose similarity or calculate the pose probability distribution for key-pose selection. Since such unsupervised methods do not consider the relationship among poses from different classes and are not able to select the very discriminative poses from each actin type, accordingly, in this paper, we propose to use a supervised method, i.e., the

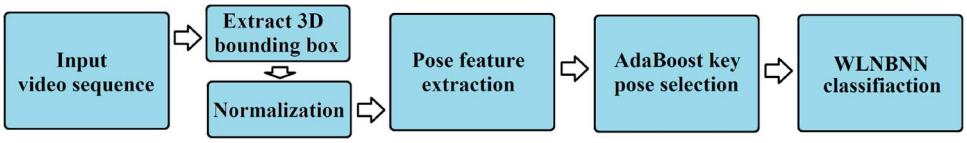


Fig. 1. Flowchart of the proposed method.

AdaBoost algorithm, to select a subset of key poses for action classification.

AdaBoost as a popular machine learning algorithm is widely used in computer vision. A pyramidal architecture was developed by Fathi and Mori [33] to extract boosted midlevel motion features for action recognition. Shen and Bai [34] combined Gabor wavelet features with the AdaBoost selection algorithm for image classification. Furthermore, an efficient approach for retrieving actions in movies based on AdaBoost selection was proposed by Laptev and Prez [35]. Due to its remarkable performance on various vision tasks, we adopt the AdaBoost algorithm to select discriminative key poses for action representation.

III. Methodology

Our recognition system is composed of three principal stages. 1) Pose description: For each video sequence, the EPFs are extracted from each frame to represent the pose appearing in it. 2) Pose selection: The AdaBoost algorithm is adopted to select the most discriminative key poses for each video sequence to represent the corresponding action. 3) Action recognition: Based on the boosted poses, a newly proposed classifier, i.e., WLNBNN, is employed for action classification. The flowchart of the proposed method is illustrated in Fig. 1. We will detail the three stages in the following sections.

A. EPFs

Given a frame containing a pose, we would like to describe it with informative features extracted from it. The descriptor is expected to capture the orientation, intensity, and contour information, which is the main cue of a pose. We therefore employ Gabor filters, the Gaussian pyramid, and the wavelet transform to obtain EPFs for pose representation.

1) Gabor Feature Map: Gabor filtering is regarded as the most effective method to obtain the orientation information, which is widely used in feature extraction due to its property of orientation selection. To mimic the biological mechanism of the visual cortex, Riesenhuber and Poggio [36] proposed the HMAX model composed of four hierarchical feedforward layers, namely, S1, C1, S2, and C2, in which S1 is obtained by Gabor filtering. Inspired by their work, we convolve each pose frame with a bank of Gabor filters with multiple scales and orientations to extract the S1 feature map. More specifically, we adopt Gabor filters with six different scales: 7×7, 9×9, 11×11, 13×13, 15×15, 17×17, and four different orientations: $0°$, $45°$, $90°$, and $135°$. As a result, by convolving with $6 \times 4 = 24$ Gabor filters, we obtain 24 S1 feature maps. The Gabor filter function is defined as follows:

$$G(x, y) = \exp\left(-\frac{(X^2 + \gamma^2 Y^2)}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} x\right) \quad (1)$$

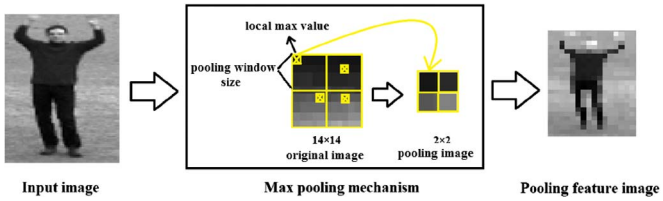


Fig. 2. Illustration of the max pooling mechanism.

$$\text{where } X = x \cos\theta - y \sin\theta,\ Y = x \sin\theta + y \cos\theta \quad (2)$$

and (x, y) is the coordinate relative to the center of the filter.

To obtain our Gabor feature maps that are equivalent to C1 in the HMAX model, we perform max pooling operations among the different scales. In other words, we pick the maximum value across all the S1 feature maps with filter scales in each orientation. The pooling among different scales is defined as follows:

$$I_{\text{MAX}} = \max_{(x,y)} [I_{7\times7}(x, y, \theta_s), I_{9\times9}(x, y, \theta_s), \ldots, I_{17\times17}(x, y, \theta_s)] \quad (3)$$

where I_{MAX} is the output of max pooling, and $I_{i\times i}(x, y, \theta_s)$ denotes the feature map with scale $i \times i$ and orientation θ_s.

Max pooling is also performed over local neighborhoods with windows varying from 8×8 to 12×12 and a shifting step of 2 pixels. This max-like feature selection operation is a key mechanism for object recognition in the cortex and provides a more robust response in the case of recognition in clutter or with multiple stimuli in the receptive field [36]. It successfully achieves invariance to image-plane transforms such as translation and scaling. An example of max pooling over local neighborhoods is given in Fig. 2. The procedure of Gabor feature map extraction is illustrated in Fig. 3.

2) Gaussian CS Feature Map: Center-surround (CS) fields have long been identified in the human visual system as having properties of edge enhancement that facilitate the detection, location, and tracking of small objects [37]. After the CS operation, features with different scales, such as edges and boundaries, are enhanced and segregated into a series of subband images. The Gaussian CS feature map is also inspired from neuroscience [37] similarity by mimicking perception of nerve cells that commonly respond to the dramatic change of colors (i.e., the dark pixels surrounded by bright ones or the bright pixels surrounded by dark ones). CS operation has been successfully used to capture the intensity information for scene classification in [38]. We first construct a seven-level Gaussian pyramid on each frame (pose) of the input action sequence. For one given pose that is viewed as the first level of pyramid, the Gaussian pyramid can be built by successfully convolving a Gaussian filter (with $\sigma = 2$) with several copies of the original pose image with reduced resolutions obtained by downsampling. This way, we obtain the corresponding Gaussian

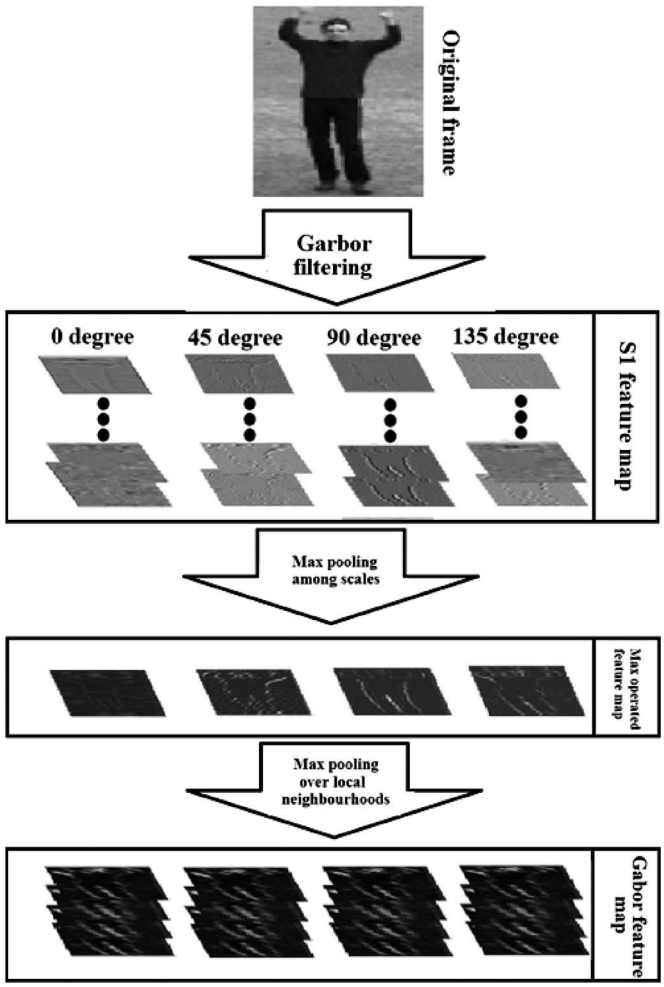


Fig. 3. Outline of the Gabor feature map extraction.

pyramid. To be precise, the construction of the Gaussian pyramid is shown as follows:

$$W(x,y) = \frac{1}{(\sqrt{2\pi}\sigma)^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

$$\text{Gaussian}_{\text{level}=l}(i,j) = \sum_x \sum_y W(x,y)\text{Gaussian}_{\text{level}=l-1}(2i+x, 2j+y) \tag{5}$$

where l denotes the levels of a Gaussian pyramid, and (i,j) represents the position of a pixel in the pose image.

The Gaussian CS feature map is then computed by subtracting point-by-point between different center levels (we use center level $= 2,\ 3$) and surrounded levels (we use surround level $=$ center level $+\ d$, where $d = 3,\ 4$). However, because scales are different between center levels and surround levels, images of surround levels are interpolated to the same size as the corresponding center level, and then, they are subtracted point-by-point by the corresponding center levels to generate the relevant subband images. As a result, four levels of feature map (i.e., levels of 2-5, 2-6, 3-6, and 3-7) are calculated as our Gaussian CS feature map. An example of the CS operation is illustrated in Fig. 4.

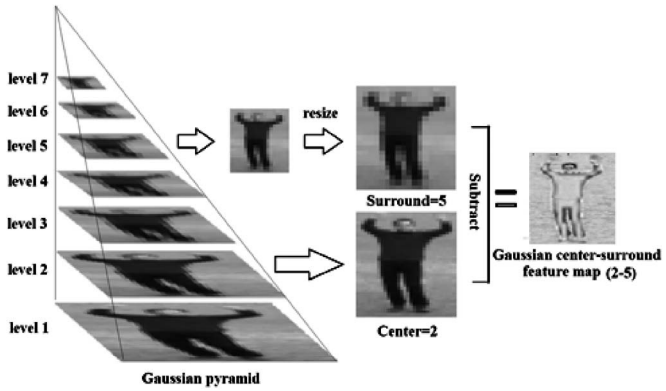


Fig. 4. Illustration of the Gaussian CS feature map.

3) Wavelet Laplacian Pyramid Feature Map: Wavelet transform [39], [40] has been an efficient way of feature extraction. It decomposes the input image into low- and high-frequency bands that carry the coarse and detail information, respectively. In our method, we consider to use the CDF "9/7" wavelet [41] to construct a wavelet Laplacian pyramid, which is proved to be an effective technique for multiresolution analysis, successfully obtaining the contour information of action poses.

Similar to the Gaussian CS feature map, we build our wavelet Laplacian pyramid by first using the five-level 2-D CDF "9/7" wavelet decomposition that generates the coefficient matrices of the approximation (cA) and horizontal, vertical, and diagonal details (cH, cV, and cD, respectively). We utilize the five-level wavelet decomposition on each frame (pose) from a given action sequence and only adopt the approximation (cA) of each level to build a CDF "9/7" pyramid. To further compute the wavelet Laplacian pyramid feature map, we make the difference between each two CDF "9/7" pyramid adjacent levels, which have been already interpolated into the same size (i.e., $l_i = W_{i+1} - W_i$, where W_i and W_{i+1} are the adjacent levels from the multilevel CDF "9/7" pyramid; l_i denotes the obtained level of the wavelet Laplacian pyramid). For this case, a four-level wavelet Laplacian pyramid feature map has been calculated to extract the contour information of action poses.

4) EPF Representation: We obtain our EPF representation by flattening the Gabor, Gaussian CS, and wavelet Laplacian pyramid feature maps into a 1-D feature vector. The obtained EPFs provide an informative representation of poses capturing multiple features, including orientation, intensity, and contour. In addition, Gabor filtering, the Gaussian pyramid, and the wavelet pyramid incorporate a multiresolution analysis and therefore enjoy the properties of invariance to scaling. To make a compact representation, we further adopt principal component analysis to reduce the high-dimensional feature vector (1656D) into a low-dimensional space by keeping the 99% principal components. The outline of our EPF extraction framework is visualized in Fig. 5.

B. Key-Pose Selection by AdaBoost

Each frame in a raw N-frame action sequence has been represented by the EPFs. The AdaBoost learning algorithm is

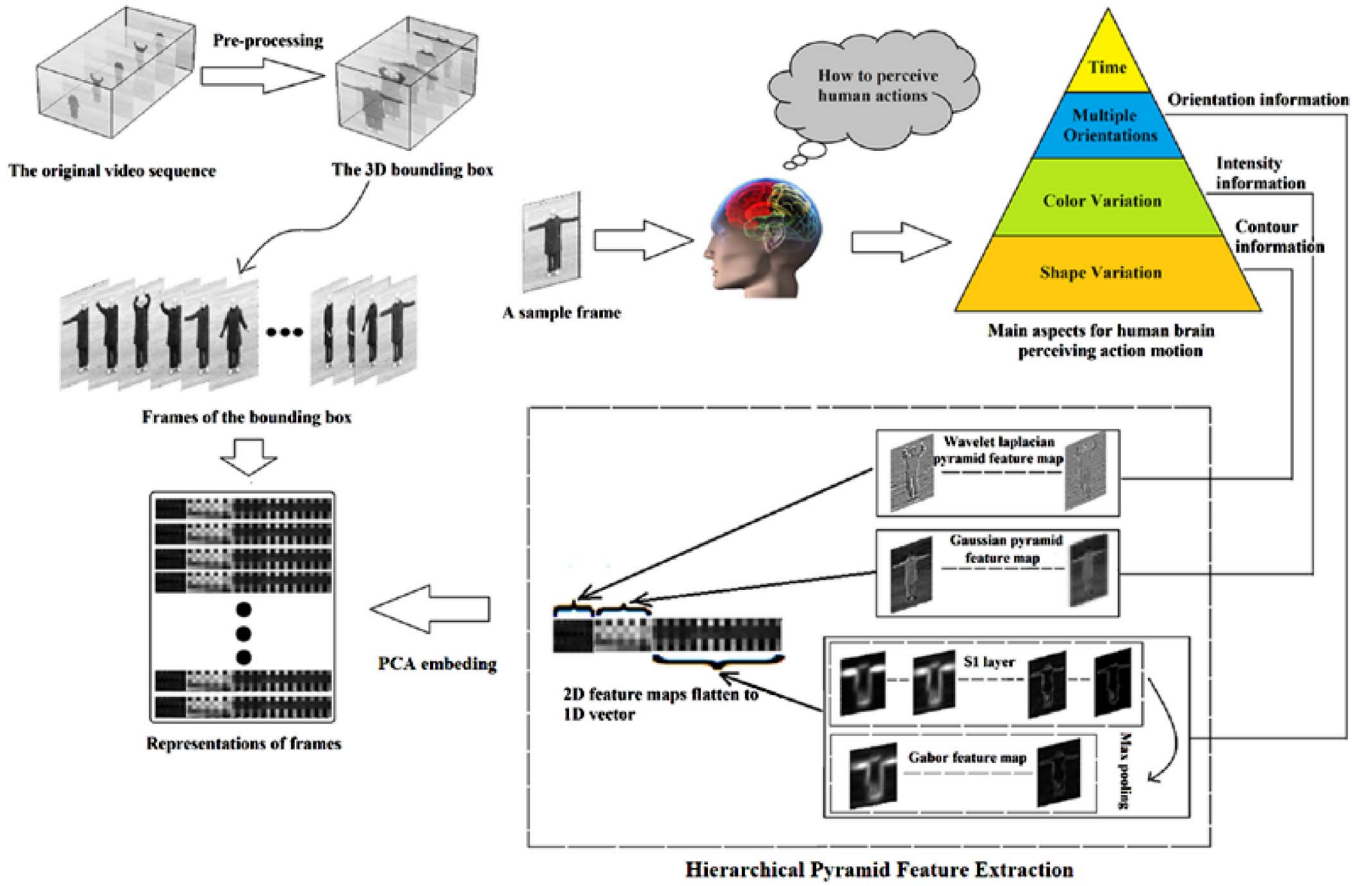


Fig. 5. Proposed EPF extraction procedure.

adopted to select the most discriminative poses from a large pose feature pool to increase the final classification accuracy and reduce the computational cost.

AdaBoost is one of the most popular machine learning algorithms, which aims to construct a strong classifier from several weak ones. Given a training set $\{(x_1, y_1) \ldots, (x_N, y_N)\}$, where x_i is a pattern and $y_i \in \{+1, -1\}$ is the class label of the corresponding pattern. At first, all training patterns are assigned with equal weights. In the learning phase, one weak classifier is trained, and all patterns are updated. Patterns that are incorrectly classified have their weights increased, and on the contrary, the weights of those patterns that are correctly classified are decreased. After all iterations of training, patterns that are consistently difficult to classify acquire larger weights, whereas easily classified patterns acquire even smaller weights. Here, we adopt the "classification and regression trees" [42] as our basic weak classifiers. The outline of the AdaBoost selection algorithm is shown in Algorithm 1.

Algorithm 1 AdaBoost Pose Selection

The training set: $(x_1, y_1), \ldots, (x_N, y_N)$, where x_i denotes the sample data, and $y_i \in \{1, -1\}$ stand for positive and negative samples, respectively.

First step

Initialize weights: $D_1(i) = (1/N)$;

Second step

For $t = 1, \ldots, T$:

1. For each feature i, a weak classifier $h_t : X \to \{-1, 1\}$ is trained and calculated. The error is evaluated with respect to $\varepsilon_t = P_{D_t}(h_t(x_i) \neq y_i)$.
2. Choose the classifier with the lowest error in each iteration.
3. Calculate the weight of this weak classifier: $\alpha_t = (1/2)\ln((1 - \varepsilon_t)/\varepsilon_t)$;
4. Update the weight of training data: $D_{t+1}(i) = (D_t(i)\exp(-\alpha_t y_i h_t(x_i))/Z_t)$;

End

Final step

Select those features with the smallest weight values.

Since poses from different classes would share similar features, for instance, some poses from Running and Jogging in the KTH data set are quite similar, these poses tend to confuse the classifier in the recognition stage. We would like to select the most discriminative poses for each class. The AdaBoost learning algorithm is then employed to select a subset of poses that are assigned with lower weights and are the most easily classified patterns in the boosting stage. For each action category, we select the top 25% most discriminative poses (with the lowest weights) from a whole sequence as the key poses. It is demonstrated that applying the AdaBoost learning algorithm can successfully reject those unrepresentative poses and dramatically increase recognition rates.

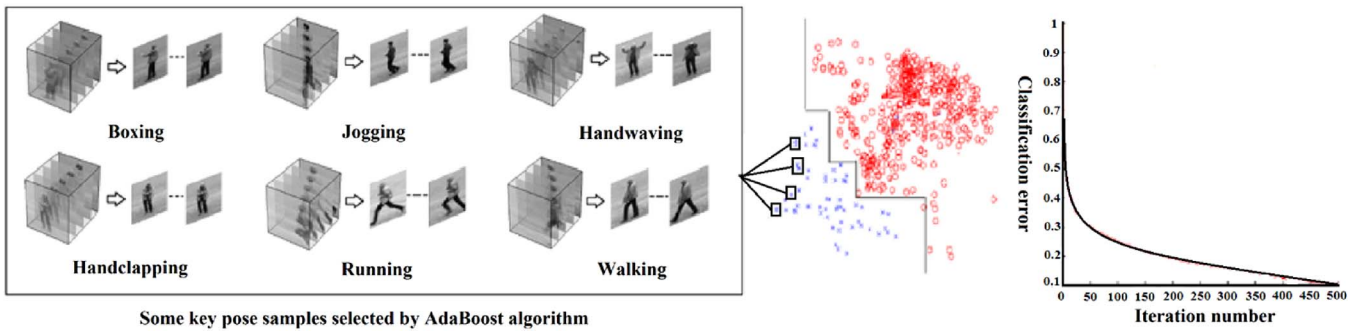


Fig. 6. (Left) Some selected key-pose samples from six different actions in the KTH data set. (Middle) Final AdaBoost boundary for pattern classification (e.g., patterns in one side are from a certain type of action, and patterns in the other side are from all other action types). It is noteworthy that the key poses are remote from the decision boundary. (Right) Classification error with the increase in iteration during AdaBoost learning.

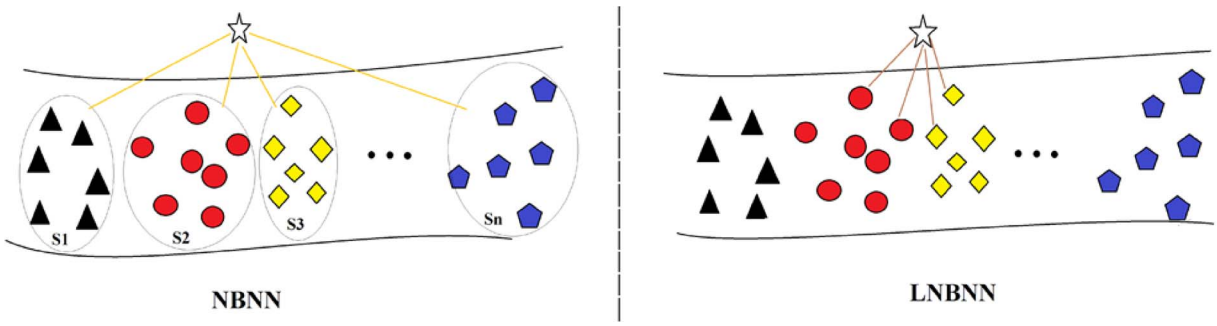


Fig. 7. (Left) NBNN algorithm finding the nearest neighbors from each of the classes (different shapes mean different classes in this figure). (Right) LNBNN algorithm only searching the local neighbors. The kNN may come from only some of the classes.

As the traditional AdaBoost learning algorithm is for binary classification, for the classification of multiple classes, a technique named "one-against-all decomposition"[1] has been proposed, in which the AdaBoost classifier is trained between one action type and all the other action types in the training set, and this procedure is repeated for every action type in the data set. This decomposition technique is an extension of the arbitrary binary method to a multiclass one. The decomposition method splits the original multiclass problem to a series of simpler binary problems that can be solved by the given binary method. The resulting binary classifiers are then properly combined together to form the multiclass classifier. The related algorithm is shown in the following equation:

$$q(x) = \arg\max_{y \in \varphi} f_y(x). \tag{6}$$

where $q : \chi \to \varphi$ is the trained multiclassifier, $f_y \in \chi \to R$, and $y \in \varphi$ are the real labels. Therefore, the most discriminative poses can be selected from the corresponding sequences for each action type. The key-pose selection procedure is illustrated in Fig. 6.

C. WLNBNN

The SVM kernel machines on the bag-of-features model have dominated the classification on local features; however, it suffers from the quantization error. Boiman *et al.* [23] proposed a simple nearest-neighbor-based classifier named NBNN, which employs nearest neighbor distances in the space of the local image descriptors instead of in the space of images. NBNN computes direct image-to-class distances without descriptor quantization. Although NBNN is extremely simple, efficient, and requires no learning/training phase, its performance ranks among the top leading learning-based image classifiers.

Recently, McCann and Lowe [24] have introduced locality to NBNN and proposed the LNBNN classifier for image classification, which outperforms the original NBNN classifier. For a query from the test set, LNBNN searches kNN regardless of their class labels, instead of finding the nearest neighbor in each class. As a result, the classification problem converts from "Does this feature descriptor look like one from class A?, class B?, class C?, ..." to "What does the feature descriptor look like?" Fig. 7 shows the schematic of LNBNN, and more relative details can be found in [24].

Based on the LNBNN algorithm, we further improve the NBNN classifier by assigning weights to the Euclidean distances between a query descriptor and the nearest exemplar set. The weights are related to the number of descriptors of each action type appearing in kNN search. If m poses in the kNN of the descriptor d_i fall in a certain class, the weight for this class is k/m. Our detailed WLNBNN is visualized in Algorithm 2. Since the size of the exemplar set has been dramatically reduced by the AdaBoost selection process compared with the primal number of raw data, and the WLNBNN classifier applies kNN search, which is much faster than searching the nearest

[1]http://cmp.felk.cvut.cz/cmp/courses/recognition/eprc/node7.html.

TABLE I
COMPARISON OF ACTION RECOGNITION ACCURACY (%) ON THE KTH DATA SET WITH DIFFERENT METHODS

Methods \ Actions	Boxing	Handclapping	Handwaving	Jogging	Running	Walking	Average
EPF+AdaBoost+WLNBNN (k=25)[2]	95	97	**98**	**89**	87	98	**94.8**
EPF+AdaBoost+LNBNN (k=25)[3]	96	96	98	89	88	97	94.0
EPF+AdaBoost+NBNN	93	95	95	86	83	96	91.8
EPF+WLNBNN (k=25)	91	93	94	84	81	93	89.3
EPF+AdaBoost[4]	88	90	92	79	74	91	85.7
EPF+BOW+SVM[5]	89	91	93	84	80	94	88.5
Dollár *et al.* [2]	93	77	85	57	85	90	81.2
Niebles *et al.* [4]	**98**	86	93	53	**88**	82	83.3
Taylor *et al.* [48]	-	-	-	-	-	-	90.0
Ji *et al.*[15]	90	94	97	84	79	97	90.2
Jhuang *et al.* [49]	92	**98**	92	85	87	96	91.7
Schindler and van Gool [17]	-	-	-	-	-	-	92.7
Le *et al.* [50]	-	-	-	-	-	-	93.9
Liu and Shah [51]	**98**	95	96	**89**	87	**100**	94.2

neighbor in each class, the running time of our final classification is greatly reduced.

Algorithm 2 WLNBNN classification

Require: An exemplar set $\{p_i\}$ consisting of descriptors from all classes.
Require: A query Q consisting of descriptors from a certain class.
For all descriptors $d_i \in Q$ **do**
  $\{p_1, p_2, p_3, \ldots, p_k\} \leftarrow \mathrm{NN}(d_i, k)$ (NN means finding the nearest neighbor)
  For all categories C found in the kNN **do**
    $\mathrm{dist}_C = (\min_{p_j|\mathrm{Class}(p_j)=C} \|d_i - p_j\|^2) \times \mathrm{weight}$
    $(\mathrm{weight} = [k/\#(\mathrm{Class}(p_j) = C)])$
    $\mathrm{totals}[C] \leftarrow \mathrm{totals}[C] + \mathrm{dist}_C$
  End
End
Return $\arg\min_C \mathrm{totals}[C]$.

IV. EXPERIMENTS AND RESULTS

We systematically test our proposed method on four different action data sets, namely, KTH [43], Weizmann [44], IXMAS [45], and HMDB51 [46].

A. Data Sets

The *KTH* data set is a commonly used benchmark action data set with 599 video clips. Six human action classes, including walking, jogging, running, boxing, hand waving, and hand clapping, are performed by 25 subjects in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3), and indoors with lighting variation (s4). We adopt the leave-one-person-out cross validation, i.e., videos of 24 subjects are used as training data, and videos of the remaining one subject are used for testing. Following the preprocessing step mentioned in [47], the coarse 3-D bounding boxes are extracted from all the raw action sequences and further normalized into an equal size of $100 \times 100 \times 60$.

Fig. 8. Some failure recognition samples on the KTH data set.

TABLE II
COMPARISON OF ACTION RECOGNITION ACCURACY (%) ON THE WEIZMANN DATA SET WITH DIFFERENT METHODS

Methods	Accuracy
EPF+AdaBoost+WLNBNN (k=25)[2]	**100**
EPF+AdaBoost+LNBNN (k=30)[3]	100
EPF+AdaBoost+NBNN	99.2
EPF+WLNBNN (k=25)	98.5
EPF+AdaBoost[4]	95.2
EPF+BOW+SVM[5]	97.8
Niebles *et al.* [4]	72.8
Jhuang *et al.* [49]	98.8
Yang *et al.* [52]	99.4
Fathi and Mori [33]	**100**

The *Weizmann* data set contains 93 video sequences showing nine different people, each of which performing ten different actions. We extract the bounding boxes by using foreground masks that are provided with the original data set and normalize them into the size of $100 \times 100 \times 60$. The same leave-one-person-out evaluation scheme is adopted on this data set.

The *IXMAS* data set is a multiview data set that contains 11 action classes. Each action is repeatedly executed three times by ten actors and recorded by five cameras simultaneously.

TABLE III
COMPARISON OF ACTION RECOGNITION ACCURACY (%) ON THE IXMAS DATA SET WITH DIFFERENT METHODS

Methods \ Actions	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5	Camera 1-5 fusion
EPF+AdaBoost+WLNBNN (k=30)[2]	**85.6**	**91.2**	**88.6**	**86.2**	**82.3**	**94.5**
EPF+AdaBoost+LNBNN (k=25)[3]	83.1	90.0	86.2	84.9	82.0	94.0
EPF+AdaBoost+NBNN	84.1	88.2	85.8	85.1	79.4	93.1
EPF+WLNBNN (k=30)	82.0	86.4	83.5	83.1	75.9	91.7
EPF+AdaBoost[4]	79.2	82.5	83.1	80.5	70.7	87.7
EPF+BOW+SVM[5]	81.4	84.8	86.1	82.7	77.3	89.6
Varma and Babu [53]	76.4	74.5	73.6	71.8	60.4	81.3
Liu and Shah [51]	76.7	73.3	72.1	73.1	-	82.8
Wu *et al.* [54]	81.9	80.1	77.1	77.6	73.4	88.2
Weinland *et al.* [55]	-	-	-	-	-	93.3

These actions are checking watch, crossing arms, scratching head, sitting down, getting up, turning around, walking, waving, punching, kicking, and picking up. We preprocess all the action sequences into the size of $100 \times 100 \times 80$ and repeat the experiments we have mentioned earlier. The same leave-one-person-out evaluation scheme is adopted on this data set.

The *HMDB51* data set collects 6849 action sequences from various movies and online videos. In our case, we adopt 2241 sequences from 19 body action categories (i.e., cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, and wave) as our research data. In our experiments, bounding boxes have been extracted from all the sequences through masks released with the data set and initialized into the size of $250 \times 300 \times 150$. Following the methods in [46], we perform our evaluation on the three split settings.

B. Comparison Results

The experimental results on the KTH data set are illustrated in Table I. Our method achieves an average recognition rate of 94.8%, which outperforms the state-of-the-art techniques. In addition, we have evaluated the newly proposed WLNBNN classifier. In Table I, we can see that WLNBNN outperforms the baseline SVM, LNBNN, and NBNN classifiers. Fig. 8 shows some failure cases in the recognition. It can be observed that two actions, i.e., Jogging and Running, which share similar motion patterns, are hard to distinguish even by human eyes. In addition, the comparison between with and without AdaBoost indicates that the employed AdaBoost algorithm can successfully select the most discriminative poses and improve the recognition performance.

The results on the Weizmann data set are shown in Table II. As expected, our method achieves a perfect 100% recognition rate, since the Weizmann data set is a relatively "easy" data set with greater interclass variations. For comparison, we have also included results of previous work. The proposed WLNBNN classifier outperforms the SVM, NBNN, and LNBNN classifiers, and the AdaBoost pose selection contributes to the recognition performance as well.

On the IXMAS data set, we have evaluated our method not only on each single view but also on multiple views by combining the data from all five cameras. The overall accuracy we obtain by applying multiview fusion achieves a recognition rate of 94.5%, which significantly exceeds all the other results listed in Table III. For each single view, our method achieves the best results among all the methods. The WLNBNN classifier outperforms the SVM, NBNN, and LNBNN classifiers, and the AdaBoost pose selection improves the accuracy both on single and multiple views.

TABLE IV
CLASSIFICATION ACCURACY (%) ON THE HMDB51 DATA SET

Methods \ Splits	Split1	Split2	Split3	Average
EPF+AdaBoost+WLNBNN (k=40)[2]	**49.1**	**54.3**	**44.6**	**49.3**
EPF+AdaBoost+LNBNN (k=20)[3]	47.3	51.1	45.7	48.0
EPF+AdaBoost+NBNN	47.3	52.5	42.9	47.6
EPF+WLNBNN (k=40)	45.0	49.4	40.9	45.1
EPF+AdaBoost[4]	43.8	40.6	38.1	40.8
EPF+BOW+SVM[5]	45.7	48.4	39.2	44.4

The HMDB51 data set is a quite challenging data set; however, our method can still provide relatively good recognition rates on three split settings based on [46] (see Table IV). As far as we are aware, this is the first report of action recognition on this subset of 19 body action categories; therefore, we do not compare with other recognition results, and we only present ours in Table V. Consistent with the other three data sets, the proposed WLNBNN classifier performs better than the SVM, NBNN, and LNBNN classifiers on this data set. The AdaBoost pose selection improves the results as well.

C. Performance Analysis

To evaluate the parameters of the proposed method, we have also conducted analysis experiments on the four data sets. As both the WLNBNN and LNBNN classifiers use kNN search, we investigated the effects of k on the performance of WLNBNN and LNBNN. The results on the four data sets are shown in Fig. 9. The proposed WLNBNN classifier achieves higher accuracy than the LNBNN and NBNN classifiers on all the four data sets.

In addition, as in the representation of poses, we combine the Gabor features, the Gaussian pyramid, and the wavelet pyramid, and we would like to evaluate the individual

TABLE V
EVALUATION OF INDIVIDUAL FEATURES ON THE KTH, WEIZMANN, IXMAS, AND HMDB51 DATA SETS. NB: 1) FOR WLNBNN CLASSIFICATION, THE NUMBER OF NEAREST NEIGHBORS k ON EACH DATA SET IS CONSISTENT WITH PREVIOUS TABLES. 2) THE ACCURACY VALUES OF THE IXMAS DATA SET IN THIS TABLE DENOTE THE MULTIVIEW FUSION RESULTS

Methods \ Dataset	KTH	Weizmann	IXMAS	HMDB51
Gabor feature map+AdaBoost+WLNBNN	87.5	95.2	89.4	42.1
Gaussian center-surround feature map+AdaBoost+WLNBNN	89.2	94.9	90.2	43.4
Wavelet Laplacian pyramid feature map+AdaBoost+WLNBNN	85.0	93.7	86.1	35.5
EPF+AdaBoost+WLNBNN	**94.8**	**100**	**95.5**	**49.3**

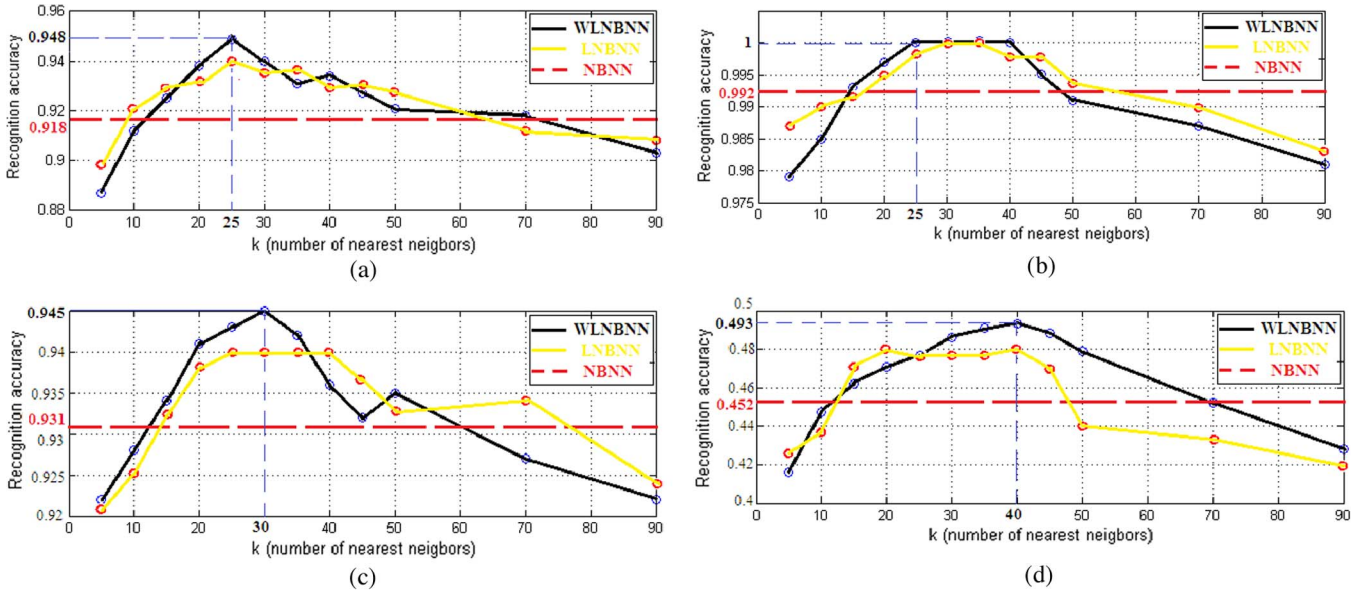


Fig. 9. Evaluation of the effects of k on the performance of the proposed method on the (a) KTH, (b) Weizmann, (c) IXMAS, and (d) HMDB51 data sets.

TABLE VI
COMPARISON OF COMPUTATIONAL COMPLEXITY (IN SECONDS) BETWEEN NBNN and WLNBNN

Methods \ Dataset	KTH	Weizmann	IXMAS	HMDB51
NBNN	37s	22s	92s	1105s
WLNBNN	23s	14s	56s	227s

contribution of each feature to the representation. The results are illustrated in Table V. It is obviously manifested that the EPFs, i.e., the combination of the Gabor features, the Gaussian pyramid, and the wavelet pyramid, outperform any individual feature. The results validate the effectiveness of the proposed EPFs.

To demonstrate the efficiency of the proposed WLNBNN classifier, we compare the classification time between NBNN and WLNBNN on four different data sets in Table VI. The results show that our WLNBNN classifier runs faster and more accurate than the original NBNN.

Additionally, we have also evaluated the computational cost of each component and the whole system. Table VII shows the time comparison of the methods with and without AdaBoost selection on the KTH data set. From the results, we can observe that the AdaBoost key-pose selection is time consuming; however, the computational time in the classification phase is significantly reduced while producing higher accuracy. The same trend would exist on the other three data sets.

V. CONCLUSION

In this paper, we have presented a new method based on key-pose selection for human action recognition. Poses from each video sequence are described by invariant and informative EPFs constructed by computing the relevant Gabor, Gaussian pyramid, and wavelet Laplacian pyramid feature maps. AdaBoost is then employed to learn the most discriminative key poses to represent actions. With the boosted poses for each action, a new classifier named WLNBNN is proposed for action classification.

We have further systematically evaluated our proposed method on four different data sets, i.e., KTH, Weizmann, IXMAS, and HMDB51 data sets, and obtained superior results for action recognition over previously published works. Typically, the AdaBoost algorithm takes much time to learn the discriminative key poses; however, it only needs to be carried out once on the training set, and the computational cost will be greatly reduced after the key-pose selection.

TABLE VII
ANALYSIS OF THE COMPUTATIONAL COMPLEXITY FOR THE ENTIRE SYSTEM ON THE kTH DATA SET

Methods \ modules	EPF extraction	AdaBoost selection	WLNBNN classification	Accuracy
EPF+AdaBoost+WLNBNN (k=40)	1.02×10^3 seconds	2.24×10^4 seconds	23 seconds	94.8%
EPF+WLNBNN (k=40)	1.02×10^3 seconds	-	85 seconds	89.3%

REFERENCES

[1] D. Peng, D. Huijun, D. Ligeng, T. Linmi, and X. Guangyou, "Group interaction analysis in dynamic context," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 275–282, Feb. 2008.

[2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. IEEE Int. Workshop Visual Surveillance Perform. Eval. Track. Surveillance*, Beijing, China, 2005, pp. 65–72.

[3] Y. Xu, D. Xu, S. Lin, T. Han, X. Cao, and X. Li, "Detection of sudden pedestrian crossings for driving assistance systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 729–739, Jun. 2012.

[4] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial–temporal words," *Int. J. Comput. Vis.*, vol. 79, no. 3, pp. 299–318, Sep. 2008.

[5] W. Bian, D. Tao, and Y. Rui, "Cross-domain human action recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 298–307, Apr. 2012.

[6] T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," in *Proc. 37th IEEE Appl. Imagery Pattern Recog. Workshop*, Washington, DC, 2008, pp. 1–8.

[7] S. Zhang, M. H. Ang, W. Xiao, and C. K. Tham, "Detection of activities for daily life surveillance: Eating and drinking," in *Proc. 10th Int. Conf. e-health Netw., Appl. Serv.*, Singapore, 2008, pp. 171–176.

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[9] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia*, Augsburg, Germany, 2007, pp. 357–360.

[10] A. Klaser and M. Marszalek, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. 19th Brit. Mach. Vis. Conf.*, Leeds, British, 2008, pp. 995–1004.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[12] I. Laptev, "On space–time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2/3, pp. 107–123, Sep. 2005.

[13] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 3, pp. 710–719, Jun. 2005.

[14] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[15] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[16] H. Lu, G. Fang, X. Shao, and X. Li, "Segmenting human from photo images based on a coarse-to-fine scheme," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 889–899, Jun. 2012.

[17] K. Schindler and L. Van Gool, "Action snippets: How many frames does human action recognition require?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[18] S. Baysal, M. Kurt, and P. Duygulu, "Recognizing human actions using key poses," in *Proc. Int. Conf. Pattern Recog.*, Istanbul, Turkey, 2010, pp. 1727–1730.

[19] X. Cao, B. Ning, P. Yan, and X. Li, "Selecting key poses on manifold for pairwise action recognition," *IEEE Trans Ind. Informat.*, vol. 8, no. 1, pp. 168–177, Feb. 2012.

[20] L. Shao, D. Wu, and X. Chen, "Action recognition using correlogram of body poses and spectral regression," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, 2011, pp. 209–212.

[21] S. Cheema, A. Eweiwi, C. Thurau, and C. Bauckhage, "Action recognition by learning discriminative key poses," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Barcelona, Spain, 2011, pp. 1302–1309.

[22] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Comput. Learn. Theory*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[23] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[24] S. McCann and D. Lowe, "Local naive Bayes nearest neighbor for image classification," *Arxiv preprint arXiv:1112.0059*, 2011.

[25] C. Thurau and V. Hlavác, "Pose primitive based human action recognition in videos or still images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, San Diego, CA, 2005, vol. 1, pp. 886–893.

[27] J. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Minneapolis, MN, 2007, pp. 1–8.

[28] W. Yang, Y. Wang, and G. Mori, "Recognizing human actions from still images with latent poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, San Francisco, CA, 2010, pp. 2030–2037.

[29] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, San Juan, Puerto Rico, 1997, pp. 762–768.

[30] M. Cooper and J. Foote, "Discriminative techniques for keyframe selection," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, 2005, p. 4.

[31] Z. Zhao and A. Elgammal, "Information theoretic key frame selection for action recognition," in *Proc. Brit. Mach. Vis. Conf.*, Leeds, U.K., 2008, pp. 95–104.

[32] Y. Zhuang, Y. Rui, T. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proc. Int. Conf. Image Process.*, Chicago, IL, 1998, vol. 1, pp. 866–870.

[33] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[34] L. Shen and L. Bai, "Adaboost Gabor feature selection for classification," in *Proc. Image Vis. Comput.*, 2004, pp. 77–83.

[35] I. Laptev and P. Prez, "Retrieving actions in movies," in *Proc. Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[36] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nat. Neurosci.*, vol. 2, pp. 1019–1025, 1999.

[37] C. Yang, M. Schmalz, W. Hu, and G. Ritter, "Center-surround filters for the detection of small targets in cluttered multispectral imagery: Background and filter design," in *Proc. SPIE*, 1995, vol. 2496, pp. 637–648.

[38] D. Song and D. Tao, "Biologically inspired feature manifold for scene classification," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 174–184, Jan. 2010.

[39] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[40] I. Daubechies, "The wavelet transform, time–frequency localization and signal analysis," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.

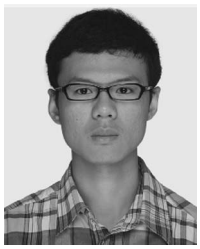
[41] A. Cohen, I. Daubechies, and J. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, Jun. 1992.

[42] G. De'ath and K. Fabricius, "Classification and regression trees: A powerful yet simple technique for ecological data analysis," *Ecology*, vol. 81, no. 11, pp. 3178–3192, Nov. 2000.

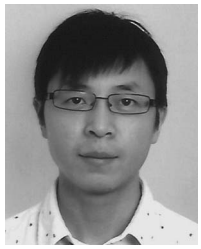
[43] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. Int. Conf. Pattern Recog.*, 2004, vol. 3, pp. 32–36.

[44] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space–time shapes," in *Proc. Int. Conf. Comput. Vis.*, Beijing, China, 2005, vol. 2, pp. 1395–1402.

[45] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–7.
[46] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. 13th IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 2556–2563.
[47] A. Yao, J. Gall, and L. Van Gool, "A Hough transform-based voting framework for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, San Francisco, CA, 2010, pp. 2061–2068.
[48] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. Eur. Conf. Comput. Vsi.*, Crete, Greece, 2010, pp. 140–153.
[49] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–8.
[50] Q. Le, W. Zou, S. Yeung, and A. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3361–3368.
[51] J. Liu and M. Shah, "Learning human actions via information maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.
[52] M. Yang, F. Lv, W. Xu, K. Yu, and Y. Gong, "Human action detection by boosting efficient motion features," in *Proc. Int. Conf. Comput. Vis. Workshops*, Kyoto, Japan, pp. 522–529.
[53] M. Varma and B. Babu, "More generality in efficient multiple kernel learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, NY, 2009, pp. 1065–1072.
[54] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Providence, RI, 2011, pp. 489–496.
[55] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Comput. Vis. Image Understand.*, vol. 104, no. 2/3, pp. 249–257, Nov. 2006.

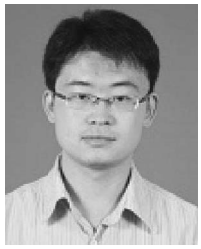


Li Liu received the B.Eng. degree in electronic information engineering from Xi'an Jiaotong University, Xi'an, China, in 2011. He is currently working toward the Ph.D. degree in the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, U.K.



Ling Shao (M'09–SM'10) is currently a Senior Lecturer (Associate Professor) in the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K. and a Guest Professor with Nanjing University of Information Science and Technology, China. Before joining The University of Sheffield, he was a Senior Scientist for four years with Philips Research, The Netherlands. He has published more than 80 academic papers in refereed journals and conference proceedings. He is the holder of more than ten awarded patents and patent applications. His research interests include computer vision, pattern recognition, and video processing.

Dr. Shao is a Fellow of the British Computer Society. He has organized several workshops with top conferences, such as ICCV, ACM Multimedia, and ECCV. He has been a Program Committee Member for many international conferences, including CVPR, ECCV, ICIP, ICASSP, ICME, ICMR, ACM MM, CIVR, and BMVC. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS and several other journals.



Xiantong Zhen received the B.S. and M.E. degrees from Lanzhou University, Lanzhou, China, in 2007 and 2010, respectively. He is currently working toward the Ph.D. degree in the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, U.K.

Xuelong Li (M'02–SM'07–F'12) is a Full Professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.