23

Generating Facial Expressions with Deep Belief Nets

Joshua M. Susskind[1,2,3], Geoffrey E. Hinton[2],
Javier R. Movellan[3] and Adam K. Anderson[1]
[1]*Department of Psychology, Univ. of Toronto,*
[2]*Department of Computer Science, Univ. of Toronto,*
[3]*Institute of Neural Computation, Univ. of California at San Diego*
[1,2]*Canada*
[3]*USA*

1. Introduction

Realistic facial expression animation requires a powerful "animator" (or graphics program) that can represent the kinds of variations in facial appearance that are both possible and likely to occur in a given context. If the goal is fully determined as in character animation for film, knowledge can be provided in the form of human higher-level descriptions. However, for generating facial expressions for interactive interfaces, such as animated avatars, correct expressions for a given context must be generated on the fly. A simple solution is to rely on a set of prototypical expressions or *basis shapes* that are linearly combined to create every facial expression in an animated sequence (Kleiser, 1989; Parke, 1972). An innovative algorithm for fitting basis shapes to images was proposed by Blanz and Vetter (1999) . The main problem with the basis shape approach is that the full range of appearance variation required for convincing expressive behavior is far beyond the capacity of what a small set of basis shapes can accommodate. Moreover, even if many expression components are used to create a repertoire of basis shapes (Joshi, Tien, Desbrun, & Pighin, 2007; Lewis, Matt, & Nickson, 2000), the interface may need to render different identities or mixtures of facial expressions not captured by the learned basis shapes. A representation of facial appearance for animation must be powerful enough to capture the right constraints for realistic expression generation yet flexible enough to accommodate different identities and behaviors. Besides the obvious utility of such a representation to animated facial interfaces, a good model of facial expression generation would be useful for computer vision tasks because the model's representation would likely be much richer and more informative than the original pixel data. For example, inferring the model's representation corresponding to a given image might even allow transferring an expression extracted from an image of a face onto a different character as illustrated by the method of expression cloning (Noh & Neumann, 2001).

In this chapter we introduce a novel approach to learning to generate facial expressions that uses a deep belief net (Hinton, Osindero, & Teh, 2006). The model can easily accommodate different constraints on generation. We demonstrate this by restricting it to generate

expressions with a given identity and with elementary facial expressions such as "raised eyebrows." The deep belief net is powerful enough to capture the large range of variations in expressive appearance in the real faces to which the net has been exposed. Furthermore, the net can be trained on large but sparsely labeled datasets using an unsupervised learning approach that employs an efficient contrastive form of Hebbian learning (Hinton, 2002). The unsupervised approach is advantageous because we have access to large corpuses of face images that are only sparsely labeled. Furthermore, since the human brain learns about faces through exposure in addition to explicit linguistic labeling, the unsupervised approach may lead to a better understanding of how the brain represents and processes faces for expression interpretation. It is unlikely that neural representations are learned by ignoring everything in the facial signal other than what correlates with occasional linguistic labels, because the labels do not provide enough information to organize a flexible and powerful representation of the face. The deep belief net approach to facial expression generation should be of interest to neuroscientists and psychologists concerned with facial expression representation in the brain because the multiple layers of representation that it uses are all learned from the data rather than being pre-specified.

2. Strategies for facial expression generation

A good criterion for determining the usefulness of a facial expression animation program is whether generation can be controlled easily. The challenge is finding a class of generative model that is powerful enough to generate realistic faces but simple enough to be learned from sparsely labeled data. Assume for a moment that we have access to a facial animation program with sensible controls, some face images, and a corresponding set of labeled data representing the controls the animation program would need to generate those images. For example, faces can be labeled using the Facial Action Coding System (FACS), which encodes expressions in terms of configurations of facial muscles and associated changes to the surface appearance (Ekman & Friesen, 1978). FACS is a kind of universal grammar for faces that describes the many different patterns of muscle actions that faces can express. FACS-based face models have been used to control facial animation (e.g. Wojdel & Rothkrantz, 2005). Currently, state of the art methods for realistic facial animation used in video games and feature films use FACS to drive models derived from motion capture data (Parag, 2006). However, this performance-driven approach to facial animation requires more information than images and labels, including motion capture technology, extensive calibration, and processes to clean data prior to modeling. This may be infeasible in most cases where we only have face images and associated high-level animation labels. With a sufficient number of FACS-labeled images, we could learn to control our animation program to do various tasks such as mimic face images by inferring the latent variables that control the generative model given an image and then generating a reconstruction from the model. However, learning the required nonlinear mapping from pixels to animation controls is likely to be a difficult problem requiring huge amounts of data and processing time. For instance, varying the intensity of a smile has highly nonlinear effects on pixel intensities. It is an even greater challenge to tailor the animation program to be flexible enough to accommodate arbitrary facial appearance. Rather than starting with an existing face model or using human animation knowledge to develop a complicated animation program, in this chapter we will learn to animate faces by training a type of general-purpose generative model on many examples of faces and associated high-level descriptors including FACS and identity labels.

A widely used technique for learning face structure from images is principal component analysis (PCA). PCA is a dimensionality reduction method that can be used to extract components from face images for use in face recognition (Turk & Pentland, 1991) and expression coding (Calder, Burton, Miller, Young, & Akamatsu, 2001). PCA is the optimal *linear* method for data compression when measured using squared error (provided we ignore the cost of coding the projections onto the components and we force the dimensions to be orthogonal). PCA, however, may ignore subtle, low-contrast cues in the interior of a face image, especially if the contrast between the face and the background is large, so that very accurate reconstruction of the boundary location is essential for minimizing squared error. A much more powerful method can be constructed using a twist on the standard PCA approach that factors faces into separate shape and texture sources. The active appearance model (AAM) is one such technique that uses information about the positions of facial feature landmarks (i.e. eyebrows, eyes, nose, mouth and facial contour) to warp face pixels to a standard reference frame (Cootes, Edwards, & Taylor, 1998). In this model, PCA is applied separately to the facial landmark coordinates and the shape-normalized pixels. The high-level controls are latent variables that linearly combine the feature coordinates and the texture map. To produce face images from a given vector of latent variables, texture and feature vectors are extracted and the shape-normalized textures are nonlinearly warped from the reference frame to the feature locations specified in the shape vector. This is a more sensible mapping from latent variables to pixels because faces can be modeled very accurately. The overall mapping is highly nonlinear even though variables controlling texture have linear effects on shape-normalized pixels, and variables controlling shape have linear effects on feature coordinates.

While appearance modeling approaches including AAMs have been used for facial expression recognition and generation (Abboud & Davoine, 2005) their applicability is limited in a number of ways because they are restricted to hard-coded transformations of images into sources such as shape, texture or lighting. Furthermore, as with standard PCA, the generative process in this type of model is deterministic and relies on heuristics such as selecting a number of model parameters using percentage of variance cutoffs. A more fundamental problem is the major cost of hand-annotating facial features to create a shape model. Due to the reliance on manual annotation, it is difficult to extend the representational capacity of trained AAMs to model additional sources of variance such as new identities or expression types. Likewise, it is difficult to make use of unlabeled data during training because feature points are not provided. Finally, while appearance models can generate realistic facial expressions spanning the kinds of variations common in the training set, fitting the model to test data is a separate problem. In an AAM, computing the underlying representation of test faces involves a search scheme (Cootes et al., 1998). The search scheme requires an initial "guess" for the location of the face in the image and it uses an iterative refinement procedure for uncovering the underlying model representation that can often fail if the guess is not already almost correct.

Another strategy is to treat facial expression generation as an unsupervised density estimation problem without the linear restrictions of standard PCA. If we have a large source of data, we can use it to learn a model of faces even if most of the images are not labeled; however, we need a good objective for adjusting model parameters. The key assumption is that there is rich structure in face images that can be uncovered without requiring labeled training data. One objective for unsupervised learning is optimal

reconstruction from a reduced code space. PCA is a linear example of this type of unsupervised approach. It learns codes for face images by finding a subspace of a particular dimensionality that minimizes squared reconstruction error. However, beyond the problems with PCA and the associated appearance model techniques mentioned above, squared reconstruction error is not always a perceptually meaningful measure. For instance, two different views of the same person are perceptually more similar than two different people in the same view even though measuring squared error of the pixels would suggest the opposite (Prince & Elder, 2005). Thus, although minimal reconstruction error may be an obvious objective for unsupervised learning of facial structure, it may not always be the most useful. This is especially true if our goal is to generate plausible expressions for a given context rather than to "mimic" expressions. Moreover, if the purpose is not to compress data, but to develop a good animation model, our objective should be to learn good "causes" for faces that lead to sensible generation of face images. If we have a good model for how to generate face images, those causes can be used for other tasks such as mapping image causes to high-level labels or driving an animation program.

A recent breakthrough in machine learning makes it relatively easy to learn complex, probabilistic, nonlinear generative models of faces using *deep belief nets* (Bengio, Lamblin, Popovici, & Larochelle, 2007; Hinton et al., 2006). A deep belief net (DBN) is a generative model that uses multiple layers of feature-detecting neurons. The net learns to represent plausible configurations of features (Hinton, 2007b). For example, a DBN could model the useful property of faces that the eyes are always situated above the nose. Given reasonable training data, the net would be highly surprised by a new face in which all the features were face-like but the eyes were below the nose. DBNs have been used successfully to learn generative models of handwritten digits (Hinton & Salakhutdinov, 2006), of natural image patches (Osindero & Hinton, 2008), and of millions of small color images (Torralba, Fergus, & Weiss, 2008). A logical extension is to apply DBNs to modeling facial expressions, thereby demonstrating the wide applicability of the approach to learning useful structure from complicated, high-dimensional data.

3. Using labels to control the generative model

Within the context of affective computing, it is important to be able to control a face animation program to output particular expressions, identities, or other domain specific attributes. However, prototypical examples of specific categories like happy, sad, or angry do not capture the full repertoire of expressive behaviors important for realistic interaction. In fact, thousands of distinct facial expressions have been catalogued (J. F. Cohn & Ekman, 2005). In our generative approach to expression modeling, we will learn a joint model of images, FACS labels, and identities. Once it has been learned, this model can generate an image with a pre-specified blend of identities and any desired combination of FACS labels. A key facet of our approach is the use of high-level descriptions, including identity and expressive facial action labels that provide rich information to usefully constrain the underlying representations used for generating image data.

Labeling facial expressions using FACS consists of describing expressions as constellations of discrete muscle configurations known as action units (AUs) that cause the face to deform in specific ways. While FACS can code muscle configurations that people commonly recognize as emotions such as anger or fear (Ekman & Rosenberg, 1997), it describes underlying anatomy rather than expression categories per se. This extends its usefulness in

tasks such as fine-grained detection of micromomentary expressions (subtle expressions appearing for mere microseconds) (Ekman & Rosenberg, 1997), detection of facial markers of deceit (Frank & Ekman, 1997), and characterization of spontaneous emotional behavior (Schmidt, Ambadar, Cohn, & Reed, 2006). Although FACS is a popular coding system, human-based coding of AUs is a labor intensive process requiring significant expertise, especially when applied to tasks such as labeling huge image datasets or sequences of video frames. Accordingly, developing an automated method for FACS labeling is an important challenge for computer vision and machine learning. Existing automated methods for FACS labeling rely on pre-processing expression data using expert knowledge of facial features (J. Cohn, Zlochower, Lien, & Kanade, 1999), or supervised feature selection methods (M.S. Bartlett et al., 2006). One obstacle to high quality automatic FACS labeling is that only a small number of datasets with coded AUs are available publicly; yet there exist many images of faces that could be used to develop an automated model if there was a sensible way to make use of this additional unlabeled data. By using a generative approach to expression modeling, we can learn useful image structure from huge numbers of faces without the need for many labeled examples. This approach enables us to learn associations between FACS labels and image structure, but is not limited only to these associations. This is important because FACS labels alone do not code additional attributes for realistic animation such as identity characteristics or fine surface texture changes.

Although faces are complex objects with often subtle differences in appearance, deep belief nets can be applied to learn a representation of face images that is flexible enough for animating as well as visually interpreting faces. State-of-the-art discriminative performance was recently achieved using DBNs as a pretraining method for handwritten digit recognition (Hinton et al., 2006) and for determining face orientation from images (Salakhutdinov & Hinton, 2008). In this chapter we apply an analogous method to learning a model for facial expressions. The DBN approach is capable of generating realistic expressions that capture the structure of expressions to which it is exposed during training, and can associate the high-level features it extracts from images with both identity and FACS labels.

4. Learning in deep belief nets

Images composed of binary pixels can be modeled by using a "Restricted Boltzmann Machine" (RBM) that uses a layer of binary feature detectors to model the higher-order correlations between pixels. If there are no direct interactions between the feature detectors and no direct interactions between the pixels, there is a simple and efficient way to learn a good set of feature detectors from a set of training images (Hinton, 2002). We start with zero weights on the symmetric connections between each pixel i and each feature detector j. Then we repeatedly update each weight, w_{ij}, using the difference between two measured, pairwise correlations

$$\Delta w_{ij} = \varepsilon\left(<s_i s_j>_{data} - <s_i s_j>_{recon}\right) \tag{1}$$

where ε is a learning rate, $<s_i s_j>_{data}$ is the frequency with which pixel i and feature detector j are on together when the feature detectors are being driven by images from the training set and $<s_i s_j>_{recon}$ is the corresponding frequency when the feature detectors are being driven by reconstructed images. A similar learning rule can be used for the biases.

Given a training image, we set the binary state, s_j, of each feature detector to be 1 with probability

$$p(s_j = 1) = \frac{1}{1 + \exp(-b_j - \sum_{i \in \text{pixels}} s_i w_{ij})} \tag{2}$$

where b_j is the bias of feature j and s_i is the binary state of pixel i. Once binary states have been chosen for the hidden units we produce a "reconstruction" of the training image by setting the state of each pixel to be 1 with probability

$$p(s_i = 1) = \frac{1}{1 + \exp(-b_i - \sum_{j \in \text{features}} s_j w_{ij})} \tag{3}$$

The learned weights and biases of the features implicitly define a probability distribution over all possible binary images. Sampling from this distribution is difficult, but it can be done by using "alternating Gibbs sampling". This starts with a random image and then alternates between updating all of the features in parallel using Eq. 2 and updating all of the pixels in parallel using Eq. 3. After Gibbs sampling for sufficiently long, the net reaches "thermal equilibrium". The states of pixels and features detectors still change, but the probability of finding the system in any particular binary configuration does not.

A single layer of binary features is not the best way to model the structure in a set of images. After learning the first layer of feature detectors, a second layer can be learned in just the same way by treating the existing feature detectors, when they are being driven by training images, as if they were data (Hinton, 2007a). To reduce noise in the learning signal, the binary states of feature detectors (or pixels) in the "data" layer are replaced by their real-valued probabilities of activation when learning the next layer of feature detectors, but the new feature detectors have binary states to limit the amount of information they can convey. This greedy, layer-by-layer learning can be repeated as many times as desired. Provided the number of feature detectors does not decrease and their weights are initialized correctly, adding an extra layer is guaranteed to raise a lower bound on the log probability of the training data (Hinton et al., 2006). So after learning several layers there is good reason to believe that the feature detectors will have captured many of the statistical regularities in the set of training images and will constitute a good generative model of the training data.

After learning a deep belief net, perception of a new image is very fast because it only involves a feedforward pass through the multiple layers. Generation from the multilayer model is slower. At the end of the layer-by-layer training, the weight between any two units in adjacent layers is the same in both directions and we can view the result of training three hidden layers as a set of three different RBM's whose only interaction is that the data for the higher RBM's is provided by the feature activations of the lower RBM's. It is possible, however, to take a very different view of exactly the same system (Hinton et al., 2006). We can view it as a single generative model that generates data by first letting the top-level RBM settle to thermal equilibrium using alternating Gibbs sampling (which may take a very long time), and then performing a single top-down pass to convert the binary feature activations in the penultimate layer into an image. In the top-down, generative direction, the weights between the lower layers form part of the overall generative model, but in the bottom-up, recognition direction they are not part of the model. They are merely an efficient way of inferring what hidden states probably caused the observed image.

5. A deep belief net for facial expressions

5.1 Facial expression dataset

In order to learn a generative model from a large and varied corpus of faces, we combined datasets that capture a significant degree of expression variation. Spontaneous expressions were collected during interviews in which participants were either deceptive or truthful (M. S. Bartlett et al., 2005). Additionally, a mixture of spontaneous and posed facial actions were collected from subjects in the MMI database (Pantic & Rothcrantz, 2000). Finally, posed facial actions were collected from the Cohn-Kanade FACS database (Kanade, Cohn, & Tian, 2000), the Ekman and Hager directed facial actions set (M.S. Bartlett, Hager, Ekman, & Sejnowski, 1999), and the Pictures of Facial Affect database (Ekman & Friesen, 1976). Identity labels accompany almost all faces. A subset of the data was coded by expert human raters, providing FACS labels for training the model to associate AUs with image features.

5.2 Preprocessing

We extracted over 100,000 face patches from the combined datasets using an automatic face detector (I. Fasel et al., 2004), which extends a previous approach (Viola & Jones, 2001). Modifications include employing a generative model framework to explain the image in terms of face and non-face regions, Gentleboost instead of Adaboost for feature selection, estimated eye and mouth corner feature detection, and a cascading decision procedure (I. R. Fasel, Fortenberry, & Movellan, 2005). Face patches were resized and cropped to 24x24 pixels. We then randomly selected 30,000 unlabeled faces and 3,473 labeled faces from the pool of detected face patches (see Table 1 below). The 8 AUs chosen for this experiment are common facial actions representing changes to the top and bottom half of the face (see Figure 4 below). The AUs representing the top half of the face are AU 1 and AU 2, which code inner, and outer eyebrow raises, respectively, AU 4, which codes brow lowering, and AU 5, which codes upper eyelid raise. The AUs representing the bottom half of the face are AU 10, which codes for upper lip raise, AU 12, which codes for lip corner raise, AU 14, which codes for cheek dimpling, and AU 20, which codes for horizontal mouth stretching.

		FACS Action Units (AUs)								
Faces	**ID**	**1**	**2**	**4**	**5**	**10**	**12**	**14**	**20**	**--**
3,473	151	0.28	0.22	0.15	0.08	0.08	0.18	0.07	0.05	0.35
27,863	205	--	--	--	--	--	--	--	--	--

Table 1. Faces, unique identities, and AU labels in the dataset. The first row describes faces with labeled action units, including the number of unique faces, and identities, and the proportion of labeled faces displaying a particular AU. The second row indicates the number of unique faces and identities in the larger set of faces with missing AU labels.

The extracted face images exhibited a large range of lighting conditions because of the differences in lighting control across datasets. To avoid learning lighting features at the expense of face details, pixel brightness was first normalized within and then across faces. First, all pixel values in a given face image were standardized across all pixels (i.e. separate linear transformations for each face). Then, each pixel value was normalized across faces to unit variance (i.e., each pixel intensity was divided by a separate constant). Finally, pixels that were beyond ±3 standard deviations from the average pixel brightness were truncated, and the images were rescaled to range between [0-255]. These preprocessing steps produced

symmetric brightness histograms that were roughly normally distributed with minimal preprocessing artifacts.

To facilitate training the first-level RBM with binary visible units, which are easiest to train, a heuristic procedure was used to convert brightness normalized face images into "soft" binary images. The method involved: (1) stretching the brightness-normalized face images using a hand-tuned logistic function resulting in contrast-enhanced pixel values, maintaining a range between [0-255], and (2) multiplying the pixels by 2 and truncating values exceeding 255. The ensuing images had dark edge features and bright regions elsewhere and retained perceptually important identity and expression attributes (see Figure 1a).

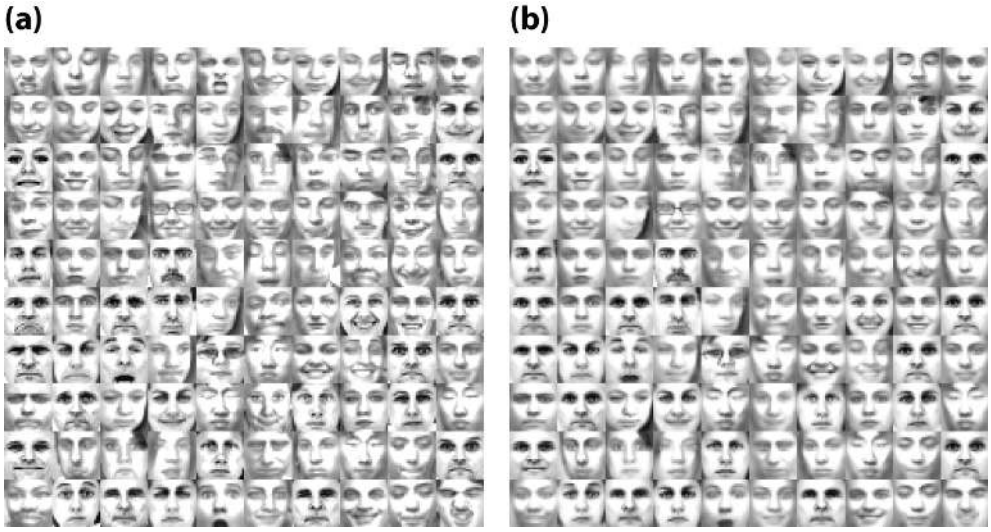

Figure 1. (a) Randomly selected soft binary training images. (b) RBM reconstructions (probabilities are shown instead of binary samples to produce smoother images).

To see how much critical information about expression is lost by the soft binarization procedure, two different neural nets were trained with backpropagation to discriminate FACS labels from real-valued versus soft binary images. The performance was slightly worse for the net trained using soft binary images (see Appendix).

5.3 Net architecture

Figure 2 depicts the deep belief net used to model the joint distribution of face images, identities, and FACS labels. In this model, 576 soft binarized pixel inputs are connected to a hidden layer of 500 logistic units, which is connected to a second layer of 500 hidden units. The penultimate hidden layer is then concatenated with identity and FACS label units, and the resulting vector serves as the visible layer of a top-level RBM with 1000 logistic hidden units. During training, each ascending pair of layers in the deep belief net is trained as an RBM, using the hidden activities computed from the previous RBM below as visible units for training the next RBM. After greedy layer-wise training, the complete net forms a hybrid model consisting of directed connections between lower layers and an undirected

associative memory at the top (Hinton et al., 2006). This top-level associative memory binds features and labels, and can thus be sampled by letting the RBM settle on likely configurations of features and label units. To generate from the net, the FACS AU label units in the penultimate layer can be clamped to particular values, and the associative memory can be sampled to find features that satisfy the label constraints, or the features can be clamped to values computed from an up-pass starting from an image, and the associative memory will fill in FACS labels. Given a particular configuration of features in the penultimate layer, the generative directed connections (pointing from higher to lower layers) convert the deep layers of feature activities into observed pixel face images.

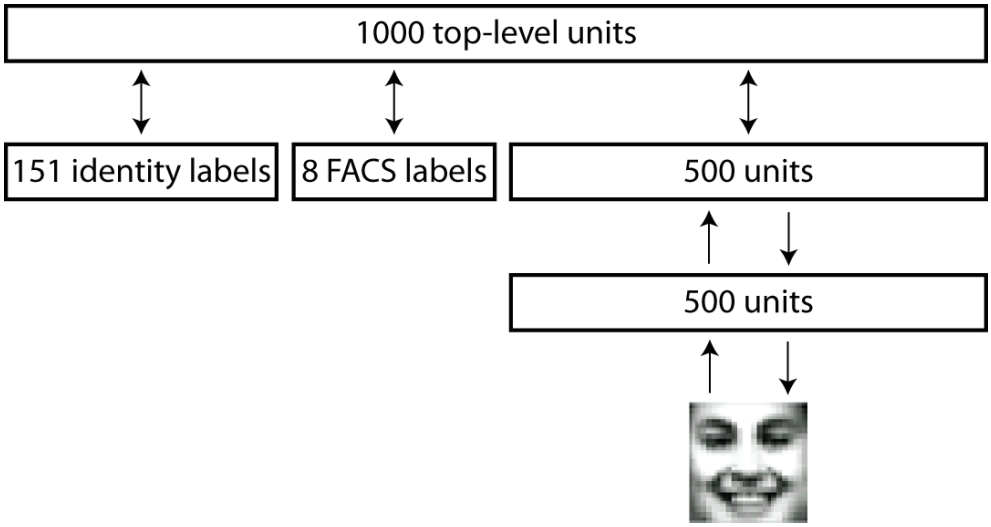

Figure 2. Architecture for deep belief net trained to generate facial expressions, identity labels, and FACS labels.

5.4 Greedy layer-wise training of the deep belief net

For training each RBM, the biases and weights connecting visible to hidden units were initialized to randomly sampled values from a normal distribution (μ=0, σ =.03). Training consisted of multiple passes through a full set of minibatches of 100 visible vectors with the weight being updated after each minibatch. To encourage the hidden layers to develop sparse representations, a penalty term was added to the weight updates for the first and second-level RBMs to pressure features to turn on 20% of the time. To reduce over-fitting, a weight decay of .00005 was used.

The first-level RBM connecting pixel visible units to 500 hidden units was trained for 200 epochs through the training data. A learning rate of .01 was used for the visible-to-hidden connections and .05 was used to update the weights on the visible and hidden biases. Figure 3 below shows receptive fields of some of the features learned by the first-level RBM. Since all the faces in the dataset were roughly aligned and scaled based on a consistent face and eye detection scheme, the positions of local features learned by the RBM tended to correspond to recognizable face parts, often characterizing local receptive fields comprising the eyebrows, eyes, cheeks, nose, or mouth.

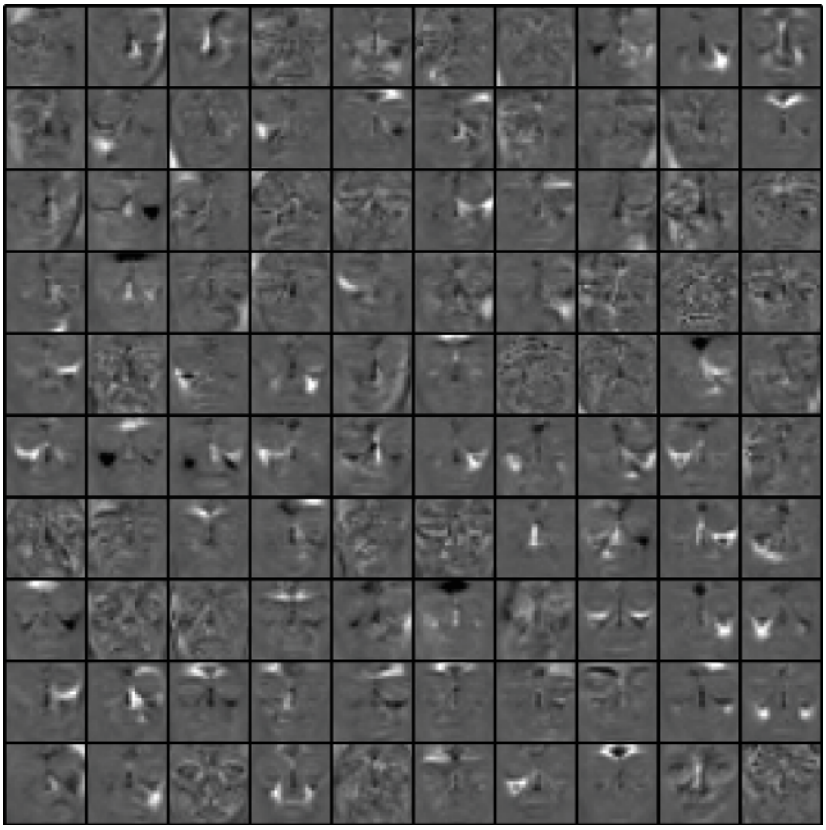

Figure 3. Receptive fields of some features learned by the first-level RBM. White indicates a positive weight to a pixel and black indicates a negative weight. Many features contain moderately to highly local receptive fields, indicating componential structure useful for representing distinct features and edges. Other features are more global.

Even though the first-level RBM was trained completely unsupervised, it learned useful structure in its features relating to different FACS AU label units.

Figure 4 shows a subset of features that correlate the most with different AUs. One interesting set of features includes salient positive weights to the whites of the eyes and negative weights to the pupils for detecting AU 5 (which codes for raised upper eyelids). These detailed features were likely learned because the automatic face detector aligns face patches to have roughly the same eye positions. Also evident from

Figure 4 is that some action units correlate with the same features because the changes in facial anatomy overlap. For instance, the same wide-eyed feature is highly correlated with AUs 1, 2, and 4, which code for raised inner brow, raised outer brow, and raised upper eyelid, respectively. Similarly, AUs 10 and 12, which code upper lip raise and upturned mouth corners, both correlate negatively with bright regions lateral to the mouth corners. A raised upper lip is a typical feature of disgust faces while upturned lip corners relate to

smiling; both of these actions may occur along with activation of the Zygomaticus muscle, serving to raise the cheeks, which leaves darker creased regions below.

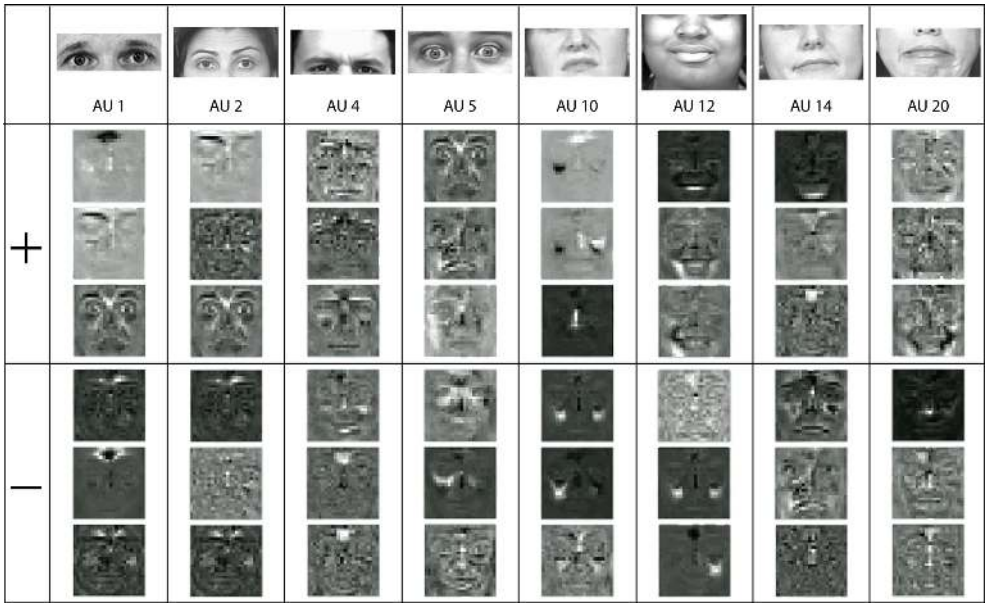

Figure 4. Unsupervised features (rows) that are highly correlated with particular action units (columns). The first 3 rows show positively correlated features with a particular AU, and the bottom 3 rows show negatively correlated features.

The visible units of the second-level RBM were initialized during training to the hidden probabilities computed by the first-level RBM after training. The second-level RBM was also trained for 200 epochs through the training data using the same learning rates and sparsity targets as were used to train the first-level RBM.

5.5 Learning the joint distribution of FACS labels and penultimate features

After training the second-level RBM, its feature activities were concatenated with discrete label units representing both identity and AU labels. The combined vector then became the visible units of the top-level RBM. Since a face image can only be associated with a single identity, a 1-of-K "softmax" coding scheme was used to represent the identities. Once binary states have been sampled for the hidden units, we generate identity labels by setting the state of each identity element to be 1 with probability

$$p(ID_i = 1) = \frac{\exp(b_i + \sum_{j \in \text{features}} s_j w_{ij})}{\sum_{k \in \text{identities}} \exp(b_k + \sum_{j \in \text{features}} s_j w_{kj})} \tag{4}$$

On the other hand, more than one FACS unit can be active for a face. Thus, the AU label vector comprises independent binary units for each of the 8 AUs. AU labels and feature activities are both generated independently for each unit using Eq. 3.
The third-level RBM was trained using N-step contrastive divergence (Carreira-Perpignan & Hinton, 2005) in 100 epoch sets using CD-3, CD-5, CD-7, CD-9, and CD-10 with the learning rate annealed in step increments from .001 to .0005 across the 500 epochs, and a weight decay of .00005. Training continued at CD-10 for a total of 4000 epochs.

5.6 Expression generation

After training, the deep belief network was tested as a face animation program by generating faces given different configurations of identity and AU labels. To generate from the DBN, one or more identity and/or AU labels are first clamped to particular values (where 0 = "off" and 1 = "on"). Next, the remaining visible units of the top-level RBM are sampled randomly according to their bias terms. This initializes the visible data vector for the top-level RBM to a reasonable unbiased starting point. Next, alternating Gibbs sampling is run for 1,000 steps, after which it is assumed the network has settled close to its equilibrium distribution given the clamped labels. Then, a single top-down pass converts the binary feature activations in the penultimate layer into an image consistent with the sample from the top-level RBM.
An innovative facial animation program would allow a user to specify some diffuse attributes, such as facial actions and/or identities, without requiring very specific controls. In other words, one should be able to specify a high-level description to the animation program without specifying every detailed feature contributing to that composition. The trained DBN is capable of this type of high-level control over face generation. Figure 5 below shows examples of the DBN generating faces after specifying a particular facial action unit. Although the network is capable of highly specific combinations of facial actions, such as all AUs off except for raised eyebrows, here we allow the net to determine its own combinations of facial actions given a single clamped unit. Thus, for example, when AU 1 is on (inner brow raise), the network often fills in other facial actions in addition to AU 1 such as AU 2 (outer brow raise), AU 4 (lowered brow), and AU 5 (raised eye lids). Note that AUs 1 and 4 can co-exist because there are multiple muscles involved in brow movement. The network's ability to generate combinations of AUs is evident in many other instances in Figure 5, such as the combination of AU 20 (horizontal lip stretcher) with AU 12 (raised lip corners), which is consistent with grinning, and AU 20 with AU 1 (inner brow raise), AU 2 (outer brow raise), and AU 5 (raised upper eye lid) which is characteristic of fear.
In addition to generating faces by specifying particular facial actions, the DBN can generate faces conforming to particular identities. Figure 6 below shows examples of the DBN generating faces after specifying a particular identity label. Since identity labels are "softmax" units, the network has learned during training not to blend identities. This is evident in Figure 6, where most faces generated for a particular identity look like that identity. Since the DBN is capable of settling on different combinations of AUs given a particular identity label, the generated faces vary in expression, sometimes exhibiting multiple AUs. However, since not all identities in the training set exhibited all facial actions, some expressions occur more often for some identities than others.

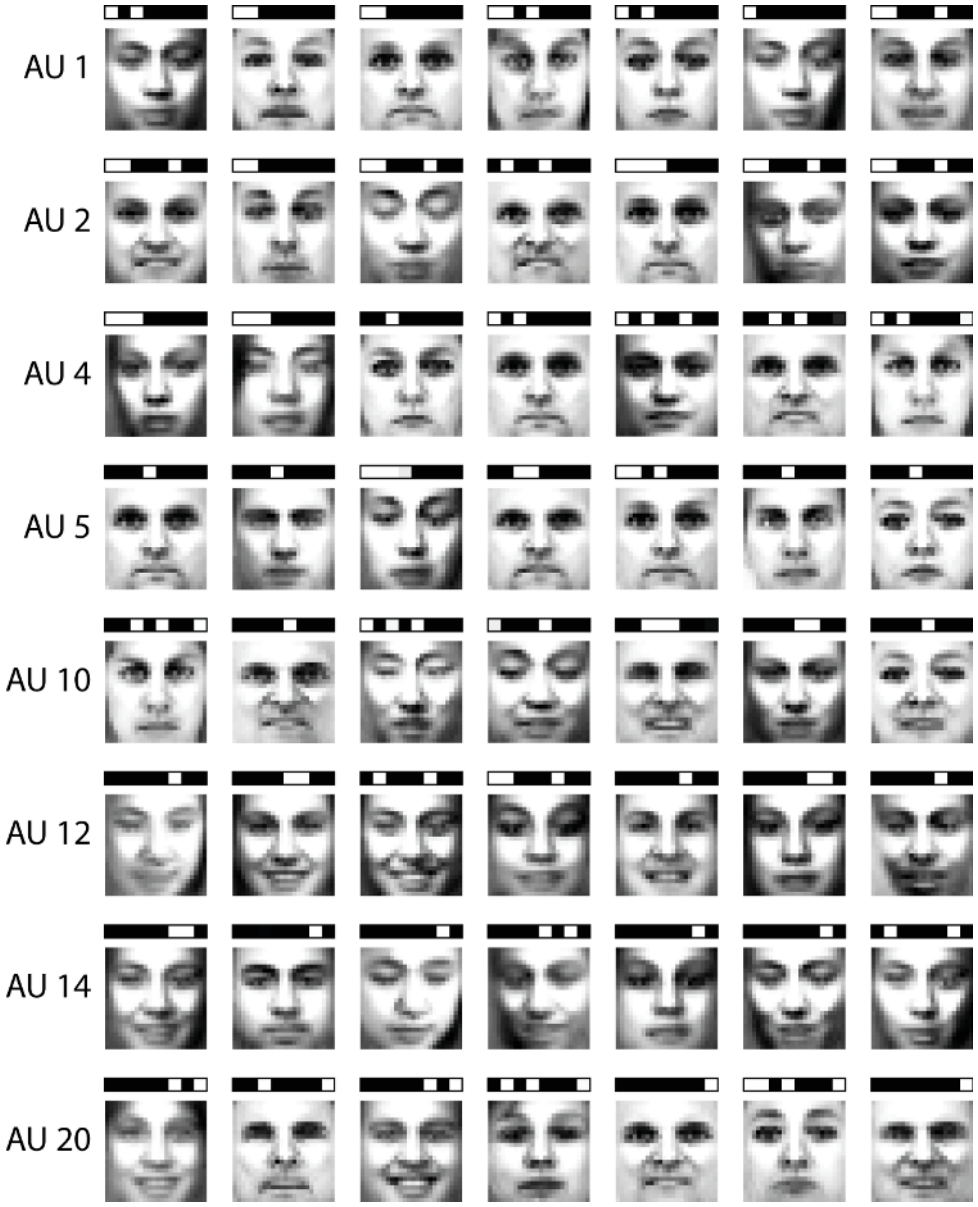

Figure 5. Face images sampled from the conditional distribution of features and AUs given an AU label. Each row shows 7 results after clamping a particular AU label to "on" and running alternating Gibbs sampling at the top-level RBM for 1000 iterations, and generating an image via a directed down-pass through the network, resulting in pixel probabilities observed at the visible image layer. Above each face image is the associated AU vector that the network settled on, indicating from left to right AUs 1, 2, 4, 5, 10, 12, 14, and 20.

Figure 6. Face images sampled from the conditional distribution of features and AUs given an identity label. Each row corresponds to faces generated with a particular identity label clamped on. The first column shows example faces from the training set representative of the clamped identity for the corresponding row. The samples vary in expression in ways representative of the expressions posed by that identity in the training set.

The DBN is also capable of generating a face given both a set of AUs and a specific identity. This is an important ability for an animation program to possess because often the intent is to animate the expressions of a particular individual. Figure 8 below shows examples of the DBN generating faces after clamping specific identity and AU labels. For example, the first row of Figure 8 shows 5 different examples of the same identity exhibiting AU 10 (upper lip raise), which is a facial action often associated with disgust. Sometimes the network also filled in other AUs such as AU 12, which can occur together with AU 10 during happiness. Note that the middle face in the first row appears to change identity even though the identity label is clamped. Since the DBN is stochastic, this will occasionally happen. The second row of Figure 8 demonstrates both a consistent identity and the likely co-occurrence of AU 1 (inner brow raise) with AU 2 (outer brow raise), and AU 5 (upper eye lid raise), which are combinations that often occur in conjunction with fear and surprise.

Finally, we demonstrate the DBN has the capacity to generate faces that are restricted to a subset of more than one identity. Occasionally in this case the network will generate blends of identities since the feature representation contains many local features consistent with both identities. Figure 7 below shows examples of the DBN generating faces after specifying two different identity labels with equal probability. These examples demonstrate that the

DBN is capable of generalizing beyond the training examples to create novel blends of identities that vary in expression.

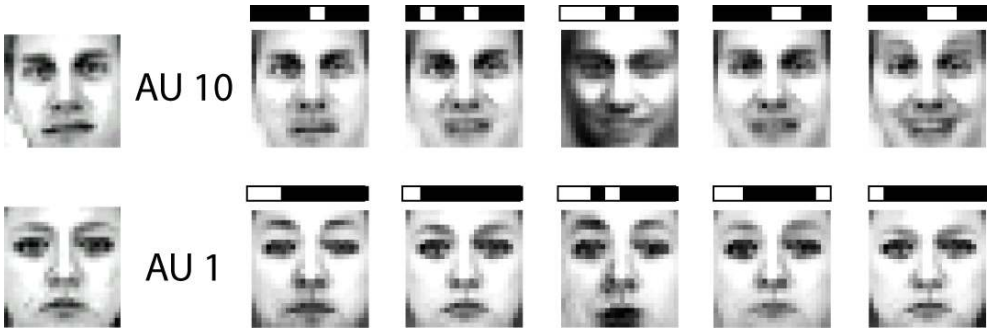

Figure 8. Face images sampled from the conditional distribution of features and AUs given both an identity label and an AU label. Each row corresponds to faces generated with both a particular identity label and an AU label clamped on. Row 1 shows that the network settles on faces congruent with a consistent identity label (corresponding to the face in column 1) that all exhibit variations on the upper lip raise (AU 10). Similarly, row 2 shows different variations on the inner brow raise (AU 1) consistent with the corresponding given identity.

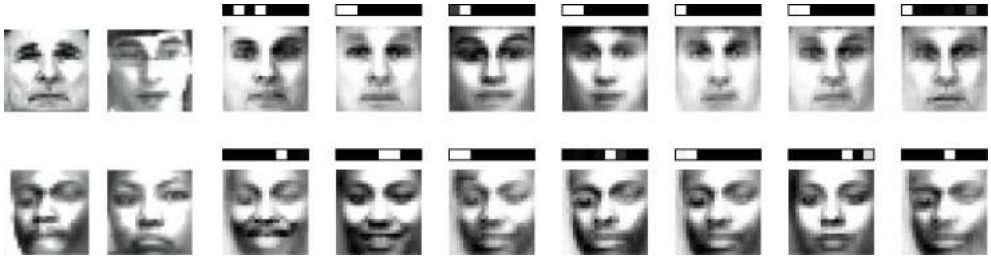

Figure 8. Face images sampled from the conditional distribution of features and AUs given a blend of two identity labels clamped on. Each row shows sample faces consistent with the identity labels for the two left-most faces in that row. Some faces are more consistent in visual appearance with one of the identities, while other faces seem to settle on blends of the two identities, indicating the identities contain compatible features.

6. Conclusion

In this chapter we showed that it is possible to train a deep belief net with multiple layers of features to function as an animation system capable of converting high-level descriptions of facial attributes into realistic face images. By specifying particular labels to the DBN, we were able to generate realistic faces displaying specific identities and facial actions. In addition, the DBN could generalize from the associations it learned during training to synthesize novel combinations of identities and facial actions. Thus, like the human brain,

the DBN can remember faces it has seen during training, can associate faces with particular identities, and can even "imagine" faces it has never seen before by blending identities and/or facial actions. By sampling from the DBN, we demonstrated that it is possible to investigate how a neural net represents faces and associates them with high-level descriptions. Samples that the DBN generates represent beliefs it has about faces. In particular, the top-level RBM acts as a constraint satisfaction network, finding sets of image features that the network considers likely to be associated with a given set of identities and action units. A question for future research is the extent to which the representations that the DBN learns resemble the neural representations used by humans. For instance, humans often confuse certain facial expressions like fear and surprise, presumably because these expressions share underlying muscle configurations and are thus visually similar (Dailey, Cottrell, Padgett, & Adolphs, 2002; Susskind, Littlewort, Bartlett, Movellan, & Anderson, 2007). Likewise, humans may confuse some identities more than others due to ways in which the faces are perceptually similar. Does the DBN capture human-like perceptual similarity? In order to answer this question we would need to measure how similarly the network represents different faces. One way this could be done is by correlating average feature activities in the DBN to different faces and comparing the degree of similarity between faces to human judgments of similarity.

Our DBN results demonstrate that different types of facial attributes can be represented by the same distributed set of image features, suggesting in particular that identity and expression are not entirely independent facial attributes. The current study did not attempt to investigate the interdependence of identity and expression directly, but the ability of the network to associate identities and facial actions with facial appearance suggests these different attributes can make use of the same distributed neural representation. One way to examine the relative interdependence of expression and identity in the network is to examine whether some facial actions are more likely to be generated given one identity label rather than another, which would indicate that expression depends on identity. The DBN can model the notion that different people smile in different ways, expressing the same facial action with different constellations of visual features. Some evidence that the brain encodes facial expression in an identity-specific manner comes from behavioral studies examining high-level facial expression adaptation to different identities (Ellamil, Susskind, & Anderson, in press; Fox & Barton, 2007).

The deep belief network approach demonstrates that given a large enough set of training data, a neural network can learn sensible representations of face images directly from image pixels, without requiring expert feature selection methods to pre-process the image. Although in this approach the DBN was trained to generate facial expressions given high-level identity and FACS AU labels, the representation of faces that it learned may also be useful for recognizing these and other expressive attributes when presented with a face image. In fact, after learning multiple layers of image features using RBMs, the DBN can be further fine-tuned for discriminating high-level labels from images using the backpropagation algorithm (Hinton & Salakhutdinov, 2006). However, the discriminative network would lose its capacity to generate faces. More appropriately for the purposes of facial animation, the DBN can be fine-tuned to recognize high-level descriptions of faces while maintaining its generative capacity to animate facial expressions using generative

fine-tuning methods such as a contrastive version of the wake-sleep algorithm (Hinton et al., 2006). The authors show that this approach works well for generating and recognizing hand-written digits. Although better generation and recognition performance might be achieved with fine-tuning, we have demonstrated that a relatively simple unsupervised learning algorithm can develop a powerful internal representation of faces.

7. References

Abboud, B., & Davoine, F. (2005). Bilinear factorisation for facial expression analysis and synthesis. *VISP, 152*(3), 327-333.

Bartlett, M. S., Hager, J., Ekman, P., & Sejnowski, T. (1999). Measuring facial expressions by computer image analysis. *Psychophysiology, 36*, 253-264.

Bartlett, M. S., Littlewort, G. C., Frank, M. G., Lainscsek, C., Fasel, I. R., & Movellan, J. R. (2006). Automatic Recognition of Facial Actions in Spontaneous Expressions. *Journal of Multimedia, 1*(6), 22-35.

Bartlett, M. S., Movellan, J. R., Littlewort, G., Braathen, B., Frank, M. G., & Sejnowski, T. J. (2005). Towards automatic recognition of spontaneous facial actions. In P. Ekman (Ed.), *What the Face Reveals*: Oxford University Press.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. In B. Scholkopf, J. Platt & T. Hoffman (Eds.), *Advances in Neural Information Processing Systems 19* (pp. 153-160). Cambridge, MA: MIT Press.

Blanz, V., & Vetter, T. (1999). *A morphable model for the synthesis of 3D faces*. Paper presented at the Proceedings of the 26th annual conference on Computer graphics and interactive techniques.

Calder, A. J., Burton, A. M., Miller, P., Young, A. W., & Akamatsu, S. (2001). A principal component analysis of facial expressions. *Vision research, 41*(9), 1179-1208.

Carreira-Perpignan, M. A., & Hinton, G. E. (2005). On Contrastive Divergence Learning. *Artificial Intelligence and Statistics*.

Cohn, J., Zlochower, A., Lien, J. J., & Kanade, T. (1999). Automated face analysis by feature point tracking has high concurrent validity with manual FACS coding. *Psychophysiology, 36*.

Cohn, J. F., & Ekman, P. (2005). Measuring facial actions. In J. A. Harrigan, R. Rosenthal & K. Scherer (Eds.), *The New Handbook of Methods in Nonverbal Behavior Research* (pp. 9-64). New York, USA: Oxford University Press.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (1998, 1998). *Active Appearance Models.* Paper presented at the European Conference on Computer Vision.

Dailey, M. N., Cottrell, G. W., Padgett, C., & Adolphs, R. (2002). EMPATH: A Neural Network that Categorizes Facial Expressions. *Journal of cognitive neuroscience, 14*(8), 1158-1173.

Ekman, P., & Friesen, W. (1976). Pictures of facial affect.

Ekman, P., & Friesen, W. V. (1978). *The facial action coding system*. Palo Alto, California: Consulting Psychologists Press.

Ekman, P., & Rosenberg, E. L. (1997). *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. New York: Oxford University Press.

Ellamil, M., Susskind, J. M., & Anderson, A. K. (in press). Examinations of identity invariance in facial expression adaptation. *Cognitive, Affective, & Behavioral Neuroscience*.

Fasel, I., Dahl, R., Hershey, J., Fortenberry, B., Susskind, J. M., & Movellan, J. R. (2004). Machine perception toolbox, http://mplab.ucsd.edu/software/mpt.html.

Fasel, I. R., Fortenberry, B., & Movellan, J. R. (2005). A generative framework for real-time object detection and classification. *Computer Vision and Image Understanding, 98*.

Fox, C. J., & Barton, J. S. (2007). What is adapted in face adaptation? The neural representations of expression in the human visual system. *Brain Research, 1127*, 80-89.

Frank, M. G., & Ekman, P. (1997). The ability to detect deceit generalizes across different types of high-stake lies. *Journal of Personality and Social Psychology, 72*, 1429-1439.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*(8), 1711-1800.

Hinton, G. E. (2007a). Learning Multiple Layers of Representation. *Trends in Cognitive Sciences, 11*, 428-434.

Hinton, G. E. (2007b). To recognize shapes, first learn to generate images In P. Cisek, T. Drew & J. Kalaska (Eds.), *Computational Neuroscience: Theoretical Insights into Brain Function*.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(1527-1554).

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science, 313*(5786), 504-507.

Joshi, P., Tien, W., Desbrun, M., & Pighin, F. (2007). Learning Controls for Blendshape-based Realistic Facial Animation. In *Data-Driven 3D Facial Animation* (pp. 162-174).

Kanade, T., Cohn, J. F., & Tian, Y. (2000). *Comprehensive database for facial expression analysis*. Paper presented at the Proceedings of the fourth IEEE International conference on, Grenoble, France.

Kleiser, J. (1989). *A fast, efficient, accurate way to represent the human face.* Paper presented at the SIGGRAPH '89 Course Notes 22: State of the Art in Facial Animation.

Lewis, J. P., Matt, C., & Nickson, F. (2000). *Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation*. Paper presented at the Proceedings of the 27th annual conference on Computer graphics and interactive techniques.

Noh, J.-y., & Neumann, U. (2001). *Expression cloning*. Paper presented at the Proceedings of the 28th annual conference on Computer graphics and interactive techniques.

Osindero, S., & Hinton, G. E. (2008). *Modeling image patches with a directed hierarchy of Markov random fields*. Paper presented at the Advances in Neural Information Processing Systems 20.

Pantic, M., & Rothcrantz, J. M. (2000). Automatic analysis of facial expressions: State of the art. *IEEE Trans on Pattern Analysis and Machine Intelligence, 22*(12), 1424-1445.

Parag, H. (2006). *Sony Pictures Imageworks*. Paper presented at the ACM SIGGRAPH 2006 Courses.

Parke, F. I. (1972). *Computer generated animation of faces*Paper presented at the Proceedings ACM annual conference.

Prince, S. J. D., & Elder, J. H. (2005). Creating invariance to "nuisance parameters" in face recognition. *Computer Vision and Pattern Recognition*.

Salakhutdinov, R., & Hinton, G. (2008). Using Deep Belief Nets to Learn Covariance Kernels for Gaussian Processes. In J. C. Platt, D. Koller, Y. Singer & S. Roweis (Eds.), *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press.

Schmidt, K. L., Ambadar, Z., Cohn, J. F., & Reed, L. (2006). Movement differences between deliberate and spontaneous facial expressions: Zygomaticus major action in smiling. *Journal of Nonverbal Behavior, 30*, 37-52.

Susskind, J. M., Littlewort, G., Bartlett, M. S., Movellan, J., & Anderson, A. K. (2007). Human and computer recognition of facial expressions of emotion. *Neuropsychologia, 45*(1), 152-162.

Torralba, A., Fergus, R., & Weiss, Y. (2008). *Small codes and large image databases for recognition*. Paper presented at the Computer Vision and Pattern Recognition (CVPR-08).

Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience, 3*(1).

Viola, P., & Jones, M. (2001). *Robust real-time object detection.* Paper presented at the ICCV Second International Workshop on Statistical and Conceptual Theories of Vision.

Wojdel, A., & Rothkrantz, L. J. M. (2005). Parametric Generation of Facial Expressions Based on FACS. *Computer Graphics Forum, 24*(4), 743-757.

Appendix

In a control experiment to ensure no critical loss of information, we compared two different classifiers trained with backpropagation to predict FACS labels, using faces preprocessed with and without the soft binarization step. Results are shown below in
Table 2 for nets trained with 100 hidden units[1]. Area under the ROC curve was computed separately as an

Net	Area under ROC							
Architecture	**AU1**	**AU2**	**AU3**	**AU4**	**AU5**	**AU6**	**AU7**	**AU8**
MLP100	0.78	0.74	0.76	0.90	0.78	0.87	0.75	0.77
BINMLP100	0.77	0.74	0.76	0.88	0.76	0.87	0.75	0.75

Table 2. FACS classification results for feedforward nets trained with backpropagation, with and without the soft binarization preprocessing step (top and bottom row, respectively).

index of classification accuracy for each FACS label. The net trained with soft binarized inputs (BINMLP100) achieved comparable results to the net trained without this extra

[1] Separate nets were tested with 50-500 hidden units. The 100 hidden unit nets were optimal as assessed by error on the labeled validation cases.

preprocessing step (MLP100). In addition, Figure 1b shows reconstructions from a trained RBM showing that treating the set of soft binarized pixel intensities as Bernoulli probabilities is appropriate for capturing essential identity and expression features in the training images, even though the RBM does not optimize image reconstruction. These results indicate that the soft binarization step does not eliminate diagnostic expression features, which validates the use of binary visible units to train the first-level RBM.