Chapter 6

Pose Representation using Deep Poselets

6.1 Introduction

In this chapter, we propose “Deep Poselets”, a pose representation based on deep neural networks. This representation implicitly encodes human pose information, thus doing away with explicitly locating the body parts. The ‘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 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 [58] 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 is on par with the state-of-the-art pose descriptors. This work [52] is published in FG 2015.

Here we briefly give the outline of the various sections. Deep poselets described in section 6.3, inspired by poselets [13], model a subset of parts (e.g, left upper and lower arm) appearing in a particular pose. The specific poselets and their positive instances are obtained using a data driven process described in section 6.3.1. 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. Motivated by Razavian et al. [75], we use an architecture similar to [58] to learn features. The details of the feature extraction and training are described in section 6.3.3. Given an input test image, all the poselet classifiers are run using the procedure described in section 6.3.4. 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 6.3.5. 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 6.4. In the experimental section 6.5, we evaluate the deep poselet method and the pose search method by comparing them with relevant baselines.
6.2 Related Work

**Poselets:** Poselets [13] are classifiers which model a subset of body parts. The key difference between Bourdev et al. [13] and our method is that Bourdev while et al. [13] is for person detection, and ours is for pose detection. A poselet, for example, can model the head and the left shoulder together. The different poselet types 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 theory, such a method can be used for human pose estimation when the detected poselets model arms in a specific pose. This is of course requires large amounts of data and hence not feasible. The poselets method has recently [14] been improved using CNNs.

To alleviate this problem, Gkioxari et al. [45] propose to discover the poselets by using only the image patches corresponding to the arms. Thus each poselet, termed armlet in et al. [45], models an arm in a particular pose. These classifiers are then run on the image to obtain detection bounding boxes. Using a pre-trained transfer model, the human joint locations are transferred to the armlet detections. This work by Gkioxari et al. [45] is the closest to ours. Both our approach and [45] use body part detectors which are sensitive to pose. While the main focus of [45] 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. [45] have used HOG features.

**Human pose:** The pose retrieval methods of [40, 51, 50] use HPE algorithms. Among the many HPE algorithms, pictorial structures [34] based methods [27, 41, 103] are very popular. Methods such as [70] have integrated a modified version of Berkeley poselets [13] with pictorial structures, while other methods such as [79] have used the poselets for inferring the pose. With the success of convolutional neural networks, a few methods [93, 91] have been proposed using CNN architectures. 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.

6.3 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. Figure 6.1 illustrates a few example deep poselets.
Figure 6.1: **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. Figure 6.2 illustrates all these four steps.

### 6.3.1 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.

In a hypothetical scenario where all poses are equally likely, all possible subsets do occur equally likely. But in reality, the pose distribution of an arm is uneven owing to the factors such as feasibility, energy conservation, popular gestures (e.g., Namaste) and actions. Further, the poses of left arm and right arm are largely decoupled. Thus given any dataset drawn I.I.D. from natural world, poses of just the left arm or the right arm occur much more frequently than say a combination of upper left arm and lower right arm. The head and torso do not contribute much to the human pose (the pose 'Namaste' is the same whether head is straight or tilted) but help in improving the spatial context. With these observations in mind, the following seven subset of body parts, represented by $S_i$, are clustered. 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,
Figure 6.2: 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 a detections while only small number of them are correct. Using linear regression we re-score the poselets detections using the context of other poselet detections. (Best viewed in color)

(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 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. It is observed that when the number of clusters are low, perceptually dissimilar poses fall into the same cluster. But when the number of clusters are large, the similar poses fragment. It is empirically found that for
subsets \( s_1, \ldots, s_6 \), number of clusters as 20 worked well and for \( s_7 \) the number of clusters as 40 worked well. All the clusters which have less than 50 members are discarded. These cluster means are taken as the canonical deep poselets. In our experiments, a total of 122 deep poselets are obtained. Figure 6.1 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 procedure described in section 3.3.3.

### 6.3.2 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 by using unsupervised object proposal methods like objectness [2] and selective search [96]. 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.

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. 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.
6.3.3 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. [23]. The ImageNet-CNN [58] is a deep neural network with five convolutional layers and three fully connected layers. Below, the feature extraction and training are explained.

6.3.3.1 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 6.5) 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 [23]. 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 [88] of 0.5 is used. For training the network, the cuda-convnet software [57] is used.

6.3.3.2 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 [3]: 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. [37].
Figure 6.3: **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.

### 6.3.4 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 6.3.2). 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.

### 6.3.5 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. Figure 6.3 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 6.3) have a score of nearly 1 and incorrect ones (detections
1 and 3 in the figure 6.3) have a score of nearly 0. For this rescoring, a RBF kernel based regression model [25] 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 re-score 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 [72], 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 $\sigma : X \to Y$ where $X, Y \in R$ is learnt. The function $\sigma(x)$ is parameterized by $w_0, w_1$ and is given by,

$$\sigma(x) = \frac{1}{1 + e^{(w_1 x + w_0)}}. \quad (6.1)$$

**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 6.3, 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 6.3.3) of all the 122 deep poselets are collected. These deep poselets are then clustered using the cluster partitioning algorithm proposed by Ferrari et al. [42]. The algorithm returned 31 clusters, where poselets in each cluster form a mutually exclusive set.

### 6.4 Pose Representation and 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 chapter (section 6.5.3), we compare the proposed pose search method against standard retrieval schemes described below. All the methods below take an expanded bounding box as input.

**Proposed deep poselets:** Given a test image, all the deep poselets are run on it using the procedure described in section 6.3.4 and the detection scores are noted. Using the human detector’s output, 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 6.3.5). 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.
### Table 6.1: The contributions of various datasets before adding the flipped versions.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3D [13]</td>
<td>238</td>
<td>0</td>
<td>0</td>
<td>238</td>
</tr>
<tr>
<td>ETH PASCAL [27]</td>
<td>0</td>
<td>0</td>
<td>548</td>
<td>548</td>
</tr>
<tr>
<td>Buffy [41]</td>
<td>747</td>
<td>0</td>
<td>0</td>
<td>747</td>
</tr>
<tr>
<td>Buffy-2 dataset [50]</td>
<td>396</td>
<td>0</td>
<td>0</td>
<td>396</td>
</tr>
<tr>
<td>Movie dataset [50]</td>
<td>1098</td>
<td>491</td>
<td>2172</td>
<td>3756</td>
</tr>
<tr>
<td>FLIC [79]</td>
<td>2724</td>
<td>2279</td>
<td>0</td>
<td>5003</td>
</tr>
<tr>
<td>MPII Human pose [4]</td>
<td>6742</td>
<td>0</td>
<td>0</td>
<td>6742</td>
</tr>
<tr>
<td>Poses in wild [19]</td>
<td>660</td>
<td>0</td>
<td>0</td>
<td>660</td>
</tr>
<tr>
<td>We are family [29]</td>
<td>1290</td>
<td>0</td>
<td>0</td>
<td>1290</td>
</tr>
<tr>
<td>Synchronic Activities [30]</td>
<td>1112</td>
<td>0</td>
<td>0</td>
<td>1112</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>15007</td>
<td>2764</td>
<td>2720</td>
<td>20491</td>
</tr>
</tbody>
</table>

**Bag-of-visual words models [85]:** 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.

**Human pose estimators [103], [18], [68]:** Following the method proposed by Jammalamadaka et al. [51], the HPE algorithms are used for the pose search task as described below. First the pose estimation algorithms Yang and Ramanan [103], Chen and Yuille [18] and Pfister et.al, [68] are run on all the expanded versions of the upper body detections in the database to obtain the pose estimates. These HPE algorithms give 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 [13]:** 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>
<thead>
<tr>
<th>Layer</th>
<th>Before fine tuning</th>
<th>After fine tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool5</td>
<td>67.5</td>
<td>69.5</td>
</tr>
<tr>
<td>fc6</td>
<td>59.7</td>
<td>69.6</td>
</tr>
<tr>
<td>fc7</td>
<td>47.4</td>
<td><strong>69.6</strong></td>
</tr>
</tbody>
</table>

Table 6.2: Performance of five randomly chosen deep poselets on various CNN features over the test data.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG poselets</td>
<td>32.6</td>
</tr>
<tr>
<td>Deep poselets before fine-tuning</td>
<td>48.6</td>
</tr>
<tr>
<td>Deep poselets after fine-tuning</td>
<td><strong>56.0</strong></td>
</tr>
</tbody>
</table>

Table 6.3: Comparison of our method with state-of-art poselet methods on the test data.

### 6.5 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.

#### 6.5.1 Data

Training deep poselet classifiers require moderately large amounts of data. 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 20,491 fully visible annotations. The statistics are given in the Table 6.1. 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 [41], H3D dataset [13], Buffy-stickmen II dataset [50], five movies from the movie stickmen dataset [50] and twenty movies from FLIC dataset [79]. The validation dataset consists of one movie from movie stickmen dataset [50] and ten movies from FLIC dataset [79]. The testing dataset consists of ETH PASCAL dataset [27] and the remaining five movies from the movie stickmen dataset [50]. 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 6.1.
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.

6.5.2 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 than 0.35 IOU, a value used in [13], 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 6.3.3, deep poselets are trained using CNN features extracted from the ImageNet network [23], 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 6.2 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 [37]. 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 6.3.3) are discarded. This step improves the average AP by 10%.
Table 6.3 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 6.4 compares the AP curves of HOG poselets and deep poselets. Figure 6.6 shows the example detections of three deep poselets. As illustrated in the figure, the performance of the deep poselet improves with more training data.

### 6.5.3 Pose Representation and 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 3.3.3. 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 6.4 shows the mAPs of various methods over all the queries and the dimension of the pose representation. As is evident, the proposed deep poselet method, with a mAPs of 34.6%, significantly
Figure 6.6: 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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>#Dimension</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag of Visual Words [85]</td>
<td>1000</td>
<td>14.2</td>
</tr>
<tr>
<td>Berkeley Poselets [13]</td>
<td>150</td>
<td>15.3</td>
</tr>
<tr>
<td>Human Pose Estimation [103]</td>
<td>8</td>
<td>17.5</td>
</tr>
<tr>
<td>CNN-HPE I [68]</td>
<td>8</td>
<td>23.8</td>
</tr>
<tr>
<td>CNN-HPE II [18]</td>
<td>8</td>
<td>37.1</td>
</tr>
<tr>
<td>Ours - Deep Poselets</td>
<td>122</td>
<td>32.9</td>
</tr>
<tr>
<td>+ Spatial Reasoning</td>
<td>122</td>
<td>34.6</td>
</tr>
</tbody>
</table>

Table 6.4: Pose search performance (mAP) and pose representation’s dimensions of various methods.

outperforms 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 modelling while ours uses a standard and relatively small neural network. We strongly believe that our method will significantly benefit from initializations with a pre-trained model and increasing the depth of the network. The pose representations of HPE algorithms are obtained using the method described in section 4.5. For this experiment, we have encoded just the arms and obtain 8D feature vector. Figure 6.5, 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 figure 6.7 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 6.5 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 6.7.

Each class of methods used for baselining in table 6.5 have weaknesses, analysis of which is presented here.

**Bag of visual words [85]:** 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 [13]:** 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 [13] 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 [45] output a compact set of entirely pose sensitive poselets.

**Human pose estimators (HPE) [103]:** 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.

6.6 Summary

In this chapter, we proposed a novel pose representation, “Deep Poselets”, which is based on the
convolutional neural networks. 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 [23] 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 pose retrieval method based on representations from “Deep Poselet”
method are on par with competing pose retrieval methods.

In the next chapter, we describe another novel pose representation based on convolutional neural
networks. This method, termed as “deep pose embedding”, maps an image to a low dimensional pose-
sensitive space. This pose representation too performs on par with other state-of-art pose descriptors on
the pose retrieval task.