Multi-Task Learning of Facial Landmarks and Expression

Terrance Devries[1], Kumar Biswaranjan[2], and Graham W. Taylor[1]

[1]School of Engineering, University of Guelph, Guelph, Canada N1G 2W1
[2]Department of Electronics and Electrical Engineering, IIT Guwahati, Guwahati, Assam India 781039

Abstract—**Recently, deep neural networks have been shown to perform competitively on the task of predicting facial expression from images. Trained by gradient-based methods, these networks are amenable to "multi-task" learning via a multiple-term objective. In this paper we demonstrate that learning representations to predict the position and shape of facial landmarks can improve expression recognition from images. We show competitive results on two large-scale datasets, the ICML 2013 Facial Expression Recognition challenge, and the Toronto Face Database.**

Keywords-**computer vision; representation learning; deep learning; expression recognition; emotion recognition; multi-task learning; convolutional neural networks; facial landmarks;**

I. Introduction

The ability to automatically recognize facial expression from images has been a long-standing goal of computer vision. The problem is typically cast as classification, specifically, recognizing one of a discrete number of expression classes. It is a step toward recognizing emotion and is a key consideration in fields such as human-computer and human-robot interaction. Despite having received much attention since the advent of robust face detection and tracking systems in the mid-1990's, the problem has remained challenging and still the focus of much research (see [1] for a recent survey of the field).

A central challenge in recognizing facial expression is that of untangling the many factors of variation that explain an image [2]. In particular, factors such as pose, subject identity (facial morphology), gender, and other visual appearance (e.g. facial hair, glasses) are intertwined with that of expression. In fact, these so-called nuisance factors tend to dominate the pixel-based representation of images: two images of the same individual with different expressions will likely lie much closer together in pixel space than two images of different individuals with the same expression.

To date, the dominant methodology for addressing this challenge has been to engineer a feature extraction pipeline, usually containing multiple stages of processing. The first stage of processing in a typical pipeline consists of extracting sets of low-level features such as SIFT [3], HoG [4], or responses to oriented Gabor filters from image patches. Next, these features are pooled over local spatial regions and sometimes across multiple scales to reduce the size of the representation and also develop local shift/scale invariance. Finally, the aggregate features are mapped to a vector which is then input to a standard classifier such as a support vector machine (SVM) or K-Nearest Neighbour (KNN). Much work is devoted to engineering the system such to produce a vector representation that is sensitive to the factor of interest (in our case, facial expression) while remaining invariant to the various nuisance factors.

An alternative approach is "representation learning": relying on the data, instead of feature engineering to *learn* a good representation, in this case, one that is invariant to nuisance factors. It is common to learn multiple layers of representation, which is referred to as "deep learning". Bengio et al. [5] provide a comprehensive review of this field of research. Several such techniques have used unsupervised or semi-supervised learning to extract multi-layer domain-specific invariant representations leading to success on challenging facial expression benchmark datasets [2], [6]. As [2] points out, one advantage to these techniques is that while feature engineering targets specific invariances (e.g. shift/scale invariance described above), feature learning can address *all* significant factors, including those that cannot be *a priori* identified.

In this paper, we exploit the ability of representation learning techniques to learn features that are good at accomplishing multiple tasks. Specifically we explore two synergistic tasks: recognizing facial expressions and localizing facial landmarks. Ultimately we are interested in the task of expression recognition, but we show that a system trained to reason about facial geometry while recognizing expressions outperforms a system trained to recognize expressions alone.

II. Related Work

Until recently, the availability of labeled facial expression datasets has been a major barrier to learning-based approaches. Benchmark datasets, such as those surveyed in [1] have typically been limited to a few subjects (less than 100) displaying the "basic" six expressions: happiness, sadness, anger, surprise, disgust and fear. Not only are the datasets historically small, the examples are "posed" rather than spontaneous expressions, which is clearly an issue for training systems that must generalize to performing "in the wild". The recent introduction of the Toronto Face Database (TFD) [7], which collects and normalizes a number of smaller databases has changed the field. In addition to thousands of images labeled with subject and identity, it contains more than 100,000 unlabeled images which makes it amenable to representation learning methods [6], [2], [8], a number of which we now describe.

Ranzato et al.'s feature learning approach [6] was one of the first to consider TFD. They learn features using a variant of Deep Belief Networks (DBNs) with a mean of product of Student's t (mPoT) front-end which is effective at modeling covariance in images. Their approach is essentially unsupervised feature learning to extract a good representation, followed by supervised learning on this representation.

Rifai et al. propose a semi-supervised "deep learning" approach which is evaluated on TFD [2]. In contrast to our approach, they focus explicitly on learning invariance, in particular, they split their higher-level features into a "discriminative" block, used for predicting emotion, and a "nuisance" block which captures the remaining factors of variation. While their method is convolutional, like ours, their filters are learned by an unsupervised method known as Contractive Auto-encoding (CAE) [9].

Tang won a recent competition on facial expression recognition held at the International Conference on Machine Learning (ICML) 2013 [8]. The focus of his work was on the use of the SVM hinge loss as an alternative to the more traditional softmax activation function with cross-entropy loss. We use his convolutional network architecture as a baseline and adopt similar training practices such as normalization and data augmentation.

Another application of deep learning to the facial expression domain has been in the context of *generating* facial expressions [10]. Recognition and generation are intimately related [11], though we are unaware of work in the facial domain which exploits this relationship.

Also related to the present work are methods which perform synergistic "mult-task learning" in the facial domain. There are a number of works which exploit the synergy between facial pose estimation and face detection, however we are not aware of any methods which consider expression recognition. Zhu et al. propose a model for face detection, pose estimation, and landmark estimation in real-world, cluttered images [12]. In fact, we use their approach to estimate the locations of landmarks in our facial expression datasets. Osadchy et al. apply a convolutional network to simultaneously detect and estimate the pitch, yaw and roll of a face [13]. Perhaps the most recognized example of multi-task learning from the deep learning community is Collobert and Weston's convolutional net-based framework for natural language processing [14], however, they work in a completely different domain than the one we consider.

III. Method

Our technique is based on a convolutional neural network (convnet). Convnets, like their standard deep neural network counterparts, perform end-to-end feature learning and are trained with the backpropagation algorithm. However, they differ in a number of respects.

First, each layer of a standard neural network can be represented in vector form. This is true for the input layer as well: when training on images, the inputs are vectorized. The convnet instead maintains 2d structure from the input through several layers of feature extraction. The hidden layers of the convnet are referred to as "feature maps" because of their 2d structure. Standard neural networks are *fully connected*, meaning that each unit in a given layer (starting with the inputs) is connected to each unit in the next layer. Convnets, however, use a form of parameter tying which massively reduces the total number of free parameters, making them suitable to model large images instead of just patches. A typical convnet layer consists of three components:

1) *filtering*, whereby the "feature maps" of the previous layer[1] are convolved with a series of filters. Each filter connects an input map to an output map. The 2d filter responses corresponding to a given output map are summed over all the input channels. Some convnets use sparse connectivity in which output maps sum filter responses over only a few input maps. Note that the filters are the trainable parameters of the convnet. The learned filters act as local receptive fields which respond to a specific input structure.
2) *an elementwise nonlinearity*, which is applied to the output maps. Popular choices for the nonlinearity are the logistic sigmoid, $tanh$, and rectified linear unit (ReLU) [15].
3) *downsampling*, whereby the output maps of a layer are downsampled using a simple operator (typically average or max) which operates on small non-overlapping windows.

Usually, after two or more repetitions of convolution, nonlinearity and pooling, the feature maps are vectorized and input to one or more "fully-connected" layers (i.e. those found in a typical multi-layer perceptron).

In the following section, we describe our convnet architecture for predicting facial expressions. We then describe how to extend this network to simultaneously capture facial landmark geometry.

A. Architecture

Our convnet is based on the winning architecture from the 2013 ICML Facial Expression Recognition Competition [8]. This network consisted of three convolutional layers with

[1] The "feature maps" at the first layer are simply the channels of the input. In the case of grayscale data, it is a single feature map.

full connectivity, a “fully-connected” ReLU hidden layer, and finally an output layer utilizing an L2SVM activation function. Augmented data was generated by randomly mirroring, rotating, zooming, or shifting the input images. The net also used several other well-known techniques such as adding Gaussian noise, momentum, and drop-out [16]. This net serves as our baseline to which we compare in Sec. IV.

Similar to [8], our net consisted of three fully connected convolutional layers, each one with a rectified linear activation function (ReLU) and max pooling. This was followed by a “fully-connected” ReLU hidden layer which fed the output layer. Unlike Tang we opted to use a top-level softmax layer and a negative log likelihood cost function instead of the L2SVM. We experimented with the L2SVM but found that it took nearly twice as long to train with no added performance benefit. A depiction of our architecture is given in Figure 1. More details of specific architectural settings are given in Sec. IV.

We preprocessed our data using the same pipeline as [8]. First we subtracted each image’s mean value and then set the image norm to be 150. The pixels were then standardized by subtracting the mean and dividing by the standard deviation of that pixel, across the entire dataset. Each minibatch was also normalized by subtracting from each image the batch mean and then dividing by the batch’s standard deviation. To generate additional examples the 48×48 images were randomly mirrored, shifted ± 3 pixels in either direction, and if the dataset contained a lot of variation in the face position, also rotated up to ± 45 degrees, and zoomed at up to $1.2\times$. The transformed images were then cropped to 42×42. In this way several different variations of a single image could be generated and subsequently processed by the net each epoch.

Following Krizhevsky et al. [17], at test time the input images were mirrored and then five 42×42 patches were extracted (corners and center) for a total of 10 variations per image. The output layer’s softmax output for the ten patches was then averaged, and the one with maximal value was chosen as the prediction.

B. Landmark Prediction Layer

In incorporating landmark prediction into our convolutional net architecture, we faced two main challenges: 1) obtaining labeled ground truth for facial landmarks for training and 2) with facial landmarks much more complicated than the 1 of K expression labels, determining the corresponding output representation. In fact, the form of facial landmark prediction performed by the net and its representation at the output depends on the type of labeled data available. Therefore we will first discuss the types of labels we obtained, and then how they were represented in the net.

To the best of our knowledge, there is no publicly available dataset which contains labeled facial expressions and landmarks. Therefore, we decided to re-label existing facial expression datasets with landmarks. Our motivation was that facial landmark prediction has been an active area of research, yielding several successful methods, many of which have publicly available implementations. We experimented with several techniques, but ultimately selected Zhu and Ramanan’s facial landmark detector [12] which returned the coordinates for 68 landmarks (at subpixel resolution) per face. These points traced facial features such as the face outline, the mouth, nose, eyes, and eyebrows.

Training the net to predict the precise location of each of 68 landmarks is not only challenging, but overkill for learning features of facial geometry that might be useful for emotion recognition. We simplified the problem by requiring the net to predict aggregate locations of major features most indicative of expression, namely the eyebrows and mouth. From the 68 landmark locations, we created coarse ground-truth “label maps” the same size of the image, which were predicted by “output maps” of the same size at the top layer of the network.

The raw landmark locations were used to create three binary mask images, corresponding to the locations of the left eyebrow, right eyebrow, and mouth (see Figure 2). The network’s output consisted of three binary output maps (each with logistic units). The net therefore was tasked with modeling the location and shape of each of these features. We used a mean per (output) pixel cross entropy error between the label maps and output maps. Figure 3 visualizes the landmark output layer.

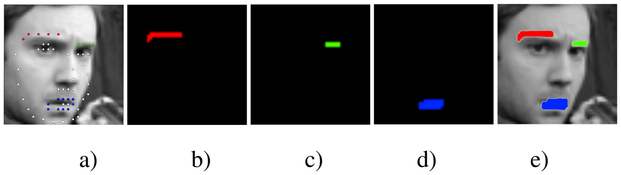

Figure 2. This sequence of images visualizes the approach we used to create labels for training the landmark prediction layer. From left to right: a) a face from the ICML dataset and its 68 facial landmarks obtained using [12]. Coloured markers indicate landmarks used in creating aggregate landmarks; b) the left eyebrow created by aggregating the red landmarks; c) right eyebrow; d) mouth; and e) overlay of aggregated landmarks on the original image. Best viewed in colour.

C. Cost Function

Our baseline convnet (no multi-task) uses a multi-class cross-entropy error function, which is a standard cost when using a softmax output. In our case, the output represents a probability distribution over facial expression labels. The cost for expression prediction can be written as:

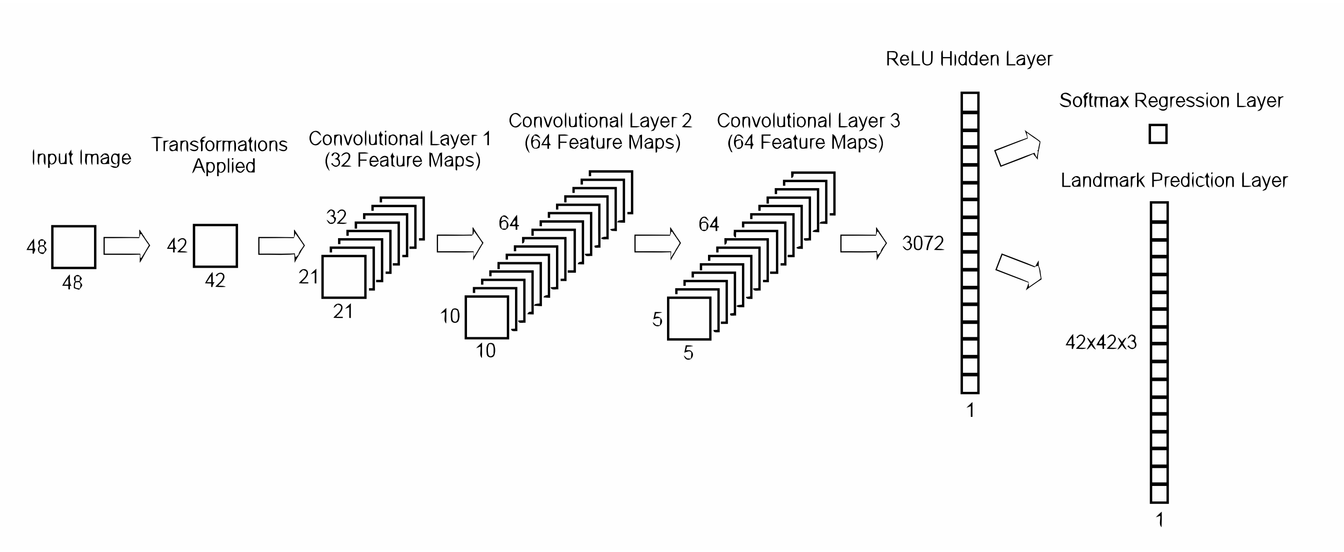
Input Image
48
48
Transformations Applied
42
42
Convolutional Layer 1 (32 Feature Maps)
32
21
21
Convolutional Layer 2 (64 Feature Maps)
64
10
10
Convolutional Layer 3 (64 Feature Maps)
64
5
5
ReLU Hidden Layer
3072
1
Softmax Regression Layer
Landmark Prediction Layer
42x42x3
1

Figure 1. An illustration of our network architecture. Random transformations are applied to the image to diversify the training set. Each image is processed by three convolutional layers and a ReLU hidden layer. From there the output goes through two parallel paths: a softmax layer for predicting the facial emotion, and an output layer for predicting the facial landmarks.

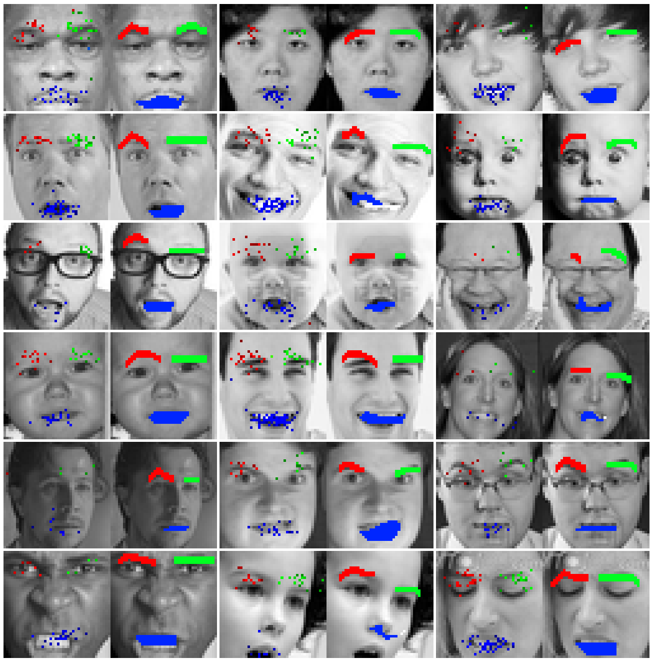

Figure 3. A visualization of the landmark output layer superimposed on images. The leftmost image in each column displays coloured pixels wherever the network had a confidence (of a landmark being present in that pixel) above 50%. The images on the right show the original image's label maps.

$$
\begin{aligned}
\mathcal{C}_e &= \frac{1}{|\mathcal{D}|}\mathcal{L}(\theta, \mathcal{D}) \\
&= -\frac{1}{|\mathcal{D}|}\sum_{n=0}^{|\mathcal{D}|-1} \log\left(p(y_n = t_n | x_n, \theta)\right) \qquad (1)
\end{aligned}
$$

where $\mathcal{D}$ is the training data, θ are the free parameters of our model (i.e. the weights and biases), $\mathcal{L}$ is the negative log likelihood function for a single input, x is the image input, y is the model output, t is the ground truth expression label, and n indexes examples. Note that $p(y_n = t_n|x_n, \theta)$ is the output produced by the convnet.

When training in the multi-task setting, our cost function includes a second term, corresponding to the landmark prediction layer. This term is the sum over the per-pixel cross-entropy errors for each output map. It can be written as:

$$
\begin{aligned}
\mathcal{C}_l = \frac{1}{|\mathcal{D}|}\sum_{n=0}^{|\mathcal{D}|-1}\Bigg[\frac{1}{\mathcal{K}}\sum_{k=0}^{\mathcal{K}-1} -\Big(l_n^{(k)} \log\left(u_n^{(k)}\right) \\
+ \left(1 - l_n^{(k)}\right)\log\left(1 - u_n^{(k)}\right)\Big)\Bigg] \times v_n \qquad (2)
\end{aligned}
$$

where $\mathcal{K}$ is the number of sigmoid units in the landmark layer (the number of facial landmarks × image size), $l_n^{(k)}$ is the landmark prediction of the k^{th} sigmoid unit for the n^{th} training image, $u_n^{(k)}$ is the corresponding ground truth value computed by aggregating the raw facial landmarks.

Our landmark detector did not return landmarks for all images, so we incorporated the binary indicator variable v_n

which would negate the second part of the cost function if the landmark detector returned no landmarks for a given image. This allowed us to train the net on images that did not have landmarks in addition to those that did. Assuming the images for which landmarks could be detected were uniformly distributed, the average landmark cost for each mini-batch would be comparable.

The total cost in the multi-task setting is the sum of the emotion cost and landmark cost:

$$\mathcal{C} = \mathcal{C}_e + \lambda \mathcal{C}_l \tag{3}$$

where λ is an empirically determined parameter which weights the two components of the cost. In all of our reported results we used $\lambda = 1$. However, we tried other values of λ and did not find our model to be sensitive to its exact setting.

Parameters were learned by mini-batch stochastic gradient descent on the total cost (Eq. 3).

IV. Experiments

The performance of both the baseline and "multi-task" networks were evaluated on the ICML 2013 Facial Expression Recognition Challenge dataset[2], and also the Toronto Face Database (TFD) [7].

For both datasets, we used the same network architecture. The first layer had 32 5×5 filters, the second 64 4×4 filters, and the third 64 5×5 filters. All convolutional layers used ReLUs as their activation functions, and also max-pooled over 2×2 neighbourhoods. This resulted in the original 42×42 input images producing 32 21×21 feature maps in the first layer, 64 10×10 feature maps in the second, and 64 5×5 feature maps in the third, before finally being flattened and processed by the ReLU hidden layer with 3072 hidden units (see Figure 1).

From here the penultimate-layer representation was processed by two parallel output layers. The first layer performed a softmax regression to generate a prediction for the emotion that the original images were displaying. The second layer contained $5,292$ ($42 \times 42 \times 3$) logistic units where each unit represented the probability of a particular landmark being present on a specific pixel of the original input images.

A. ICML Dataset

The ICML dataset consists of 28,709 48×48 training images, each with one of seven emotion labels, as well as a 7177 image test set. Images have been taken "from the wild", so the faces are in many different orientations and not always facing forward. We experimented with two convolutional neural networks, where the only difference between them was that one had an extra output layer and was also trained in the "multi-task" setting to predict facial landmarks. Each batch of images was preprocessed and transformed using the methods described in Sec. III. Four new variations of each original image were generated every epoch.

Our tests show that the net with an additional landmark prediction layer consistently performed better than the baseline convolutional net, increasing the accuracy by 0.5% to 3.0%, depending on whether or not transformations were applied to the dataset or (see Table I). The difference in increase between transformations and no transformations may be due to the benefit of data augmentation and landmark prediction not being completely orthogonal. We note that state-of-the-art performance is 71.2% achieved by Tang [8]. We re-implemented Tang's baseline rather than use his implementation in our experiments, though we used that implementation as a reference during development. We are unsure of the reason for the nearly 4% difference in performance between our baseline and his implementation using softmax output (70.1%). We suspect it has to do with our implementation of the pipeline generating transformed examples as this is where our implementation deviates the most from Tang's implementation.

	Baseline convnet	Multi-task convnet
No Transformations	63.71 ± 0.63	$\mathbf{66.83 \pm 0.66}$
w/ Transformations	66.39 ± 0.58	$\mathbf{67.21 \pm 0.70}$

Table I

Expression classification test set accuracy of models trained on the ICML dataset, averaged over 8 runs with random weight initializations.

B. TFD

TFD is structured similarly to the ICML dataset (48×48 images and seven emotion classes), however all faces are looking directly forward. As well, the images have been normalized and cropped around the edges of the face. It contains 4,178 images in five official folds. Each fold contains a training, validation, and test set consisting of 70%, 10%, and 20% of the images respectively.

Similarly to our experiments with the ICML dataset, we compared the baseline convnet with the multi-task convnet. The dataset was again normalized, but as the TFD contains only faces looking straight ahead, only mirroring and shifting transformations were applied to the images. Five variants were made for each image every epoch. The same network architecture used for the ICML dataset was also used for the TFD. Similarly to the ICML dataset, the multi-task convnet yielded better results (see Table II), about 2% improvement on the validation set and 1% on the test set. These results rival previous state of the art results on the TFD. Most notably, Rifai et al. [2] achieve 85.0 ± 0.47 % using a much more complex semi-supervised deep architecture which uses multiple stages of learning.

[2] http://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge

	Baseline convnet	Multi-task convnet
Valid	85.72 ± 1.28	$\mathbf{87.80 \pm 1.16}$
Test	84.14 ± 1.86	$\mathbf{85.13 \pm 1.84}$

Table II
CLASSIFICATION ACCURACY OF MODELS TRAINED ON THE TFD, AVERAGED OVER THE 5 OFFICIAL FOLDS.

C. Computational efficiency

If we did not use data augmentation, the addition of the landmark prediction layer would not significantly increase training time. However, when using data augmentation, each of the landmark's ground truth label maps had to be transformed in the same way as the original image. For the three landmarks we considered, this increase was about 30% per epoch if only mirroring and shifting transformations were used and 200% if rotation and zoom were also included. As the landmark layer was only used to learn a better feature pipeline, it can be completely discarded at test time. Therefore during testing, there is no difference in speed between the baseline and the multi-task net trained to predict facial landmarks.

V. CONCLUSIONS

We have introduced a multi-task convolutional network that simultaneously predicts facial landmarks and facial expression. In experiments on the ICML 2013 Facial Expression Recognition Challenge dataset, and also the Toronto Face Database, we have shown that reasoning about facial landmark position improves expression recognition. In our landmark localizer, we have employed very simple output structures which correspond to independent predictions per pixel. Future work will consider structured outputs that better capture spatial dependencies among landmarks. The local pooling layers found in typical convnets, including ours, are suited for invariant recognition but make it more difficult to preserve precise spatial information for tasks like landmark localization. We plan to investigate other architectures that are less destructive to spatial information and relationships.

REFERENCES

[1] V. Bettadapura, "Face expression recognition and analysis: the state of the art," *arXiv preprint arXiv:1203.6722*, 2012.

[2] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza, "Disentangling factors of variation for facial expression recognition," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 808–822.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, 1999, pp. 1150–1157.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[5] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[6] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On deep generative models with applications to recognition," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2857–2864.

[7] J. M. Susskind, A. K. Anderson, and G. E. Hinton, "The toronto face database," Department of Computer Science, University of Toronto, Toronto, ON, Canada, Tech. Rep. TR-2010-001, 2010.

[8] Y. Tang, "Deep learning using support vector machines," *arXiv preprint arXiv:1306.0239*, 2013.

[9] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 833–840.

[10] J. M. Susskind, G. E. Hinton, J. R. Movellan, and A. K. Anderson, "Generating facial expressions with deep belief nets," *Affective Computing, Emotion Modelling, Synthesis and Recognition*, pp. 421–440, 2008.

[11] G. E. Hinton, "To recognize shapes, first learn to generate images," *Progress in brain research*, vol. 165, pp. 535–547, 2007.

[12] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2879–2886.

[13] M. Osadchy, Y. L. Cun, and M. L. Miller, "Synergistic face detection and pose estimation with energy-based models," *The Journal of Machine Learning Research*, vol. 8, pp. 1197–1215, 2007.

[14] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[15] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, vol. 15, 2011, pp. 315–323.

[16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[17] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.