Human Pose Search using Deep Poselets

Nataraj Jammalamadaka[1] Andrew Zisserman[2] C. V. Jawahar[1]

CVIT[1]
IIIT Hyderabad

Visual Geometry Group[2]
Department of Engineering Science
University of Oxford

***Abstract*— Human pose as a query modality is an alternative and rich experience for image and video retrieval. We present a novel approach for the task of human pose retrieval, and make the following contributions: first, we introduce 'deep poselets' for pose-sensitive detection of various body parts, that are built on convolutional neural network (CNN) features. These deep poselets significantly outperform previous instantiations of Berkeley poselets [2]. Second, using these detector responses, we construct a pose representation that is suitable for pose search, and show that pose retrieval performance exceeds previous methods by a factor of two. The compared methods include Bag of visual words [24], Berkeley poselets [2] and Human pose estimation algorithms [28]. 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.**

I. Introduction

As an atomic unit of gesture and action, pose is an important aspect of human communication. Accordingly it has been the focus of many works [6], [10], [14], [17], [20], [23], [27], [28] 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. Figure 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 a novel approach to pose search using 'deep poselets'. 'Deep poselets' 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 work 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 [18] 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.

The pose search task was originally proposed by Ferrari *et al.* [9] 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.* [16] extended [9] by demonstrating pose search on 3.1 Million frames taken from 22 Hollywood movies. In [16], 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.* [15] detects wrong pose estimates, and hence is able to filter them out.

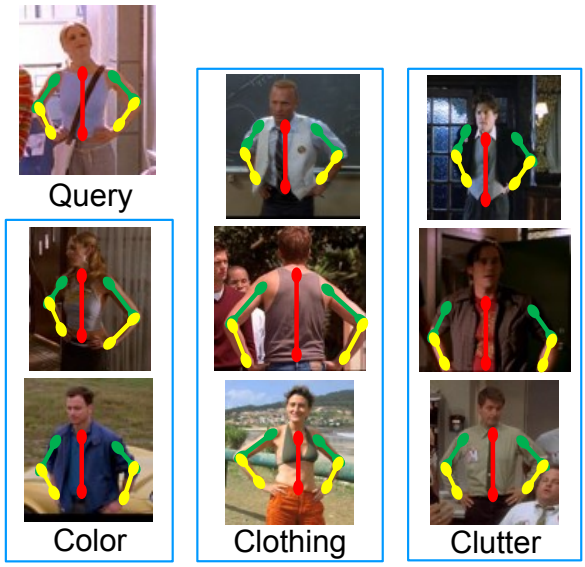

Fig. 1. **Pose Search:** For the query image (top-left corner), the pose search system retrieves people in the database who are in the same pose as the query image. The system has to be invariant to the color and type of the clothes, the clutter in the background and presence of other people in the image.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

The pose retrieval methods of [9], [16], [15] use HPE algorithms. Among the many HPE algorithms, pictorial structures [8] based methods [6], [10], [28] in particular are very popular. Methods such as [20] have integrated a modified version of Berkeley poselets [2] with pictorial structures, while other methods such as [23] have used the poselets for inferring the pose. With the success of convolutional neural networks, a few methods [25] have been proposed using CNN architectures. The work by Gkioxari *et al.* [13] is the closest to ours. Both our approach and [13] use body part detectors which are sensitive to pose. While the main focus of [13] 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.* [13] have used HOG features. Even though the performance of HPE is improving, it is not good enough to be used as base technology for tasks such as action recognition and pose retrieval. A single mistake by the algorithm, say a mistaken wrist position, renders the whole pose estimate wrong. Our proposed approach addresses this by softly encoding several locations for each body part.

Deep poselets, inspired by poselets [2], model a subset of parts (e.g, left upper and lower arm) appearing in a particular pose. The key difference between [2] and our method is that [2] is for person detection, and ours is for pose detection. The different poselet types in [2] are derived from data by randomly selecting a large number of potential candidates, and then successively pruning them using various heuristics. Several such classifiers are trained with the objective of detecting a person. All these classifiers are then run on a test image. Based on the relative locations between the detections, the location of the person is estimated. In our approach, we obtain specific poselets and the positive instances belonging to them using a data driven process described in section II-A. Given the poselets and instances belonging to them, a classifier is trained to discriminate positive instances from the negatives ones. The features for these classifiers are learnt using CNNs. CNNs have significantly improved the performance of image classification [4], [12], [18] on the challenging ImageNet dataset [3]. Motivated by Razavian *e*t al. [22], we use an architecture similar to [18] to learn features. The details of the feature extraction and training are described in section II-C. 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 III. Using these deep poselets and their detection scores, a representation for a pose is constructed. The representation is then used to perform pose search as described in section IV. In the experimental section V, we evaluate both the deep poselet method and the pose search method by comparing them with relevant baselines.

II. 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. Figure 2 illustrates a few example deep poselets.

A. 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.

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.

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 modelled, 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 modelled 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 preprocessed 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 with 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. Figure 2 illustrates a few deep poselets discovered using the above process.

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 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 degrees respectively.

B. 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 by using unsupervised object proposal methods like objectness [1] and selective search [26]. 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. EPA gives the highly probable location of the deep poselet within the bounding box of the person.

Deep poselets typically occur in a localized region within expanded bounding box. For example, a deep poselet modelling the left arm typically lies in the left half of the bounding box. We term this localized region as 'expected poselet area'. The search space of the deep poselet can be restricted to this 'expected poselet area' 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 are encompassed by expected poselet area. 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.

C. 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. Convolutional neural networks, first proposed by Lecun *et al.* [19], model an object as composition of patterns starting from edges to higher level parts like faces. A CNN consists of convolutional layers, pooling layers and fully connected layers. A convolutional layer consists of K 3-D filters which are applied to the input to obtain K feature maps. At each location of the feature map, a nonlinear function called a neuron activation function is applied. The convolutional layers are followed by pooling layers which pool the inputs in a local neighbourhood and typically down-sample the input, thus introducing translation invariance. Finally, fully connected layers take input from all the neurons of the previous layer and act as the reasoning units. Taking input from such a wide context helps in making better informed decisions about the class labels. The network is trained using the back propagation algorithm. In our experiments, we use the implementation of the ImageNet-CNN network by Donahue *et al.* [4]. The ImageNet-CNN [18] is a deep neural network with five convolutional layers and three fully connected layers. Below, the feature extraction and training are explained

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 V) 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 [4]. 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 dropout rate of 0.5 is used. For training the network, the cuda-convnet software is used.

2) Learning SVMs: The SVM training follows an iterative procedure. 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 the AP on validation set converges. In practice, it is found that three iterations suffice. Empirically, this procedure improved the AP by 7% over the method in which the candidate window is used as-is for training. This procedure is reminiscent of best positive bounding box selection used in Felzenswalb *et al.* [7].

D. Testing

Given a test image, it is processed using the human detector algorithm to obtain upper body detections. Each upper

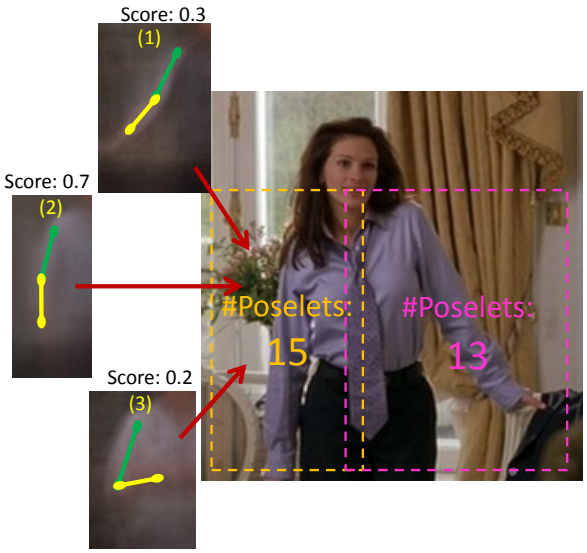

Fig. 3. **Spatial reasoning:** For a given test sample, three deep poselet detections and their scores are shown as belonging to the area marked by an orange rectangle. Detections 1 and 3 are partially correct as the pose of the left upper arm matches that of the test sample. Detection 2 is the correct 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 (detection 1 and 3) get a score of nearly 0. The image also shows that area around the left arm (orange rectangle) has 15 unique deep poselets while area around the right arm (pink rectangle) has 13 unique deep poselets.

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 II-B). 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.

III. 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. Figure 3 illustrates this behavior. In the figure, three deep poselet 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 figure 3) have a score of nearly 1 and incorrect ones (detections 1 and 3 in the figure 3) have a score of nearly 0. For this rescoring, a RBF kernel based regression model [5] is learnt for each deep poselet type P. The input to this model is a feature vector comprising of calibrated scores 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.

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 [21], 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,

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

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 figure 3, 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 II-C) of all the 122 deep poselets are collected. These deep poselets are then clustered using the cluster partitioning algorithm proposed by Ferrari *et al.* [11]. The algorithm returned 31 clusters, where poselets in each cluster form a mutually exclusive set.

IV. Pose Search

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

Our pose search approach: Given a test image, all the deep poselets are run on it using the procedure described in section II-D 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 III). 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.

Bag-of-visual words models [24]: 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

Dataset	Train	Validation	Test	Total
H3D dataset [2]	238	0	0	238
ETH PASCAL dataset [6]	0	0	548	548
Buffy stickmen dataset [10]	747	0	0	747
Buffy stickmen-2 dataset [15]	396	0	0	396
Movie stickmen dataset [15]	1098	491	2172	3756
FLIC [23]	2724	2279	0	5003
Total	5198	2764	2720	10682

TABLE I

THE CONTRIBUTIONS OF VARIOUS DATASETS BEFORE ADDING THE FLIPPED VERSIONS.

Fig. 4. **Images from the dataset:** These images show the pose variation in the dataset.

is then compared against all the images in the database. The distances or similarity scores are sorted to obtain the ranked list.

Human pose estimator [28]: Following the method proposed by Jammalamadaka *et al.* [16], the HPE algorithms are used for the pose search task as described below. First the pose estimation algorithm [28] is 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.

Berkeley poselets [2]: 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.

Layer	Before fine tuning	After fine tuning
pool5	**67.5**	69.5
fc6	59.7	69.6
fc7	47.4	**69.6**

TABLE II

PERFORMANCE OF FIVE RANDOMLY CHOSEN DEEP POSELETS ON VARIOUS CNN FEATURES OVER THE TEST DATA.

V. 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.

A. Data

Training deep poselet classifiers 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 [10], ETH PASCAL dataset [6], the H3D dataset [2], Buffy stickmen-2 dataset [15], Movie stickmen dataset [15] and FLIC dataset [23]. Each of these datasets contains images and stick figure annotations of the humans. Figure 4 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 a positive instance for retrieval is ambiguous. In all, there are $10,682$ fully visible annotations. The statistics are given in the Table I. To further enhance the dataset size, each image and annotation is horizontally flipped effectively doubling the corpus to $21,364$ 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 $21,364$ samples is divided into training, validation and test datasets. The training dataset consists of Buffy stickmen dataset [10], H3D dataset [2], Buffy-stickmen II dataset [15], five movies from the movie stickmen dataset [15] and twenty movies from FLIC dataset [23]. The validation dataset consists of one movie from movie stickmen dataset [15] and ten movies from FLIC dataset [23]. The testing dataset consists of ETH pascal dataset [6] and the remaining five movies from the movie stickmen dataset [15]. 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 I.

B. 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

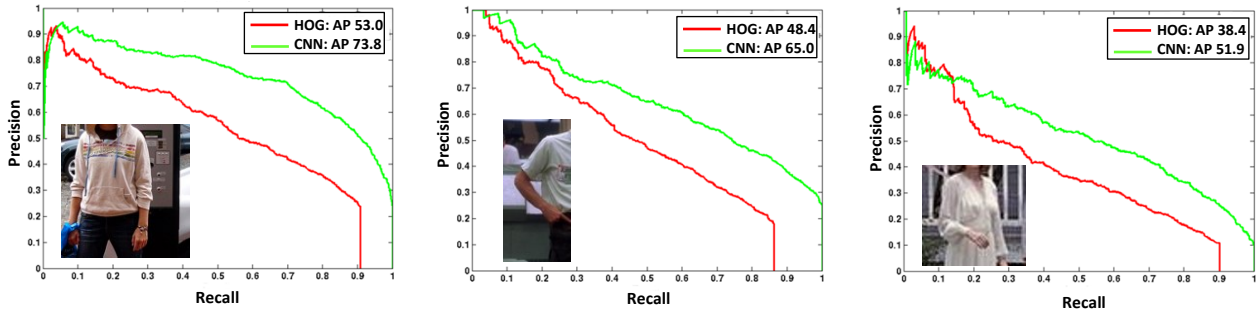

Fig. 5. **Deep poselets vs HOG poselets:** The graphs show the performance of three deep poselets on test data. The red curve in each graph corresponds to HOG poselet while the green curve corresponds to the deep poselet. As can be seen, the deep poselet outperforms the HOG poselet.

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

TABLE III

COMPARISON BETWEEN HOG AND DEEP POSELETS (CNN-FEATURES) ON THE TEST DATA.

have more 0.35 IOU, a value used in [2], are considered as positive. All the detections are then sorted in the decreasing order of score and AP is calculated using the labels.

Deep poselets: Using the procedure described in section II-C, deep poselets are trained using CNN features extracted from the ImageNet network [4], before and after fine-tuning it. The hyper-parameters are set using 3-fold cross validation. We experiment with the features from last pooling layer (pool5), the first (fc6) and second (fc7) fully connected layers. Table II shows the performance of deep poselets using features from different layers averaged over five randomly chosen deep poselets on the testset. For deep poselets using features before fine tuning the network, the last pooling layer (pool5) works best. This is expected as the network is trained on a very different task of object detection. For the deep poselets using the features after fine tuning the network, the features from second fully connected layer (fc7) works best. The deep poselets using features after fine tuning consistently outperform those which use features before fine tuning.

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 [7]. 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 II-C) are discarded. This step improves the average AP by 10%.

Table III 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. Figure 5 compares the AP curves of HOG poselets and deep poselets. Figure 7 shows the example detections of three deep poselets. As illustrated in the figure, the performance of the deep poselet improves with more training data.

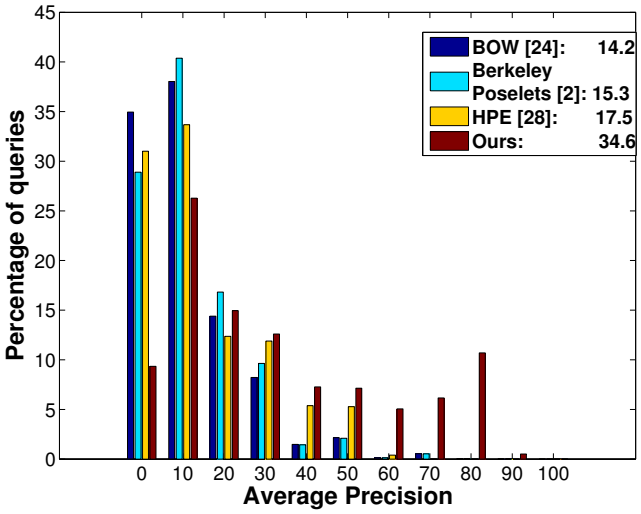

Fig. 6. **Posesearch performance:** The distribution of query performances by various retrieval methods are shown. Each bar in the graph shows the percentage of queries (Y-axis) having an average precision (X-axis). Thus the more the number of queries on the right side of the graph the better the method. 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.

Methods	#Dimension	mAP
Bag of Visual Words [24]	1000	14.2
Berkeley Poselets [2]	150	15.3
Human Pose Estimation [28]	8	17.5
Ours - Deep Poselets	122	32.9
+ Spatial Reasoning	122	**34.6**

TABLE IV

POSE SEARCH PERFORMANCE (MAP) AND POSE REPRESENTATION'S DIMENSIONS OF VARIOUS METHODS.

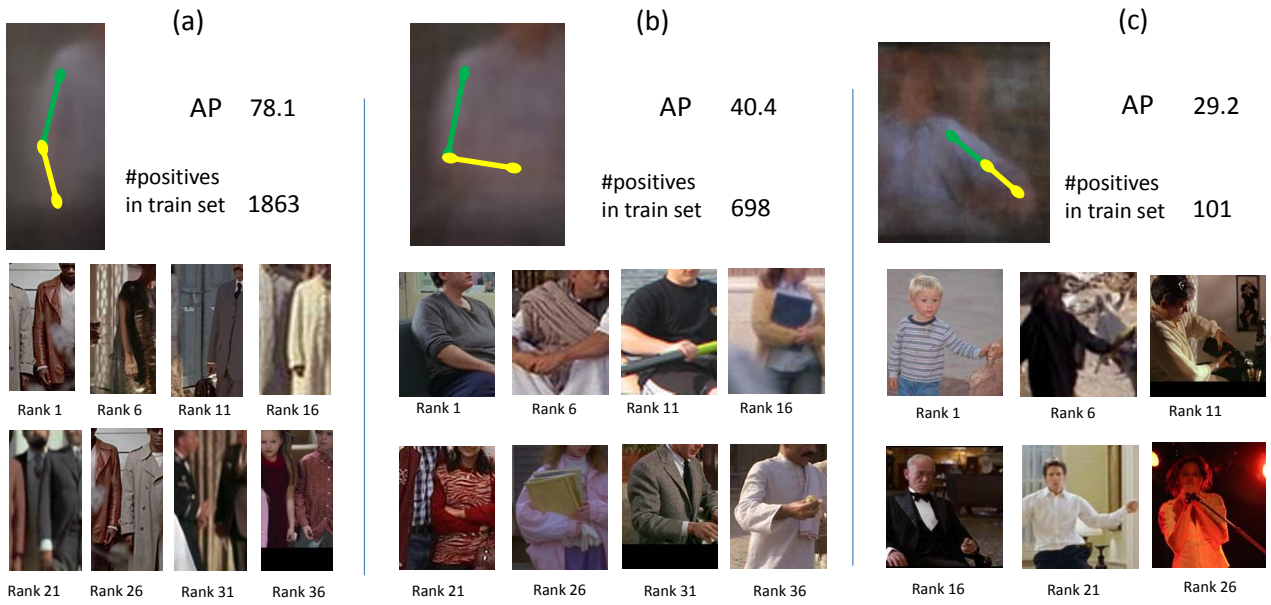

Fig. 7. **Top deep poselet detections:** Three deep poselets and top detections by them are shown. For each deep poselet, every fifth detection is displayed. In the top 50 detections, while there are no mistakes in deep poselet (a), there are 4 mistakes in deep poselet (b) and 20 mistakes in deep poselet (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.

C. *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 II-A. 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 IV shows the mAPs of various methods over all the queries and the dimension of the pose representation. As is evident, the proposed approach, with a mAP of 34.6% significantly outperforms other methods with the best of them at 17.5%. The table also shows that applying spatial reasoning has improved the mAP from 32.9% to 34.6%, an improvement of 1.7%. Figure 6, which shows the distribution of pose search APs over all the queries, gives an insight into our method's better performance. Our method performs extremely well and outperforms other methods on queries such as query 3 in figure 8 with APs in the excess of 50%. Such queries have low intra-class variation and high frequency. The second mode on the right in figure 6 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 figure 8.

Each class of methods used for baselining in table 6 have weaknesses, analysis of which is presented here.

Bag of visual words [24]: While these methods perform very well for general object retrieval, their performance on pose search suffers because, (a) the loss of geometric context when histogramming the visual words, (b) distracting SIFT detections on clothes, and (c) disproportionately small area of arms and legs with respect to the rest of the bounding box. Our method overcomes this problem by learning to ignore distracting patterns like clothing and identifying the key areas in the bounding box where the arms and outline of the human are present.

Berkeley poselets [2]: A pose sensitive poselet describes the body pose of a person. For example, a poselet corresponding to the whole left arm in a certain pose is pose sensitive while that of face and shoulder is not. A scan through the set of poselets detected by [2] shows that most of the detected poselets are not pose sensitive. This renders the method incapable of detecting the human pose. While, in theory, this method is capable of discovering poselets which model the arms in various poses, it would output far more pose-insensitive poselets. Our method and [13] output a compact set of entirely pose sensitive poselets.

Human pose estimators (HPE) [28]: Most HPE algorithms are modelled as a CRF and the pose estimate is obtained by inferring a maximum a posteriori estimate. Typically maximum a posteriori estimation algorithms decide on one particular location for each body part and can potentially make a wrong choice. Clearly this affects the pose retrieval as a mistake in one part effectively renders this detection useless and can potentially worsens the performance of the retrieval system. Our method solves this by taking into account several likely alternative locations, while constructing a representation for the pose. Soft coding of pose is the key to the performance of our algorithm.

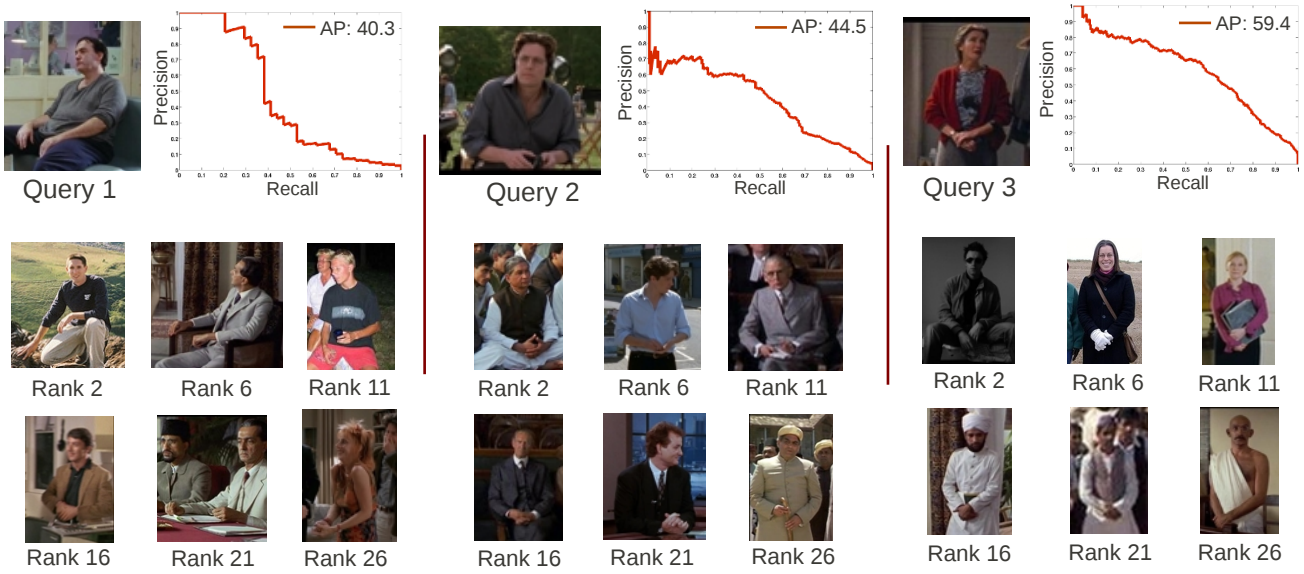

Fig. 8. **Example retrievals:** Top retrievals and AP curves for three queries are displayed. For the top retrievals every fifth sample from the top in retrieved list is displayed. The first mistake occurs at ranks 11, 4 and 33 respectively for the above queries.

VI. Conclusions

In this work, we successfully demonstrated a novel approach for image and video search using pose as a query modality. 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 [4] features, these detectors perform very well. They have been used as a basic building blocks in constructing a feature representation for pose. We then empirically demonstrated that our pose retrieval method outperforms other competing pose retrieval methods by a factor of 2.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, Nov 2012.

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

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[5] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *NIPS*, 1996.

[6] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *BMVC*, 2009.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[9] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. In *CVPR*, 2009.

[10] V. Ferrari, M. J. Marín-Jiménez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[11] V. Ferrari, T. Tuytelaars, and L. J. V. Gool. Real-time affine region tracking and coplanar grouping. In *CVPR*, 2001.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[13] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *CVPR*, 2013.

[14] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*, 2014.

[15] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar. Has my algorithm succeeded? an evaluator for human pose estimators. In *ECCV*, 2012.

[16] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar. Video retrieval by mimicking poses. In *ICMR*, 2012.

[17] M. Kiefel and P. Gehler. Human pose estimation with fields of parts. In *ECCV*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

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

[21] J. Platt. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In *Advances in Large Margin Classifiers*, 2000.

[22] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[23] B. Sapp and B. Taskar. MODEC: multimodal decomposable models for human pose estimation. In *CVPR*, 2013.

[24] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[25] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[26] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[27] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *CVPR*, 2011.

[28] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.