Learning Invariance through Imitation

Graham W. Taylor, Ian Spiro, Christoph Bregler and Rob Fergus
Dept. of Computer Science, Courant Institute of Mathematical Sciences, New York University
{gwtaylor,spiro,bregler,fergus}@cs.nyu.edu

Abstract

Supervised methods for learning an embedding aim to map high-dimensional images to a space in which perceptually similar observations have high measurable similarity. Most approaches rely on binary similarity, typically defined by class membership where labels are expensive to obtain and/or difficult to define. In this paper we propose crowdsourcing similar images by soliciting human imitations. We exploit temporal coherence in video to generate additional pairwise graded similarities between the user-contributed imitations. We introduce two methods for learning nonlinear, invariant mappings that exploit graded similarities. We learn a model that is highly effective at matching people in similar pose. It exhibits remarkable invariance to identity, clothing, background, lighting, shift and scale.

1. Introduction

Effective systems for visual reasoning must cope with input variability caused by clutter, occlusion, viewpoint and illumination changes and intra-class differences. In response, a variety of approaches have attempted to learn image representations that are invariant to photometric and geometric distortions. Methods like Boost-SSC/PSH [22], variants of Neighborhood Components Analysis (NCA) [7, 21, 28] and Dimensionality Reduction by Learning an Invariant Mapping (DrLIM) [9] use supervised learning to map high-dimensional images to a low-dimensional space in which nearest neighbors are easily computable, and observations that are perceptually similar have high measurable similarity. However, these methods employ a binary notion of pairwise similarity, either through predefined classes or by thresholding real-valued labels. Such labelings are expensive to obtain, often difficult to define and cannot represent graded similarity which may benefit learning.

In this paper we propose a new paradigm for learning invariant mappings: imitation. Consider the problem of learning an embedding in which people in similar pose lie close-by. The first step in this task is to obtain many images of people in similar pose but with different clothing, backgrounds, lighting and other appearance changes. Obtaining this data is time-consuming. Moreover, judging the degree of similarity between observations is non-trivial and inconsistent across observers. Other works (*e.g.* [22, 1, 8]) have used synthetic renderings to a modest degree of success, but we believe there is a better source of real data that exhibits the same amount of variability a model would observe at test time. Given an image of a person in pose, people have a profound ability to mimic its content. Therefore, we can exploit the abundance of webcams to quickly crowd-source a massive dataset of people in similar pose by asking people on the web to imitate images (Fig. 1).

A question arises as to how one should select the seed images which users are asked to imitate. A key aspect of our approach is the use of temporal coherence in video which

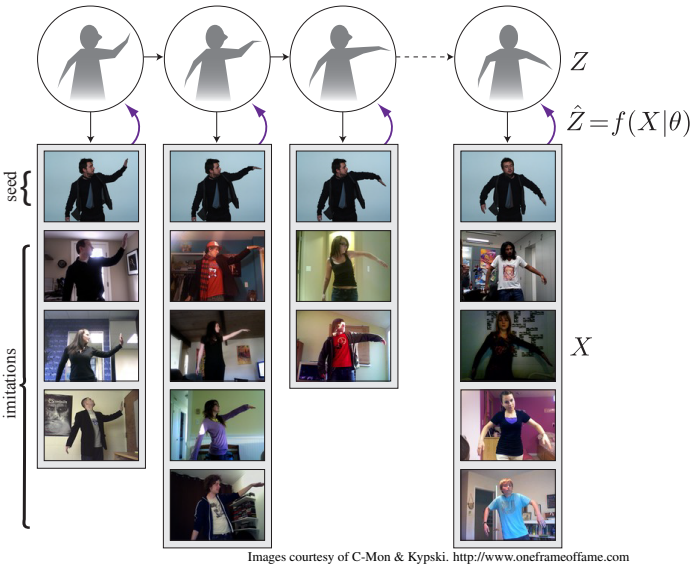


Figure 1: Schematic of our approach. We assume for each frame of video, there exists an unobserved low-dimensional representation of pose, Z. A seed image is generated by mapping from pose space to pixels, X, through an unobserved interpretation function. Our method learns a nonlinear embedding, $f(X|\theta)$ which approximates Z with a low-dimensional vector. In the example above, users are asked to imitate seed images taken from a music video [5].

greatly increases the number of similar examples. Temporal coherence has been used to learn invariant features [2, 15] but in a very different context. These methods directly learn from frames of video while we use video only as a source of seed images presented to users. Our model learns only from user-contributed imitations, many of which could correspond to a single frame of the original video. We use scene cuts and correspondence by frame number to determine the graded similarity of the imitations.

Embedding algorithms that rely on binary pairwise similarities are unable to cope with the graded, or "soft" similarities obtained by imitation. This paper contributes two novel learning algorithms that exploit soft similarities. Furthermore, we propose a simple model for transforming the type of discrete distances obtained through temporal coherence into more perceptually coherent distances.

2. Related Work

In recent years, Amazon Mechanical Turk and other crowd-sourcing platforms have emerged as a way of accelerating vision and other tasks [24], often for rapid labeling of massive image datasets [20, 25, 6, 19, 3, 26]. Most of these techniques ask the participants to explicitly provide desired segmentations, feature points, configurations, pose, or class labels. For continuous domains, especially for similarity measures, people have more difficulty supplying consistent labels. The process is also very taxing to the participant. To the best of our knowledge, this is the first attempt at using "imitation-based" crowd-sourcing, and a learning framework that is tailored to this type of data collection.

Our learning framework has some overlap with other methods mainly in two areas: dynamical models and metric learning. As outlined in Fig. 1, we can describe our data collection process as a chain of unobserved variables Z that encode, for example, image invariant body-pose, and observed imitations X. Such a structure could be represented probabilistically with dynamical models like LDS, HMMs, CRFs, and variants thereof (*e.g.* [23, 18, 29, 27] to name a few). But those models do not explicitly impose the constraint that linked hidden states must lie close-by in state-space. Also the mapping from hidden state to observed data (or vice-versa) is usually represented by linear or mixture models. While we experiment with linear variants of our model, we advocate for nonlinear, multilayered feature extractors and employ a discriminative estimation framework.

Our approach is related to supervised methods for dimensionality reduction like NCA [7], its nonlinear variants [21, 28] and DrLIM [9]. These methods implicitly learn a distance metric by learning a mapping from high-dimensional (*i.e.* pixel) space to low-dimensional feature space such that perceptually similar observations are mapped to nearby points on a manifold. However, these methods are restricted to binary pairwise similarities and, in the case of NCA, optimized for nearest neighbor classification. This is unsuitable for continuous notions like pose.

At least two works from the vision literature have attempted non-parametric pose estimation by learning pose-sensitive hash functions. Shakhnarovich *et al.* [22] learn a hash-function by boosting. Jain *et al.* [12] simultaneously learn a Mahalanobis metric sensitive to pose and encode this information into randomized hash functions. Both works use an explicit notion of neighbours and non-neighbours; this requires them to define a similarity threshold based on pose distance. We attempt to integrate the notion of "soft similarity" into metric learning. In addition, both of these works only consider synthetic renderings of humans on static backgrounds while we consider real images.

3. Methods

In this section we present two methods for learning an invariant mapping that can exploit the graded similarities obtained through imitation. The first is a probabilistic approach that is suitable for batch learning. However, it requires normalization by a term involving all the points in the training set, making it unsuitable for very large datasets. Therefore we also present an online, energy-based method.

We first introduce notation. We are given M sets of user-contributed imitations, each of which contains N_m images[1] $\{X_1^1, \ldots, X_{N_1}^1\}, \ldots, \{X_1^M, \ldots, X_{N_M}^M\}$. We construct sets based on our data collection efforts. For example, if the seed images used to solicit imitations are from different videos (or different scenes within the same video), we divide the respective imitations into sets that respect these boundaries. Each image X_i^m has an integer label y_i^m indicating its seed.

We assume that sets are independent of one another, but points within a set have a degree of similarity based on their seed. We seek to learn a functional mapping $Z_i^m = f(X_i^m|\theta)$, parameterized by θ, such that if points X_i^m and X_j^m come from nearby seed images (*i.e.* $|y_i^m - y_j^m|$ is small), then Z_i^m and Z_j^m will lie close-by in the output space. The mapping can be linear, $f(X_i^m) = AX_i^m$, or it may be nonlinear and more complex. We discuss the specific form of mapping in §3.4.

To measure similarity in the output space, we use a Euclidean distance metric, $d_{ij}^{mn} = ||Z_i^m - Z_j^n||_2$. To simplify notation, we use a single superscript, *e.g.* $d_{ij}^m = d_{ij}^{mm}$ when points i and j are in the same set as well as drop the superscript on labels within a set: $|y_i^m - y_j^m| = |y_i - y_j|$.

3.1. Probabilistic approach

We adopt the same stochastic notion of neighbors employed by SNE [11] and NCA [7]. Each training point, X_i^m, selects another point, X_j^n, as its neighbor with probability

[1] Our method is not restricted to images. It naturally extends to video imitation provided a relationship is established among the seeds.

$$p_{ij}^{mn} = \exp\left(-\left(d_{ij}^{mn}\right)^2\right) \Big/ \sum_{l,k} \exp\left(-\left(d_{ik}^{ml}\right)^2\right). \quad (1)$$

Note that $p_{ii}^{mm} = 0$, *i.e.* a point cannot be neighbors with itself and points in different sets may have nonzero probability of being neighbors. NCA minimizes nearest neighbor classification error. But in our setting, we are not concerned with classification. Instead we define a target distribution,

$$q_{ij}^{m} = \exp\left(-\left(\hat{d}_{ij}^{m}\right)^2\right) \Big/ \sum_{k} \exp\left(-\left(\hat{d}_{ik}^{m}\right)^2\right) \quad (2)$$

where for all points not in the same set, $q_{ij}^{mn} = 0$. The target distances $\hat{d}_{ij}^{m}$ are a function of y_i^m and y_j^m, and should reflect the perceptual similarity of X_i^m and X_j^m. We postpone discussing their specific form until §3.3.

Our objective is to minimize the Kullback-Leibler (KL) divergence of the discrete distribution over neighbors induced by the embedding, p, and the target distribution due to seed identity, q, for every datapoint:

$$D_{KL}(p,q) = \sum_{i,m} \sum_{j,n} q_{ij}^{mn} \log \frac{q_{ij}^{mn}}{p_{ij}^{mn}}. \quad (3)$$

The parameters of the mapping $f(\cdot)$ are found by minimizing $D_{KL}(p,q)$ for every datapoint with respect to θ:

$$\frac{\partial D_{KL}}{\partial \theta} = \sum_{i,m} \frac{\partial D_{KL}}{\partial Z_i^m} \frac{\partial Z_i^m}{\partial \theta} \quad (4)$$

where $\frac{\partial Z_i^m}{\partial \theta}$ depends on the form of $f(X_i^m|\theta)$ (see §3.4). The gradient $D_{KL}(p,q)$ with respect to an output, Z_i^m, can be written as

$$\frac{\partial D_{KL}}{\partial Z_i^m} = -2 \sum_{j,n} (Z_i^m - Z_j^n)\left[q_{ij}^{mn} - p_{ij}^{mn} + q_{ji}^{nm} - p_{ji}^{nm}\right]. \quad (5)$$

We can apply the chain rule once for the case of linear $f(\cdot)$ or use backpropagation for multi-layered $f(\cdot)$.

3.2. Energy-based approach

Although normalization prevents the embedding from collapsing to a single point, the probabilistic method is not suitable for the online setting. We seek an approach that acts only on pairs of images, making updates without having to consider all points. Moreover, we may want to target the selection of pairs (*e.g.* balance the similar and dissimilar examples when the dataset is dominated by one or the other). DrLIM [9] is an energy-based method that is trained by stochastic gradient descent. It can be used with arbitrary nonlinear f. Unfortunately it relies on a binary notion of similarity which does not suit our data. We propose a different loss function which uses a *soft* notion of similarity:

$$L = s_{ij}^{mn} L_S\left(X_i^m, X_j^n\right) + \delta\left(s_{ij}^{mn}, 0\right) L_D\left(X_i^m, X_j^n\right) \quad (6)$$

where s_{ij}^{mn} is a measure of similarity, $s \gg 0$ for points that have similar seeds, and $\delta(\cdot)$ is the Dirac delta function. We set $s_{ij}^{mn} = 0$ for $m \neq n$, *i.e.* points in different sets. We discuss s_{ij}^{mn} further in §3.3. Our loss has the effect of "pushing together" similar points with a force equal to their similarity through the similarity loss:

$$L_S(X_i^m, X_j^m) = \frac{1}{2}\left(d_{ij}^{m}\right)^2. \quad (7)$$

However, to keep the embedding from collapsing, L has a contrastive component whereby dissimilar points are "pulled apart" through the dissimilarity loss:

$$L_D(X_i^m, X_j^n) = \frac{1}{2}\left[\max\left(0, \alpha - d_{ij}^{mn}\right)\right]^2, \quad (8)$$

where $\alpha > 0$ is a margin which ensures that dissimilar points contribute to the loss only if they lie close-by in the embedded space. Following [9], we fix $\alpha = 1.25$ for all of our experiments. The model is trained by stochastic gradient descent. The gradient of the loss is given by:

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial Z_i^m} \frac{\partial Z_i^m}{\partial \theta} + \frac{\partial L}{\partial Z_j^n} \frac{\partial Z_j^n}{\partial \theta} \quad (9)$$

$$\frac{\partial L}{\partial Z_i^m} = \begin{cases} s_{ij}^m \left(Z_i^m - Z_j^m\right) & \text{if } s_{ij}^m > 0 \\ \left(d_{ij}^{mn} - \alpha\right) \frac{Z_i^m - Z_j^n}{d_{ij}^{mn}} & \text{if } s_{ij}^{mn} = 0, d_{ij}^{mn} < \alpha \\ 0 & \text{if } s_{ij}^{mn} = 0, d_{ij}^{mn} \geq \alpha \end{cases} \quad (10)$$

Note that $\frac{\partial L}{\partial Z_j^n}$ is obtained from Eq. 10 by substituting i for j and m for n.

3.3. From discrete labels to similarity

The probabilistic approach requires a mapping from discrete seed identity, y, to a real-valued target distance, $\hat{d}$. A simple heuristic is to set $\hat{d}_{ij}^m = |y_i - y_j|$ for images in the same set. However, this mapping does not reflect the underlying dimensionality of the data nor the distribution of seed images. We thus consider a simple generative model that treats the seed images as a random walk on some underlying manifold of dimension D much lower than the dimensionality of the input. The generative process is zero-mean Gaussian with isotropic noise σ:

$$p(Z_i|Z_{i-1}) = \mathcal{N}(Z_{i-1}, \sigma^2 I) \quad (11)$$

where Z_i is point on a low-dimensional manifold that generated X_i. I is the identity matrix. The expected distance between points Z_i and Z_j is the mean of a χ distribution:

$$\mathbb{E}\left[d\left(Z_i, Z_j\right)\right] = \sqrt{2\sigma|y_i - y_j|} \frac{\Gamma\left(\left(D+1\right)/2\right)}{\Gamma\left(D/2\right)} \quad (12)$$

where Γ is the Gamma function. Setting $\hat{d}_{ij}^m = \mathbb{E}\left[d\left(Z_i, Z_j\right)\right]$ ensures that the target depends on the underlying dimensionality of the input and the noise of the generative process.

The energy-based approach requires an analagous mapping from discrete seed identity, y, to a real-valued similarity score, s. For images in the same set, the simplest

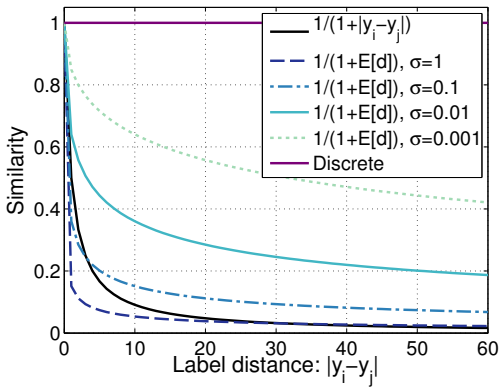


Figure 2: Soft similarity between two points based on seed labels. The "Discrete" similarity ignores seeds and treats all points in the same set as equally similar. This is analagous to methods like DrLIM. $s_{ij} = (1 + |y_i - y_j|)^{-1}$ is a simple empirical similarity. The remaining curves use $s_{ij} = (1 + \mathbb{E}[d(Z_i, Z_j)])^{-1}$ (see Eq.12) with $D = 32$.

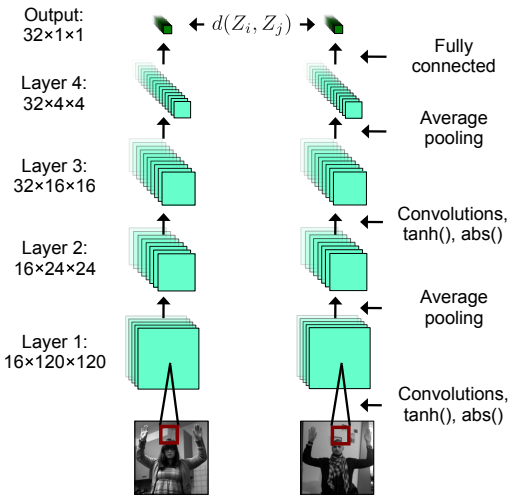


Figure 3: Convnet architecture for learning a mapping. Dimensions reflect the experiments in §4.2.

mapping we consider is $s^m_{ij} = (1 + |y_i - y_j|)^{-1}$. We also consider $s^m_{ij} = (1 + \mathbb{E}[d(Z_i, Z_j)])^{-1}$. Figure 2 shows similarity as a function of $|y_i - y_j|$ for $D = 32$ and a few choices of σ. We see that σ controls the spread of a soft window of affinity over nearby points. Small values of σ spread out the affinity, where large values induce a tight window that only considers immediate neighbours.

3.4. Nonlinear mapping

Related approaches have used a multi-layered neural network for $f(X^m_i|\theta)$ [21, 28]. However, for images larger than about 64×64 pixels, representing each image as a vector is inefficient. The number of parameters at the first layer grows quadratically with the image; moreover, the same features must be re-learned at different locations in the image. An alternative is to use a *convolutional architecture* [14]. By employing weight sharing and feature pooling, convolutional networks (convnets) require far fewer parameters and have "built in" invariance to small geometric distortions of the input.

Our mapping, shown in Fig. 3, is a siamese network that processes pairs of images through two identical pathways, each of which is a standard convnet, similar to ones used for object recognition. The first convolutional layer is composed of 16 $H_1 \times H_1$ learned filters, each of which is applied to the two input images. A $\tanh$ followed by abs is applied elementwise to the filtered images. This produces 16 feature maps which are averaged over non-overlapping $R_1 \times R_1$ windows. The second convolutional layer consists of 32 feature maps, each of which is connected to 4 feature maps of the previous downsampling layer. Connectivity is chosen uniformly randomly prior to learning. The second convolutional layer uses $H_2 \times H_2$ filters and the same nonlinear operations but is downsampled by a factor of R_2. The resulting 32 downsampled feature maps are concatenated into a vector which is fully connected to a 32-dimensional output, Z, by a learned matrix of weights. No nonlinearity is applied to the output. Task-dependent settings H_1,R_1,H_2,R_2 and other parameters related to learning are discussed in §4.

4. Experiments

We carry out two groups of experiments to evaluate our approach. The first uses a synthetic but structured dataset to validate our proposed learning algorithms but does not use any real imitations. The images we use are small enough to compare linear mappings directly to the convnet mappings using the probabilistic, batch model.

The second dataset, obtained from the web, consists of images of people performing imitations. The images are of sufficient resolution to impede linear or vector-based nonlinear mappings. The number of images is too large to work suitably with the batch method thus we use online learning.

4.1. Synthetic data

In [10] Hinton and Nair proposed a generative model of handwritten digits that mimicked a pen connected to two pairs of opposing springs. A motor program, which specified the stiffnesses of each of the springs at discrete time steps was used to generate images that looked very much like MNIST digits. Adding noise to the motor program of a given digit could produce very different images of the same class. They used a neural network to invert the generative model and recover the motor programs of the true MNIST digits. This low-dimensional representation, they suggested, was a semantic representation of the digit that could then be used for classification or producing more data.

We obtained Hinton and Nair's code to 1) create trajectories from motor programs, and 2) render 28×28 pixel

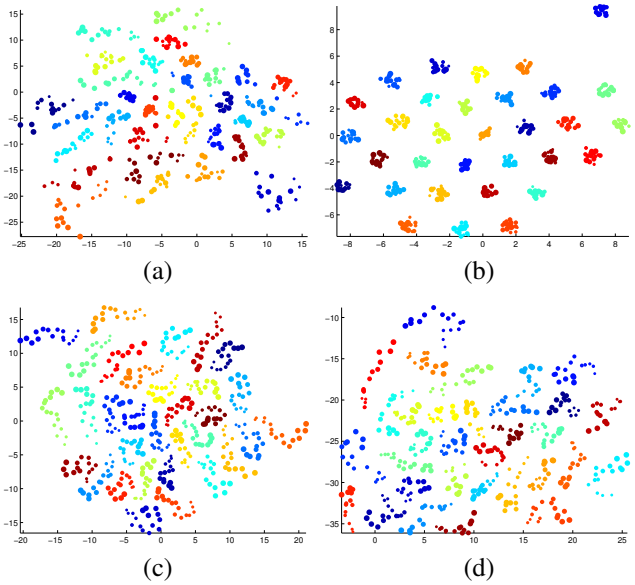


Figure 4: 2D embeddings learned by a) Standard NCA. Points in the same set are labeled similar. Points not in the same set are dissimilar. b) Our method, linear mapping, $\sigma = 0.01$. c) Our method, linear mapping, $\sigma = 1$. d) Our method, convnet mapping, $\sigma = 1$. Each colour represents a set of images. Marker size represents order within a set. Note how c) and d) preserve the chain structure of the data.

images from these trajectories. Through trial and error we found settings that produced motor programs which resembled pen strokes (see the first row of Fig. 5). Given a prototype, we could then create a chain-structured set of motor programs by repeatedly adding Gaussian noise ($\sigma = 0.04$) according to Eq. 11. We simulated the "imitation function" by rendering a single image from each motor program. Each column of Fig. 5 shows a set generated by this process.

To first visualize the different learning methods, we generate 32 image sets of length 16 each. We learn several embeddings to map the 28×28 images to D =2 dimensions. As a baseline we consider standard NCA which ignores the chain structure, treating all images in a set as similar, and images in different sets as dissimilar. We also try the linear and convnet variants of our probabilistic method with $\hat{d}_{ij}^m = \mathbb{E}\left[d\left(Z_i, Z_j\right)\right]$ and different settings of σ. Convnets use $H_1 = H_2$ =5,$R_1 = R_2$ =2. All methods are trained using nonlinear conjugate gradient in full batch. We see in Fig. 4 that all methods are able to separate the sets. However, only the approaches that utilize soft similarity (*i.e.* the linear and convnet variants of our method with a sufficiently large setting of σ) preserve the chain structure in the embedding.

To quantitatively evaluate the methods, we consider a larger dataset, generating 96 sets of length 16 each for training, validation, and test. Instead of visualization, which is difficult to quantify, we consider the image retrieval task. The most common evaluation metric used by the information retrieval community is Discounted Cumulative Gain (DCG) [4]. Typically DCG is used to measure search engine performance: a user submits a query and is presented with a ranked list of results. The DCG at rank K for a given query is computed as:

$$\text{DCG@}K = \sum_{j=1}^{K} \frac{2^{g_j} - 1}{\log\left(j+1\right)} \tag{13}$$

where we only consider the first K results and g_j is the relevance grade of the document at rank j. The numerator rewards documents of high relevance, while the denominator discounts the reward at lower ranks.

We learn a mapping using the training set. We then project each image of the test set into D dimensions, using the learned mapping. We then consider each test image, X_i^m as a "query" and its K nearest neighbors in the embedded space as the ranked list of results. DCG@K is then computed where the relevance of each result, X_j^n, is given by $g_j = \left(1 + |y_i - y_j|\right)^{-1}$ if the query and the result are in the same set (*i.e.* $m = n$) and $g_j = 0$ otherwise. DCG is computed on the validation set every 10 line searches to determine early stopping and prevent overfitting. We report mean DCG over the entire test set.

We experiment with several variants of computing target distance from the seed id, for points in the same set:

1. Simple: $\hat{d}_{ij}^m = |y_i - y_j|$.
2. Block: $\hat{d}_{ij}^m = 1$ if $|y_i - y_j| <= w$, otherwise $\hat{d}_{ij}^m = 0$ where w is some window. This is a discrete approach that considers seed id.
3. Random walk: $\hat{d}_{ij}^m = \mathbb{E}\left[d\left(Z_i, Z_j\right)\right]$ as given by Eq. 12. We use D =2 and σ =0.01,0.1,1.

Results are shown in Table 1. We show the mean and standard deviation over 10 repetitions with randomly initialized parameters. P-Lin and P-Conv are the linear and convnet versions of our probabilistic method, respectively. Two trends are apparent. First, using soft similarity is beneficial as standard NCA is consistently the worst performing method. Second, the nonlinear (convnet) variant of our method always outperforms the linear embedding. Both of these trends are more prominent in the lower dimensional embedding. We repeated these experiments using the online energy-based method. Results were similar to the probabilistic method and are provided as supplementary material.

This dataset is interesting since we have access to the underlying generative model, but it has a fundamental flaw. Examples within a set are still extremely close in the input space due to the simple approximation of the imitation function. That is why retrieval based on pixels, although slow, performs well according to DCG. We now turn to a more realistic and challenging task where imitations based on the same seed may appear very different in the input space.

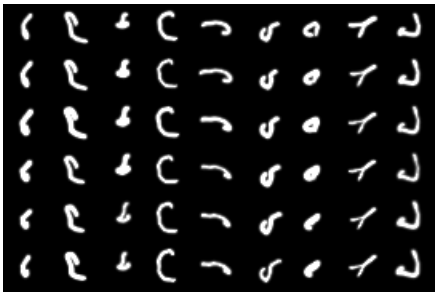


Figure 5: Examples of the synthetic strokes used in our experiments.

	K=1 NN		K=5 NN		K=10 NN	
	2D	16D	2D	16D	2D	16D
NCA	0.27 ± .03	1.13 ± .01	0.70 ± .07	2.39 ± .01	0.96 ± .04	2.82 ± .00
P-Lin (block) w=5	0.28 ± .02	1.14 ± .01	0.74 ± .06	2.40 ± .01	1.00 ± .06	2.81 ± .01
P-Lin (simple)	0.33 ± .02	1.16 ± .01	0.80 ± .06	2.43 ± .01	1.11 ± .07	2.83 ± .01
P-Lin $\sigma = 0.01$	0.30 ± .03	1.15 ± .01	0.77 ± .04	2.42 ± .01	1.00 ± .06	2.84 ± .01
P-Lin $\sigma = 0.1$	0.33 ± .02	1.16 ± .01	0.80 ± .04	2.43 ± .01	1.07 ± .05	2.84 ± .01
P-Lin $\sigma = 1$	0.34 ± .03	1.15 ± .02	0.81 ± .05	2.43 ± .01	1.17 ± .06	2.84 ± .01
P-Conv (block) w=5	0.43 ± .02	1.15 ± .01	1.10 ± .03	2.44 ± .01	1.44 ± .04	2.87 ± .01
P-Conv (simple)	0.46 ± .03	1.17 ± .01	1.16 ± .05	2.47 ± .01	1.53 ± .07	2.89 ± .01
P-Conv $\sigma = 0.01$	0.43 ± .03	1.15 ± .03	1.12 ± .04	2.43 ± .06	1.45 ± .06	2.89 ± .01
P-Conv $\sigma = 0.1$	0.44 ± .03	1.15 ± .02	1.09 ± .08	2.47 ± .02	1.51 ± .05	2.88 ± .01
P-Conv $\sigma = 1$	0.45 ± .04	1.11 ± .08	1.08 ± .14	2.44 ± .04	1.50 ± .09	2.87 ± .02
Pixels (784D)	1.23		2.53		2.95	

Table 1: Image retrieval performance (DCG@K) using the synthetic dataset.

4.2. Learning an invariant mapping to capture pose

In the next set of experiments, we consider the problem of matching people in similar pose but with different clothes, background, lighting, and other appearance changes. As an alternative to collecting imitation data ourselves, we leverage an existing project in an unintended way. *One Frame of Fame* [5] is a music video created by the Dutch band C-Mon & Kypski. It contains members of the band performing a variety of poses choreographed to music. However, there is a twist: the band aims to replace selected frames of their video with imitations captured from the webcam of an anonymous visitor. The band has created a web application to solicit frames. A visitor to the website is presented with a frame of the original video and they are asked to imitate that pose using their own webcam. At the time of writing, the band had collected 31,152 frames.

We downloaded all images with submission numbers 1-24,255, less 30 that were missing. The images were 450×338 jpegs. Frame numbers in the filename of each image allowed us to establish correspondence to the original frame and generate seed indices, y. Of the 6,029 frames in the original video, 1,400 had one or more audience imitations. The number of imitations per frame ranged from 1-39 with a median of 17. We manually determined scene cuts in the original video, which allowed us to divide the audience submissions into M =47 sets. Each image was downsampled, converted to greyscale and underwent local contrast normalization [17]. The images were then zero-padded vertically to be 128×128 at the original aspect ratio. We removed 160 images that were either all-zero or corresponded to seed frames that did not contain a person. The resulting training set of 24,065 images still contains many corrupted or incomplete uploads as well as images that either do not contain people or contain people not making an attempt to imitate. We estimate that 4% of the images fall into one of these categories, but we avoided manually cleaning the database to demonstrate that our method is robust to noise.

Several weeks after our initial collection, we downloaded all images with submission numbers 24,256-25,687, 2 of which were missing. We preprocessed these identically to the training set. As we use this set for testing, we manually removed 161 frames that were either corrupt, did not contain people, or contained people not imitating the seed frame. The resulting test set was 1,269 images, corresponding to 793 frames of the original video. The number of copies per frame ranged from 1-6 with a median of 1. The frame numbers of the training and test images, corresponding to the original video, were well-distributed and contained a significant degree of overlap.

Again we evaluate image retrieval performance by DCG. The seed frame numbers provide ground-truth correspondence from each test image to the training set. Therefore, we project each test image, X_i^m to a 32-dimensional space using the learned embedding and compute its distance to every embedded *training* image, X_j^n. We then find the K nearest neighbors and compute DCG@K (Eq. 13) using $g_j = (1 + |y_i - y_j|)^{-1}$ if $m = n$ (*i.e.* the points are in the same set) and $g_j = 0$ otherwise. We report mean DCG over the test set. We consider the following methods:

Pixel distance requires no learning. It is not practical in real situations due to the intractability of computing distances in high-dimensional input space.

PCA. We compute nearest neighbours in the linear space spanned by the 32 principal components of the training data.

Conv DrLIM [9] is identical to the method we describe in §3.2 except that graded similarity is not used. Pairs from the same scene are assigned a similarity of 1. Other pairs are assigned 0.

PSE. Our pose-sensitive embedding. We try several variants of computing similarity from seed id:

1. Simple: $s_{ij}^m = (1 + |y_i - y_j|)^{-1}$.
2. Block: $s_{ij}^m = 1$ if $|y_i - y_j| <= w$, where w is some window. Pairs of points on the same set but outside w are ignored. Pairs not in the same set have $s_{ij}^{mn} = 0$.
3. Random walk: $s_{ij}^m = (1 + \mathbb{E}[d(Z_i, Z_j)])^{-1}$ as given by Eq. 12. We use D =32 and σ =0.1,0.01.

PSE linear is a linear variant of our approach which uses a global descriptor (GIST [16]) as input instead of pixels.

The convnet is replaced by a 32×512 matrix.

All convnets used $H_1 = H_2 = 9$, $R_1 = 5$ and $R_2 = 4$. We train each model online, presenting pairs of inputs and taking a step proportional to the negative of the gradient given by Eq. 10. We use a learning rate of 0.01 which was roughly set as high as possible before observing oscillations. We use a momentum of 0.9. Balancing the training set by alternating similar pairs with dissimilar pairs accelerated learning considerably over simply choosing pairs at random. This was true for every model that we considered. Training takes less than 24h on a modern workstation with eight cores. The forward pass (from pixels to 32D embedding) takes 0.034s for a 128×128 image. Nearest-neighbour search (against a database of 24,065 images) takes 0.005s.

Fig. 6 shows DCG@K retrieval performance vs. number of weight updates for each of the learned models and the two baselines: Pixel matching and PCA. PSE, which considers soft similarity, demonstrates a clear performance gain over DrLIM. However, the linear variant, which relies on a global image descriptor, performs poorly. Using the stochastic noise process to compute similarity shows a modest gain over the simple approach. Contrary to the synthetic task, pixel-based matching performs terribly. This is also true for PCA. Most striking is to look at the nearest neighbors returned by the query (Fig. 7). Our embedding is highly effective at finding people in similar pose. It is invariant to identity, clothing, lighting, shift and scale.

5. Conclusion

In this paper we focus on the problem of learning invariant embeddings from image data. As an alternative to assigning discrete or real-valued labels which are expensive and often difficult to define, we propose crowd-sourced imitations as a way to quickly amass a large, varied set of examples with graded similarity. We demonstrate our approach by learning a complex, nonlinear mapping in which people in similar pose lie close-by regardless of identity, background, lighting or other appearance changes. Quantitatively our method outperforms existing supervised embedding approaches. We also demonstrate impressive qualitative results in retrieval. Our data is sourced entirely from the web and requires no manual labeling, other than determining scene cuts from video – a task that can be automated.

Our mapping has been trained discriminatively end-to-end. Future work will explore unsupervised learning using forms of convolutional sparse coding[13, 30] to initialize the parameters of the convnet. This will allow us to exploit even larger image databases. We are also interested in the multitask setting, supplementing our dataset with real or synthetic data labeled with articulated 2D or 3D pose.

References

[1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. *CVPR*, 2004. 2729

[2] S. Becker. Learning to categorize objects using temporal coherence. In *NIPS*, pages 361–368, 1993. 2730

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *CVPR*, pages 1365–1372, 2010. 2730

[4] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *SIGIR*, pages 63–70, 2007. 2733

[5] C-Mon and Kypski. One frame of fame. http://oneframeoffame.com, 2010. 2729, 2734

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2730

[7] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004. 2729, 2730

[8] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV*, pages 641–648, 2003. 2729

[9] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742, 2006. 2729, 2730, 2731, 2734

[10] G. E. Hinton and V. Nair. Inferring motor programs from images of handwritten digits. In *NIPS*, 2006. 2732

[11] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *NIPS*, pages 833–840, 2003. 2730

[12] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, Los Alamitos, CA, USA, 2008. 2730

[13] K. Kavukcuoglu, M. Ranzato, and Y. LeCun. Fast inference in sparse coding algorithms with applications to object recognition. Technical report, NYU, 2008. CBLL-TR-2008-12-01. 2735

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. 2732

[15] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *ICML*, pages 737–744, 2009. 2730

[16] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 2734

[17] N. Pinto, D. Cox, and J. DiCarlo. Why is real-world visual object recognition hard? *PLoS Comput Biol*, 4(1), 2008. 2734

[18] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *PAMI*, 29(10):1848–1852, 2007. 2730

[19] B. Russell and A. Torralba. Building a database of 3D scenes from user annotations. In *CVPR*, pages 2711–2718, 2009. 2730

[20] B. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1):157–173, 2008. 2730

[21] R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, volume 11, 2007. 2729, 2730, 2732

[22] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, pages 750–759, 2003. 2729, 2730

[23] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual human motion recognition. *CVIU*, 104(2-3):210–220, 2006. 2730

[24] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. of the Conf. on Empirical Methods in NLP*, pages 254–263. Association for Computational Linguistics, 2008. 2730

[25] A. Sorokin and D. Forsyth. Utility data annotation with Amazon Mechanical Turk. *Proc of First IEEE Workshop on Internet Vison at CVPR 2008*, 2008. 2730

[26] I. Spiro, G. Taylor, G. Williams, and C. Bregler. Hands by hand: Crowd-sourced motion tracking for gesture annotation. In *IEEE CVPR Workshop on Advancing Computer Vision with Humans in the Loop*, 2010. 2730

[27] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *NIPS*, pages 1345–1352, 2007. 2730

[28] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008. 2729, 2730, 2732

[29] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *NIPS*, pages 1441–1448, 2006. 2730

[30] M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010. 2735

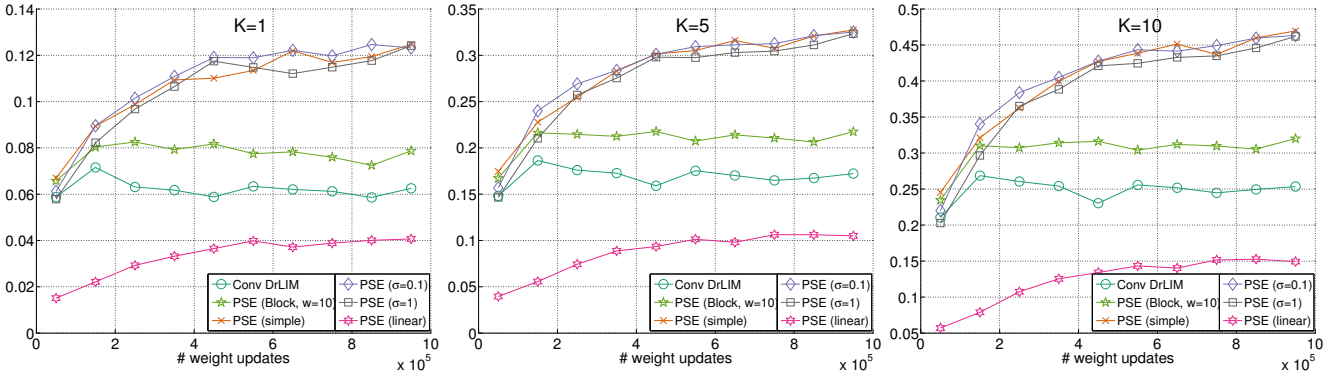
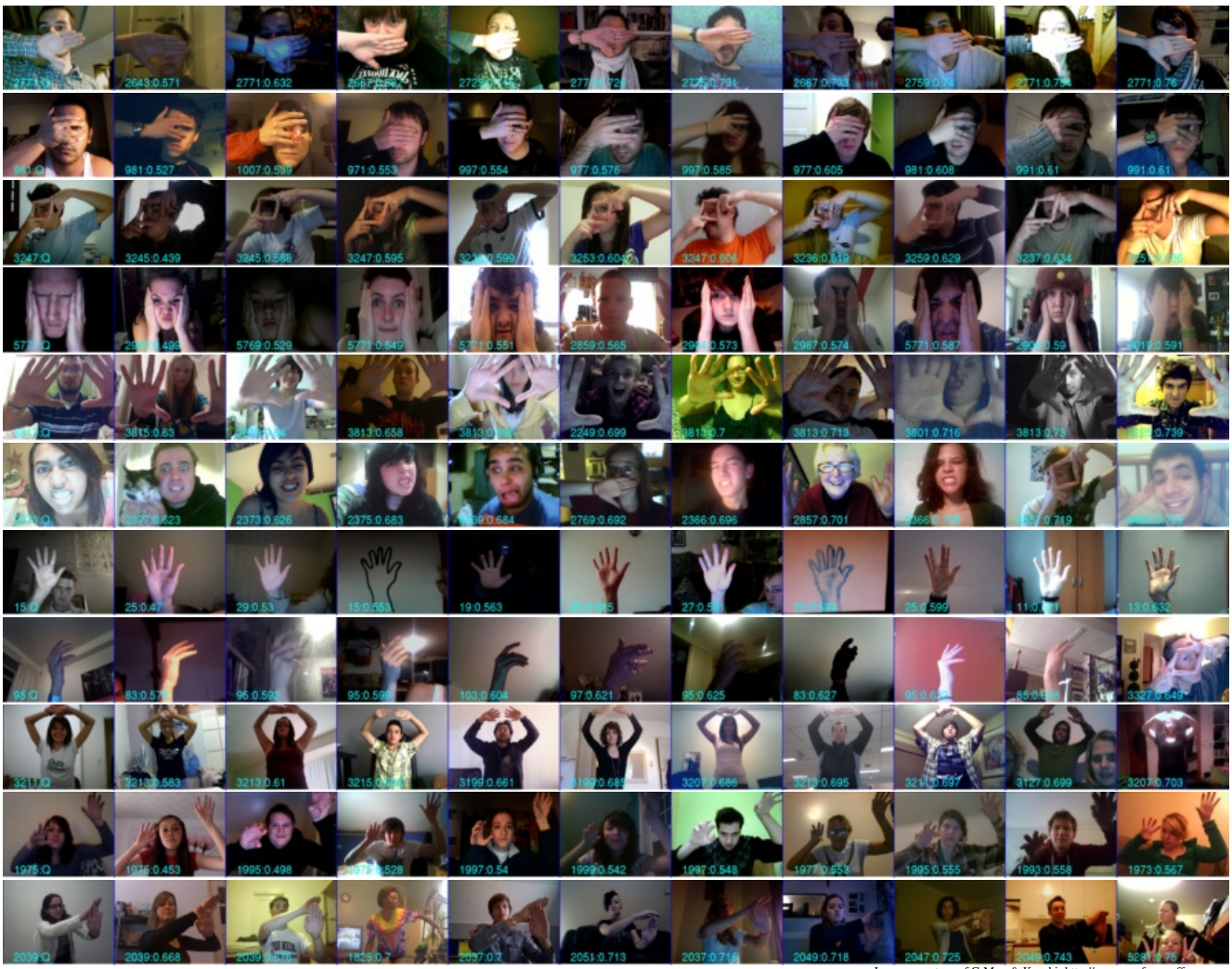


Figure 6: Retrieval performance. DCG@K on the test set vs. # of weight updates for various learned 32D embeddings of the *One Frame of Fame* dataset. Pixel-based matching performance is well below the curves, at (**0.005,0.015,0.021**) for K=1,5,10 respectively. PCA (32 components) achieves (**0.005,0.018,0.026**). We also tried (but do not show) the Block method with $w = 5$. It performed slightly worse than $w = 10$. Models are trained for 1,000,000 weight updates. DCG was computed at intervals of 10,000 weight updates. To avoid clutter, we report the mean over each block of 100,000 weight updates.



Images courtesy of C-Mon & Kypski. http://www.oneframeoffame.com

Figure 7: Sample retrieval results. Each row is a query. We select a test image (column 1) and find its 10 nearest neighbors using our learned embedding: PSE (simple). Text indicates seed id (left) and distance from the query (right).