Modeling the joint density of two images under a variety of transformations

Joshua Susskind
Institute for Neural Computation
University of California, San Diego
United States
josh@mplab.ucsd.edu

Roland Memisevic
Department of Computer Science
University of Frankfurt
Germany
ro@cs.uni-frankfurt.de

Geoffrey Hinton
Department of Computer Science
University of Toronto
Canada
hinton@cs.toronto.edu

Marc Pollefeys
Department of Computer Science
ETH Zurich
Switzerland
marc.pollefeys@inf.ethz.ch

Abstract

We describe a generative model of the relationship between two images. The model is defined as a factored three-way Boltzmann machine, in which hidden variables collaborate to define the joint correlation matrix for image pairs. Modeling the joint distribution over pairs makes it possible to efficiently match images that are the same according to a learned measure of similarity. We apply the model to several face matching tasks, and show that it learns to represent the input images using task-specific basis functions. Matching performance is superior to previous similar generative models, including recent conditional models of transformations. We also show that the model can be used as a plug-in matching score to perform invariant classification.

1. Introduction

The ability to judge whether two images, or image patches, are "the same" is one of the most basic and central operations in most computer vision tasks. Matching images that show the same *content* across different views, for example, is the key operation in most retrieval and classification tasks. Matching patches that represent the same *position* across different images is the main task in virtually all geometric inference, tracking, stereo, and related tasks.

What makes matching hard is the fact that it is intimately tied to the presence of *invariants* in a task at hand: Two images are the same exactly if they are invariants under some class of allowable transformations. In many common tasks the "allowable transformations" are, unfortunately, much too complex to be modeled sufficiently well with rigid geometric transformations or simple photometric effects. They involve homographies or affine transformations in many geometry tasks, and highly complex, and often subtle, non-rigid transformations in the case of face modeling and retrieval. Modeling these transformations by hand is usually either impractical or extremely time-consuming and difficult.

Figure 1. Subset of filter pairs learned from non-rigid transformations of faces. To compute the joint density of two images, filters in the top row are applied to the first image and filters in the bottom row are applied to the second image. The responses of corresponding filters are multiplied and projected into a feature space that represents relations. Both the mappings and the filters are learned from data (see main text for details).

The task of *learning* about invariances has received a fair amount of attention in the past. In pure matching tasks, *metric learning* (see, for example, [27], [2], [1]) has been used with varying levels of success. Many metric learning methods, unfortunately, are either based on adapting a Mahalanobis distance (in which case they are not expressive enough to model the subtle variations required by datasets like face images), or they require a considerable amount of hand-crafting to achieve the desired invariances (such as the use of convolutional neural networks in [2], or carefully designed Haar-features in [1], among others). Invariances

have also been addressed with bilinear models (for example, [23], [18], [4]).

Matching of large images (for registration, retrieval, or other tasks) is typically performed by resorting entirely to hand-crafted systems operating on interest points. Since interest point positions need to be matched as a sub-routine, these approaches still rely on, and thus profit from, being able to globally match regions around interest points well.

An entirely different, probabilistic approach to modeling allowable transformations was recently suggested by [15]. The approach has been extended and deployed in various applications, such as [16], [21], [8], [17]. The idea behind this line of work is to use binary latent variables to learn the conditional distribution of one image (the "output") given another image (the "input"). The conditional distribution is modeled by a conditional Restricted Boltzmann Machine (RBM) which makes it possible to deploy efficient learning methods known as contrastive divergence learning (CD) [5]. Unfortunately, in inference tasks that involve matching one cannot use the conditional distribution directly as a score, because the RBM distribution is defined only up to an unknown normalizing constant. In matching applications one uses instead a measure of how well the model transforms one image into another as a score function which has been shown to work fairly well in some cases [15], [8].

In this paper, we describe an approach to modeling the *joint* distribution over image pairs, and show that, despite the apparent inapplicability of CD learning, it is possible to train the model efficiently using an appropriate variation of contrastive divergence which we call 3-way CD. We then show how the joint model allows us to perform matching using probabilities over image pairs, and that the probabilistic score yields a much higher accuracy than conditional models. Learning a joint density also allows us to overcome filter normalization issues (due to having a conditional partition function), which often lead to high-frequency noise-artifacts on the input filters (see, for example, [8]).

We apply the model to face images as a major case study. Faces represent a particularly difficult class of objects for many vision tasks, because they show a very wide range of meaningful configurations that cannot be captured with simple rigid transformations. One way to analyze how faces change is to represent transformations between pairs of face images of the same type, such as same identity pairings that occur in neighboring video frames, or same expression pairings that occur across people experiencing the same emotional context. A model's capacity at representing transformations can be judged by attempting to transfer an expression from one individual onto the face of another individual whom the model has never seen before. This analogy task is known as expression transfer [24]. An immediate application of a model of face transformations is verification, where two or more images are compared in order to determine if they are the same identity or expression.

In our approach we assume no a priori knowledge of what features would be useful for these, or similar, tasks. While most approaches to face transformation employ morph bases, we learn a distributed representation of component "morphlets", which capture the non-rigid nature of face transformations. We apply joint density models to two very different kinds of image pairs, same identity/different expressions versus same expression/different identity. In both cases, modeling the relation between pairs requires the model to extract highly non-linear/non-rigid transformations of faces, yet the two tasks differ in the importance of identity versus expression.

2. The model

We define a probabilistic model that captures the *relationship* between two real-valued images $\boldsymbol{x}$ and $\boldsymbol{y}$. In many real-world tasks, such as dealing with facial transformations, there can be a multitude of subtle, non-linear interactions between the pixel intensities. We can model these interactions probabilistically by marginalizing over a set of binary latent variables $h_1, \ldots, h_K$. In our case, each of the latent variables can contribute a "basis relationship" to an overall model of the dependency between $\boldsymbol{x}$ and $\boldsymbol{y}$.

Following [16], we define the joint probability distribution over triplets ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{h}$) using a *matching score*

$$S(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = \sum_{f=1}^{F} (\sum_{i=1}^{I} v_{if} x_i)(\sum_{j=1}^{J} w_{jf} y_j)(\sum_{k=1}^{K} u_{kf} h_k) \qquad (1)$$

that first projects $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{h}$ onto F basis functions (or "filters") $u_{\cdot f}$, $v_{\cdot f}$ and $w_{\cdot f}$, respectively, and then sums over *products* of corresponding filter-responses. As a result the score function assigns large values to triplets whose filter-responses tend to *match* well. The filters themselves will be learned from training data as we shall show, allowing the score function to assign meaning to the co-occurrence of subsets of filter responses by letting the hidden variables weight these co-occurrences appropriately.

To turn the matching score into a probability distribution we first add "bias" terms including quadratic containment of the visibles

$$\begin{aligned} E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = &-S(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) \\ &- \sum_{k=1}^{K} u_k h_k + \frac{1}{2} \sum_{i=1}^{I} (x_i - v_i)^2 + \frac{1}{2} \sum_{j=1}^{J} (y_j - w_j)^2 \end{aligned} \qquad (2)$$

which we then exponentiate and normalize

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) &= \frac{1}{Z} \exp\left(-E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h})\right), \\ Z &= \sum_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}} \exp\left(-E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h})\right) \end{aligned} \qquad (3)$$

The particular choice of bias terms yields Gauss conditional distributions to model continuous data as we discuss below. The number F of triplets and the number K of latent variables have to be set by hand or by cross-validation, and typically range in the 100's to 1000's.

To obtain the distribution over an image pair $(\boldsymbol{x}, \boldsymbol{y})$, we can now marginalize over $\boldsymbol{h}$:

$$p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{h} \in \{0,1\}^K} p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) \tag{4}$$

Note that the sum over $\boldsymbol{h}$ is not tractable for large K, because it contains 2^K terms. As we shall discuss below, however, all that is required for maximum likelihood learning and for matching are *conditional* distributions under the model, which can be computed efficiently.

By plugging the definition of energy (Eq. 2) into Eq. 3, and normalizing appropriately, we get

$$p(\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{x}) = \prod_k \text{bernoulli}\big(u_k + \sum_f u_{kf} \sum_i v_{if} x_i \sum_j w_{jf} y_j\big) \tag{5}$$

$$p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{h}) = \prod_i \mathcal{N}\big(v_i + \sum_f v_{if} (\sum_j w_{jf} y_j)(\sum_k u_{kf} h_k); 1.0\big) \tag{6}$$

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{h}) = \prod_j \mathcal{N}\big(w_j + \sum_f w_{jf} (\sum_i v_{if} x_i)(\sum_k u_{kf} h_k); 1.0\big) \tag{7}$$

This shows[1] that, among the three sets of variables, computation of the conditional distribution of any *one group* ($\boldsymbol{x}$, $\boldsymbol{y}$, or $\boldsymbol{h}$), given the other *two*, is tractable. Furthermore, the variables within a group are conditionally independent given the variables in the other two. An illustration of the model is shown in Figure 2. Variables (pixels and hidden variables) are shown as circles, filters as triangles.

Our model is related to previous work on three-way interactions, such as [16], [19]. In particular, an energy function similar to Eq. 2 was used in [16], with the difference that we are modeling real-valued rather than binary images and that we are concerned with non-rigid transformations. More importantly, the normalizing constant Z in Eq. 3 in contrast to previous methods (such as [16], [21], [8]) is a sum over $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{h}$. We thus define the *joint* distribution for an image pair, rather than a conditional distribution of an output-image given an input image. This has the effect that learning, unlike in these approachs, does not boil down to training a set of case-specific RBMs, making it impossible here to deploy standard contrastive divergence learning [5]. As we demonstrate in our experiments, however, it is possible to efficiently train this model using a "three-way" version of contrastive divergence learning, which we discuss in the next section.

[1] It would be possible to modify the model, such that the variances in Eqs. 7 and 6 are different from one. Here, instead, we normalize images, such that each pixel, independently, has mean zero, and standard deviation one on average, before deploying or training the model.

2.1. 3-way Contrastive Divergence

To train the model we maximize the log-likelihood $L = \sum_\alpha \log p(\boldsymbol{y}^\alpha, \boldsymbol{x}^\alpha)$ for a set of training pairs $\{(\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha)\}$. The derivative of L wrt. a single model parameter θ is given by

$$-\frac{\partial L}{\partial \boldsymbol{\theta}} = \sum_\alpha \left\langle \frac{\partial E(\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right\rangle_{\boldsymbol{h}} - \left\langle \frac{\partial E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right\rangle_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}} \tag{8}$$

The second term in this sum is an average wrt. to the model distribution over $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{h}$, and thus it cannot be computed in closed form. It is possible, however, to approximate the average by drawing samples from the distribution. Since the model exhibits a "tri-partite" structure, drawing samples from any of the distributions $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{h})$, $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{h})$, $p(\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{x})$ is straightforward, using the decoupling into products in Equations 5 to 7. This suggests using Gibbs sampling, by repeatedly sampling from the distributions in one of these groups, conditioned on the other two.

The tri-partite structure facilitates Gibbs-sampling in this model just as the *bi-partite* structure in a standard RBM [5]. In contrast to a standard RBM, to sample each variable once requires visiting (all) three groups of variables rather than just two. "Alternating" Gibbs sampling in an RBM is replaced by three-way iterations in this case.

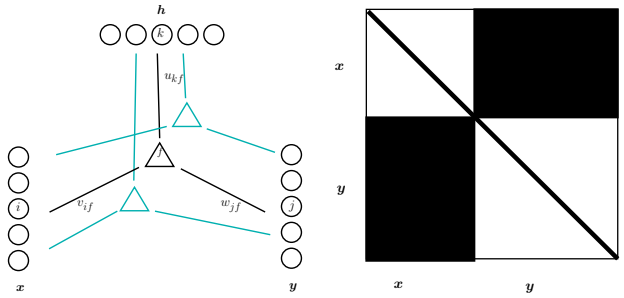

Figure 2. Left: Schematic representation of the model. Right: The conditional inverse covariance matrix over image-pairs, given hidden variables. Note that the number of pixels is not necessarily the same in images $\boldsymbol{x}$ and $\boldsymbol{y}$.

Like using CD in a standard RBM, it is possible to initialize the sampling at the training-data and to cut it short before reaching the equilibrium distribution [5]. Because of the three-way structure, a single iteration involves sampling and updating each of the three groups of variables. The *order* with which sites are visited therefore becomes a choice that one has to make (unlike in a two-way model). Here we make this choice randomly in every iteration and show that this is a viable approach in our experiments. The overall

algorithm is shown in figure 2.1, where we use the matrices U, V, W containing the F basis functions in their rows and vectors $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$ containing the biases. In practice, one can add a regularization term $-\lambda(\|U\|_2^2 + \|V\|_2^2 + \|W\|_2^2)$ to the objective function where λ is chosen by hand or by cross-validation.

Algorithm 1 Three-way contrastive divergence learning

Require: Data-set $(\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha)_{\alpha=1}^N$, learning-rate ϵ
  repeat
    for α from 1 to N **do**
      compute $A^x = U\boldsymbol{x}^\alpha$, $A^y = V\boldsymbol{y}^\alpha$
      set $h_k = p(h_k|\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha)$ for each k
      compute $A^h = W\boldsymbol{h}$
      apply **positive phase** updates:
      $U = U + \epsilon(A^x \boldsymbol{x}^{\mathrm{T}})$, $V = V + \epsilon(A^y \boldsymbol{y}^{\mathrm{T}})$,
      $W = W + \epsilon(A^h \boldsymbol{h}^{\mathrm{T}})$,
      $\boldsymbol{u} = \boldsymbol{u} + \epsilon\boldsymbol{x}^\alpha$, $\boldsymbol{v} = \boldsymbol{v} + \epsilon\boldsymbol{y}^\alpha$, $\boldsymbol{w} = \boldsymbol{w} + \epsilon\boldsymbol{h}$
      sample $\hat{\boldsymbol{h}}$ from $p(\boldsymbol{h}|\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha)$
      sample g from bernoulli(0.5)
      if $g > 0.5$ **then**
        sample $\hat{\boldsymbol{x}}$ from $p(\boldsymbol{x}|\hat{\boldsymbol{h}}, \boldsymbol{y}^\alpha)$, set $A^x = V^{\mathrm{T}}\hat{\boldsymbol{x}}$
        sample $\hat{\boldsymbol{y}}$ from $p(\boldsymbol{y}|\hat{\boldsymbol{h}}, \hat{\boldsymbol{x}})$, set $A^y = W^{\mathrm{T}}\hat{\boldsymbol{y}}$
      else
        sample $\hat{\boldsymbol{y}}$ from $p(\boldsymbol{y}|\hat{\boldsymbol{h}}, \boldsymbol{x}^\alpha)$, set $A^y = W^{\mathrm{T}}\hat{\boldsymbol{y}}$
        sample $\hat{\boldsymbol{x}}$ from $p(\boldsymbol{x}|\hat{\boldsymbol{h}}, \hat{\boldsymbol{y}})$, set $A^x = V^{\mathrm{T}}\hat{\boldsymbol{x}}$
      end if
      set $h_k = p(h_k|\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ for each k
      compute $A^h = W\boldsymbol{h}$
      apply **negative phase** updates:
      $U = U - \epsilon(A^x \hat{\boldsymbol{x}}^{\mathrm{T}})$, $V = V - \epsilon(A^y \hat{\boldsymbol{y}}^{\mathrm{T}})$,
      $W = W - \epsilon(A^h \boldsymbol{h}^{\mathrm{T}})$,
      $\boldsymbol{u} = \boldsymbol{u} - \epsilon\hat{\boldsymbol{x}}$, $\boldsymbol{v} = \boldsymbol{v} - \epsilon\hat{\boldsymbol{y}}$, $\boldsymbol{w} = \boldsymbol{w} - \epsilon\boldsymbol{h}$
      renormalize $U, V, W, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$
    end for
  until convergence criterion reached

It is interesting to note that, by defining their model as a *conditional* model, [16] and related approaches side-step difficulties related to three-way structure and use contrastive divergence training like in an RBM. We show that using three-way iterations in learning allows us to obtain a fully probabilistic model, which, in turn, can be used to define a highly accurate invariant metric[2] for use in matching and classification tasks.

Plugging Eq. 2 into Eq. 3 and assuming $\boldsymbol{h}$ to be fixed shows that the *joint* conditional distribution over image pairs $(\boldsymbol{x}, \boldsymbol{y})$ given $\boldsymbol{h}$, is a Gaussian distribution. The precision (inverse covariance) matrix $P^{\boldsymbol{h}}$ of this Gaussian is the identity matrix with additional off-diagonal terms that are non-zero only *across* a pair of images (but not within a single image), and that are given by

$$P^{\boldsymbol{h}}_{ji} = P^{\boldsymbol{h}}_{ij} = -\sum_k h_k \sum_f u_{kf} v_{if} w_{jf}. \tag{9}$$

The structure of this matrix is illustrated in Figure 2, where white denotes zero and black non-zero.

In practice, it is important that the matrix stays positive-definite during the optimization. An analogous requirement for a single image model was described in [19]. We found that an effective way to keep $C^{\boldsymbol{h}}$ positive definite and to avoid numerical instabilities is to simply normalize the filter-coefficients (*i.e.* the columns of U, V and W) after each gradient update. In contrast to [19] we do not take any additional measures to avoid instabilities[3].

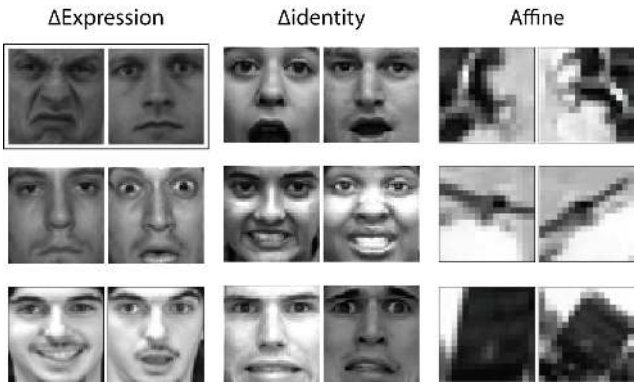

Figure 3. Examples of training image pairs. Same identity pairs were used in tasks 1 and 3, same expression pairs were used in task 2, and affine image patches were used in task 3.

2.2. Learning a semi-metric

Note that the block structure of the precision matrix (Figure 2 (b)) is consistent with our goal to encode relationships *between* images rather than the structure within a single image. In particular, the model does *not* try to faithfully represent any covariances within an image, but only cross-covariances between pixels x_i in one image and pixels y_j in the other image. We now describe how the model allows us to assess the similarity between images after it has been trained on image pairs.

The log-probability that the model assigns to an image pair $(\boldsymbol{x}, \boldsymbol{y})$ is given by

$$\log p(\boldsymbol{y}, \boldsymbol{x}) = -\log Z - \frac{1}{2}\Big(\sum_i (v_i - x_i)^2 + \sum_j (w_j - y_j)^2\Big) + \sum_k \log\Big(1 + \exp\big(u_k + \sum_f u_{kf} \sum_i v_{if} x_i \sum_j w_{jf} y_j\big)\Big) \tag{10}$$

[2] Strictly speaking the model defines a "semi-metric", because it does not strictly enforce the triangle-inequality. In matching tasks a semi-metric is all we need, however.

[3] Since diagonal dominance implies positive definiteness [7], one way of ensuring positive definiteness is by keeping the off-diagonal entries sufficiently small, which is achieved implicitly by the normalization.

We can compute this quantity only up to $\log Z$. However, $\log Z$ is the same for all image pairs $(\boldsymbol{x}, \boldsymbol{y})$, so it cancels when *comparing* image pairs and when the unnormalized log-probability is used as a compatibility score.

Naively using this score as a metric on images would be problematic, because the similarity judgement for a pair $(\boldsymbol{x}, \boldsymbol{y})$, could be made arbitrarily bad, for example, by rescaling both images with the same constant factor. To remove the sensitivity to how well images themselves are modelled, we use the following definition of dissimilarity between two images $\boldsymbol{x}$, $\boldsymbol{y}$:

$$d(\boldsymbol{x}, \boldsymbol{y}) = -\log p(\boldsymbol{x}, \boldsymbol{y}) - \log p(\boldsymbol{y}, \boldsymbol{x}) + \log p(\boldsymbol{x}, \boldsymbol{x}) + \log p(\boldsymbol{y}, \boldsymbol{y}) \tag{11}$$

A similar approach is taken in [22] to define a semi-metric using a RBM without three-way interactions. We compare to this method in Section 3, where we show that three-way connections can greatly improve the performance on matching tasks.

It is important to note that in conditional models (for example, [16]), the normalizing constant Z is a function of the input image, so that the unnormalized log-probability (Eq. 10) cannot be used to define a metric. The authors suggest using a squared "one-step-reconstruction error" instead, and report fairly good results on some classification problems. We show in Section 3.1 that much better results can be achieved using unnormalized log-probabilities.

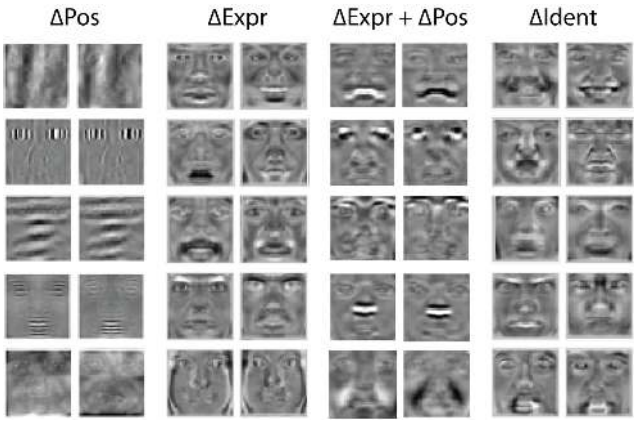

Figure 4. Morphlets learned from different types of training pairs. In each panel, the left column shows input filters w_{if}, and the right column shows output filters w_{jf}. Position morphlets were learned from pairs where the image was shifted randomly by +/-2 pixels in x, y, holding all else constant.

Note that the definition of the semi-metric (Eq. 11) also ensures symmetry (*i.e.* that $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$). An alternative way to obtain a symmetric dissimilarity measure is by using *weight-sharing* during learning, to ensure that each input-filter (column of V) also occurs as an output-filter (column of W) and vice versa. Furthermore, parameters involving the hidden variables (U and the biases $\boldsymbol{u}$), which weight the filter responses, then need to be adjusted accordingly. We use both the basic and the symmetric models in our experiments.

3. Experiments

We experimented[4] with two kinds of data. For both, we compared quantitatively to methods such as conditional models, bilinear models, as well as models of the joint distribution of a concatenation of two images.

The first type of data contains patches of natural images related by *affine* transformations. We generated data-sets based on (i) patches of size 16×16 cropped from images in the CIFAR database [9], and (ii) patches of larger 28×28 patches cropped from the van-Hateren database (first used in [25]).

The second type of data contains transformations of face images. We used $12,973$ images from the Toronto Face Database (TFD) [20] containing frontally posed facial expressions of the 6 basic emotions [3], and neutral. In order to test the model on a potentially more difficult face database, we also used $12,072$ images from the Pubfig database [10] which contains images extracted from the Internet by a face detector. These images tend to vary greatly in lighting, pose, and occlusions. Examples of image pairs used in the experiments are shown in figure 3.

3.1. Invariant classification

As a proof of concept, we tested the model as a plug-in matching score for classification using the "rotated MNIST" dataset described in [11]. The dataset contains *rotated handwritten digits* ("0" to "9") of size 28×28 pixels as well as labels; the task is classification. There are 12000 cases for training and 50000 cases for testing. We trained the symmetric model with 100 mapping units and 500 factors on the 28×28 affine van Hateren patches. For classification, we used k-nearest neighbors with the matching score defined by the trained model. We kept 2000 cases from the training data as a validation set used to choose k. For nearest neighbors using Euclidean distance the test-set performance is 13.21% false (at $k = 7$). For the semi-metric inferred by the model the performance is 10.13% false (at $k = 5$). This shows that the model is able to effectively encode the invariances in a way suitable for classification. For reference, the performance of other methods on this dataset is shown in Table 3.1. The model knows the type of invariances in this task and has thus an advantage over black-box methods. However, it is important to note that there were no digits in the training data used for training the model. We did not vary any of the parameters (number of hiddens/factors) in the experiment, so performing a search over these using cross-validation is likely to improve

[4] A GPU-based implementation of the model in the Python language is available at http://www.cs.toronto.edu/~rfm/code/gbmcuda.py.

Model	**3-way**	Knn	SVM	DEEP	NNet
Performance	**10.13**	13.21	11.11	10.30	18.11

Table 1. Error rates in % for an invariant classification task. Keys used in the table: *3-way:* Model described in this paper; *Knn:* K-Nearest Neighbors; *SVM:* Best performing SVM (RBF kernels); *DEEP:* Best performing deep belief net; *NNet:* Multilayer Perceptron.

performance. A model trained on rotations, rather than general affine transformations, may also improve performance. In any case it is clear that the model can effectively transfer knowledge from the domain of general natural images to the domain of hand-written digits.

3.2. Learning expression and identity morphlets

We trained two models on face pairs from the TFD in order to extract features that represent either changes in facial identity holding expression constant (identity morphlets) or changes in facial expression holding identity constant (expression morphlets). For the expression changes, we trained a model on 8 million same identity image pairs using 512 factors, and 512 hidden units. We did not use weight-decay. For the identity changes, we trained an identical model on 8 million same expression image pairs. In addition, we trained models on rigid image shifts and on shifts combined with expression changes to demonstrate that very different kinds of change can be represented by the same type of model. Figure 4 shows image filters for the various types of transformation, Figure 1 shows some additional filters for shifts combined with expression changes. The plots demonstrate that the morphlets take on distinct forms that depend on the class of transformation. A lot of the learned filters are reminiscent of the localized filters that one obtains with non-negative matrix factorization [12], but there are no positivity constraints, and these filters come in pairs. There are also some filters that are more global.

3.3. Facial expression transfer

The problem of remapping a facial expression change from one person onto another is known as expression transfer [24]. A model trained to represent expression transformations by extracting identity morphlets can perform expression transfer by applying a transformation inferred from a source pair to a target input image, which can be either x or y. This is a type of analogy making [16], which in this case is complicated by the fact that it relies on highly delicate, localized deformations. In particular, the target input image is a different identity than the source pair, and we need to deform the target identity such that the output image matches the expression of the corresponding source image, while retaining the identity of the target (see Figure 5). The results demonstrate a crucial capacity of face representation, which is the ability to separate static identity information of a face from its dynamic expression features.

Figure 5. Facial expression transfer for random test pairs using a model trained on same identity face pairs. Face pairs in the top row are shown to the model at test time in order to infer a facial expression transformation. The inferred transformation is then applied to left images of the bottom rows to obtain synthetic expressions that retain facial identity.

3.4. Quantitative comparisons

We examined the performance of the model quantitatively on four tasks. Each task amounts to determining for a set of image pairs, whether the images are "the same" or not, where the definition of sameness is different in each task. For each task we used the area under the ROC curve (AUROC) as a measure of discrimination performance. We set F, K and λ using an independent validation set.

First we compared to the conditional model described in [16], using the reconstruction error metric as suggested by the authors. Second, we compared to a bilinear variation of the joint model to assess whether the nonlinear representation of transformations captured by the binary hidden units in our model is more suited to metric learning than an equivalent bilinear approach. We also compared to RBMs on the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y}$ as an alternative probabilistic model for which we can obtain a free energy score for discrimination [22]. This also allows us to assess the importance of using three-way connections. We used cosine similarity applied to the pixel vectors as a baseline discrimination measure. In addition to the basic formulation of the model, we examined the symmetric version using weight-sharing to enforce symmetry (see Section 2.2).

The first two tasks used face images from the TFD [20]. For Task 1 (TFD Same ID), models were trained on pairs of images of the same identity only, and the discrimination task was same versus different person. For Task 2 (TFD Same Exp), models were trained on pairs of images of the same expression from two different identities, and the discrimination task was same versus different expression. Task 3 (Pubfig Same ID) was trained on images from the Pubfig database [10], and like Task 1 also performed same versus different identity discrimination. For the fourth task

Model/Task	TFD ID	TFD Exp	PUBFIG ID	AFFINE
cosine	0.848	0.663	0.649	0.721
RBM	0.869	0.656	0.647	0.799
conditional	0.805	0.634	0.557	0.825
bilinear	0.905	0.637	**0.774**	0.812
3-way	0.932	**0.705**	0.771	0.930
3-way symm	**0.951**	0.695	0.762	**0.931**

Table 2. Area under the ROC curve on four matching tasks.

(AFFINE) we used affine transformations of the CIFAR patches of size 16-by-16 pixels (examples also shown in Figure 3). The task is to determine whether two patches are affine transforms of each other or not.

The ROC curves are plotted in Figure 6, and AUROC measures are shown in Table 3.4. The nonlinear 3-way joint model outperformed both the RBM and the conditional model by a fairly large margin in all tasks. The joint nonlinear models outperformed the bilinear model on all tasks except Task 3 (PUBFIG Same ID), for which performance was basically equivalent for the non-symmetric and bilinear models. The symmetric and non-symmetric formulations tended to perform similarly in all tasks. Interestingly, the conditional model using the reconstruction metric [16] performed significantly worse than the RBM, even though the RBM did not model any 3-way interactions. Critically, for the face tasks, the matching score applied by the joint 3-way models always significantly outperformed the cosine baseline, but the reconstruction error metric used by the conditional model did not. On the other hand, all models, whether using the matching score (joint models and RBM) or reconstruction error (conditional model), outperformed the cosine measure for Task 4 (AFFINE).

4. Discussion

A trained model can extract two pieces of information from an image pair: (i) the matching score (*i.e.* how likely is this pair "the same"); (ii) an implicit encoding of the relationship between the two images in the form of the inferred hidden variables. In this paper, we focused on the first. Combining the score with the encoding of the relation, could be useful in geometry tasks, where it could help eliminate false matches and thus potentially speed up iterative schemes, such as RANSAC, and is an interesting direction for further research. Another interesting direction for future research is the introduction of sparsity, for example by adopting the approach in [13].

The binary hidden units in the model could be stacked to produce deep architectures in the same way that deep belief nets [6] are constructed. This would give rise to a hierarchical model of image transformations, analogous to a hierarchical factored model of natural images proposed by [19], but more powerful as it would represent spatiotemporal geometry rather than spatial geometry alone. This ability to stack nonlinear transformation modules is an advantage over bilinear models, where the hidden units are linear. It is also an advantage relative to standard optical flow models developed to track face deformations [14], which lack any internal feature representational layer. It could be the case that optical flow in the right feature space would work well for representing face transformations; however, the 3-way model already solves the problem of what the features are and how they transform within the framework of a single unsupervised learning procedure.

Acknowledgments

This research was funded by grants from NSERC, CFI, CIFAR and the German Federal Ministry of Education and Research (BMBF), Bernstein Fokus Neurotechnologie, Frankfurt Vision Initiative 01GQ0841, as well as by gifts from Google and Microsoft.

References

[1] B. Babenko, P. Dollár, and S. Belongie. Task specific local region matching. In *ICCV*, Rio de Janeiro, 2007. 2793

[2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of Computer Vision and Pattern Recognition Conference*. IEEE Press, 2005. 2793

[3] P. Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3):169–200, 1992. 2797

[4] D. Grimes and R. Rao. Bilinear sparse coding for invariant vision. *Neural Computation*, 17(1):47–73, 2005. 2794

[5] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:2002, 2000. 2794, 2795

[6] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006. 2799

[7] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990. 2796

[8] G. B. Huang and E. Learned-Miller. Learning class-specific image transformations with higher-order Boltzmann machines. In *In Workshop on Structured Models in Computer Vision at IEEE CVPR, 2010*. 2794, 2795

[9] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009. 2797

[10] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2009. 2797, 2798

[11] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, New York, NY, USA, 2007. ACM. 2797

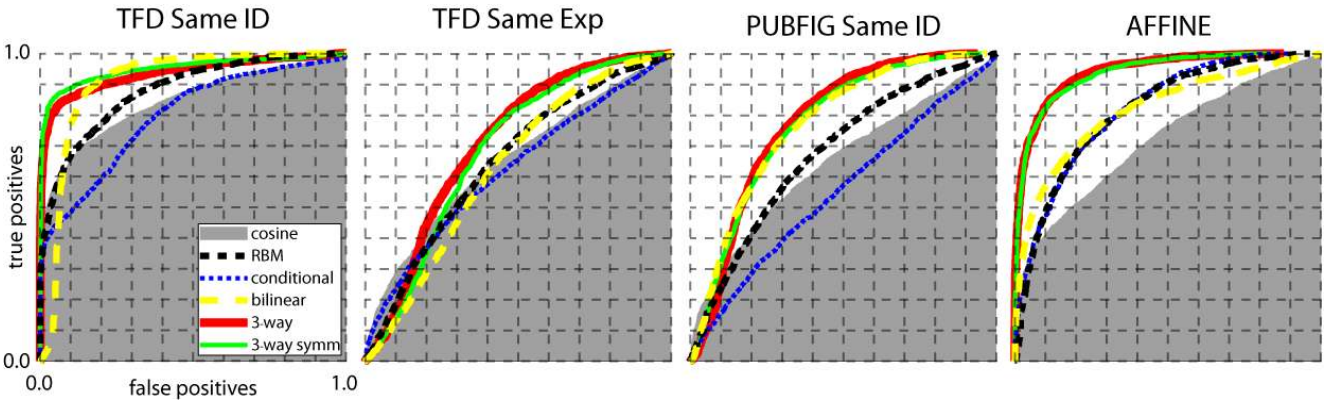

Figure 6. ROC curves on four matching tasks.

[12] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. 2798

[13] H. Lee, C. Ekanadham, and A. Ng. Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems 20*. MIT Press, 2008. 2799

[14] A. M. Martinez. Matching expression variant faces. *Vision Research*, pages 1047–1060, April 2003. 2799

[15] R. Memisevic and G. Hinton. Unsupervised learning of image transformations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 2794

[16] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 22(6):1473–92, 2010. 2794, 2795, 2796, 2797, 2798, 2799

[17] R. Memisevic, C. Zach, G. Hinton, and M. Pollefeys. Gated softmax classification. In *Advances in Neural Information Processing Systems 23*. 2010. 2794

[18] B. Olshausen, C. Cadieu, J. Culpepper, and D. Warland. Bilinear models of natural images. In *SPIE Proceedings: Human Vision Electronic Imaging XII*, San Jose, 2007. 2794

[19] M. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2795, 2796, 2799

[20] J. M. Susskind, A. K. Anderson, and G. E. Hinton. The Toronto face database. Technical report, Toronto, ON, Canada, 2010. 2797, 2798

[21] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proc. European Conference on Computer Vision (ECCV'10)*, 2010. 2794, 2795

[22] Y. W. Teh and G. E. Hinton. Rate-coded restricted Boltzmann machines for face recognition. In *Advances in Neural Information Processing Systems*, volume 13, 2001. 2797, 2798

[23] J. Tenenbaum and W. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000. 2794

[24] B. Theobald, I. Matthews, M. Mangini, J. Spies, T. Brick, J. Cohn, and S. Boker. Mapping and manipulating facial expression. *Lang Speech*, 52. 2794, 2798

[25] L. van Hateren and J. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings: Biological Sciences*, 265(1412):2315–2320, 1998. 2797

[26] E. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *In Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)*. AUAI press, 2005.

[27] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. *MIT Press*, pages 505–512, 2002. 2793