Image and Vision Computing 59 (2017) 31–43

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Human pose search using deep networks ☆

Nataraj Jammalamadaka[a,*], Andrew Zisserman[b], Jawahar C.V.[a]

[a] CVIT, IIIT, Gachibowli, Hyderabad, Telangana 500020, India
[b] Visual Geometry Group, Department of Engineering Science, University of Oxford, Oxford, UK

ARTICLE INFO

Article history:
Received 30 September 2015
Received in revised form 19 September 2016
Accepted 12 December 2016
Available online 23 December 2016

Keywords:
Pose retrieval
Pose estimation
Video and image retrieval
Deep networks

ABSTRACT

Human pose as a query modality is an alternative and rich experience for image and video retrieval. It has interesting retrieval applications in domains such as sports and dance databases. In this work we propose two novel ways for representing the image of a person striking a pose, one looking for parts and other looking at the whole image. These representations are then used for retrieval. Both the representations are obtained using deep learning methods.

In the first method, we make the following contributions: (a) We introduce 'deep poselets' for pose-sensitive detection of various body parts, built on convolutional neural network (CNN) features. These deep poselets significantly outperform previous instantiations of Berkeley poselets [6], and (b) Using these detector responses, we construct a pose representation that is suitable for pose search, and show that pose retrieval performance is on par with the previous methods. In the second method, we make the following contributions: (a) We design an optimized neural network which maps the input image to a very low dimensional space where similar poses are close by and dissimilar poses are farther away, and (b) We show that pose retrieval system using these low dimensional representation is on par with the deep poselet representation and is on par with the previous methods.

The previous works with which the above two methods are compared include bag of visual words [44], Berkeley poselets [6] and human pose estimation algorithms [52]. All the methods are quantitatively evaluated on a large dataset of images built from a number of standard benchmarks together with frames from Hollywood movies.

1. Introduction

Pose is an atomic unit of gesture and action, and an important aspect of human communication. Accordingly it has been the focus of many works [15,21,25,31,39,42,51,52] in the recent past. With the exponential growth of videos and images online, it has become very critical to develop interfaces which allow easy access to human pose. Text queries as an interface for image and video search will gradually become untenable with massive growth in videos and images on the Internet. With computer vision improving, content based retrieval is becoming a reality. Pose is one such content, and human pose retrieval is of great interest as it indicates action and gesture. Real-life applications of human pose retrieval include baseball or cricket shot retrieval from a sports database and a dance pose retrieval from say a ballet collection. Thus a gesture and pose as a query modality gives an alternative and rich experience for search.

Fig. 1 illustrates an example pose retrieval. As shown in the figure, a pose search system aims to retrieve people in a similar pose to the query irrespective of the gender of the person, color of the clothing, the type of clothes worn or the clutter and crowd in which the person is standing.

In this work, we propose two novel deep learning based approaches to pose search. In the first method, we propose 'deep poselets' which can be described as classifiers which detect a subset of body parts in a specific pose. The response of these deep poselets are used to construct a feature representation of the pose, which is used for the pose retrieval. The main contributions of this method are, (a) demonstrating that explicitly clustering the pose space of arms is useful for encoding the pose, (b) demonstrating that a similar architecture to ImageNet-CNN [33] is able to work on the unrelated task of poselet classification, (c) finding areas in the image that have high probability of deep poselets being present, and thereby improving their performance, and (d) empirically demonstrating that deep poselet based pose search outperforms competing methods. In the

☆ This paper has been recommended for acceptance by Richard Bowden.

* Corresponding author.
E-mail address: nataraj.j@research.iiit.ac.in (N. Jammalamadaka).

http://dx.doi.org/10.1016/j.imavis.2016.12.002
0262-8856/

Fig. 1. Pose search: For the query image (top-left corner), the pose search system retrieves people in the database who are in the same pose as the query image. The system has to be invariant to the color and type of the clothes, the clutter in the background and presence of other people in the image.

second method which is inspired by the work of Taylor et al. [46], we propose 'deep pose embedding' model which takes an image triplet consisting of a reference image, an image with the similar pose and an image with the dissimilar pose and learns a projection function to a pose-sensitive lower dimensional space. In contrast to our 'deep poselet' method, this method looks at the complete image and maps it to a lower dimensional space. The main contributions of this method are, (a) demonstrating that an image can be mapped to a pose space using deep networks, and (b) the projection is pose sensitive and performs well on pose retrieval task.

The pose search task was originally proposed by Ferrari et al. [20] where it was demonstrated on a database containing six episodes of the popular TV show 'Buffy the vampire slayer'. In their work, first, all the people in a frame are detected using an upper body detector, and a human pose estimation (HPE) algorithm is run on the detected upper bodies. Using the marginals computed during the inference, a feature representation is constructed for the pose. The work by Jammalamadaka et al. [28] extended [20] by demonstrating pose search on 3.1 Million frames taken from 22 Hollywood movies. In [28], a HPE algorithm is used to estimate pose and a very low dimensional feature vector is built using the angles of the various body parts. Furthermore, the algorithm proposed by Jammalamadaka et al. [27] detects wrong pose estimates, and hence is able to filter them out.

Here we briefly give the outline of the paper. Deep poselets are described in Section 3. The data driven process to obtain specific poses and their positive instances are described in Section 3.1. The details of the feature extraction and training are described in Section 3.3. Given an input test image, all the poselet classifiers are run using the procedure described in Section 3.4. During the detection stage, mutually exclusive poselet types (e.g., those corresponding to the left arm) fire at the locations with a significant overlap in their detections. This conflict is resolved by spatial reasoning, described in Section 3.5. Using these deep poselets and their detection scores, a representation for a pose is constructed. The deep pose embedding model described in Section 4 projects an image onto a lower dimensional pose-sensitive space. The properties of this pose sensitive space, the details of the CNN projection function and training methodology are described in Section 4.1. The projection function CNNs is trained using image triplets. In Section 4.2, we describe how to handle the exponentially large triplet combinations. The representations obtained from the above two methods are then used to perform pose search as described in Section 5. In the experimental Section 6, we evaluate the deep poselet method, the deep pose embedding method, and the pose search method by comparing them with relevant baselines.

This work is a continuation of our conference paper [29], where the primary focus was deep poselets. Here, we extend [29] by proposing an alternative deep pose embedding model, and make connections between the two models. Furthermore, we train both the models on much larger data (2×) than used in [29] to improve the pose diversity and appearance invariance. We evaluate both the models individually to demonstrate their performance and compare them against other pose retrieval baselines to show significant progress on this problem.

2. Related work

2.1. Convolutional neural networks (CNNs)

Convolutional neural networks are first proposed by Lecun et al. [34] where an object is modeled as a composition of patterns

starting from edges to higher level parts like faces. In the recent past, convolutional neural networks [13,23,33] have been shown to outperform and significantly improve image classification on the challenging ImageNet dataset [12]. Furthermore, features from this network, trained only for image classification, have been shown [41] to improve the state-of-art on several other unrelated tasks like scene recognition, fine grained detection and so on. In our implementation, rectified linear unit [36] is used as the activation function and the drop-out regularization scheme [26] is used while training.

2.2. Siamese network

Siamese network, introduced in [8], is a pair of neural networks who, at any given time instance, have the same architecture and same weights. The network takes in a pair of images, forward propagates them using a pair of neural networks to obtain the projections and learns the weights based on a loss function on these projections. Typically the supervision given is whether the pair of images are similar or dissimilar. This method has been applied for signature verification [8], face verification [11], and image retrieval [50]. The work by Taylor et al. [46] using a loss function which takes a label in [0,1] interval. They have applied this method for character recognition and pose retrieval. Our work extends this method by taking image triplet and defining a ranking loss [30] on them. We further demonstrate the method on a large dataset.

2.3. Poselets

Poselets [6] are classifiers which model a subset of body parts. The key difference between [6] and our method is that [6] is for person detection, and ours is for pose detection. A poselet, for example, can model the head and the left shoulder together. The poselet method has recently [7] been improved using CNNs. Gkioxari et al. [24] adapts poselets for HPE problem by proposing to discover the poselets by using only the image patches corresponding to the arms. This work by Gkioxari et al. [24] is the closest to ours. Both our approach and [24] use body part detectors which are sensitive to pose. While the main focus of [24] is on key point detection, ours is on implicit pose encoding. Further, while we train CNN features specifically for body part detection task using CNNs, Gkioxari et al. [24] have used HOG features.

2.4. Human pose

The pose retrieval methods of [20,27,28] use HPE algorithms. Among the many HPE algorithms, pictorial structures [19] based methods [15,21,52] are very popular. Methods such as [38] have integrated a modified version of Berkeley poselets [6] with pictorial structures, while other methods such as [42] have used the poselets for inferring the pose. With the success of convolutional neural networks, a few methods [47,48] have been proposed using CNN architectures.

3. Deep poselets

In this work, a deep poselet is defined as a model which consists of subset of the seven body parts present in a particular pose. The seven body parts used are the left and the right upper arms, the left and the right lower arms, the left and right hip, and the head. Fig. 2 illustrates a few example deep poselets.

Fig. 2. Discovered deep poselets: Six deep poselets and instances belonging to them are shown. For each deep poselet, an average image marked with stickman and example instances are displayed. A deep poselet is composed of subset of body parts in a particular pose as indicated by the stick figure on the average image. The body parts and their poses in each example instance matches its corresponding deep poselet.

Deep poselet method consists of discovering the deep poselets from the data, training the poselets and finally detecting and post-processing them on test images. All these steps are described in great detail in the next few sections. Fig. 3 illustrates all these four steps.

3.1. Deep poselet discovery

The deep poselet framework can be understood as a discretization of the pose space, where each state is captured by one deep poselet. We formulate this discretization as a data driven process by clustering the body joints. Clustering all the body parts jointly needs huge amounts of data to fully represent the pose space. Instead we cluster on seven subset of body parts, where subset i is represented by S_i. The seven subsets used are (1) the left arm and the left hip, (2) the left arm, left hip, and the head, (3) the left arm and the right hip, (4) the right arm and the right hip, (5) the right arm, right hip, and the head, (6) the right arm and the left hip, and (7) all body parts minus the head. The left and the right arm are modeled, in three different spatial contexts, by the subsets $\{S_1, S_2, S_3\}$ and $\{S_4, S_5, S_6\}$ respectively. These three spatial contexts are (a) itself, (b) with torso, and (c) with head and torso. The subset S_7 models both the arms and captures the popular poses in the database. The resultant cluster means form an atomic unit of pose and a combination of them describes an upper body pose. Since the body parts modeled by a subset S_i can only take one of N distinct poses and clustering algorithms give unique means, *these cluster means are mutually exclusive to each other*.

Clustering each subset S_i is performed in the following way. First the dataset is pre-processed by computing a bounding box of the person from the stickman annotation. This bounding box is then expanded by extents learnt from the data such that all possible human poses, with their various articulations and extensions of body parts, are contained within the *expanded bounding box*. Next, body parts annotations of subset S_i are x–y normalized by the dimensions of the expanded bounding box. These normalized coordinates are concatenated and passed onto a K-means algorithm for clustering. The cluster means are taken as the canonical deep poselets. In our experiments, a total of 122 deep poselets are obtained. Fig. 2 illustrates a few deep poselets discovered using the above process.

3.1.1. Pose comparison

While it is sensible to consider the samples belonging to the deep poselet cluster as positive samples, some of these are perceptually dissimilar to the cluster mean. Further, there are samples whose membership is perceptually ambiguous. Thus for a deep poselet, each sample is classified as belonging to positive class, negative class or ignore class using body part angle (angle made by a body part with the image axis). The samples belonging to the ignore class are neither considered while training nor while testing. The classification is done using the following procedure: (a) All the samples whose individual part angles do not deviate by more than τ_1 from the canonical deep poselet are taken as positive samples, (b) all the samples whose individual part angles deviate by more than τ_2 degrees from the canonical deep poselet are considered as negative samples, and (c) finally all the samples whose individual part angles deviate by less than τ_2 degrees but with at-least one part which deviates between τ_1 and τ_2 degrees are considered as ignore class. Using cross validation, the thresholds τ_1 and τ_2 are set at 20 and 30° respectively.

3.2. Expected poselet area (EPA)

As deep poselets use CNNs, the sliding window approach for locating the body parts is very expensive during test time. Previous CNN based methods for image classification have solved this problem

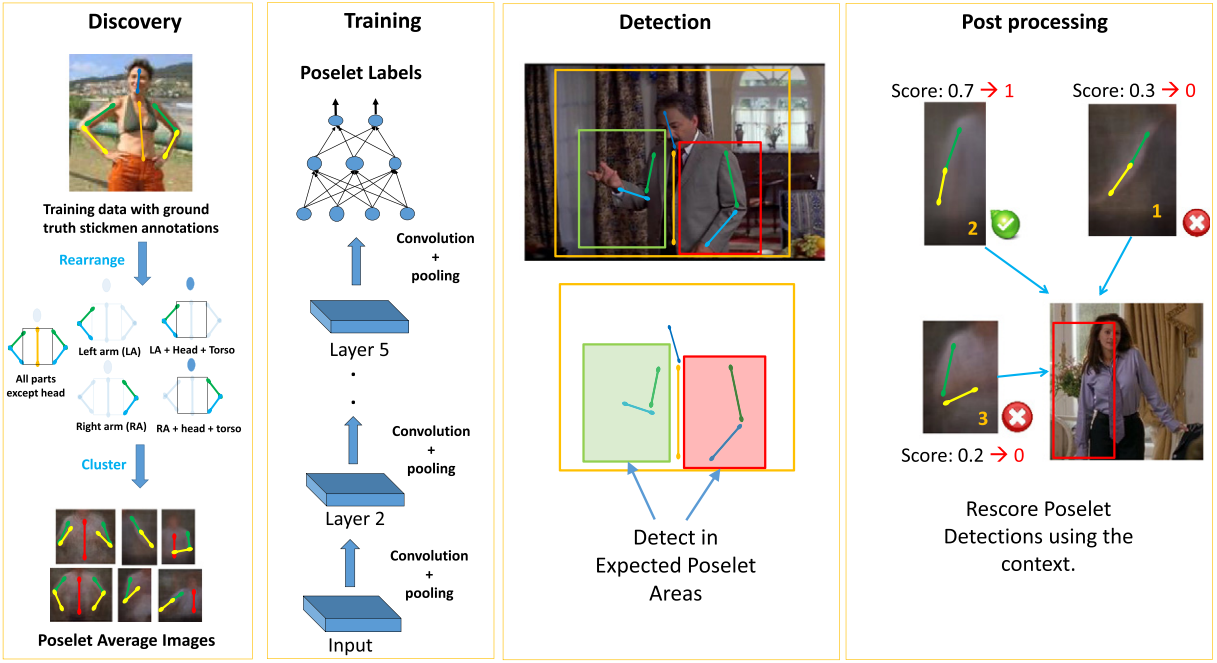

Fig. 3. Deep poselet method: The proposed deep poselet method has four parts: (a) Discovery: First, poselets of various body joint configurations (illustrated in the figure) are discovered by clustering in the pose space. (b) Training: These poselets are then trained using convolutional neural networks. (c) Detection: Each poselet has been observed to have a localized area within the upper body bounding box. We term this area as "expected poselet area (EPA)". The poselet detection is performed within this area. (d) Post-processing: The EPA of several poselets intersect (e.g., all poselets belonging to the left arm). Thus within the same area, several poselets have detections while only small number of them are correct. Using linear regression we rescore the poselet detections using the context of other poselet detections. Parts (c) and (d) are our contributions while parts (a) and (b) have some overlap with previous works.

by using unsupervised object proposal methods like objectness [1] and selective search [49]. Unfortunately, poselets are not whole objects but parts of a specific object (e.g. arms as part of human). Thus the above object proposal methods are not useful for the task. We solve this problem by finding the 'expected poselet area (EPA)' in an image. The EPA gives the highly probable location of the deep poselet within the bounding box of the person. For example, a deep poselet modeling the left arm typically lies in the left half of the bounding box. The search space of the deep poselet can be restricted to this EPA, which improves both the performance and time complexity. The extent of the EPA of a deep poselet is learnt from the positives in the training data. This is done by taking 5 percentile and 95 percentile of the normalized coordinates (normalized w.r.t expanded bounding box) as the extent of EPA respectively. Experiments show that over 95% of the positive instances in both training and test data are encompassed by expected poselet area. This highly precise spatial locality property of poselets ensures that searching only in this area and avoiding the rest of the image (exhaustive search) decreases the probability of false positive occurrence. Thus this improves the accuracy and since the search space is reduced it is computationally efficient.

While EPA encompasses the positives instance well, it also has background area within it. Thus the ground truth area can be any of the possible sub-windows of the EPA. A way to deal with this would be to search for the true detection in the EPA over all possible scales and locations. We simplify the search procedure by fixing the scale of deep poselet to 90% of the EPA and translations to 9 equally spaced sub-windows.

3.3. Training

As mentioned before, each deep poselet models a subset of parts in a specific pose. We train a discriminative classifier which can tell apart image regions belonging to this deep poselet from other image regions. We use linear SVMs to train the deep poselets. For the features, we use the representations from CNNs.

In our experiments, we use the implementation of the ImageNet-CNN network by Donahue et al. [13]. The ImageNet-CNN [33] is a deep neural network with five convolutional layers and three fully connected layers. Below, the feature extraction and training are explained

3.3.1. Feature extraction

The nine sub-windows of the EPA are passed through ImageNet-CNN in a feed forward manner and the feature maps of the fifth pooling layer (pool5), the first and the second fully connected layers (fc6 and fc7 respectively) are noted. From these three feature maps, the best performing one (details in Section 6) is used as the representation for the deep poselet.

Further, we fine-tune the ImageNet-CNN to the task of poselet classification so that the CNN takes an image region as input and outputs the poselet class label or background. For fine-tuning, the last fully connected layer of the ImageNet-CNN is replaced by a 123 (122 deep poselets and a background class) neuron fully connected layer. The weights of the newly added layer are randomly initialized. The weights of the rest of the layers are initialized from the ImageNet-CNN [13]. It has been observed that the sample strength ratio between the largest poselet class and the smallest poselet class is 80. To compensate for this skew, the data of the classes with low strength are augmented by their translated versions. The original learning rates are decreased by a factor of 10 so that the existing weights do not significantly change. For the first two fully connected layers, a drop-out rate [45] of 0.5 is used. For training the network, the cuda-convnet software [32] is used.

3.3.2. Learning SVMs

Typically an EPA has significant background areas. Thus the ground truth area can be any of the possible sub-windows of the EPA. To select the correct sub-window a multiple instance learning (MIL) approach is used [2]: after extracting the feature representations from the nine sub-windows of all EPAs, an initial linear SVM model is trained. For this, all the sub-windows are given the same label as the EPA. Using this initial SVM, the best scoring sub-windows are selected and a new SVM model is trained. This process is repeated until there is no change in the AP on the validation set. In practice, it is found that three iterations suffice. Empirically, this procedure improved the AP by 7% over the method in which all candidate windows are used for training. This procedure is reminiscent of best positive bounding box selection used in Felzenswalb et al. [18].

3.4. Testing

Given a test image, it is processed using the human detector algorithm to obtain upper body detections. Each upper body detection is then transformed to obtain the expanded bounding box. For each deep poselet, the corresponding EPA (expected poselet area) is computed using the learnt transformation (Section 3.2). The EPA is then divided into nine equally spaced sub-windows with the scale of each sub-window at 90% of EPA. Each sub-window is passed onto the deep poselet model to obtain a score. The sub-window with the best score is noted as the deep poselet detection.

3.5. Spatial reasoning

On an image with a person in it, typically most of the deep poselets fire, when only a few of them are correct. Many of these deep poselet detections significantly overlap, while being mutually exclusive. Fig. 4 illustrates this behavior. In the figure, three deep poselet

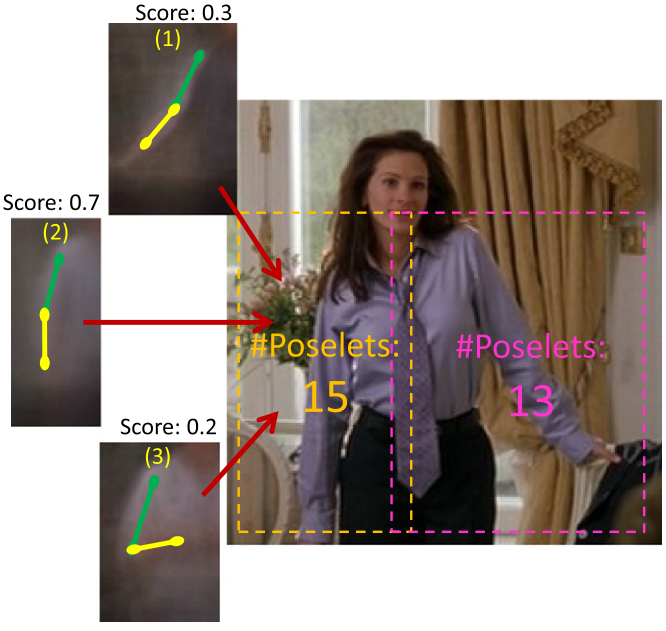

Fig. 4. Spatial reasoning: For a given test sample, three deep poselet detections and their scores are shown as belonging to the area marked by an orange rectangle. Detections 1 and 3 are partially correct as the pose of the left upper arm matches that of the test sample. Detection 2 is the correct one. Typically many such deep poselet detections, often mutually exclusive, have significant overlap. Using spatial reasoning, these detections are rescored such that correct ones (detection 2) get a score of nearly 1 and the partially or totally incorrect ones (detections 1 and 3) get a score of nearly 0. The image also shows that the area around the left arm (orange rectangle) has 15 unique deep poselets while the area around the right arm (pink rectangle) has 13 unique deep poselets.

detections corresponding to the left arm are displayed. Clearly they are mutually exclusive because the arm can be present in only one of the three poses represented by them. This conflict is resolved by rescoring the deep poselet detections using other mutually exclusive deep poselet detections as context. The expected outcome is that the correct detections (detection 2 in the Fig. 4) have a score of nearly 1 and incorrect ones (detections 1 and 3 in the Fig. 4) have a score of nearly 0. For this rescoring, a RBF kernel based regression model [14] is learnt for each deep poselet type P. The input to this model is a feature vector comprising of calibrated scores (procedure in the next paragraph) of the P's own detection and its mutually exclusive deep poselets and the output is the new score. For training, the above feature is provided as input and the binary label of the deep poselet detection is provided as target value. Given a test sample, first all the deep poselets are run on the sample and then the above regression models are applied to rescore each deep poselet detection. Below the procedure for calibration and finding mutually exclusive poselets are described.

3.5.1. Calibration

Calibration ensures that scores of various deep poselets are comparable. This is achieved by mapping the scores of all deep poselets to the [0,1] interval. We use the method proposed by Platt [40], in which a logistic regression model is learnt with the deep poselet score as input. Let $X \in R$ be the scores of the deep poselet detections D. A mapping $\sigma : X \rightarrow Y$ where $X, Y \in R$ is learnt. The function $\sigma(x)$ is parameterized by w_0, w_1 and is given by,

$$\sigma(x) = \frac{1}{1 + e^{(w_1 x + w_0)}}. \tag{1}$$

3.5.2. Mutually exclusive deep poselets

For each deep poselet type P, a mutually exclusive poselet is defined as one which occupies the same area in the person bounding box. For example, the three detections in Fig. 4, which are mutually exclusive, occupy the same area. The following procedure is used to find the mutually exclusive deep poselets. First the 'expected poselet areas' (Section 3.3) of all the 122 deep poselets are collected. These deep poselets are then clustered using the cluster partitioning algorithm proposed by Ferrari et al. [22]. The algorithm returned 31 clusters, where poselets in each cluster form a mutually exclusive set.

4. Deep pose embedding

In this section, we describe the second way of representing the pose from an image. We describe the CNN projection function and the triplet ranking loss used for training.

4.1. Model and training procedure

Given an image of a person in a particular pose, we project it into a low-dimensional pose-sensitive space. This low-dimensional space has the following structure: (a) The projections of all images with similar poses are near-by, and (b) The projections of images with dissimilar poses are far away.

For learning the projection function, image triplets (x_i, x_i^p, x_i^n) are given which consists of a reference image, an image with similar pose and an image with dissimilar pose respectively. The projection function $f : X \rightarrow Y$ parameterized by w, is learnt such that the L_2 distance d_i^p between the pair (y_i, y_i^p) (here $y = f(x)$) is less than the L_2 distance d_i^n between the pair (y_i, y_i^n) by a margin of 1. Formally, the parameters w are learnt by minimizing the following equation,

$$L = \frac{\alpha}{2}\|w\|^2 + \sum_{i \in I} \max(0, 1 - \left(d_i^n - d_i^p\right)) \tag{2}$$

where α is a hyper-parameter to control the amount of regularization and I is the set of indices of the training samples. The above equation is minimized using stochastic gradient descent where the gradient is given by,

$$\frac{\partial L}{\partial w} = \alpha * w + \sum_{i \in I} \frac{\partial g}{\partial w}. \tag{3}$$

Here $g = \max(0, 1 - (d_i^n - d_i^p))$ and its gradient is given by,

$$\frac{\partial g}{\partial w} = \begin{cases} 0, & \text{if } 1 - \left(d_i^n - d_i^p\right) \leq 0 \\ -\frac{\partial d_i^n}{\partial w} + \frac{\partial d_i^p}{\partial w}, & \text{otherwise.} \end{cases}$$

The gradient of the L_2 distance d between two points (y_1, y_2) is given by,

$$\frac{\partial d(y_1, y_2)}{\partial w} = \frac{1}{d(y_1, y_2)}(y_1 - y_2)^T \left[\frac{\partial y_1}{\partial w_k} - \frac{\partial y_2}{\partial w_k}\right] \tag{4}$$

For the projection function, we use convolutional neural networks (CNNs) which are highly non-linear and can handle the articulations of the poses. The architecture of our network is again based on Krizhevsky et al. [33] and is shown in Fig. 5. The network has three identical CNNs, both in terms of architecture and the parameters and is illustrated in Fig. 6. Each CNN has five convolutional layers followed by two fully connected layers. For the non-linearity, we use leaky relu unit [35] which is effective against saturation. The weights of all the layers are randomly initialized from the Gaussian distribution. For the convolutional layers, (0,0.01) are used as mean and standard deviation respectively. For the fully connected layers (0,0.005) are used as mean and standard deviation respectively. The size (*width* × *height*) of the first, second and fifth max pooling layers are 3 × 3 with a stride of 2. The third and fourth convolutional layers are not followed by any max-pooling layers. Both the first and second max-pooling layers are followed by cross map local response normalization with a size of 5 × 5 and its parameters are given in [33].

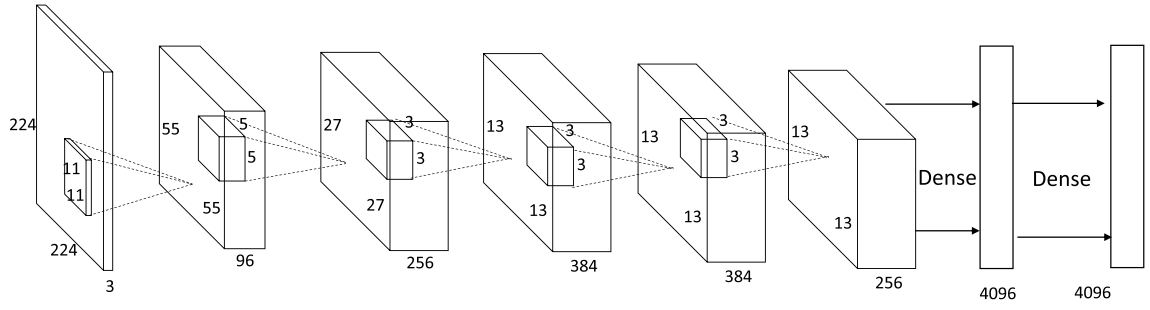

Fig. 5. CNN architecture: This architecture is a minor variation of the CNN architecture proposed in [33]. The number of layers and the number of the parameters are depicted.

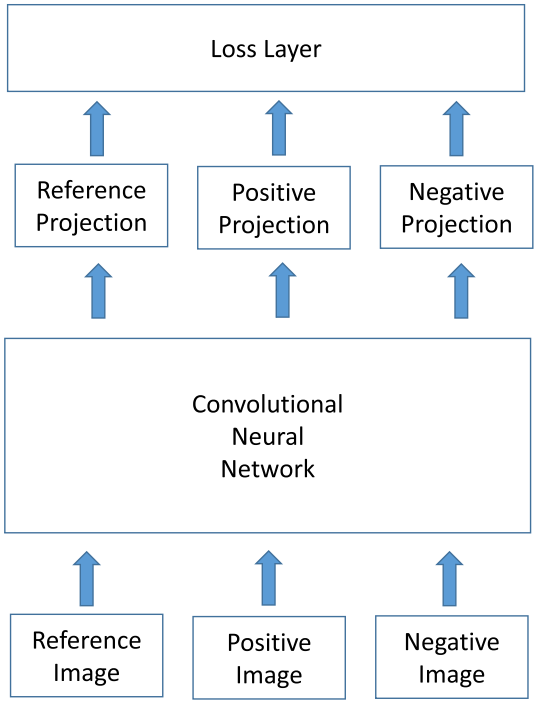

Fig. 6. Triplet architecture: A training sample to this network contains a tuple of three images. The three images are reference image, a positive image which contains the similar pose as reference image, and a negative image which contains a dissimilar pose to the reference image. The images are forward propagated and passed to the loss function which measures how well the network separates reference-positive pair and the reference-negative pair. This architecture can be visualized as containing three networks, each for an image, with the same CNN architecture and same set of parameters at any given time.

For training the network, a mini-batch of 128 image triplets X are presented to the network. The sampling strategies and augmentations are discussed in the next sections. For a given image triplet X_i, the three images are passed through the network to obtain feature maps Y_i of the final layer. These feature maps are then used to obtain the gradient (Eq. (3)) of the loss function defined in Eq. (2). The weight vectors are then updated using the following equation:

$$w^t = w^{t-1} - \eta\frac{\partial L}{\partial w} + \beta w_m^{t-1} \tag{5}$$

$$w_m^t = w_m^{t-1} - \eta\frac{\partial L}{\partial w}. \tag{6}$$

Here w_m is the standard momentum term, $\beta = 0.9$ is the rate of momentum and η is the learning rate. Note that the gradient of the L_2 regularization term in Eq. (2) is accounted for in Eq. (3). The network is trained using the popular deep learning library Theano [4,5].

4.2. Data

The challenge with models that use triplets is the data explosion. A data set of just ten thousand images can produce a training set of one trillion triplets. While training the CNNs, which requires thousands of data augmentations [33] (through minor translations, scaling and rotations), this problem is further compounded. Given the computational constraints, clearly it is not possible to train the network on all the triplets and their augmentations. Yet, it is not prudent to discard them from which the network can learn. To solve this problem, we propose a method which mines the difficult triplets and their augmentations. By presenting these difficult examples to the network and leaving the simpler examples out, both the computational efficiency and better utilization of data are achieved. The method works by first mining for difficult triplets and then difficult augmentations per triplets. Both these mining steps are described below.

4.2.1. Triplet mining

Many real datasets follow power law and have significantly higher samples for certain classes. To correct this skew, we organize the data set into K clusters $\{C_1, \ldots, C_K\}$ where samples belonging to each cluster have similar poses. We obtain these clusters using K-means algorithm described in Section 3.1.1. For the first few epochs (5 in our implementation), we randomly sample image triplets in the following way. First R images are randomly sampled from each cluster without replacement. For each sample, T positives and negatives are randomly sampled to form triplets. Thus the total number of triplets are RTK. At the end of these epochs the network would have reasonable estimate of parameters.

After the initial epochs, the data is mined to obtain difficult examples for training. As before, from each cluster C_i, R_i samples are randomly sampled. Each sample is then forward propagated to obtain the loss value. All the sample values whose loss is 0 are discarded. The remaining samples, which the network found difficult, are sent for training. Similar strategies have been used in other applications [43].

4.2.2. Augment mining

Given a triplet T_i, it is augmented with minor translated, rotated and scaled versions of itself. Each image in the triplet is transformed by small random translation, rotation and scaling several times to obtain the augmentations T_i^j, where $j = 1 \ldots N$. In our implementation, we obtain 4 augmentations per triplet. These augmented triplets are then forward propagated to obtain the loss given in Eq. (2). The M augmented triplets (5 in our implementation) with the non-zero loss are retained for training.

5. Pose search

In this section, we first describe our pose search approaches. We then review three standard retrieval methods for the pose search task. Later in the paper (Section 6.4), we compare the proposed pose search method against standard retrieval schemes described below. All the methods below take an expanded bounding box as input.

5.1. Proposed deep poselets

Given a test image, all the deep poselets are run on it using the procedure described in Section 3.4 and the detection scores are noted. All the deep poselet detections are clustered by the person to which they belong. These deep poselet detections are then rescored using spatial reasoning (Section 3.5). Finally a feature vector of K dimensions, where K is the number of deep poselet detectors, is constructed by max pooling the detections. The feature is then l_2 normalized. Thus for each upper body in the dataset, a feature vector is constructed.

Given a query image, a feature representation is created using the method described above and it is compared against all the samples in the dataset using Euclidean distance. The samples in the dataset are sorted by distance and presented to the user.

5.2. Proposed deep pose embedding network

Given a query image, it is passed through the trained convolutional neural network and the output representation is noted. This representation is compared against all the samples in the dataset using Euclidean distance. The samples in the dataset are sorted by distance and presented to the user.

5.3. Bag-of-visual words models [44]

Given a training data composed of images with people in various poses, the SIFT features are extracted at the key points and 1000 visual words are obtained. Given a test upper body detection, the SIFT features are extracted in the expanded bounding box and bag of words representation is obtained using the visual words computed from the training data. This representation is then compared against all the images in the database. The distances or similarity scores are sorted to obtain the ranked list.

5.4. Human pose estimators [9,37,52]

Following the method proposed by Jammalamadaka et al. [28], the HPE algorithms are used for the pose search task as described below. First the pose estimation algorithms Yang and Ramanan [52], Chen and Yuille [9] and Pfister et al. [37] are run on all the expanded versions of the upper body detections in the database to obtain the pose estimates. This HPE algorithm gives the locations of various body joints by efficiently searching over multiple scales and all possible translations. For each pose estimate, the sine and cosine of upper and lower parts of both the arms are extracted to form a pose representation. Given a test upper body bounding box, the above procedure is applied to obtain the pose representation. It is then compared against all the instances in the database and the ranked list is obtained after sorting the scores.

5.5. Berkeley poselets [6]

Here, all the poselet classifiers are run on an image to obtain poselet detections. These poselet detections are then pooled into clusters based on the person bounding box, and are max pooled to obtain a description of the human pose. The above procedure is applied on the database and the representations are stored. Given the query sample the above representation is obtained and is compared against all the samples in the database. The ranked list is obtained by sorting the scores.

Table 1
The contributions of various datasets before adding the flipped versions.

Dataset	Train	Val	Test	Total
H3D [6]	238	0	0	238
ETH PASCAL [15]	0	0	548	548
Buffy [21]	747	0	0	747
Buffy-2 dataset [27]	396	0	0	396
Movie dataset [27]	1098	491	2172	3756
FLIC [42]	2724	2279	0	5003
MPII human pose [3]	6742	0	0	6742
Poses in wild [10]	660	0	0	660
We are family [16]	1290	0	0	1290
Synchronic activities [17]	1112	0	0	1112
Total	15,007	2764	2720	20,491

6. Experiments

In this section, we present the experimental evaluation of the deep poselet method and the pose search method. First the data used for both the tasks is described in detail. Then the experimental setup and results for the deep poselet method and pose search method are described.

6.1. Data

Training deep poselet classifiers and deep pose embedding algorithms require moderately large amounts of data. We thus pool several existing datasets to create training and test data for deep poselets and pose search. The datasets used are Buffy stickmen dataset [21], ETH PASCAL dataset [15], the H3D dataset [6], Buffy stickmen-2 dataset [27], movie stickmen dataset [27], FLIC dataset [42], MPII human pose dataset [3], Poses in the wild dataset [10], We are family dataset [16] and Synchronic Activities dataset [17]. Each of these datasets contains images and stick figure annotations of the humans. Fig. 7 shows some examples from these datasets. For the convenience of pose search method, we consider only those annotations in which all parts are visible. For a partially occluded person, defining

Fig. 7. Images from the dataset: These images show the pose variation in the dataset.

Table 2
Comparison of our method with state-of-art poselet methods on the test data.

Method	AP-test
HOG poselets	32.6
Deep poselets before fine-tuning	48.6
Deep poselets after fine-tuning	**56.0**

Bold in table indicates the proposed method's best result.

a positive instance for retrieval is ambiguous. In all, there are 20,491 fully visible annotations. The statistics are given in the Table 1. To further enhance the dataset size, each image and annotation is horizontally flipped effectively doubling the corpus to 40,982 stickmen. Using the stickman annotations, the bounding box of the upper body is constructed and transformed into the expanded bounding box. To understand the efficacy of various pose representation schemes, the ground truth bounding box is assumed.

The combined dataset of 40,982 samples is divided into training, validation and test datasets. The training dataset consists of Buffy stickmen dataset [21], H3D dataset [6], Buffy-stickmen II dataset [27], five movies from the movie stickmen dataset [27] and twenty movies from FLIC dataset [42]. The validation dataset consists of one movie from movie stickmen dataset [27] and ten movies from FLIC dataset [42]. The testing dataset consists of ETH PASCAL dataset [15] and the remaining five movies from the movie stickmen dataset [27]. This division of data ensures that training and testing datasets have no overlap in movies and helps in evaluating the methods on unseen data. The individual contributions of various datasets to the train, validation and test data are given in Table 1.

6.2. Deep poselets

Given a set of deep poselet detections and ground truth bounding boxes, the deep poselet performance is reported in terms of average precision (AP) in the following way. First all the deep poselet detections in an image are compared against the ground truth bounding boxes using the intersection over union measure (IOU). All the detections which have more 0.35 IOU, a value used in [6], are considered as positive. All the detections are then sorted in the decreasing order of score and AP is calculated using the labels.

6.2.1. HOG poselets

To baseline the performance of the deep poselets, we compare it with poselets which use HOG features. In this method, a linear SVM is trained using the standard hard-negative mining approach [18]. For the positive samples, the HOG feature is extracted in the bounding box. For the negative samples, the HOG feature of all possible bounding boxes in scale and translation space are considered. Given a test sample, the classifier is run on all scales and locations. All the detections which are above a pre-determined threshold (95% recall on the training data) are deemed as positive detections. Further, all the poselet detections which do not overlap more than 0.35 IOU with the 'expected poselet area' (Section 3.3) are discarded. This step improves the average AP by 10%.

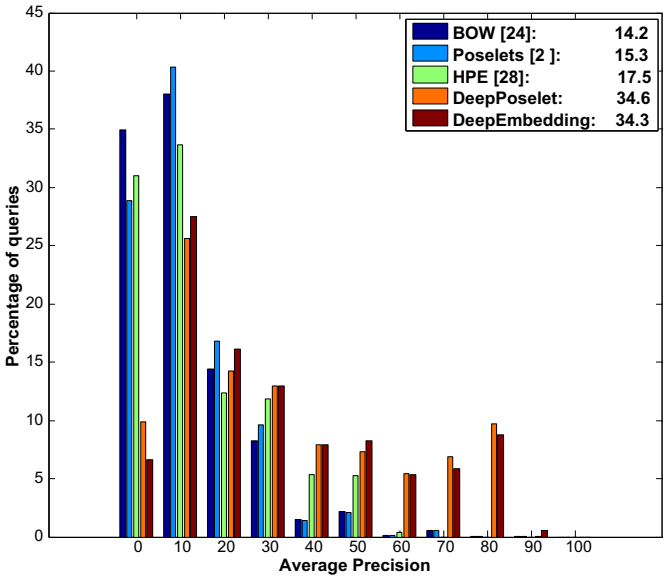

Fig. 9. Posesearch performance: The distribution of query performances by various retrieval methods are shown. Each bar in the graph shows the percentage of queries (Y-axis) having an average precision (X-axis). Thus the more the number of queries on the right side of the graph the better the method. This is also reflected by the mean of the distribution (mAP) of various methods given in the top right corner. It is clear that the proposed method significantly outperforms other methods.

Table 2 shows the performances of HOG poselets and deep poselets. These values are averaged across all the 122 classifiers. It is apparent from the numbers that deep poselets outperform the HOG poselets. It is also observed that out of 122 deep poselets, 118 of them using features before fine-tuning and 120 of them using features after fine-tuning outperform the HOG poselets. Fig. 8 compares the AP curves of HOG poselets and deep poselets. Fig. 10 shows the example detections of three deep poselets. As illustrated in the figure, the performance of the deep poselet improves with more training data.

6.3. Deep pose embedding

Given the training and validation data described in Section 6.1, the performance of the deep pose embedding method is measured using average value of the loss function. The training data is chosen as described in Section 4.2. A total of 44 epochs have been generated

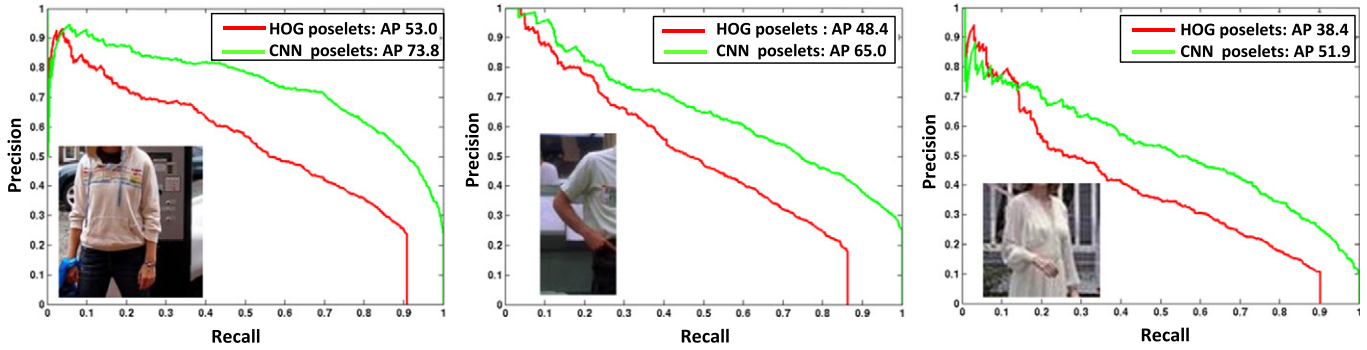

Fig. 8. Deep poselets vs HOG poselets: The graphs show the performance of three deep poselets on test data. The red curve in each graph corresponds to HOG poselet while the green curve corresponds to the deep poselet. As can be seen, the deep poselet outperforms the HOG poselet.

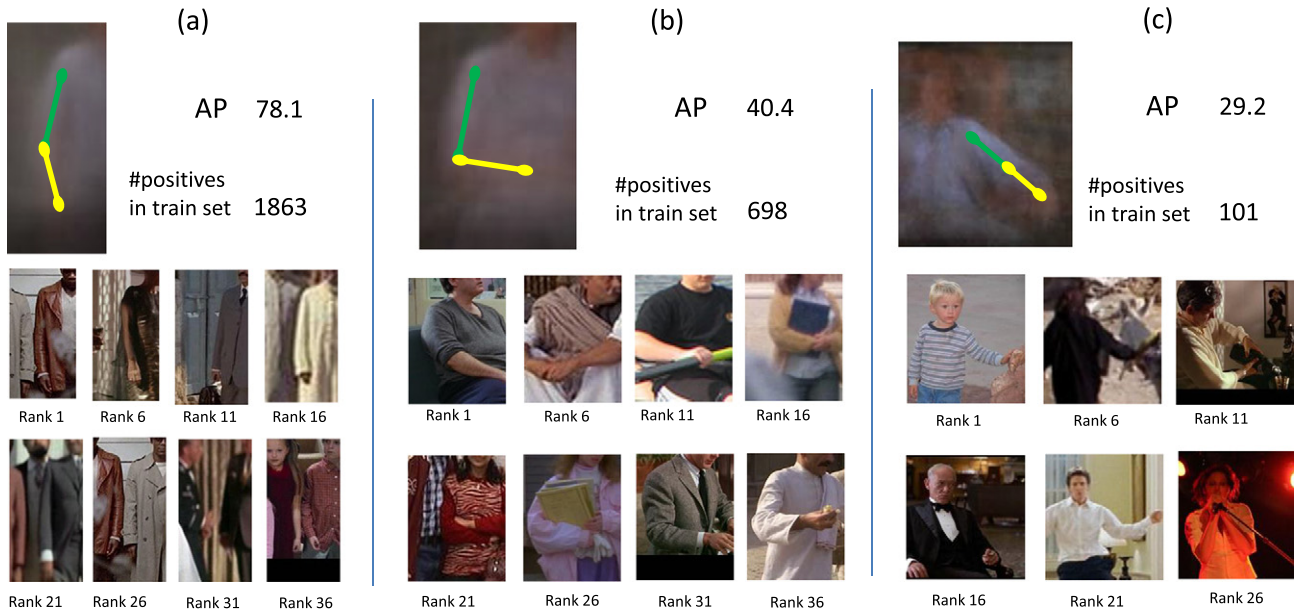

Fig. 10. Top deep poselet detections: Three deep poselets and top detections by them are shown. For each deep poselet, every fifth detection is displayed. In the top 50 detections, while there are no mistakes in deep poselet (a), there are 4 mistakes in deep poselet (b) and 20 mistakes in deep poselet (c). In the deep poselets (b) and (c), the first mistakes occur at ranks 20 and 10 respectively. It can be seen that the performance of deep poselets improve as the number of training samples increases.

to train the neural network. For the validation data, we use a total of 20,000 triplets using the procedure described in Section 4.2.1 for the initial set of training epochs. Note that for validation data, we neither discard triplets with zero loss nor do any augmentations. The Fig. 11 displays the performance of both training and validation data over the epochs. The minimum possible loss value is 0 and there is no upper bound on the loss function. Loss value of 0 indicates that the network is able to separate the positive and negative sample by a margin of at least 1. Loss value (0,1) indicates that the network is still able to separate reference-positive pair and the reference-negative pair but by a margin less than 1. A loss $[1, \text{inf})$ on a training sample indicates that the distance between the reference-positive pair is more than or equal to the reference-negative pair.

The plot in Fig. 11 demonstrates that the learning is converging. It also shows how the loss function on validation data has similar error values as on the training data. This indicates that the network is able to generalize well. As described earlier, the first five epochs do not use any data augmentations. After the fifth epoch, augmentation and data mining applied which explains the sudden spike in the plot.

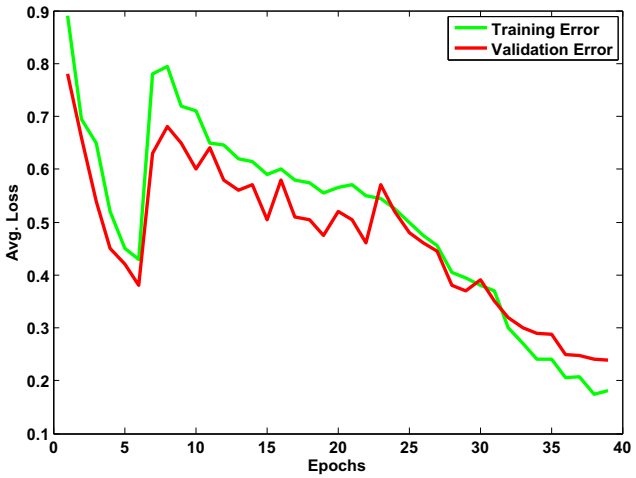

Fig. 11. Deep pose embedding network's performance.

6.4. Pose search

Given a query image, the feature representation is computed and its similarity score or distance is computed with all samples in the test data. These scores are then sorted to obtain a ranked list. The label for each sample in this list, which indicates if the sample has a similar pose as the query, is determined using the part angles as described in Section 3.1.1. Using the ranked list and labels, average precision (AP) is calculated. Each sample in the test data is used as a query to retrieve the results, thus evaluating the various retrieval methods on a total of 5440 queries, the size of test data. The pose search task is evaluated using mean average precision (mAP), which is the average of APs over all the queries.

Table 3 shows the mAPs of various methods over all the queries and the dimension of the pose representation. As is evident, the proposed approaches deep poselet method and deep pose embedding method, with a mAPs of 34.6% and 34.3% respectively, significantly outperform the traditional methods with the best of them at 17.5%. The table also shows that applying spatial reasoning for deep poselets has improved the mAP from 32.9% to 34.6%, an improvement of 1.7%. The new CNN based human pose estimation algorithms 'CNN-HPE I' and 'CNN-HPE II', which are currently the state-of-the-art on several datasets, do better than their traditional counter-parts. The 'CNN-HPE II' algorithm mildly outperforms our algorithm by 2.5%. We have to note that these both algorithms have used sophisticated

Table 3
Pose search performance (mAP) and pose representation's dimensions of various methods.

Methods	#Dimension	mAP
Bag of visual words [44]	1000	14.2
Berkeley poselets [6]	150	15.3
Human pose estimation [52]	8	17.5
CNN-HPE I [37]	8	23.8
CNN-HPE II [9]	8	**37.1**
Ours – deep pose embedding	4096	34.3
Ours – deep poselets	122	32.9
+Spatial reasoning	122	**34.6**

Bold in table indicates the proposed method's best result and also best result from competing methods.

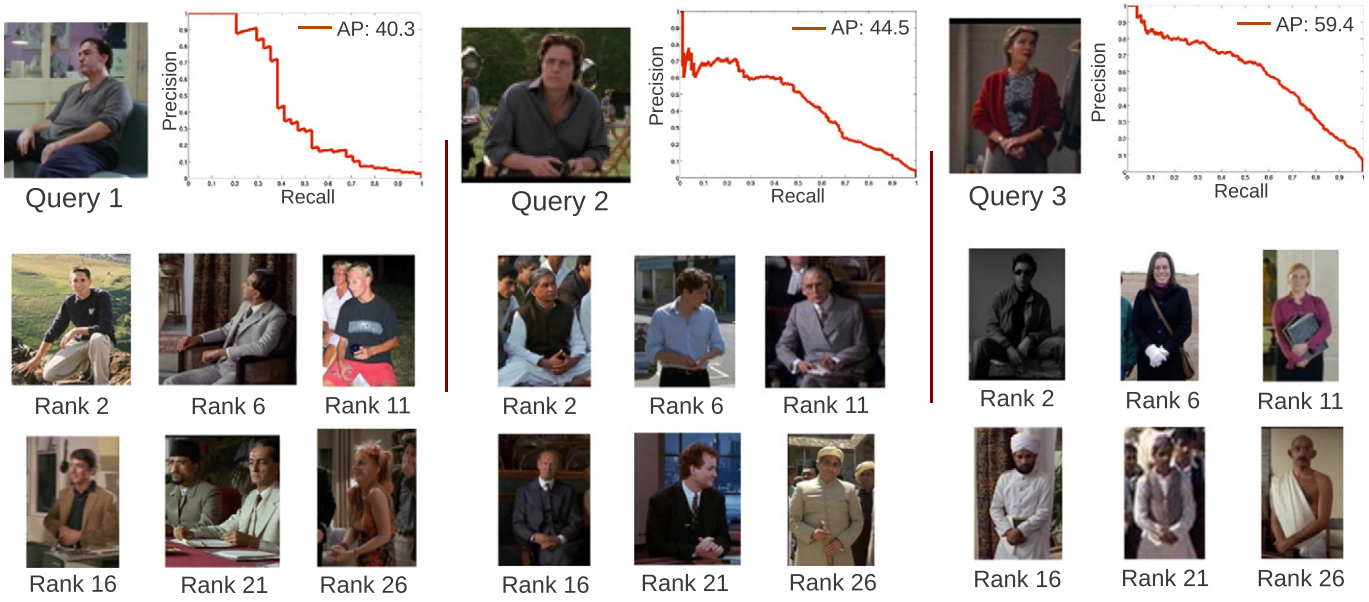

Fig. 12. Example retrievals by deep poselets: Top retrievals and AP curves for three queries are displayed. For the top retrievals every fifth sample from the top in retrieved list is displayed. The first mistake occurs at ranks 11, 4 and 33 respectively for the above queries.

modeling while ours uses a standard and relatively small neural network. We strongly believe that our method will significantly benefit from initializations with a pretrained model, increasing the depth of the network and improving the data mining strategies. Fig. 9, which shows the distribution of pose search APs over all the queries, gives an insight into our method's better performance. Our methods perform extremely well on queries such as query 3 in Fig. 12 with APs in the excess of 50%. Such queries have low intra-class variation and high frequency. The second mode on the right in Fig. 9 corresponds to these poses. On queries with rare poses, our method gives better APs, while other methods post near zero APs. Few examples queries and their top retrievals are displayed in Fig. 12.

Fig. 9 shows an interesting pattern where the AP distributions for the deep poselet method and deep pose embedding method are very similar. This observation throws up questions and we attempt to answer them here: (i) Do they have similar failure cases? If we consider all the queries which have an AP less than 10% as failures, the number of common failure queries between both the methods are about 70% of the total failure queries for either of the methods. This clearly suggests that both the methods have similar failure cases.

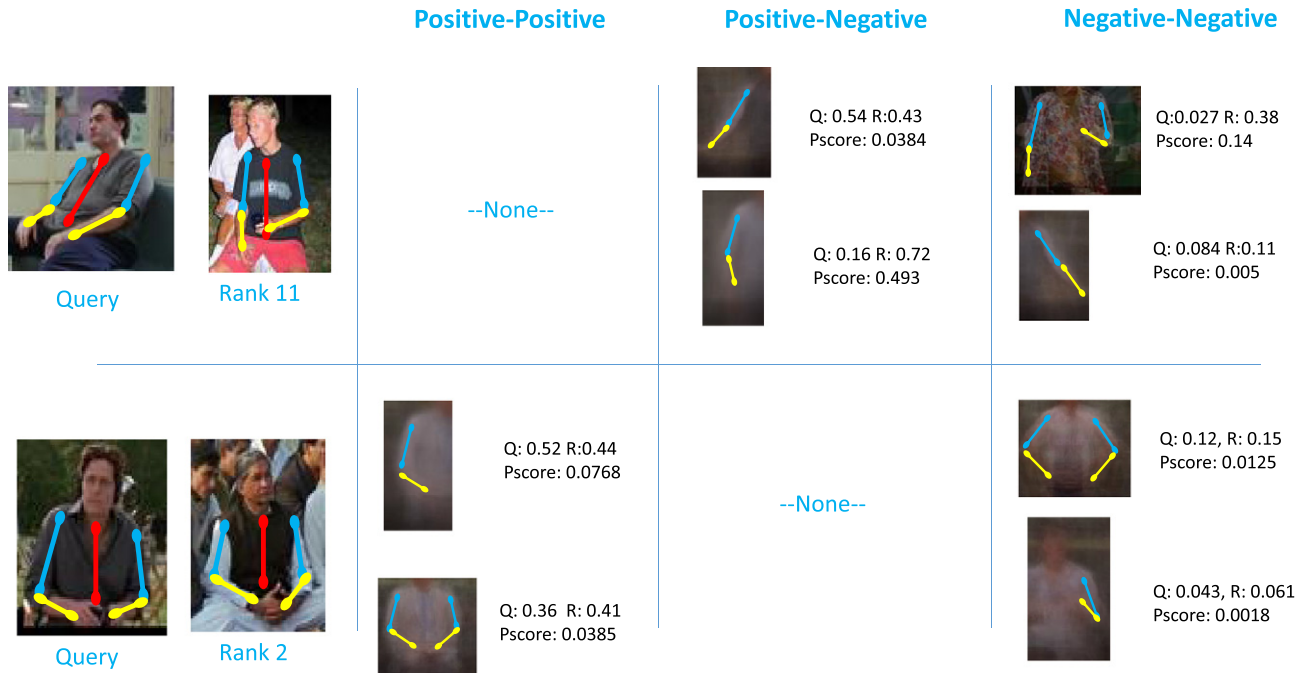

Fig. 13. Deep poselet analysis: For the two query–retrieval pairs, the top and bottom poselets based on prominence scores are displayed for all three poselet categories. For the first pair (above), the retrieval is incorrect. This analysis clearly shows that the top poselet in positive–negative category has misfired for the retrieval image. Similarly the top poselet in negative–negative category also misfired. For the second pair (below), the retrieval is correct and all the prominence scores reflect it.

(ii) What are their strengths and weaknesses? The deep poselet method is easily comprehensible. A failure can be easily understood and can typically be attributed to either a poselet misfiring or the query's pose not being covered by any of the poselets. The flip side is that there are several steps involved in training and building the feature vector. Further a unified method for locating the parts and reasoning the spatial consistency could have better performance.Consider the following analysis of pose retrieval using deep poselets. For both query and the retrieved image, the poselet labels and detection scores are noted. First, each poselet is classified into one of the three categories: (a) Positive–positive, (b) positive–negative or (c) negative–negative where, for example, "positive–negative" label would mean one of query/retrieval is positive and other is negative. In order to understand which of the poselet detections have affected the result most, the product of detection-score-sum and detection-score-absolute-difference is noted and is termed as prominence score. For positive–positive and negative–negative poselets, the prominence score should be small and for positive–negative poselets the prominence score should be large. Note that each poselet category are independently analyzed. The prominence scores of poselets belonging to the same category are appropriately sorted (ascending order for positive–negative poselets and descending order for other pairs). The poselets at the top of the sorted list are responsible for wrong retrieval and poselets at the bottom are responsible for correct retrieval. Fig. 13 demonstrates this analysis on two query–retrieval pairs.

For the deep embedding though, the training and building the feature vector is straight forward. But what is being learnt is not very clear and understanding the failures can be very difficult.

7. Conclusions

In this work, we successfully demonstrated a novel approach for image and video search using pose as a query modality. We proposed two ways, deep poselet method and deep pose embedding method, to obtain the pose descriptors and perform the pose retrieval. In the first method, we have shown that pose space can be discretized by using 'pose-sensitive' deep poselets. These deep poselet detectors model a subset of body parts in a particular pose. We have shown that using the state-of-the-art CNN [13] features, these detectors perform very well. They have been used as a basic building blocks in constructing a feature representation for pose. In the second method, we have shown how an image can be directly mapped to a lower dimensional pose-sensitive space. We then empirically demonstrated that pose retrieval using our both methods are on par with competing pose retrieval methods.

Acknowledgments

We would like to thank Aniket Singh for helping with the implementation of the triplet network in the Theano framework and for other helpful discussions during the implementation. We also would like to thank James Charles for sharing the trained models of a HPE algorithm. We are grateful for financial support from the UKIERI and EPSRC (EP/M013774/1) program grant Seebibyte.

References

[1] B. Alexe, T. Deselaers, V. Ferrari, Measuring the objectness of image windows, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 34 (11) (2012) 2189–2202.
[2] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, Advances in Neural Information Processing Systems (NIPS), 2002.
[3] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D human pose estimation: new benchmark and state of the art analysis, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[4] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I.J. Goodfellow, A. Bergeron, N. Bouchard, Y. Bengio, Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.
[5] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: a CPU and GPU math expression compiler, Proceedings of the Python for Scientific Computing Conference (SciPy), 2010. oral Presentation.
[6] L. Bourdev, J. Malik, Poselets: body part detectors trained using 3D human pose annotations, IEEE International Conference on Computer Vision (ICCV), 2009.
[7] L.D. Bourdev, F. Yang, R. Fergus, Deep poselets for human detection, CoRR (2014) abs/1407.0717.
[8] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a siamese time delay neural network, Advances in Neural Information Processing Systems Conference (NIPS), 1993. pp. 737–744.
[9] X. Chen, A. Yuille, Parsing occluded people by flexible compositions, Computer Vision and Pattern Recognition (CVPR), 2015.
[10] A. Cherian, J. Mairal, K. Alahari, C. Schmid, Mixing body-part sequences for human pose estimation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[11] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005. pp. 539–546.
[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: a deep convolutional activation feature for generic visual recognition, 2013. arXiv preprint arXiv:1310.1531
[14] H. Drucker, C.J.C. Burges, L. Kaufman, A.J. Smola, V. Vapnik, Support vector regression machines, Advances in Neural Information Processing Systems Conference (NIPS), 1996.
[15] M. Eichner, V. Ferrari, Better appearance models for pictorial structures, British Machine Vision Conference (BMVC), 2009.
[16] M. Eichner, V. Ferrari, We are family: joint pose estimation of multiple persons, European Conference on Computer (ECCV), 2010. pp. 228–242.
[17] M. Eichner, V. Ferrari, Human pose co-estimation and applications, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 34 (11) (2012) 2282–2288.
[18] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 32 (9) (2010) 1627–1645.
[19] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, Int. J. Comput. Vis. (IJCV) 61 (1) (2005) 55–79.
[20] V. Ferrari, M. Marin-Jimenez, A. Zisserman, Pose search: retrieving people using their pose, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[21] V. Ferrari, M.J. Marín-Jiménez, A. Zisserman, Progressive search space reduction for human pose estimation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[22] V. Ferrari, T. Tuytelaars, L.J.V. Gool, Real-time affine region tracking and coplanar grouping, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2001.
[23] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[24] G. Gkioxari, P. Arbelaez, L. Bourdev, J. Malik, Articulated pose estimation using discriminative armlet classifiers, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
[25] G. Gkioxari, B. Hariharan, R. Girshick, J. Malik, Using k-poselets for detecting people and localizing their keypoints, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[26] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, CoRR (2012) abs/1207.0580.
[27] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, C.V. Jawahar, Has my algorithm succeeded? An evaluator for human pose estimators, IEEE European Conference on Computer Vision (ECCV), 2012.
[28] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, C.V. Jawahar, Video retrieval by mimicking poses, International Conference on Multimedia Retrieval (ICMR), 2012.
[29] N. Jammalamadaka, A. Zisserman, C.V. Jawahar, Human pose search using deep poselets, International Conference on Automatic Face and Gesture Recognition, 2015.
[30] T. Joachims, Optimizing search engines using clickthrough data, Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23–26, 2002, Edmonton, Alberta, Canada, 2002. pp. 133–142.
[31] M. Kiefel, P. Gehler, Human pose estimation with fields of parts, European Conference on Computer Vision (ECCV), 2014.
[32] A. Krizhevsky, Cuda-Convnet: a fast C++/CUDA implementation of convolutional neural networks.
[33] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems Conference (NIPS), 2012.
[34] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE (1998)

[35] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL), 2013.
[36] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, IEEE International Conference on Machine Learning (ICML), 2010.
[37] T. Pfister, J. Charles, A. Zisserman, Flowing ConvNets for human pose estimation in videos, IEEE International Conference on Computer Vision, 2015.
[38] L. Pishchulin, M. Andriluka, P.V. Gehler, B. Schiele, Poselet conditioned pictorial structures, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
[39] L. Pishchulin, M. Andriluka, P.V. Gehler, B. Schiele, Strong appearance and expressive spatial models for human pose estimation, IEEE International Conference on Computer Vision (ICCV), 2013.
[40] J. Platt, Probabilistic outputs for support vector machines and comparison to regularize likelihood methods, Advances in Large Margin Classifiers, 2000.
[41] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN Features off-the-shelf: an astounding baseline for recognition, CoRR (2014) abs/1403.6382.
[42] B. Sapp, B. Taskar, MODEC: multimodal decomposable models for human pose estimation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
[43] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[44] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, IEEE International Conference on Computer Vision (ICCV), 2003.
[45] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[46] G.W. Taylor, I. Spiro, C. Bregler, R. Fergus, Learning invariance through imitation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[47] J.J. Tompson, A. Jain, Y. LeCun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, Advances in Neural Information Processing Systems (NIPS), 2014.
[48] A. Toshev, C. Szegedy, DeepPose: human pose estimation via deep neural networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[49] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. (IJCV) 104 (2) (2013) 154–171.
[50] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, Y. Wu, Learning fine-grained image similarity with deep ranking, IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 1386–1393.
[51] Y. Wang, D. Tran, Z. Liao, Learning hierarchical poselets for human parsing, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[52] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of–parts, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.