Smile Detection in the Wild Based on Transfer Learning

Xin Guo*, Luisa F. Polanía†, Kenneth E. Barner*

*Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716, USA
Email: {guoxin, barner}@udel.edu
†American Family Mutual Insurance Company, Madison, WI 53783, USA
Email: polania@amfam.com

Abstract—Smile detection from unconstrained facial images is a specialized and challenging problem. As one of the most informative expressions, smiles convey basic underlying emotions, such as happiness and satisfaction, and leads to multiple applications, such as human behavior analysis and interactive controlling. Compared to the size of databases for face recognition, far less labeled data is available for training smile detection systems. This paper proposes an efficient transfer learning-based smile detection approach to leverage the large amount of labeled data from face recognition datasets and to alleviate overfitting on smile detection. A well-trained deep face recognition model is explored and fine-tuned for smile detection in the wild, unlike previous works which use either hand-engineered features or train deep convolutional networks from scratch. Three different models are built as a result of fine-tuning the face recognition model with different inputs, including aligned, unaligned and grayscale images generated from the GENKI-4K dataset. Experiments show that the proposed approach achieves improved state-of-the-art performance. Robustness of the model to noise and blur artifacts is also evaluated in this paper.

Keywords-smile detection; CNN; deep learning; transfer learning;

I. INTRODUCTION

A smile is considered the most common human facial expression to convey emotions of joy, happiness, and satisfaction ([1]). Smile detection has multiple applications in different domains, such as human behavior analysis ([2]), photo selection ([3]), product rating ([4]), and patient monitoring ([5]). Recent longitudinal studies have used smile information from images to predict future social and health outcomes ([6], [7], [8]). For example, [8] showed that the smile intensity in Facebook profile pictures is correlated with satisfying social relationships and is a predictor of self-reported life satisfaction after 3.5 years. Another application is related to the smile shutter function of modern consumer cameras. In 2007, Sony released its first camera Cyber-shot DSC T200 equipped with a smile shutter function that perceives three human faces in the scene and takes a photograph if a smile is perceived. Similarly, in 2011, Samsung released its first smart phone with a smile shutter functionality, the Samsung Galaxy mini S5570. It is reported that the smile shutter in both the Sony and the Samsung devices is only capable of detecting big smiles but unable to detect slight smiles ([5]). All these applications motivate the development of robust and automatic smile detection algorithms.

During the last two decades, the image processing and computer vision communities have developed many smile detection algorithms ([9]). For example, in [10], local binary patterns (LBP) were used as main image descriptors for smile detection. The authors reported a classification accuracy of 90% using support vector machines (SVM) and a small dataset of 5781 images. A smile detector based on the Viola-Jones cascade classifier was proposed in [11]. The detector achieved a classification accuracy of 96.1% on a small testing dataset of 4928 images. Although the classification accuracy was high, the employed images were mainly frontal and were captured under tightly controlled conditions.

An important contribution to the field of automatic smile detection was introduced by [9]. They collected the GENKI-4K database, which contains 4000 real-life face images, downloaded from publicly available Internet repositories, which have been labeled as either smiling or non-smiling by human coders. The relevance of this dataset is that it contains a large number of images that span a wide range of imaging conditions and camera models, as well as variability in ethnicity, gender, age, and background. Prior to the GENKI-4K dataset, the employed datasets for smile detection were overly constrained and led to non-generalizable results. The GENKI-4K database has become the standard dataset for evaluating smile recognition algorithms in the wild. For example, [12] proposed a smile detection approach that uses intensity differences between pixels in the grayscale representation of the GENKI-4K face images as features. They used AdaBoost to choose and combine weak classifiers and reported a classification accuracy of 88%.

Among the latest algorithms for smile detection are the convolutional neural networks (CNN)-based algorithms, which learn hierarchical feature representations with higher level features formed by the composition of lower level features ([13]). The high classification accuracy achieved by CNNs has led them to become the state-of-the-art in smile detection in the wild. For example, in [14], a CNN architecture of 4 convolutional layers and 1 fully-connected layer was trained from scratch for smile detection. The authors
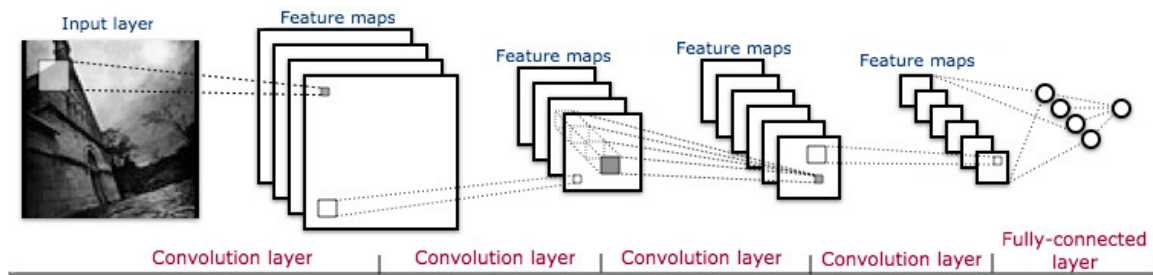
679

Figure 1. Example of a CNN architecture.

achieved a classification accuracy of 94.6%, which is greater than any accuracy attained by previous methods on the GENKI-4K dataset. Similarly, [15] proposed to use a CNN for smile detection. They used model selection for choosing the CNN parameters and used both the face and mouth regions as inputs. They used the DISFA database ([16]), whose images were captured under laboratory-controlled conditions. A CNN architecture, referred to as Smile-CNN, was recently proposed by [17] to perform smile detection. The architecture consists of 3 convolutional layers and 1 fully-connected layer, and was trained from scratch on images from the GENKI-4K dataset. The authors attained an average classification accuracy of 92.4% and 91.8% using an SVM and an AdaBoost classifier, respectively.

In this paper, we introduce a CNN-based approach that uses transfer learning to achieve a classification accuracy of 95.38% on the GENKI-4K dataset, which is greater than that obtained by previous CNN-based smile recognition methods. Specifically, our contributions include the fine-tuning of the VGG-face model ([18]) with images from the GENKI-4K dataset, incorporation of face alignment to enhance the performance of the CNN model, and evaluation of the model robustness to image artifacts, such as noise and blur. The motivation for fine-tuning a pre-trained model is that available datasets for smile recognition are small to train a deep neural network from scratch. Even the GENKI-4K dataset has a limited size of 4000 images, which differs from the large-scale datasets typically used to train CNNs from scratch, which are in the order of millions of images ([18], [19]).

II. BACKGROUND

Remarkable progress has been made in image recognition in recent years, mainly due to the availability of large-scale labeled datasets, modern graphics processors, the revival of deep CNNs, and the capability of CNNs to enable transfer learning. This section briefly describes CNNs and transfer learning.

A. Convolutional Neural Networks

In the context of visual recognition, a CNN is a type of feed-forward neural network that learns image features

from pixels through convolutions, matrix multiplications, and nonlinear transformations, constructing a non-linear mapping between the input and the output. The lower convolutional layers extract and combine local features from the input image, and the top convolutional layers are able to learn more complicated structures by combining features from previous layers. Fully-connected layers convert the features of the top convolutional layers into a 1-dimensional vector that is categorized by a trainable classifier. CNNs are trained using backpropagation ([20]). Fig. 1 illustrates, as an example, a CNN architecture of 4 convolutional layers and 1 fully-connected layer.

Feature learning for images is not a trivial problem because there are scale, orientation, and position variations for individual images. To mitigate for the high variability in the data and ensure some degree of scale, translation, and orientation invariance, CNNs combine local receptive fields, shared weights, and downsampling. Local receptive fields mean that a layer receives inputs from a set of units located in a small neighborhood in the previous layer, which allow neurons to extract primitive visual features, such as oriented edges and corners. Since units in a layer are organized in planes, weight sharing means that units within a plane share the same set of weights and perform the same operation on different parts of the image. Such planes and set of weights are usually referred to as feature maps and filter banks, respectively.

The filtering operation performed by a feature map is equivalent to discrete convolution, which earned the CNN its name. Apart from enforcing shift-invariance, weight sharing is essential for reducing the number of trainable parameters, which otherwise may grow very rapidly for high-dimensional inputs and lead to intractable networks. Downsampling refers to the introduction of layers to reduce the resolution of the feature maps, which in turn reduces the sensitivity to small translations and distortions. A typical downsampling technique is known as max-pooling and consists of computing the maximum of a local patch of units in one feature map. A standard CNN architecture contains four types of layers: convolutional, fully-connected, activation, and pooling.
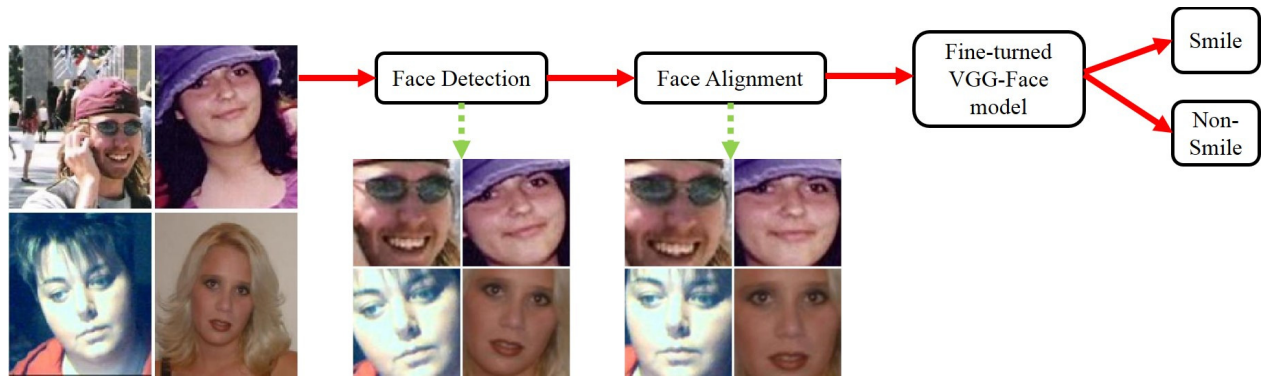
Figure 2.   System diagram. Original facial images are first detected and aligned, then fed to the CNN model for classification into two categories: smile and non-smile.

B. Transfer Learning

Two difficulties of training CNNs are the high number of needed training samples and annotations and the long time required to fully train the networks ([21]). There are many applications that suffer from deficit of training samples, and therefore, fully training a CNN becomes impractical for those cases. For example, medical imaging applications.

The concept of transfer learning, as applied to visual recognition tasks, refers to transferring image representations learned with CNNs on large datasets to other visual recognition tasks with limited training data ([22]). The intuition behind this idea is that convolutional layers provide generic mid-level image representations that can be transferred to new tasks. The natural hierarchical feature representation of CNNs, going from low-level features to more complex features, allows features to be shared between unrelated tasks.

Features learned from large-scale datasets, *e.g.*, ImageNet, can replace hand-crafted features in other tasks. Specifically, the process starts by removing the last fully-connected layer and then, use the rest of the CNN as a fixed feature extractor for the new dataset. Alternatively, the features learned from large-scale datasets can serve as a better weight initialization for networks being trained with limited data. This process is known as fine-tuning and can be performed in two different ways. That is, either all the layers of the CNN can be fine-tuned or the high-level portion of the CNN can be fine-tuned while some of the earlier layers can be kept fixed to prevent overfitting. The motivation for fine-tuning is that the low-level features of a CNN contain more generic features, *e.g.*, edge detectors, while higher-level features become more specific to the given task.

III. Methods

This section describes the face detection and alignment methods used in our experiments to preprocess the raw facial images, and the details for fine-tuning the VGG-face model. The pipeline of the method is shown in Fig. 2.

A. Preprocessing

Faces are first detected using the method described in [23]. The motivation for using this method is that it provides facial landmarks which can be used for face alignment via a 2D affine transformation where the left and right eye corners of all the images are aligned to the same positions. The next step is to crop and rescale the face regions to 256×256 pixels. Samples of cropped and aligned face patches are shown in Fig. 2.

Smile detection requires the modeling of subtle and localized variations between images. By eliminating some of the other type of variability in the data that is not relevant for smile detection, *e.g.*, head pose variations, face alignment is expected to help the network better learn the optimal features for smile detection.

B. VGG Net

The 16-layer VGG architecture was presented in [21] as a new architecture that bids the performance of AlexNet ([19]) in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). To this end, the authors increased the depth of AlexNet by adding more convolutional layers and used smaller convolution filters to keep the number of parameters tractable.

The 16-layer VGG architecture is as follows. The input to VGG is a fixed-size RBG image of 224×224 pixels. All the convolutional layers have a fixed small receptive field which is 3×3. The convolution stride and the spatial padding are both fixed to 1 pixel. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) layer as in [19]. Spatial downsampling is performed through max-pooling over a 2×2 pixel window, with stride 2, after 2 or 3 convolutional layers. A stack of 13 convolutional layers are followed by three fully-connected layers, where the first two have 4096 channels each, and the third has a number of channels that depends on the classification task. The last layer is the soft-max layer. The activation function for each fully-connected layer is a rectified linear unit as well.

Figure 3. Examples of smiling (top two rows) and non-smiling (bottom two rows) faces in the wild. Images are from the GENKI-4K database.

In [18], the VGG-face model was presented as the result of training the 16-layer VGG architecture on a very large-scale dataset for face recognition. The dataset contained 2.6M images of 2.6K celebrities and public figures. The VGG-face model became the state-of-the-art for face recognition on the YouTube Faces dataset.

C. Fine-tuning

The architecture of the VGG-face model is modified by changing the number of neurons in the last fully-connected layer to 2, indicating a binary classification having as targets smile and non-smile facial expressions. With the exception of the last fully-connected layer, the modified architecture is initialized with the VGG-face model ([18]), which is expected to be better than random Gaussian weights initialization since it was trained on 2.6M facial images. The last fully-connected layer is initialized with weights sampled from a Gaussian distribution of zero mean and variance 1×10^{-4}.

Because the features learned from CNN layers typically correspond to generic features, such as contours and edges, the weights of all the convolutional layers are kept the same as in the VGG-face model, while the weights of the first two fully-connected layers are fine-tuned and the last fully-connected layer is trained from scratch.

The goal is to train the model that minimizes the average error of the final soft-max layer. The learning parameters of the model, such as learning rate and weight decay of the network, are set the same as in the VGG-face model, then gradually adjusted based on grid search. As a result, all the model learning parameters are the same as in the

Subset	1	2	3	4
Number of smile faces	540	541	540	541
Number of non-smile faces	460	459	460	459

VGG-face model except the initial learning rate, which is scaled by a factor of 10. More precisely, the learning rate is initially set to 1×10^{-3} and then decreased by a factor of 10 when the accuracy on the validation set stops increasing. The weight decay coefficient is set to 5×10^{-4}. Stochastic gradient descent is used to optimize the network with mini-batches of 64 samples and momentum coefficient of 0.9.

The training images are rescaled to 256×256 and randomly cropped to 224×224 patches to generate the input to the network. The training data is further augmented by flipping the images horizontally with 50% probability.

IV. EXPERIMENTAL RESULTS

This section describes the experiments that validate the performance of the fine-tuned VGG-face model for smile detection using three different inputs, including aligned, unaligned and grayscale images generated from the GENKI-4K dataset. Evaluation of the models is performed in terms of classification accuracy and robustness to noise and blur artifacts. All the experiments were carried on 2 NVIDIA K40C GPUs, each with 12GB GDDR5.

A. Database

The VGG-face model is fine-tuned and tested on the GENKI-4K ([9]) database. The images were taken by ordinary people for their own purpose, thus resulting in a wide range of imaging conditions, both outdoors and indoors, as well as variability in illumination, pose (yaw, pitch and roll parameters of the head of most of the images is within $\pm 20°$ from frontal position), background, age, gender, ethnicity, facial hair, hat, and glasses. All the images are manually labeled. GENKI-4K contains 2162 and 1838 smiling and non-smiling facial images, respectively. Sample images are shown in Fig. 3.

B. Comparison with State-of-the-art Methods

To perform a fair comparison with state-of-the-art methods, the GENKI-4K dataset is first randomly divided into four subsets, having 1000 samples each, and then, those subsets are used for four-fold cross-validation. The number of smiling faces and non-smiling faces for each fold are shown in Table. I. Each time, one subset is used for testing and the other three are used for training. The average detection rate and the standard deviation of the four-fold are reported as the final performance. All the images are preprocessed as described in Section III-A.

As discussed in Section III-C, the weights of all the convolutional layers are set the same as the VGG-face
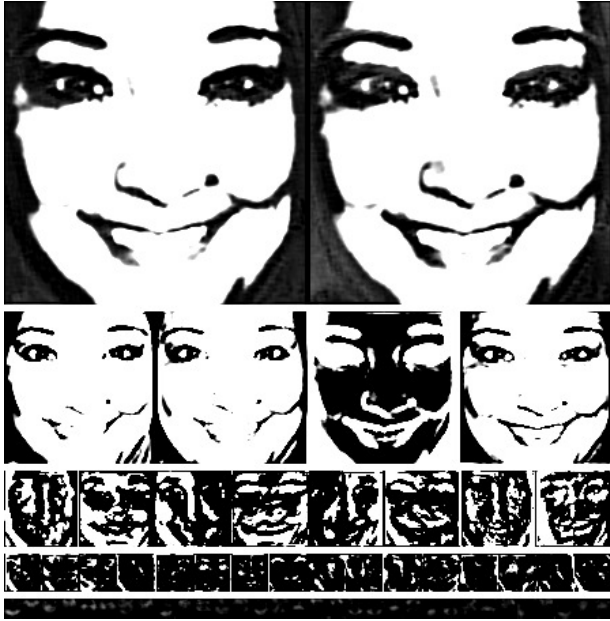
Figure 4. Random feature maps learned from CNN layers of a sample aligned face (refer to ([18]) for a description of the VGG layers). First row: 2 feature maps learned from conv1_2, each feature map of size 224×224. Second row: 4 feature maps learned from conv2_2, each feature map of size 112×112. Third Row: 8 feature maps learned from conv3_3, each feature map of size 56×56. Fourth row: 16 feature maps learned from conv4_3, each feature map of size 28×28. Fifth row: 32 feature maps learned from conv5_3, each feature map of size 14×14.
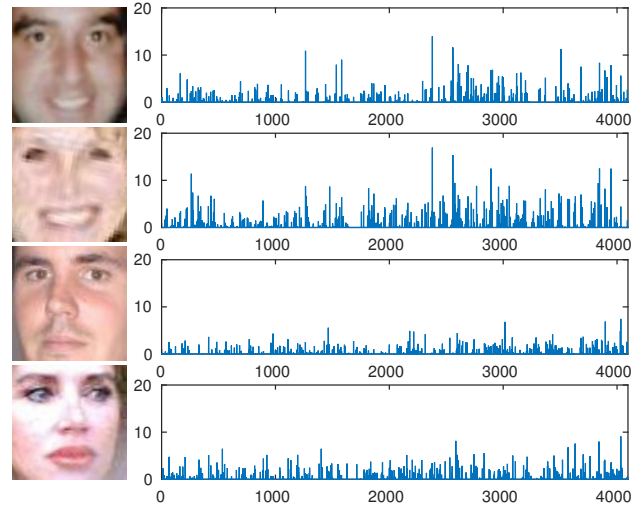

Figure 5. Features of fc7 layer for 2 smiling faces and 2 non-smiling faces. Left column: the aligned faces; right column: the values of features extracted after the fc7 layer, the horizontal and vertical axes correspond to the feature index and the feature values, respectively.

model weights because features extracted from CNN layers are generic features and because the VGG-face model was trained on 2.6M facial images. Sample feature maps from CNN layers of a smile image are shown in Fig. 4. It can be seen that low-level convolutional layers generally learn edges and outlines while the features become more abstract and sparse for upper layers. The first two and the last fully-connected layers are fine-tuned and trained from scratch on the GENKI-4K dataset, respectively. The features of the last fully-connected layer, fc7 (4096 features), are extracted and compared for 2 smiling and 2 non-smiling faces in Fig. 5. As shown in the figure, features within the same class have similar feature values and trends, despite variations in gender, illumination and head pose.

The number of weight and bias parameters of the original VGG-face model is around 138M. However, the number of trainable parameters of the fine-tuned VGG-face model is 120M since the parameters of all the convolutional layers are kept fixed during training. Despite having a reduction of only 13% in the number of parameters, training takes only 30 minutes to converge at 1000 iterations.

The performance of the fine-tuned VGG-face model is compared with state-of-the-art methods in Table II. These methods are related to either the extraction of handcrafted features, such as histogram of oriented gradients (HOG)

and local binary patterns (LBP), or to CNN models trained from scratch. The proposed fine-tuned VGG-face model outperforms all the methods in Table II in terms of classification accuracy and also exhibits a small variance. It is worth noting that most of the other methods use classifiers that are more sophisticated than the softmax classifier, such as SVM and extreme learning machine (ELM). However, the representational power of the features learned by CNN models reduces the need for using sophisticated classifiers.

C. Impact of Face Alignment and Color Channels

In a real-world scenario, facial landmarks may be hard to detect due to occlusion and illumination, resulting in misaligned faces. Similarly, color images may not always be available, and we have to content ourselves with grayscale images. In this section, the impact of face alignment and color information in the performance of fine-tuned VGG-face models is evaluated.

Our approach to evaluate the impact of face alignment requires training and testing the model with the original unaligned GENKI-4K images using only face detection and cropping as preprocessing. On the other hand, to evaluate the impact of color information, the GENKI-4K images are first preprocessed as described in Section III-A, then converted to grayscale and fed to the fine-tuned VGG model for training and testing. Note that since the input of the VGG-face model requires the three color channels, the same grayscale image is fed to the red, green, and blue channels to convert a single-channel image to a 3-channel one.

The same cross-validation partitioning described in Table I is employed for this experiment and the results are compared

683

Table II

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE GENKI-4K DATABASE.

Method	Features	Classifier	Accuracy(%)
[24]	LBP	ELM	85.2
	HOG	ELM	88.2
[12]	LBP	SVM	87.1±0.76
	Pixel comparison	AdaBoost	89.7±0.45
[25]	HOG (labeled)	SVM	91.8±0.97
	HOG (labelled + unlabelled)	SVM	92.3±0.81
[17]	Raw pixels	SVM	84.0±0.91
	Raw pixels	AdaBoost	80.0±0.76
	Learned features	SVM	92.4±0.59
	Learned features	AdaBoost	91.8±0.95
[26]	Guassian	SVM	93.2±0.92
[14]	CNN-Basic	Softmax	93.6±0.47
	CNN-2Loss	Softmax	94.6±0.29
[27]	HOG31 + GSS + Raw pixel	AdaBoost	92.51±0.40
	HOG31 + GSS + Raw pixel	Linear SVM	94.28±0.60
	HOG31 + GSS + Raw pixel	Linear ELM	94.21±0.35
	HOG31 + GSS + Raw pixel	Adaboost + Linear SVM	94.56±0.62
	HOG31 + GSS + Raw pixel	Adaboost + Linear ELM	94.61±0.53
The proposed method	Fine-tuned VGG-face model	Softmax	95.38±0.52

Table III

CLASSIFICATION ACCURACY (%) ON UNALIGNED FACIAL IMAGES AND ALIGNED GRAYSCALE FACIAL IMAGES

	Fold1	Fold2	Fold3	Fold4	Avg
RGB & aligned	95.0	95.5	95.7	95.3	95.4
RGB & unaligned	94.8	95.2	95.1	94.1	94.8
Grayscale & aligned	94.4	95.1	94.9	95.0	94.9

with the classification accuracy attained using the original GENKI-4K images after cropping and face alignment. As expected, experimental results (Table III) show that the models trained with the aligned RGB images perform better that the models trained with the unaligned and grayscale images. However, the decrease is small, less than 1%, which means that the fine-tuned VGG-face model can achieve high classification accuracies even when the data exhibits high variability of head poses (within $\pm 20°$ from frontal position) and loss of color information.

D. Evaluation under Image Quality Distortion

Image distortions, such as image noise and blur, have demonstrated their power to fool a well-trained deep learning network ([28], [29]). In this section, the fine-tuned VGG-face model is evaluated on distorted images to evaluate its robustness.

Noise in the images may result from using low-quality camera senors in the real world. Noise is modeled as Gaussian noise added to each color channel of each pixel separately. The standard deviation of the noise is varied from

1 to 10 in steps of 1.

Blur may occur either because a camera is not focused properly on the target of interest or because the target is moving. A Gaussian kernel with varying standard deviation from 1 to 10 in steps of 1 is used to blur the images. The size of the filter window is set to 4 times the standard deviation. Sample images of noisy and blurred images under different Gaussian variations are shown in Fig. 6.

The results in Fig. 7 indicate that the fine-tuned VGG-face model is robust to noise and blur to some degree, given that the classification accuracy remains high (above 80%) regardless of the image artifacts. It is worth noting that the accuracy of blurred images almost stays the same before the standard deviation of the Gaussian kernel reaches 6.

V. CONCLUSIONS AND DISCUSSION

A smile detection method based on transfer learning was presented in this paper. Unlike previous research works which either perform feature extraction and classification separately or train a CNN from scratch, we leveraged the large labeled datasets and well-trained deep learning models in the face recognition field to generate a CNN model that achieves improved state-of-the-art on smile detection. The motivation behind this approach is that the labeled data on real-world smile detection datasets is scarce compared to the labeled data on real-world face recognition datasets. For example, the GENKI-4K dataset has 4K images while the VGG-face recognition database has 2M images.

It was shown via experiments that the proposed method outperforms state-of-the-art methods for smile detection in
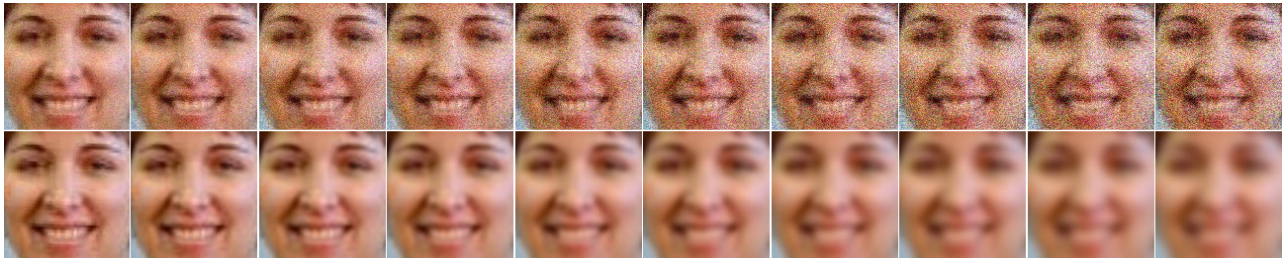
Figure 6. Sample images of noisy and blurred images. Top row: noisy images with the standard deviation of Gaussian noise varying from 1 to 10 in steps of 1. Bottom row: blurred images with the standard deviation of a Gaussian kernel varying from 1 to 10 in steps of 1.
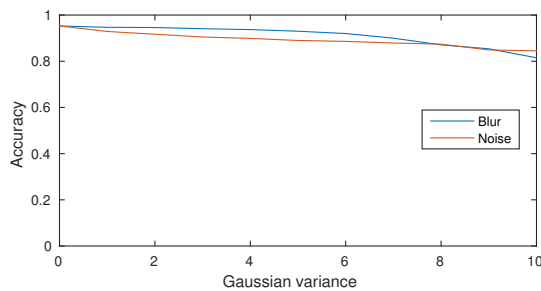

Figure 7. Performance evaluation of the fine-tuned VGG-face model in the presence of noise and blur artifacts.

terms of classification accuracy. However, transfer learning is not only important for increasing the classification accuracy, but it also reduces training time and leads to more robust systems because it exploits the high variability of large-scale datasets. For example, it was also shown that models also achieve high classification accuracy for smile recognition when fine-tuned and tested on artifact-corrupted images.

ACKNOWLEDGMENT

REFERENCES

[1] P. Ekman, "Self-deception and detection of misinformation," *Self-deception: An adaptive mechanism*, pp. 229–257, 1988.

[2] M. Pantic, A. Pentland, A. Nijholt, and T. Huang, "Human computing and machine understanding of human behavior: A survey," in *Artifical Intelligence for Human Computing*. Springer, 2007, pp. 47–71.

[3] E. Potapova, M. Egorova, and I. Safonov, "Automatic photo selection for media and entertainment applications," *GRAPHICON-2009*, pp. 117–124, 2009.

[4] C. Tavares and T. Odaka, "System and method for capturing and using biometrics to review a product, service, creative work or thing," Jun. 24 2004, uS Patent App. 10/876,848.

[5] H. Yadappanavar and S. Shylaja, "Machine learning approach for smile detection in real time images," *International Journal of Image Processing and Vision Sciences*, vol. 1, no. 1, 2012.

[6] E. Abel and M. Kruger, "Smile intensity in photographs predicts longevity," *Psychological Science*, vol. 21, no. 4, pp. 542–544, 2010.

[7] L. Harker and D. Keltner, "Expressions of positive emotion in women's college yearbook pictures and their relationship to personality and life outcomes across adulthood." *Journal of personality and social psychology*, vol. 80, no. 1, p. 112, 2001.

[8] J. Seder and S. Oishi, "Intensity of smiling in Facebook photos predicts future life satisfaction," *Social Psychological and Personality Science*, vol. 3, no. 4, pp. 407–413, 2012.

[9] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan, "Toward practical smile detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2106–2111, 2009.

[10] D. Freire-Obregón, M. Castrillón-Santana, and O. Déniz-Suárez, "Smile detection using local binary patterns and support vector machines," *Proceedings of Computer Vision Theory and Applications (VISAPP09)*, 2009.

[11] O. Déniz, M. Castrillon, J. Lorenzo, L. Anton, and G. Bueno, "Smile detection for user interfaces," in *International Symposium on Visual Computing*, 2008, pp. 602–611.

[12] C. Shan, "Smile detection by boosting pixel differences," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 431–436, 2012.

[13] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[14] K. Zhang, Y. Huang, H. Wu, and L. Wang, "Facial smile detection based on deep learning features," in *3rd IAPR Asian Conference on Pattern Recognition*, 2015, pp. 534–538.

[15] P. Glauner, "Deep convolutional neural networks for smile recognition," *arXiv preprint arXiv:1508.06535*, 2015.

[16] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J. Cohn, "DISFA: A spontaneous facial action intensity database," *IEEE Transactions on Affective Computing*, vol. 4, no. 2, pp. 151–160, 2013.

[17] J. Chen, Q. Ou, Z. Chi, and H. Fu, "Smile detection in the wild with deep convolutional neural networks," *Machine Vision and Applications*, pp. 1–11, 2016.

[18] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1717–1724.

[23] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[24] L. An, S. Yang, and B. Bhanu, "Efficient smile detection by extreme learning machine," *Neurocomputing*, vol. 149, pp. 354–363, 2015.

[25] M. Liu, S. Li, S. Shan, and X. Chen, "Enhancing expression recognition in the wild with unlabeled reference data," *Asian Conference on Computer Vision*, pp. 577–588, 2012.

[26] V. Jain and J. L. Crowley, "Smile detection using multi-scale gaussian derivatives," in *In 12th WSEAS International Conference on Signal Processing, Robotics and Automation*, 2014.

[27] Y. Gao, H. Liu, P. Wu, and C. Wang, "A new descriptor of gradients self-similarity for smile detection in unconstrained scenarios," *Neurocomputing*, vol. 174, pp. 1077–1086, 2016.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[29] S. Dodge and L. Karam, "Understanding How Image Quality Affects Deep Neural Networks," *arXiv:1604.04004*, 2016. [Online]. Available: http://arxiv.org/abs/1604.04004