Activity Recognition and Prediction with Pose based Discriminative Patch Model

Song Cao, Kan Chen and Ram Nevatia
University of Southern California, Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089, USA
songcao@usc.edu

Abstract

We describe an image based activity recognition solution which can be applied to both off-line video classification and activity prediction in frames. We propose a Pose based Discriminative Patch Model to make activity recognition and prediction on image level (only observing several frames). This model enables a general and flexible framework to add in discriminative patches and consider their mutual relations to an efficient tree structure. PDP makes contribution in two aspects: (1) PDP provides a novel solution to improve activity recognition and prediction, by utilizing pose based discriminative patches instead of pose configuration feature, and modeling the patches' mutual relations. (2) PDP is an image-based algorithm, so it can make predictions using limited frames, even a single image. PDP focuses on challenging data captured from Internet and movies, where we achieve a 6% improvement compared with state-of-the-art method on video level recognition dataset - Sub-JHMDB, and image level action recognition dataset. We also obtain good improvement on activity prediction task.

1. Introduction

Activity recognition is an important problem for a number of applications. Activities can be recognized in a video in off-line fashion, or be predicted in an on-line fashion from few frames, or even be recognized from a single image. The off-line video recognition is well explored using Dense Trajectory feature, while the rest two topics (recognition and prediction in few frames or images) remain not well explored. In this paper, our target is to provide reliable recognition within limited frames. we present a unified framework for all three variations of the same basic task. Of course, some cues, such as motion, are not available in single image task and extended trajectories may not be available for prediction. We posit that human pose can be a common feature.

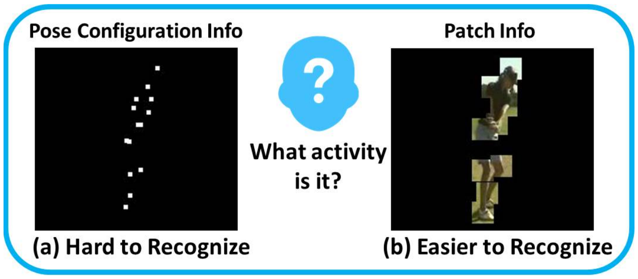

Figure 1. Pose Configuration versus Patch. (a) Groundtruth Joint Location; (b) Patches extracted at Joint Location using [38]. It is easier to recognize the activity *Playing Golf* from (b) than (a).

Utilization of pose information in activity prediction has been limited, possibly due to the difficulty of inferring accurate pose from images. Do we really need to wait for a robust pose estimation algorithm to leverage all information from human poses? Here is an inspiring observation. In Figure 1, it is difficult for a human to recognize activity from Figure 1 (a) which is the groundtruth of pose joint locations. However if we observe the patches extracted at joint locations which are predicted by [38] in Figure 1 (b), it is easier to recognize the activity as *Playing Golf*.

Motivated by this observation, we propose a method which can exploit pose information without the necessity of estimating accurate pose first. Instead, we extract discriminative image patches that are associated with human joint positions. Experiments have shown favorable improvement in activity prediction and recognition performance as well as providing satisfying pose estimation results. Besides, this approach can be further encoded into video prediction framework, which can benefit the current approaches in such field.

There has been renewed interest in using pose for activity recognition and prediction, partly due to availability of datasets with accurate pose (joint positions and angles) annotations. Jhuang *et al.* [8], compared recognition performance with groundtruth low level features (segmentation mask and optical flow), middle level features (human location and size) and high level features (human pose) and observe that high level features perform better than the other two, which shows features on semantic level can provide useful information in activity prediction and recognition.

However, current pose estimation algorithms are not robust enough to provide discriminative pose configurations. The challenges fall into two categories. First, most of activity recognition tasks in real applications suffer from frequent occlusions and blurred motion. The video quality is quite different from standard pose estimation datasets, which are often captured in a lab environment. Pose estimation algorithms developed for such condition do not necessarily fit well the activity recognition video data. Second, the current pose estimation benchmark (Probability of Correct Keypoint, PCK uses the matching threshold as 50% of the mean ground-truth segment length) is not precise enough to judge human pose estimation result for utility in activity recognition. For example, even when an estimated body joint angle is highly different from the true value, the PCK benchmark can still possibly count it as a correct pose estimation result. Thus, even a 100% correct pose estimation results under PCK benchmark might still not be a discriminative pose configuration for recognition tasks.

Our paper argues that, even though there are no robust pose estimation algorithms available yet, we can still utilize pose-based information and build a state-of-the-art activity recognition and prediction system, through the Pose based Discriminative Patch (PDP) model.

The PDP model (as shown in Figure 2) is a tree structure model that consists of multiple pose based discriminative patches. These discriminative patches are selected around the human joint locations, and relatively unique for a target activity. Inspired by Pictorial Structure model [2], we organize these patches into a graphical tree model. In the pipeline, we first train PDP activity model using initial PDP model as well as extract discriminative patches. Then we recognize the video or image by determining PDP activity model that has the best match. There are three distinct *properties* of PDP from research topics of activity recognition and prediction, human pose estimation and discriminative patches.

The *first property* of PDP is to use pose related patches to recognize activities instead of directly computing features from the pose configuration [8]. This method not only reduces the reliance on extremely accurate pose configuration, but also increases the representation capability obtained from human pose. Specifically, we observe that a classical HOG [6] features extracted from pose based patches are more helpful than features extracted from the pose configuration, based on current human pose estimation algorithms. Through modeling and learning a PDP, the patches become discriminative in recognizing activity. The PDP model enjoys advantages from human pose estimation where the patches are extracted from human joint locations, as well as classical features like HOG that can be utilized in our framework. From the representation view, it is more effective than directly extracting hand designed features from pose configuration. As shown in Figure 1, pose based patch information (b) is more effective than (a).

The *second property* of PDP is that it encodes the discriminative mutual relations between patches. Though [24, 9] discussed how to utilize patches in activity recognition before, their framework does not take the mutual relations into consideration. Besides, unlike the poselet [3] framework which adopts a voting scheme, we use a tree model to organize these patches. We model the relationship between two patches by recognizing their types and locations. By considering the pairwise information, the PDP has the capability to deal with more complex activities recognition.

The *third property* of PDP is that it can be applied into derivative applications besides recognizing human activities in videos. PDP can also recognize activities and estimate human poses in challenging still images, as shown by experiments in Section 4. In this way, PDP provides a possible solution for the activity prediction challenge.

The rest of the paper is organized as follows. The related work is in Section 2. Then we introduce the Pose based Discriminative Patch model in Section 3. This is followed by detailed experiments and analysis in Section 4. Finally, we state our conclusions in Section 5.

2. Related Work

For the purpose of pose estimation, multiple features have been adopted to better describe body parts, e.g. HOG [25], HOGcomb [1], Shape Context [2] and JRoG [28]. Information of color, shape and Poselet is also considered [26, 20, 21]. There are many limitations to such features. First, there is huge information loss between 2D images and 3D real world, which may be the reason why predominant pose estimation methods like Kinect [27] choose to use depth images. Second, self-occlusion and scene occlusion [7] in human poses are quite difficult to deal with, especially for human pose estimation which considers pairwise information. It is hard to use pictorial structure to predict the invisible parts occluded by surrounding objects [23]. Third, illumination variance, motion blur and other noise in images affect the accuracy of appearance model. All these factors have become barriers to achieve accurate and robust human pose estimation.

There are also related researches on predicting future activities in other domains of computer vision. Most of the works focused on predicting (or forecasting) the future trajectories of pedestrians [11][18]. There are also literatures on predicting motion from still images [40]. [13] utilizes hierarchical representation in activity. But their experiments focus more on interaction events only using 5 activities. However, our work focuses on predicting activity based on human pose information. The dataset used in our experi-

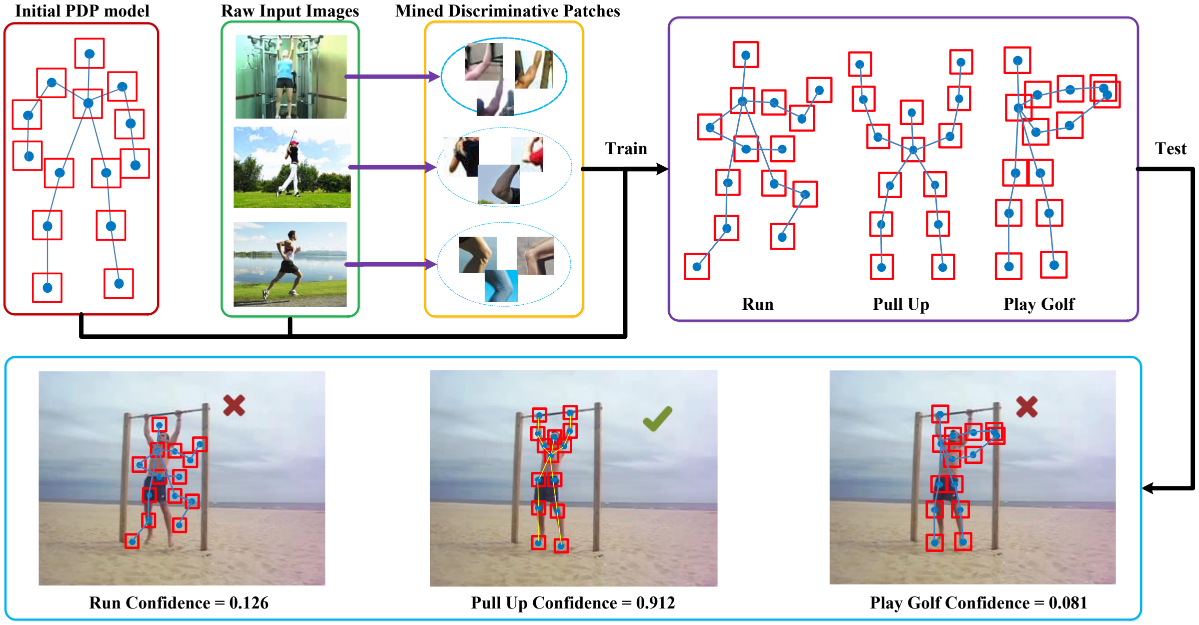

Figure 2. Illustration of PDP model. We train PDP activity models using an initial PDP model and mined discriminative patches; we recognize a video or an image by determining PDP activity model that has the best score. In this example, *pull up* gets the best confidence score for the test frame.

ments is more challenging which involves 12 activities, collected from Internet and sports.

For activity recognition, low and middle level features [34, 32] are most commonly adopted in complex datasets such as UCF101 [30], HMDB51 [12]. Other approaches decomposed videos into video segments [10, 33], action concepts [31, 19] and motion atoms [35] based on part-based idea. These approaches are promising while not fully utilize the information from human pose compared to our framework. Recently, [5] proposed an action recognition method using pose configuration which shows good results on Keck Gesture Dataset [14]. Similarly, we are trying to explore the idea of pose-based patches, but to avoid the limitations of feature-based pose configurations. In addition, there are some approaches trying to decompose an action or event into a set of discriminative patches or poselet [15]. In [9], Jain et. al. proposed mid-level discriminative patches based on Exemplar-SVM [16]. They encode the detection results into Bag-of-Words features and apply them for event classification. This method does not consider the correlation between different patches. In contrast, borrowing from the experience of human pose estimation approaches, we further takes advantage of tree-structured model to encode the mutual relations between patches.

In the aspect of human pose datasets, [29, 39] provide reliable pose estimation results which were used for experiments in most papers before. However, these dataset suffered problems of relatively simple activities and limited annotations. Recently, [8] provides a pose annotated activity recognition dataset called JHMDB, originated from HMDB51 [12]. Pose annotated JHMDB provides a sound foundation of utilizing human pose in boosting activity recognition. However, there is still urgent need for more complete and challenging human pose datasets in research areas.

3. Pose based Discriminative Patch model

Pose based Discriminative Patch (PDP) model is a general framework which shows capability in improving both activity recognition and human pose estimation. It takes advantage of discriminative patches, human pose and graphical model. In this section, we first show how to mine discriminative patches as PDP parts. Then, we introduce how to build the PDP model and compute PDP output. We demonstrate this for model representation, inference and learning. To reduce the computational load, we simplify the PDP model into an image level representation. Therefore, it can also be used in still image activity recognition and human pose estimation.

3.1. Mining Discriminative Patches

To begin with, we focus on how to find discriminative patches for each activity, using the joint annotations. A Pose Difference Feature for discriminative patch search is proposed. Similar to Poselets [3], which are also generated from joint locations, we develop a *Pose Difference Feature* as relational distances between patches after normalization. Then, we can extract features from different locations around the human body.

There are two main challenges in finding discrimina-

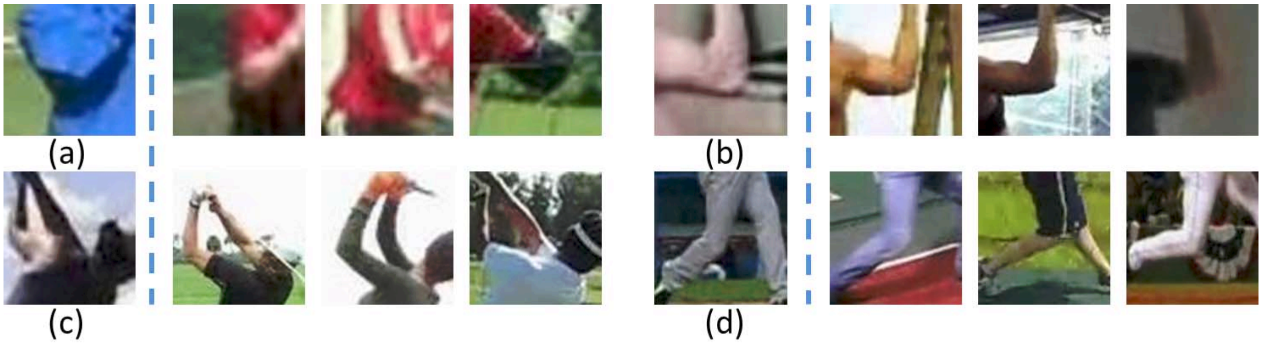

Figure 3. Discriminative Patch Samples. (a) Discriminative Patch indicates *catch*; (b) Discriminative Patch indicates *pull up*; (c) Discriminative Patch indicates *playing golf*; (d) Discriminative Patch indicates *swing baseball*.

tive patches. The first issue is the definition of *patch similarity*. As we have pose annotations for the training data, we use human joint locations to cluster patches instead of unsupervised clustering by HOG features. Specifically, we first define multiple seed patch locations (e.g. Left & Right Elbows) based on pose annotations. For the target patch P_i, we denote its neighbor patch (according to the PDP model) as the source patch P_j. The Pose Difference Feature for the target patch is determined by the relational vectors from the source patch P_j center to target patch P_i center, after body scale normalization. We define $\mathbf{d}_{ji} = [P_{jx} - P_{ix}, P_{jy} - P_{iy}]$, and $[j_1, j_2, j_3, \ldots, j_N]$ as the neighbor set of P_i. Given P_i and P_i', we define the similarity function $Similarity(P_i, P_i')$ in Equation 1, using the target patch P_i and its corresponding source pathes P_j.

$$Similarity(P_i, P_i') = \sum_{k=1}^{N} \| \mathbf{d}_{i,j_k} - \mathbf{d}_{i',j_k'} \|^2 \qquad (1)$$

where j_k comes from the neighbor set of P_i, and j_k' comes from the neighbor set of P_i'. The next issue is *how to define a discriminative patch*. Heuristically, the discriminative patch should frequently occur in the *intra-class* activity, while it should seldom occur in *inter-class* activities. Therefore, we use *patch purity* r_i and Algorithm 1 to evaluate how discriminative a patch is. We denote N_i as the total number of patches from the intra-class activity. When we sort all the N patches in non-descending order by patch similarity, we assume we get fn_i intra-class patches among the top N_i patches. We define patch purity as $r_i = fn_i/N_i$. The details of searching discriminative patches are shown in Algorithm 1.

We define patches at image level to decrease the complexity. To utilize the original pose annotation, we use the original 14 joints annotation (neglect the belly joint) as the seed patch centers. At each seed patch P_i center, we define a patch candidates set P_{jk}. Besides patch purity, we also keep the selected patch candidates to be relatively different within each seed patch set. After the patch selection procedure, we train the selected patches with HOG feature [6] and liblinear SVM [22].

In Figure 3, we show the patches we mine from JHMDB. (e.g. patch (a) is discriminative in indicating *catch* activity; patch (b) is discriminative in indicating the *pull up* activity). These discriminative patches will help us classify activity in complex videos and images.

Algorithm 1 Mining Discriminative Patches

```
Input: Candidate set for one seed patch center ${F}_M$,
Training Set ${T}_N$
for each patch $f_i$ in $F$ do
    for each patch $t_j$ in $T$ do
        Calculate score $s_{ij} \triangleq$ Similarity($f_i$, $t_j$) according to Equation (1)
    end for
    Sort $[s_{i1}, s_{i2}, ..., s_{iN}]$ in non-descending order
    Calculate the purity $r_i = fn_i/N_i$ of elements which have same label as $f_i$ in the first $N_i$ elements of sorted list ${s_i}_N$
end for
Sort $[r_1, r_2, ..., r_M]$ in non-increasing order
Greedy select $M_0$ Discriminative Patches ${P_i}_{M_0}$ in which their mutual similarity is over a threshold, in the sorted list ${r_i}_M$
Output: Discriminative Patches ${P_i}_{M_0}$ for the current seed patch center.
```

3.2. Representation of PDP

Next, we discuss how to organize discriminative patches into a Pose based Discriminative Patch model (PDP). As PDP is an image-based framework, we denote the frame to be tested as I. Assuming that PDP is composed of N patch candidate sets, we define the patch configuration as $P_{config} = (P_1, P_2, \ldots, P_N)$, where P_1 is the root patch (In experiment, we define root patch near the neck locations).

We name each patch in P_{config} as P_i, where i indicates the patch candidate set ID which the current patch belongs to. From each patch candidate set, we select only one patch, and locate it in its best location. By placing the patch P_i in a specific location, we obtain its location as $l_i = (x_i, y_i)$, as well as the patch descriptor f_i, and patch type t_i. We denote P_i to represent information from $[l_i, t_i, f_i]$.

To test frame I, we compute a Joint Feature Vector $\boldsymbol{\Phi}(I, P_{config})$ including Patch Appearance Feature, Patch Distance Feature and Patch Configuration Feature. Particularly, Patch Appearance Feature $\boldsymbol{\phi}(P_i)$ is used to describe the appearance when specific patch P_i is located in a specific location l_i. In experiments, we selected HOG as our appearance feature.

Patch Distance Feature describes the relative location of part i with respect to j. We use this feature to demonstrate mutual relation of two neighbour patches. Specifically, we define $\boldsymbol{\psi}(P_i, P_j) = [\mathrm{d}x, \mathrm{d}x^2, \mathrm{d}y, \mathrm{d}y^2]$, where $\mathrm{d}x = x_i - x_j, \mathrm{d}y = y_i - y_j$.

Patch Configuration Feature indicates the possibility of a patch configuration to be present in specific activity. We denote this feature by $\boldsymbol{\varphi}(P_{config})$. If a patch candidate or a pair of patch candidates belong to current patch configuration, we set 1 in its corresponding dimension. Specifically, We define $\boldsymbol{\varphi}(P_{config})$ in Equation 2, where $i, j \in [0, n-1]$.

$$\boldsymbol{\varphi}(P_{config}) = \begin{cases} \delta[i] & \text{If patch } i \text{ appears} \\ \delta[i \cdot \mathrm{n} + j] & \text{If patch } i, j \text{ both appear} \end{cases} \tag{2}$$

In summary, for each input image I, the Joint Feature Vector $\boldsymbol{\Phi}(I, P_{config})$ can be written as $\boldsymbol{\Phi}(I, P_{config}) = [\boldsymbol{\phi}(P_i), \boldsymbol{\psi}(P_i, P_j), \boldsymbol{\varphi}(P_{config})]$.

3.3. Inference of PDP

We define our model (demonstrated in Figure 2) as a tree-structured graph $G = (V, E)$, where inference can be done efficiently with dynamic programming. We represent model weight as $[\omega^a, \omega^d, \omega^c]$, where ω^a is the weight for Patch Appearance Feature, ω^d is the weight for Patch Distance Feature and ω^c is the weight for Patch Configuration Feature. Under the current model weight, the task for inference is to maximize the objective score by finding the best P_{config}.

Given an image input I and a specific patch configuration P_{config}, we compute the score in Equation 3.

$$\begin{aligned} score(I, P_{config}) &= \sum_{i \in V} \omega_i^a \cdot \boldsymbol{\phi}(P_i) \\ &+ \sum_{ij \in E} \omega_{ij}^d \cdot \boldsymbol{\psi}(P_i, P_j) + \omega^c \cdot \boldsymbol{\varphi}(P_{config}) \end{aligned} \tag{3}$$

Under current model weights, we maximize the score from inference of different pose configurations. We compute the message that patch i passes to its parent patch j by the following:

$$score_i(P_i) = \boldsymbol{\phi}(I, P_i) + \sum_{k \in kids(i)} m_k(P_i) \tag{4}$$

$$m_i(P_j) = \max_{P_i} [score_i(P_i) + \boldsymbol{\psi}_{ij}(P_i, P_j)] \tag{5}$$

From Equation 4, we start from the leaves and traverses to the parents; computing the local score of patch i, for all possible type t_i, and at all possible locations l_i, by collecting messages from the children of i. For every location and possible type of patch j, Equation 5 computes the best scoring location and type of its child patch i. Once messages are passed to the root patch ($i = 1$), $score_1(I, P_1)$ represents the best scoring configuration P_{config} for each P_1 position and type. Practically, we use sliding window search for each potential P_1, we can write $score_1(I, P_1) = \max(score(I, P_{config}))$.

The computations need to consider $L \times T$ parent locations and maximize $L \times T$ child locations, making a $O(L^2T^2)$ calculation for each patch selection. However, due to the quadratic form of $\psi(P_i, P_j)$, the inner maximization can be efficiently computed for each combination of t_i and t_j in $O(L)$ with a max-convolution or distance transform. Considering this, the computation time is reduced to $O(LT^2)$

3.4. Learning of PDP

We train our model with the *latent SVM* framework, in which the patch configuration is our latent variable. The parameters ω^a for *Patch Appearance Feature*, ω^d for *Patch Distance Feature* and ω^c for *Patch Configuration Feature* are learned from the latent SVM.

Given a training set of N patches configuration in which their corresponding binary class labels y_i belong to $\{-1, 1\}$, we can also compute their feature representations to obtain dataset $\{(I_i, y_i), ..., (I_N, y_N)\}$. We denote $\omega = [\omega^a, \omega^d, \omega^c]$. Therefore, The objective we would like to minimize is given by:

$$\min_{\omega} \frac{1}{2} \parallel \omega \parallel^2 + C \sum_{i=1}^{N} max(0, 1 - y_i) \tag{6}$$

$$f_\omega(I) = \max_{P_{config}} \ \omega \cdot \boldsymbol{\Phi}(I, P_{config}) \tag{7}$$

The optimization is achieved using the *Dual Coordinate Descent* algorithm; more details can be found in [38].

3.5. Discussion

Difference with Poselet [3]: Poselets and PDP are both generated from human joint locations. Poselet has property of (1) different patch ratio, (2) multiple joints combination, (3) extracted for human detection task. Poselet framework focuses on finding general human parts (e.g. Head, Upper

Body, Leg), and uses all Poselet candidates to cover human appearance. Besides, they provide a voting scheme for integrating all the poselet detection responses. In PDP, our objective and patch selection strategy are quite different from the Poselets. We use fixed square patches, and the patches are designed to distinguish different activities, not detecting humans. Our search algorithm highlights the discriminative property from patch instead of generative property.

Difference with Mid-level Discriminative Patches [9]: The middle level Discriminative Patches use a Bag of Words type idea. Among video patches, no mutual relations are considered. In PDP, after we find pose based discriminative patches, we build a tree to organize these discriminative patches, modeling their mutual relations. These relations integrate more discriminative information besides patches.

4. Experiments

In this section, we show PDP's capability in four different aspects for video level activity recognition, image level activity recognition, pose estimation in still images, activity prediction. We achieve improvements on the video level dataset by 6% compared with state-of-the-art method, and also improvements on still image activity recognition, activity prediction and human pose estimation.

4.1. Dataset

We use two datasets to demonstrate our model capability.

Sub-JHMDB originates from HMDB51 dataset [12]. It is composed of 12 activities, containing 316 videos in total. The 12 activities include *catch, climb starts, golf, jump, kick ball, pick, pull up, push, run, shoot ball, swing baseball, walk*. All these videos are captured from Internet or movies. Though Sub-JHMDB is a single actor dataset, it is very challenging where the state-of-the-art method get about 49% overall classification accuracy [8]. We do not test on the JHMDB because some of the joints are outside the image frame. However, according to experiments given in [8], the Sub-JHMDB is more difficult than JHMDB where baseline performance on Sub-JHMDB is 10.66% lower than on JHMDB.

The second dataset is **Still Image Action Dataset**. it is used for evaluating activity recognition in still images. We use the dataset from [17] which was collected from the Internet. The dataset includes still images from five activities: *running, walking, playing golf, sitting, and dancing*. There are 2458 images in total. We get the pose annotations from [37].

4.2. Experiment Setup

For sub-JHMDB, we use the same training and testing splits as in [8]. The ratio of the number of clips in the two sets and the ration of the number of distinct video sources in the two sets are both close to 7:3.

For Still Image Action Dataset, we follow the same training and testing splits as in [37]. We use 1/3 of the images from each action category to form training data, with the 14 points pose annotation, while the 2/3 rest of the dataset are testing images.

For our target activity i, we define all the training videos labeled with i as positive videos, and the rest as negative videos. When mining the discriminative patches for activity i from our training videos, we first build a positive patch bank where we store all the frames belonging to the positive videos, and a negative patch bank where we store frames coming from the negative videos.

Initialization of PDP: According to pose annotation, we select 14 patch candidate set locations corresponding to the 14 joints location (we neglect the belly joint from the original annotations). For each patch candidate set, we select $M_0 = 3$ different types of patches. In both of the datasets, we use the searching algorithm given in Section 3.1. We train patches using liblinear SVM [22] with HOG features, and get 42 patch detectors in total. These detectors initializes ω^a of PDP.

Next, we will also need positive samples and negative samples to further train and update $[\omega^a, \omega^d, \omega^c]$. For the positive samples, we pick the three frames that contribute the most number of discriminative patches from each video.

The negative sample consists of two parts. One is frames from negative videos; we use the negative training images from the INRIA person database as our negative training set, there are 1218 images in total. For the second part, we randomly select one frame from each of the negative videos.

4.3. Activity Recognition in JHMDB

We demonstrate PDP's capability in sub-JHMDB dataset. If the test video v has N frames, the output confidence will be defined as $score(v) = \frac{1}{N}\sum_{k=1}^{N} s(v_k)$, where the $s(v_i)$ is the PDP output score for each frame v_i. We predict the label of test video by finding the maximal output confidence $score(v)$.

Our experimental results are given in Table 1 and Table 2. In Table 1, the first result is using DT features, the second result is using DT features combined with pose configuration features predicted from [38]. Both of these results are provided by [8]. The third result is our PDP solution which improves the overall accuracy for 6%. We want to note that [8] also reports a result of 75.5% accuracy but this is given groudhtruth pose to represents a theoretical upper bound, and is not comparable to an automatic system.

We also show the accuracy for each specific activity in Table 2, also provided by [8]. We report improvements in 7 of the 12 activities. Specially, we show high improvement on *Jump, kick ball, Pull up and Swing baseball*, which are all complicated activities. PDP does not perform well in *Climb stairs* due to heavy occlusions. Recently, [36] has

Dance	0.47	0.02	0.31	0.07	0.13
Golf	0.25	0.27	0.34	0.04	0.09
Run	0.10	0.00	0.81	0.03	0.06
Sit	0.22	0.02	0.11	0.61	0.05
Walk	0.13	0.02	0.38	0.00	0.46
	Dance	Golf	Run	Sit	Walk

(a) Results from [17]

Dance	0.53	0.12	0.15	0.12	0.08
Golf	0.18	0.65	0.10	0.03	0.03
Run	0.13	0.07	0.66	0.07	0.08
Sit	0.13	0.06	0.02	0.79	0.01
Walk	0.15	0.12	0.24	0.01	0.48
	Dance	Golf	Run	Sit	Walk

(b) Results from [37]

Dance	0.33	0.05	0.14	0.29	0.19
Golf	0.03	0.84	0.04	0.07	0.02
Run	0.01	0.04	0.78	0.05	0.12
Sit	0.01	0.03	0.01	0.78	0.17
Walk	0.06	0.06	0.29	0.11	0.48
	Dance	Golf	Run	Sit	Walk

(c) Results from PDP

Figure 4. Confusion matrices of the classification results on the Still Image Action Dataset.

Method	[8]	[8] + [38]	PDP
Overall Accuracy	46.0	49.8	**55.43**

Table 1. Overall Accuracy on Sub-JHMDB

Class	[8]	[8] + [38]	PDP
Catch	42.86	**42.86**	33.94
Climb stairs	**60.00**	33.33	0.00
Golf	72.22	**94.44**	91.67
Jump	21.74	21.74	**45.83**
Kick ball	40.91	45.45	**60.71**
Pick	42.31	61.54	**62.50**
Pull up	57.14	53.57	**81.48**
Push	74.07	**74.07**	57.59
Run	**50.00**	33.33	21.43
Shooting ball	15.79	21.05	**33.33**
Swing baseball	0.00	5.26	**79.05**
Walk	46.15	53.85	**61.11**
Mean	43.60	45.04	**52.39**

Table 2. Accuracy per class using PDP on Sub-JHMDB

reported Detection task results; However, as our target (temporal localization) and evaluation benchmark are different, we are unable to compare. We do not test our solution with [15] on PASCAL action recognition dataset (10 activity category) or [13] on TV interaction dataset (5 activity category) as they lack of human pose annotations. However, our dataset is as challenging as their dataset, which we have 12 activities, captured from Internet and movies.

4.4. Activity Recognition in Still Image Action Dataset

As PDP is an image-based recognition algorithm, it can also be used in still image activity recognition.

Method	Overall	Mean Per-class
[17]	56.45	52.46
[37]	61.07	62.09
PDP	**63.40**	**63.90**

Table 3. Results on the Still Image Action Dataset. We report both overall and mean per class accuracies due to the class imbalance.

Method	Hea	Sho	Elb	Wri
[38] retrained	**38.28**	**78.19**	61.66	36.13
Our Method	36.84	77.92	**70.28**	**43.07**
Method	Hip	Kne	Ank	Total
[38] retrained	40.87	19.84	1.11	39.44
Our Method	**41.51**	**22.84**	**2.97**	**42.20**

Table 4. Pose Estimation PCK performance compared to retrained [38] on Sub-JHMDB.

The overall accuracy and mean per-class accuracy are shown in Table 3. PDP performs well on a complex activity like playing golf which beat previous baseline by about 20%. We do not get good result on dancing activity because the pose information from dancing are in high variance, which lower the total mean average accuracy. In total, we get 2% improvement than state-of-the-art method. We show our confusion matrices in Figure 4.

4.5. Human Pose Estimation

In this section, we use images extracted from test data in the Sub-JHMDB dataset. As patch candidates are correlated with human pose estimations, we can use PDP output to indicate human pose estimation. We use the standard PCK (Probability of Correct Keypoint) benchmark to evaluate our algorithm; all the frames coming from the test videos are used.

From Table 4, We can see that PDP model gets 3% improvement under PCK benchmark. PDP model shows improvement in *elbows, wrist, hips and knees*. This is because the discriminative patches that come from these parts are more effective than the rest. We do not get good result on Ankle because most of time it is occluded in complex activity recognition. Furthermore, we can observe from Figure 5 that pose estimation is still not nearly perfect. But, we can still achieve much improvement in activity recognition.

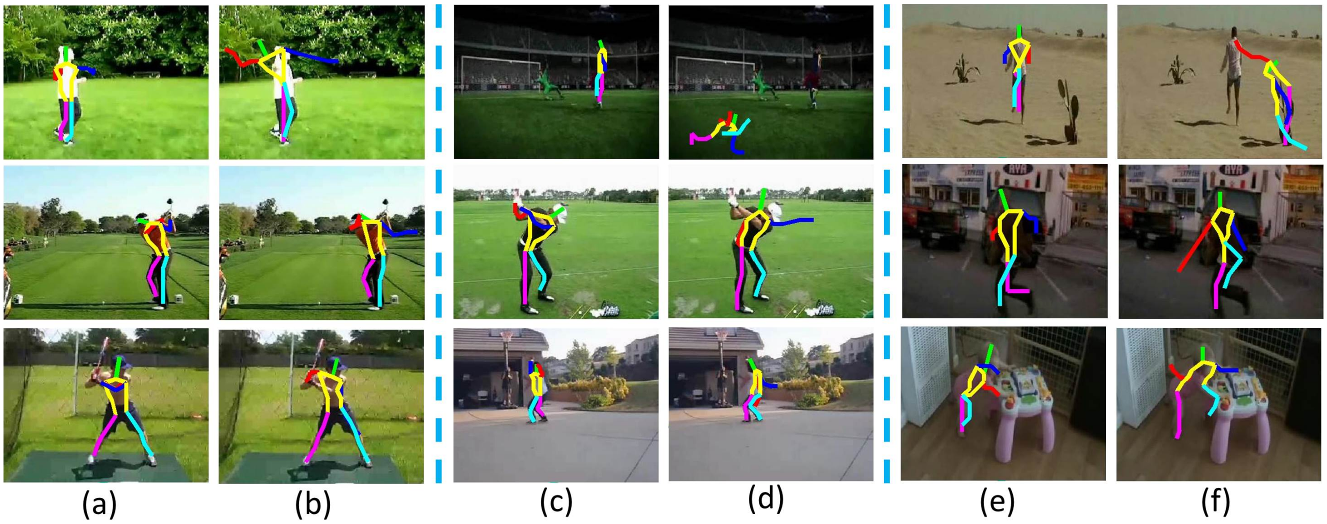

Figure 5. Pose Estimation Visualization Results. (a)(c)(e) are from PDP. (b)(d)(f) are from retrained [38].

4.6. Activity Prediction

To demonstrate our PDP's prediction performance, we use two baselines: SIFT Flow [4] and Pose Estimation Configuration based on [38]. PDP improves the baselines by a large margin of around 30%.

We use the first $1, 3, 5, 7$ observed frames to predict the future activities. The results are shown in the Table 5. By only using 1 frame (single image), PDP gets the accuracy of 41.22%.

For SIFT Flow in our baseline [4] (which is also used as baseline in [13]). Given a test image, it first finds the nearest neighbor from the training data using the SIFT Flow algorithm, which matches densely sampled SIFT features between the two images, while preserving spatial discontinuities. The future action label of the matched training image is directly transferred to the testing image. Using the single frame, the results are low as 13.94%.

The second baseline uses pose configuration similarity, and label with test video with the most similar poses' activity category. First, we use [38] to estimate pose locations. Then, we compare the similarity for the test video clips or frames to the video clips or frames in our training test. We use the PCK metrics to evaluate how similar the two poses are. We label the test video using the label of the training video with the highest PCK scores. The classification results are shown in the following Table 5.

We get the upper limit for Dense Trajectory is 43.60% with full video observed (provided by [8]). Our method beats Dense Trajectory using only the first 5 frames. From the experiment, we can see PDP model performs very well in prediction problem which improves both baselines for around 30%.

Observed	SIFT Flow [4]	Pose Config	PDP
First Frame	13.94	10.44	**41.22**
First 3 Frames	9.03	15.27	**43.07**
First 5 Frames	9.72	14.25	**44.71**
First 7 Frames	11.34	16.37	**45.33**

Table 5. Prediction Accuracy using PDP on Sub-JHMDB

5. Conclusion

In this paper, we focus on activity recognition and prediction, where we propose a Pose-based Discriminative Patch (PDP) model to utilize human pose information into activity recognition. Based on the observation that the human pose estimation algorithm is far from perfect, we suggest pose-based discriminative patches as being more effective than extracting features from pose configuration. We integrate this idea into PDP model and demonstrate its effectiveness in problems of video level activity recognition, still image activity recognition and human pose estimation. Our experiment shows that using PDP is a better way of human activity recognition than using pose configuration features. PDP's superior performance indicates the need for action related pose annotations.

For future work, we plan to extend our framework to handle data with heavy occlusions, and in long time period. Therefore, PDP can show its capability in activity detection. Also, we will include temporal information in PDP, adding in mutual relation within video segments.

Acknowledgment: This research was supported, in part, by the Office of Naval Research under grant N00014-13-1-0493.

References

[1] K. Alahari, G. Seguin, J. Sivic, and I. Laptev. Pose estimation and segmentation of people in 3d movies. In *Proc. ICCV*, 2013. 2

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, 2009. 2

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. ICCV*, 2009. 2, 3, 5

[4] a. A. T. a. J. S. C. Liu, and J. Yuen and W. T. Freeman. Sift flow: dense correspondence across different scenes. In *ECCV*, 2008. 8

[5] W. Chunyu, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013. 3

[6] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. CVPR*, 2005. 2, 4

[7] G. Ghiasi, Y. Yang, D. Ramanan., and C. Fowlkes. Parsing occluded people. In *CVPR*, 2014. 2

[8] S. Z. C. S. H. Jhuang, J. Gall and M. J. Black. Towards understanding action recognition. In *Proc. ICCV*, 2013. 1, 2, 3, 6, 7, 8

[9] A. Jain, A. Gupta, and L. S. D. M. Rodriguez. Representing videos using mid-level discriminative patches. In *Proc. CVPR*, 2013. 2, 3, 6

[10] D. K. Kevin Tang, Li Fei-Fei. Learning latent temporal structure for complex event detection. In *CVPR*, 2012. 3

[11] K. Kitani, B. Ziebart, D. Bagnell, and Hebert. Activity forecasting. In *ECCV*, 2012. 2

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 3, 6

[13] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014. 2, 7, 8

[14] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 444–451. IEEE, 2009. 3

[15] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *Proc. CVPR*, 2011. 3, 7

[16] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 3

[17] S. P. Nazli Ikizler, R. Gokberk Cinbis and P. Duygulu. Recognizing actions from still images. In *Proc. ICPR*, 2008. 6, 7

[18] S. Pellegrini, A. Ess, K. Schindler, and L. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009. 2

[19] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014. 3

[20] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, 2013. 2

[21] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *ICCV*, 2013. 2

[22] C.-J. H. X.-R. W. R.-E. Fan, K.-W. Chang and C.-J. Lin. Liblinear: A library for large linear classification. In *JMLR*, 2008. 4, 6

[23] V. Ramakrishna, D. Munoz, M. Hebert, A. J. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014. 2

[24] S. Sadanand and J. J. Corso. Action bank: A high-level representation. In *CVPR*, 2012. 2

[25] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proc. ECCV*, 2010. 2

[26] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Proc. CVPR*, 2011. 2

[27] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR*, 2011. 2

[28] V. Singh, R. Nevatia, and C. Huang. Efficient inference with multiple heterogeneous part detectors for human pose estimation. In *Proc. ECCV*, 2010. 2

[29] V. K. Singh and R. Nevatia. Action recognition in cluttered dynamic scenes using pose-specific part models. In *Proc. ICCV*, 2011. 3

[30] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 3

[31] C. Sun and R. Nevatia. Active: Activity concept transitions in video event classification. international conference on computer vision. In *ICCV*, 2013. 3

[32] C. Sun and R. Nevatia. Large-scale web video event classification by use of fisher vectors. In *WACV*, 2013. 3

[33] A. Vahdat, K. Cannons, G. Mori, I. Kim, and S. Oh. Compositional models for video event detection: A multiple kernel learning latent variable approach. In *ICCV*, 2013. 3

[34] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 3

[35] L. Wang, Y. Qiao, and X. Tang. Mining motion atoms and phrases for complex action recognition. In *ICCV*, 2013. 3

[36] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *ECCV*, 2014. 6

[37] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *Proc. CVPR*, 2010. 6, 7

[38] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, 2011. 1, 5, 6, 7, 8

[39] A. Yao, J. Gall, and L. van Gool. Coupled action recognition and pose estimation from multiple views. *International Journal of Computer Vision*, 100(1):16–37, Oct. 2012. 3

[40] J. Yuen and A. Torralba. A data-driven approach for event prediction. In *ECCV*, 2010. 2