Thin-Slicing for Pose: Learning to Understand Pose without Explicit Pose Estimation

Suha Kwak Minsu Cho Ivan Laptev
WILLOW Project Team – Inria / École Normale Supérieure, Paris, France

Abstract

We address the problem of learning a pose-aware, compact embedding that projects images with similar human poses to be placed close-by in the embedding space. The embedding function is built on a deep convolutional network, and trained with triplet-based rank constraints on real image data. This architecture allows us to learn a robust representation that captures differences in human poses by effectively factoring out variations in clothing, background, and imaging conditions in the wild. For a variety of pose-related tasks, the proposed pose embedding provides a cost-efficient and natural alternative to explicit pose estimation, circumventing challenges of localizing body joints. We demonstrate the efficacy of the embedding on pose-based image retrieval and action recognition problems.

1. Introduction

> There can be as much value in the blink of an eye as in months of rational analysis. — *Malcolm Gladwell*

How much detail do we actually need to figure out in order to *understand* a scene? While the answer largely depends on the scene and a task at hand, we often make a good decision only with a thin slice of information. In fact, a decision by a few glances is sometimes no worse and even better than hours of pondering [26]. Given a limited time and resource, this thin-slicing decision becomes crucial in particular. For example, drivers often need to rely on their immediate understanding of situations they pass by, otherwise running someone or themselves into danger.

This paper addresses learning a thin-slicing machine for human pose, relieving the need for explicit pose estimation. While a person in a specific pose is one of the most informative objects for scene understanding, the problem of pose estimation, aiming at localizing individual body joints, still remains a challenging open problem [13, 43, 50, 51, 61]. Even with a full body visible, frequent self-occlusion and huge diversity in pose make the problem hard to solve. As often formulated a complicated structured inference problem [13, 50], human pose estimation is computationally heavy in practice as well.

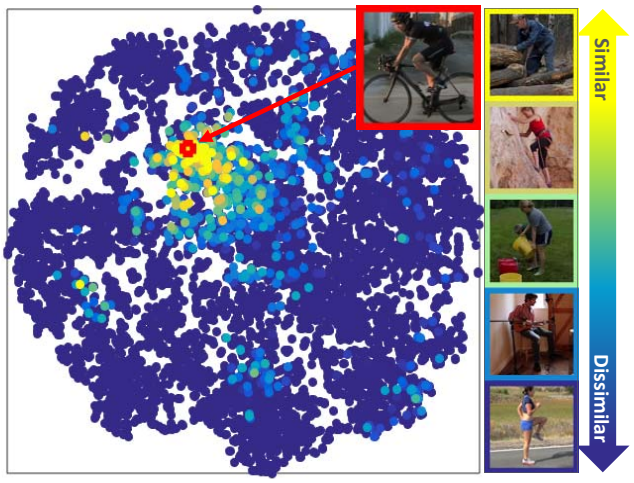


Figure 1: The manifold of our pose embedding visualized using t-SNE [53]. Each point represents a human pose image. To better show correlation between the pose embedding and annotated pose, we color-code pose similarities in annotation between an arbitrary target image (red box) and all the other images. Selected examples of color-coded images are illustrated in the right-hand side. Images similar with the target in annotated pose are colored in yellow, otherwise in blue. As can be seen, yellow images lie closer by the target in general, which shows that a position on the embedding space implicitly represents a human pose. (Best viewed in color.)

We argue that solving explicit pose estimation may not be necessary for many problems that only require an estimate of pose similarity. To bypass pose estimation, we learn an efficient pose embedding function based on a convolutional neural network (CNN) [18, 32] with a triplet rank loss [12, 44, 56, 57, 59]. Our pose embedding, trained on real data, provides a pose-aware, compact mapping that projects images with similar human poses in same neigh-

borhoods of the embedding space and far from each other otherwise. A position in the embedding space represents the corresponding pose in an implicit way, and helps in capturing pose-related semantics in the images as illustrated in Fig. 1. Even without the need for detailed labeling of pose similarity, the triplet rank loss combined with a convolutional neural network architecture enables us to learn a robust embedding space in an efficient manner.

The proposed pose embedding provides a practical alternative to explicit pose estimation in a variety of applications. Efficient pose retrieval can be performed in this embedding space to find images or video frames with similar poses in a large database. The embedding can be also used for a pose-based feature for action recognition [29, 30, 54, 60], group activities [15, 33, 34, 42], and human object interaction analysis [62]. In experiments, we transfer our learned model to the problems of pose-based image retrieval and action recognition, and demonstrate that the proposed embedding successfully generalizes to diverse poses in cluttered scenes.

2. Related work

Our approach to human pose embedding is related to a broad range of topics including pose estimation, pose-based action recognition, and similarity-based embedding. Here we briefly review representative work on those topics.

Pose estimation. Human pose understanding has been studied extensively during the last decade. Unlike our approach, most previous work has focused on localzing explicit body parts or joints. Many of them rely on pictorial structures of human body as a prior [5, 19, 23, 58, 61]. It is shown in [55] that the pictorial structure can be learned from data. Multi-modality of pose is captured at larger granularities and estimated jointly with body part locations in [43]. Following the great success in image classification [32, 45, 48], CNN has recently become popular in pose estimation. A pioneering work on this line of research was done by Toshev and Szegedy [51], where body joint coordinates are directly estimated by a regression machine based on CNN. Chen and Yuille [13] combine a graphical model with CNN, where the unary and pairwise clique potentials of the graphical model are learned by CNNs. Instead of an explicit graphical model, Tompson *et al*. [50] learn spatial dependencies among all pairs of body joints through CNNs.

Action recognition with pose. Naturally, pose-based features have shown to be effective for action recognition in images or videos. Pose is treated as a latent variable for action recognition in still images [60], and Bao and Fei-Fei [63] estimate semi-3D pose for the same purpose. For action recognition in video, estimated poses are often employed directly as descriptors [30, 54]. Cheron *et al*. [14] use pose as a location prior for CNN-based features being extracted around body parts.

Embedding by similarity. Learning embedding by pairwise similarity has been extensively studied. The siamese-type CNN architecture was proposed to learn an embedding space that reflects a semantic distance between data [9], and has been applied to face verification [16] and visual representation learning [3, 20]. For an embedding space that preserves relative similarity among triplets, Athisos *et al*. use AdaBoost with random embedding functions [6], and Chechik *et al*. propose a bi-linear similarity function with a triplet rank loss [12]. The triplet rank loss has been recently adopted as a loss function of CNNs for various tasks such as image ranking [56], face identification [44], visual representation learning [57], and joint object categorization and camera pose estimation [59].

Pose embedding. Low dimensional manifolds of articulated human pose have been employed for person tracking [37, 52] and 3D pose estimation from silhouette [2, 21]. Taylor *et al*. [49] learn CNNs with neighborhood component analysis criteria [27] to match images with similar poses and estimate pose by nearest neighbor regression. As a pose encoding, Pons-Moll *et al*. [39] propose posebits, which are binary attributes about geometric relations between body parts and used for 3D pose estimation and retrieval. There is a parallel work about pose embedding [25] motivated by a similar idea with ours. The most important difference of ours from [25] is that in testing our method does not require any pose information while [25] assumes that the pelvis location of a person is known. Also, we show how pose embedding can be adopted for action recognition in image, which is not investigated in [25].

3. Algorithm overview

Instead of explicit pose estimation, we learn a function that maps an image of a person into an embedding space. A position of the image on the embedding space represents a corresponding pose in an implicit way such that images will be placed nearby if their poses are similar, otherwise far from each other. We design the embedding function using a CNN, and train it with a triplet rank loss [12, 44, 56, 57, 59]. In this work, two separate embedding functions are learned and evaluated: (1) full-body pose embedding that takes a full body as input, and (2) upper-body pose embedding that considers only an upper part of a body as input. Both embedding functions share the same network architecture.

3.1. Network architecture

Details of the network architecture is described in Fig. 2. In particular, the structures and parameters of the five convolution layers (*i.e*., conv1–5) are transferred directly from those of the VGG-S network [11], which is pre-trained for ImageNet classification task [41]. It is demonstrated in [38] that representation learned for a large scale image classification could be also useful in other tasks as well. The

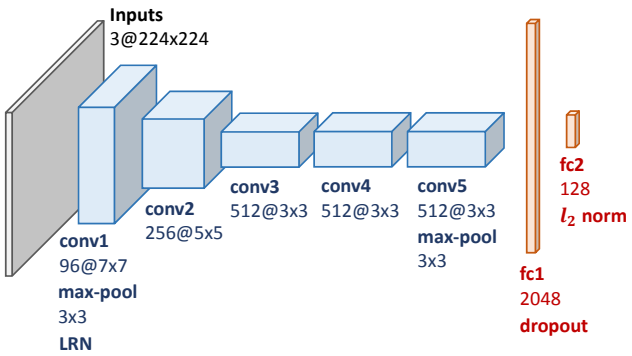


Figure 2: Our network architecture for pose embedding. The network consists of 5 convolution layers (conv) followed by 2 fully-connected layers (fc). In details, Local Response Normalization (LRN) [32] is applied after conv1, and the output of fc1 is regularized by Dropout [46]. All the convolution layers and related operations (*blue*) are transferred directly from those of the VGG-S network [11], and their parameters are fine-tuned during training.

parameters of the convolution layers are not fixed, but fine-tuned during training with a smaller learning rate. We believe that the transferred representation would speed up network learning and also result in better generalization as our training dataset is not large when compared to the number of parameters.

On the pre-trained part of the network, we add two fully-connected layers (*i.e.*, fc1–2), which are learned from scratch for adaptation to the pose embedding task. The embedding dimensionality is fixed to 128. Finally, an l_2 normalization layer is added on the top of the network so that the embedding vectors lie on a unit sphere.

3.2. Triplet rank loss

The triplet rank loss considers pose distances among a triplet of images. Let $\mathcal{X}$ be a set of human pose images and $\mathcal{Y}$ a set of pose annotations of images in $\mathcal{X}$. We define a dissimilarity between two images $x_i, x_j \in \mathcal{X}$ as a distance between their corresponding pose annotations $y_i, y_j \in \mathcal{Y}$, which is denoted by $D(y_i, y_j)$. Our goal is to learn an embedding function f such that Euclidean distances in the embedding space hold the same order relations with the pose distances:

$$\begin{aligned} &||f(x_i) - f(x_i^+)||_2^2 < ||f(x_i) - f(x_i^-)||_2^2, \\ &\forall x_i, x_i^+, x_i^- \in \mathcal{X} \quad \text{s.t. } D(y_i, y_i^+) < D(y_i, y_i^-). \end{aligned} \tag{1}$$

Based on this, our triplet rank loss L is defined as a hinge loss with a safety margin:

$$L = \sum_{i=1}^{N} \left[||f(x_i) - f(x_i^+)||_2^2 - ||f(x_i) - f(x_i^-)||_2^2 + \delta \right]_+, \tag{2}$$

where N is the number of triplets and δ indicates the margin. Given a set of triplets together with pairwise pose distances, the embedding function is learned by minimizing Eq. (2) through error back-propagation [36, 40] and stochastic gradient decent (SGD) with momentum [47]. We describe details for learning in the following sections.

4. Dataset compilation

4.1. Data sources

Learning a pose embedding space with the triplet loss requires images of human poses, and pose distances between them. To this end, we collect pose images with body joint annotations, and compute the pose distances using the annotated joint coordinates. Our dataset mainly built on the MPII Human Pose dataset [4] that contains around 25K images with diverse human poses from YouTube videos. The images have extensive key-point annotations for body joints as well as bounding box annotations for heads. Among them, we selected 19,919 images with full body joint annotations, and included them in our dataset. Note that invisible joints are also counted as valid ones if their positions are annotated, because they would be useful to learn an embedding function robust against partial occlusions. In addition, we collected more images from the H3D [8] and the VOC2009 person (trainval) [22] datasets with body joint annotations [7]. As most of human poses in these datasets are not fully annotated, we selected images that have at least 12 joint annotations.

In total, our dataset comprises 12,366 images for training (MPII: 10,000, H3D: 843, VOC2009 people: 1,523), and 9,919 images for validation (MPII: 9,919).

4.2. Image and pose standarization

In our CNN architecture, all images need to be standarized into a fixed-size input image (*i.e.*, 224×224×3) that captures a target human body. The best way of obtaining such input would be to align images using a stable center of pose (*e.g.*, pelvis), resize them with respect to a human body scale, and crop the target human body. However, this is problematic in testing because such a precise input is hard to be obtained without explicit and accurate pose estimation. Instead, we assume that the target person is localized in the form of a bounding box by a person detector in testing. To approximate this condition during training, we standarize pose images using bounding boxes of target persons, as described in Fig. 3. For full-body embedding, we find a bounding box that most tightly encloses all annotated joints, and crop the smallest square box that covers the bounding box while sharing the center position. For upper-body embedding, the cropped area is horizontally aligned using the center of the annotated head, and rescaled so that its size is

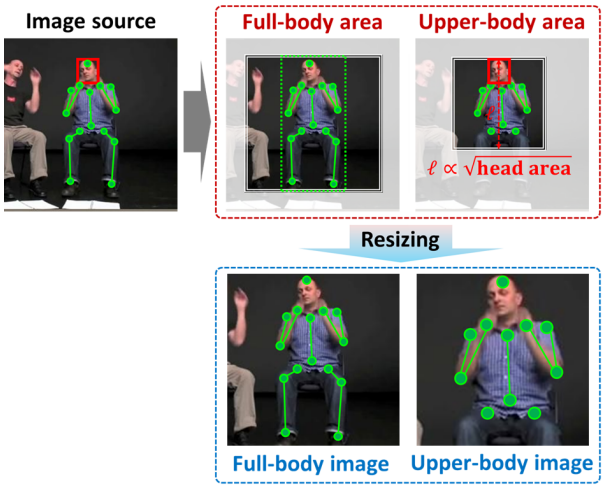


Figure 3: Visualization of the standarization procedure.

Figure 4: Examples of anchor images (*left*), and their positive (*middle*) and negative images (*right*).

proportional to that of the head bounding box[1]. We disregard annotations of ankles and knees for upper-body data. All the cropped images and their annotated joint coordinates are resized to the input size of our embedding network. The pose distance between two images is defined as the mean of Euclidean distances between standarized coordinates of corresponding joints. When the two poses do not have the same number of joint annotations, we compute the distance using only shared joints.

5. Learning pose embedding network

5.1. Triplet sampling

Instead of using all possible triplets in training, which is impractical due to the combinatorial explosion, we sample a set of triplets that would be useful to learn the embedding function. In particular, given an anchor image x_i, we divide the other images into two sets of positive (similar) images $\mathcal{P}_i = \{x_i^+\}$ and negative (dissimilar) images $\mathcal{N}_i = \{x_i^-\}$, respectively, and assemble a triplet by choosing one from each of $\mathcal{P}_i$ and $\mathcal{N}_i$. Since in our dataset there is no given positive/negative distinction a prior, one straightforward way to do this would be to divide images by thresholding their pose distances from the anchor. However, due to the imbalanced distribution of poses in the dataset, some common poses will have a lot of positive images while others could not have any positive. Furthermore, the pose distance metric may be inconsistent over anchor images since we do not know exact body scales nor 3D positions of body joints. Hence, for each anchor image x_i, we consider its p nearest neighbors in pose distance as $\mathcal{P}_i$, and other images as $\mathcal{N}_i$. Examples of positive and negative images are illustrated in Fig. 4.

Although the above sampling strategy reduces the number of combinations significantly, still there exist numerous triplets per anchor and some of them would be trivial and ineffective for learning embedding network. Hard negative mining has been commonly used to avoid such trivial triplets and boost learning procedure, especially when training images are annotated by categorical labels [44, 57, 59]. In our case, however, it is not straightforward to sample hard negatives from $\mathcal{N}_i$ since human poses are continuous and changes smoothly between $\mathcal{P}_i$ and $\mathcal{N}_i$. In other words, hard negative images are often too close to positive images in pose distance, so it is not desirable to separate such negatives from positives in the embedding space. Instead of hard negative mining, we reduce the size of $\mathcal{N}_i$ every epoch by removing negative images farthest from the anchor, while assuming that the farthest negative images would be trivial in general. This assumption is not necessarily true, but we found that it holds in most cases. We believe that our approach could allow the embedding networks to focus more on subtle differences between poses in later stages of training. Also, it practically improved performance when the learned embedding was applied to action recognition.

5.2. Learning network with random triplets

Since many triplets share identical images, it is inefficient to calculate embedding vectors and gradients for each triplet individually. For learning network efficiently, we first assemble a mini-batch with an anchor image and randomly selected positive and negative images, and generate triplets by all combinations of positive and negative images in the

[1]Since this procedure entails head box annotations, 2366 images from the H3D dataset and VOC2009 person datasets, which do not provide head annotation, are not used for learning upper-body embedding.

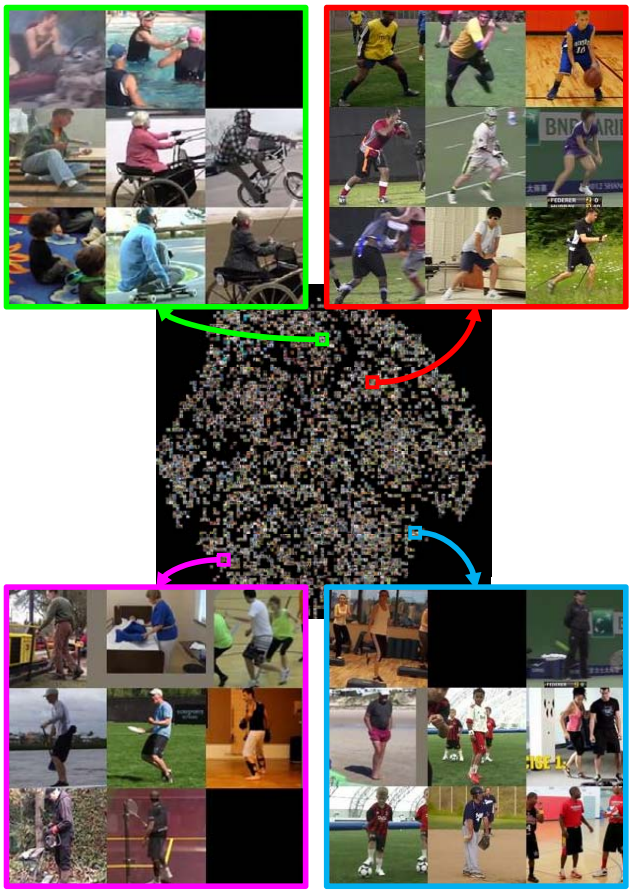


Figure 5: The MPII validation images on the manifold of our full-body pose embedding. The 2D manifold is estimated by t-SNE [53], and the pose images are deployed by [31]. The zoomed parts of the manifold show that images depicting similar poses are located in similar positions as desired, even though they are not used when learning the embedding function. More visualization results can be found in our project webpage [1].

batch. By doing this, individual images in the batch are embedded (forward-propagated) once. In this approach, multiple loss-gradients are computed for each embedding vector since an image participates in multiple triplets, so we accumulate them per embedding vector and perform back-propagation once.

6. Experiments

In this section, we first describe implementation details, and then validate our concept empirically by qualitative analysis of our pose embedding manifold. We also demonstrate the effectiveness of our approach in three tasks: 1) pose retrieval on the MPII dataset [4], 2) action recognition on the VOC2012 Action dataset [22], and 3) action recognition on the People Playing Musical Instrument (PPMI) dataset [62].

6.1. Implementation details

The number of positive images p per anchor is fixed to 30 for full-body pose embedding, and 15 for upper-body counterpart. The size of a mini-batch is set to 128, which consists of an anchor image, 5 randomly selected positives, and 122 randomly selected negatives. The size of $\mathcal{N}_i$ is decreased by 3K every epoch until it becomes 1K. During training, images in the batch are translated and scaled individually up to $\pm 10\%$ of the input size, and the entire batch is horizontally flipped with probability 0.5. The learning rate for the fully-connected layers is initialized as 0.01, and multiplied by 0.2 every epoch. The learning rates for the pre-trained part (*i.e.*, conv1–5) are 10 times smaller than that for the fully-connected layers. The momentum and weight decay parameters are set to 0.9 and 0.0005, respectively. Our system is implemented in Torch7 [17]. We run 60K SGD iterations for learning, which took about three days on a single Nvidia TITAN Black with 6GB RAM in our experiment. Code and trained networks will be released online at [1].

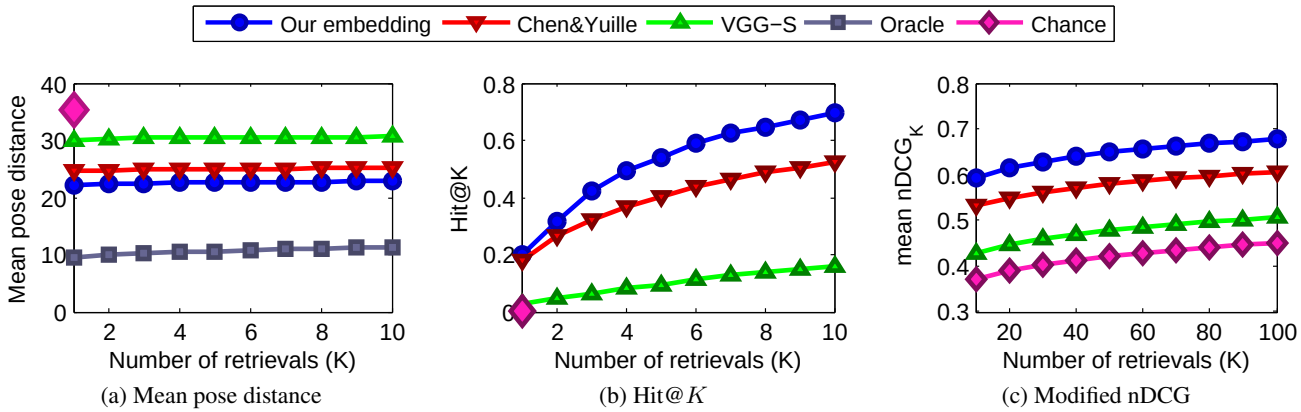
6.2. Manifold visualization

We validate our pose embedding qualitatively by visualizing its low-dimensional manifold of the standarized MPII images. We first illustrate how images depicting similar human poses are distributed on the manifold in Fig. 1. As expected, images resembling each other in pose are also close to each other on the manifold in general. This means that a position in the embedding space approximates a human pose. The manifold is also visualized by deploying the images directly on it, and the result is shown in Fig. 5.

6.3. Pose retrieval

We evaluate our full-body pose embedding network on a pose retrieval task using the MPII validation set. Among 9,919 standarized images in the validation set, the first 1,919 images are used as queries, and the remained 8K images are regarded as a test set on which retrieval is performed. We compare our full-body pose embedding with VGG-S [11], which is the base network of our embedding function, and Chen and Yuille's method [13] that estimates human pose explicitly using CNN. For VGG-S, 4,096-D output of its penultimate fully-connected layer is employed as a descriptor. For Chen and Yuille's, 26 part-locations obtained by its explicit pose estimation is directly used. Note that for Chen and Yuille's, the approximate scale of person is assumed to be known in order to boost both of its speed and accuracy.

Three performance metrics are defined to evaluate retrieval results quantitatively. We first compute the mean of pose distances between queries and their K nearest neigh-



(a) Mean pose distance (b) Hit@K (c) Modified nDCG

Figure 6: Quantitative evaluation of pose retrieval on the MPII validation set. Oracle is always 1 in (b) and (c).

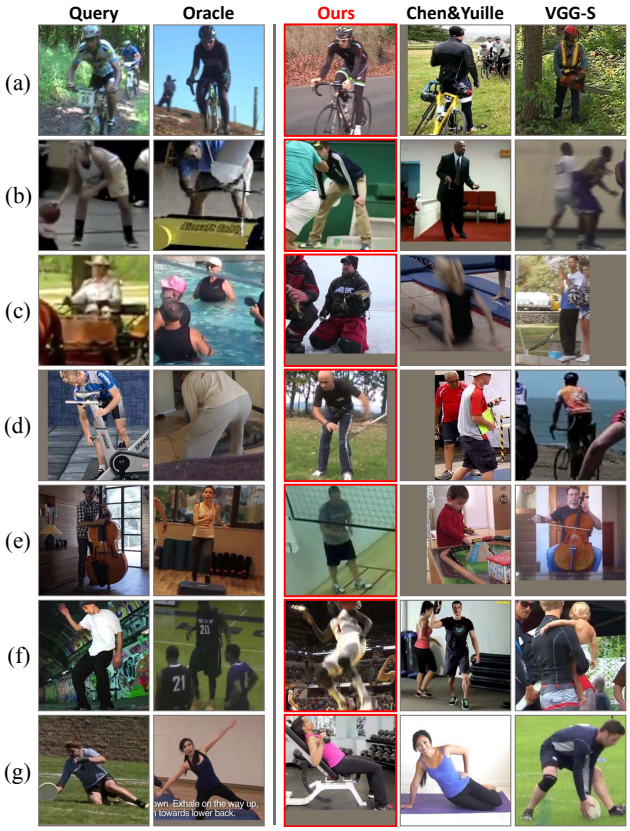


Figure 7: Qualitative examples of the 1st nearest pose.

bors. Note that the pose distance is the mean of Euclidean distances between standarized joint coordinates, defined in Section 4.2. We also measure retrieval performance by Hit@K rate, which counts how many queries have at least one correct image among their K nearest neighbors. We consider a retrieved image as correct when it belongs to the 50 nearest neighbors of the query in the pose distance. Finally, we design a modified version of normalized discounted cumulative gain (nDCG) [10]:

$$\text{nDCG}_K(q) = \frac{1}{Z_K} \sum_{i=1}^{K} \frac{2^{r_i}}{\log_2(i+1)}, \tag{3}$$

where $r_i = -\log_2(||y_q - y_i||_2 + 1)$ is the relevance between the query q and i^{th} retrieval in terms of their true poses y_q and y_i. The relevance is reduced by the discounting factor $1/\log_2(i+1)$ to place a greater emphasis on one returned at a higher rank. nDCG_K considers only top K retrievals, and Z_K is for normalization so that the maximum nDCG_K becomes 1. A higher nDCG means a better retrieval. Note that the modified nDCG evaluates ranking quality while the Hit@K measures classification accuracy.

The quantitative evaluation results are summarized in Fig. 6. Our full-body pose embedding outperforms both of VGG-S [11] and the pose estimation [13] in terms of all the three metrics. Also, our approach was three orders of magnitude faster than the pose estimation on the same hardware. Selected retrieval examples are visualized in Fig. 7. The results show that VGG-S tends to focus more on object category (cello in Fig. 7(e)) or holistic characteristics of images (Fig. 7(a,b,g)) than pose. Although a subset of its weights is initialized by VGG-S, our network successfully captures human poses out of diverse background contexts. Our embedding often makes reasonable results even if human bodies are partially occluded (Fig. 7(c,e)), and sometimes its results look better than those by oracle (Fig. 7(d,f)). Our approach could fail when query pose is rarely observed from the training data (Fig. 7(g)), and the explicit pose estimation performs better than ours in that case. However, it often fails to distinguish front- and back-facing poses (Fig. 7(a)) and is inaccurate when query pose is occluded (Fig. 7(c)).

6.4. VOC2012 Action recognition

The VOC2012 Action dataset [22] consists of 10 action classes: jumping, phoning, reading, playing instrument,

	jump	phone	instr.	read	bike	horse	run	photo	comp.	walk	Mean
PosE(full)	57.2	24.4	41.6	22.4	71.1	69.4	77.0	30.7	34.8	60.8	48.9
PosE(upper)	38.7	26.1	39.7	28.4	66.2	62.1	65.6	23.3	33.5	47.0	43.1
PosE(full,upper)	57.5	30.6	48.5	32.5	76.8	73.8	78.7	33.9	42.9	60.9	53.6
VGG(img)	78.8	56.7	91.2	62.7	87.6	93.0	78.6	56.5	88.5	44.7	73.8
VGG(img) + PosE(full,upper)	83.3	56.9	91.8	64.4	92.9	96.8	87.7	61.9	88.9	67.2	79.2
VGG(img,box)	82.4	64.8	93.1	70.0	93.7	97.4	84.4	66.7	90.4	61.2	80.4
VGG(img,box) + PosE(full,upper)	85.0	64.8	93.5	70.2	94.9	98.0	88.8	69.6	90.5	69.5	82.5

Table 1: APs (%) on the trainval set of VOC2012 Action.

	jump	phone	instr.	read	bike	horse	run	photo	comp.	walk	Mean
VGG(img)	84.4	61.0	93.2	63.1	92.4	93.8	83.9	61.4	85.2	51.3	77.0
VGG(img) + PosE(full,upper)	86.3	61.2	92.9	63.5	96.0	97.4	90.5	65.9	85.9	73.5	81.3
VGG(img,box)	88.3	68.0	94.7	68.1	96.8	97.2	89.1	70.2	86.9	63.3	82.2
VGG(img,box) + PosE(full,upper)	88.5	68.0	94.5	67.8	97.6	98.1	92.5	73.0	87.0	74.6	84.2

Table 2: APs (%) on the test set of VOC2012 Action.

riding bike, riding horse, running, taking photograph, using computer, and walking. The dataset also contains images of others class that does not belong to any of the above classes. The number of human subjects is 6,278 in the trainval set and 6,283 in the test set. The bounding boxes of people performing actions are provided in both training and testing, and we use the boxes to standarize input images of our embedding network as described in Section 4.2.

We first evaluate our pose embedding on the trainval set to analyze its characteristics. We extract two pose embedding vectors per image: one for full-body pose, denoted by PosE(full), and the other for upper-body pose, denoted by PosE(upper). For classification, SVMs with RBF kernels are trained with the embedding vectors. We compute two separate RBF kernels for PosE(full) and PosE(upper), and also compute a combined one, PosE(full,upper), that simply combines the two kernels by summation. The SVM parameter and kernel bandwidths are chosen by cross-validation. The top part of Table 1 (1st-3rd rows) summarizes average precision (AP) scores of the classification results on the trainval set. It shows that the combination PosE(full,upper) improves over either PosE(full) or PosE(upper) alone.

Our pose embeddings, however, are not strong enough to capture all action features beyond pose, *e.g.*, people often perform different actions with similar poses as illustrated in Fig. 8. Such an ambiguity issue can be handled by exploiting context information in nearby objects and background. To this end, we combine our pose embedding vectors with features from ImageNet pre-trained CNNs, VGG-16 and VGG-19 [45], as follows. We apply VGG-16 to an original image (not standarized) with multiple scales in a fully convolutional manner, and obtain a 4,096-D activation vector, which is in turn l_2 normalized. In a similar manner, we extract a 4,096-D vector through VGG-19. These two vectors are concatenated to obtain an 8,192-D feature vector per image. We denote it by VGG(img). The same procedure is done for the bounding box region so that another 8,196-D feature vector is calculated. We call it VGG(box). VGG(img,box) denotes the concatenation of VGG(img) and VGG(box), which is a 16,384-D feature vector. We refer

Figure 8: Nearest neighbors of VOC Action (trainval) images in the full-body pose embedding space. If the nearest neighbors are from the same action class with the query, they are colored in blue, otherwise in red. Even when our pose embedding captures pose configurations well, the action classes of the nearest neighbors are frequently incorrect since semantically different actions can be performed from similar poses.

	mAP
Oquab *et al*. [38]	70.2
Hoai [28]	76.3
Simonyan and Zisserman [45]	84.0
VGG(img,box)	82.2
VGG(img,box) + PosE(full,upper)	**84.2**

Table 3: Comparison with state of the arts in VOC Action.

readers to [45] for more details about feature extraction through VGG-16 and VGG-19.

For classification, we apply a linear kernel for the VGG features, and combine it with those of PosE by simple summation. The AP scores of the combinations of VGG and PosE is reported in the bottom part (4th-7th rows) of Table 1. Note that VGG(img,box) in the table means our reproduction of [45]. As can be seen from the result, VGGs are improved by combination with PosE(full,upper), especially for the classes like jumping, running, and walking, where people do not interact with nearby objects thus pose is more discriminative than context. It is also worth noting that VGG(img) + PosE(full,upper) performs on par with VGG(img,box) although the representation power of PosEs is limited by the smaller network size and dimensionality compared to VGG(box). This empirically confirms that our embedding is complementary to VGGs and useful for action recognition.

Table 2 reports APs of the combinations of VGGs and PosEs on the test set. The result shows that PosEs enhances the performance of walking class with a large margin, and also substantially improves the performance in running and taking photo classes. Our best performing model is also compared with state of the arts in VOC Action in Table 3, where VGG(img,box) + PosE(full,upper) outperforms other methods with a small margin. Note that VGG(img,box), our reproduction of [45], does not achieve the same performance reported in [45] probably due to implementation issues. We believe that our best model, that is VGG(img,box) + PosE(full,upper), will perform better when combined with the original VGG(img,box) features.

6.5. PPMI action recognition

The PPMI dataset [62] contains images of people holding 7 musical instruments: bassoon, erhu, flute, French horn, guitar, saxophone, and violin. In this experiment, we consider a task of classifying whether a person is playing or not playing. Performing this task may involve recognizing a pose of persons holding the target instrument.

For each instrument and each action category, 100 training images and 100 test images are provided. Since all images in the datasets exhibit only upper-bodies, we apply the upper-body pose embedding only. Although the images are already standarized by an upper-body detector, scales and positions of people in the images vary substantially. For each image, we thus crop multiple patches with different scales from the center of the image, and aggregate embedding vectors of the patches by max-pooling per dimension. We train linear SVMs with the aggregated pose embedding vectors, where the SVM hyper-parameter was estimated by cross-validation on the training dataset. The classification result is quantified and compared in Table 4.

	[35]	[24]	[62]	PosE	VGG	V+P
bassoon	71.5	68.5	78.0	82.0	74.5	83.5
erhu	78.0	75.5	78.5	81.0	88.0	87.0
flute	84.5	79.0	90.5	96.5	98.5	99.0
French horn	78.5	75.5	80.5	85.0	91.0	95.0
guitar	79.5	81.0	75.5	87.5	88.5	89.0
saxophone	76.0	76.5	78.5	82.0	85.0	88.0
violin	78.5	75.5	85.0	86.0	94.0	94.0
Average	78.1	75.9	80.9	85.7	88.5	**90.8**

Table 4: Classification accuracy on the PPMI dataset.

Our approach, denoted by PosE, clearly outperforms the previous results on this task [24, 35, 62] with a substantial margin even without explicitly considering instruments. Our method is also compared with a linear SVM on top of 8,192-D feature vector from VGG-16 and VGG-19, that is denoted by VGG. The feature vector is computed by the same procedure as VGG(img) of Section 6.4. VGG demonstrates a strong representation power on this task, outperforming all the others. By concatenating our pose embedding vector with VGG feature, we achieve better classification accuracies. Along with the previous experiments, this demonstrates again that our pose embedding provides a useful feature for pose-related tasks.

7. Conclusion

We have presented a deep embedding network for human pose, which leverages available pose annotations and learns a compact mapping from images to a pose-aware space. We have empirically shown that our pose embedding is useful for pose retrieval and action recognition, and that it further improves action recognition performance when combined with different types of features. We believe this thin-slicing type of approach to pose understanding can be an effective alternative for many challenging pose-related applications, in particular, that require cost-efficient and implicit pose information. In future, we will investigate action recognition in videos, spatio-temporal pose embedding, and more effective embedding with other types of loss functions.

Acknowledgments. We thank Greg Mori and Vadim Kantorov for fruitful discussions. This work was supported in part by Google Research Award and the ERC grants Activia and VideoWorld.

References

[1] http://www.di.ens.fr/willow/research/thinslice/. 5

[2] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *CVPR*, 2004. 2

[3] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, 2015. 2

[4] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 3, 5

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

[6] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *CVPR*, 2004. 2

[7] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. 3

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

[9] J. Bromley, I. Guyon, Y. Lecun, E. Sckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In *NIPS*, 1994. 2

[10] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005. 6

[11] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 2, 3, 5, 6

[12] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010. 1, 2

[13] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014. 1, 2, 5, 6

[14] G. Chéron, I. Laptev, and C. Schmid. P-CNN: Pose-based CNN Features for Action Recognition. In *ICCV*, 2015. 2

[15] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, 2011. 2

[16] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 2

[17] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 5

[18] Y. L. Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1990. 1

[19] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *CVPR*, 2013. 2

[20] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

[21] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, 2004. 2

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 3, 5, 6

[23] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 2

[24] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 8

[25] G. Mori et al. Pose embeddings: A deep architecture for learning to match human poses. *arXiv:1507.00302*, 2015. 2

[26] M. Gladwell. *Blink :the power of thinking without thinking*. New York : Little, Brown and Co., 2005. 1

[27] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004. 2

[28] M. Hoai. Regularized max pooling for image categorization. In *BMVC*, 2014. 8

[29] N. Ikizler-Cinbis, R. Cinbis, and S. Sclaroff. Learning actions from the web. In *CVPR*, 2009. 2

[30] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black. Towards understanding action recognition. In *ICCV*, 2013. 2

[31] A. Karpathy. t-SNE visualization of CNN codes. http://cs.stanford.edu/people/karpathy/cnnembed/. 5

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 1, 2, 3

[33] S. Kwak, B. Han, and J. H. Han. Multi-agent event detection: Localization and role assignment. In *CVPR*, 2013. 2

[34] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, 2010. 2

[35] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 8

[36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 3

[37] C.-S. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *ICCV*, 2007. 2

[38] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 2, 8

[39] G. Pons-Moll, D. J. Fleet, and B. Rosenhahn. Posebits for monocular human pose estimation. In *CVPR*, 2014. 2

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(Oct), 1986. 3

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, April 2015. 2

[42] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *CVPR*, 2009. 2

[43] B. Sapp and B. Taskar. MODEC: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013. 1, 2

[44] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 2, 4

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 7, 8

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 3

[47] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 3

[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[49] G. W. Taylor, R. Fergus, G. Williams, I. Spiro, and C. Bregler. Pose-sensitive embedding by nonlinear nca regression. In *NIPS*, 2010. 2

[50] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 1, 2

[51] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 1, 2

[52] R. Urtasun, D. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *CVPR*, 2006. 2

[53] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9: 25792605, Nov 2008. 1, 5

[54] C. Wang, Y. Wang, and A. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013. 2

[55] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *CVPR*, 2013. 2

[56] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 1, 2

[57] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 1, 2, 4

[58] Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *ECCV*, 2008. 2

[59] P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *CVPR*, 2015. 1, 2, 4

[60] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010. 2

[61] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 1, 2

[62] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010. 2, 5, 8

[63] B. Yao and L. Fei-Fei. Action recognition with exemplar based 2.5d graph matching. In *ECCV*, 2012. 2