Deep Learning Models for Facial Expression Recognition

Atul Sajjanhar
School of Information Technology
Deakin University
Geelong, Victoria, Australia

atuls@deakin.edu.au

ZhaoQi Wu
University of Electronic Science and Technology of China
Chengdu, China

1184852670@qq.com

Quan Wen
University of Electronic Science and Technology of China
Chengdu, China

quanwen@uestc.edu.cn

***Abstract*— We investigate facial expression recognition using state-of-the-art classification models. Recently, CNNs have been extensively used for face recognition. However, CNNs have not been thoroughly evaluated for facial expression recognition. In this paper, we train and test a CNN model for facial expression recognition. The performance of the CNN model is used as benchmark for evaluating other pre-trained deep CNN models. We evaluate the performance of Inception and VGG which are pre-trained for object recognition, and compare these with VGG-Face which is pre-trained for face recognition. All experiments are performed on publicly available face databases, namely, CK+, JAFFE and FACES.**

Keywords*— *facial expression recognition, LBP, deep learning, transfer learning, CNN, Inception, VGG, VGG-Face

I. Introduction

The human face image holds information which is useful to determine the characteristics and identity of the face. Face characteristics and identity determination have several applications. These applications include border security [1], forensic art, surveillance monitoring [2] and biometrics [3]. A face image has demographic and expression information of an individual. Facial attributes can be used to classify face images based on demography and expression. In this work, we focus on facial expression recognition. Typical applications of expression recognition are mental state recognition, human computer interaction, and human behavior understanding.

Approaches for facial expression recognition can be divided into two broad categories: first, approaches based on geometric features extracted from face images such as distance between the eyes etc.; and second, approaches based on image appearance which use the raw pixel data such as intensity histogram [4]. Convolutional Neural Networks (CNNs) are the current state-of-the-art model architecture for image classification tasks based on image appearance. Although there is extensive literature about CNNs for face recognition, CNNs have not been thoroughly evaluated for facial expression recognition. In this paper, we evaluate and compare classification models based on CNNs for facial expression recognition.

The paper is organized as follows. In Section II, we describe the approaches based on image appearance which are used to train and test a CNN model for facial expression recognition. Section III describes the pre-trained classification models used in this paper. Experimental Setup is described in Section IV. Experimental results are given in Section V. Discussion is presented in Section VI, and the paper is concluded in Section VII.

II. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) rely on image appearance for classification tasks. Image appearance for object recognition is described in sub-section A. The architecture of a CNN is described in sub-section B.

A. Image Appearance

We use CNN for facial expression recognition (FER) based on image appearance, as shown in Fig. 1. In the context of appearance of face images, we consider region-of-interest (ROI) images, difference images and Local Binary Pattern (LBP). Each of these are explained in this section.

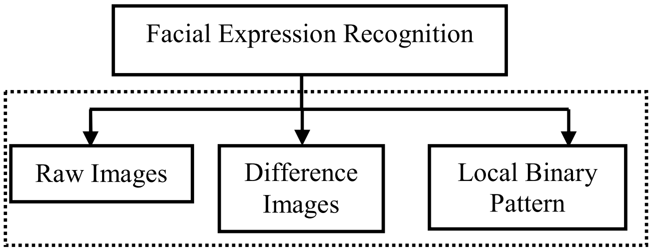

Fig. 1. Facial Expression Recognition Based on Image Appearance

The classification accuracy achieved for ROI images is validated against the classification accuracy for difference images and LBP images. Deep Learning approach is used for classification based on appearance.

ROI images are obtained from raw images by extracting the face region from raw face images.

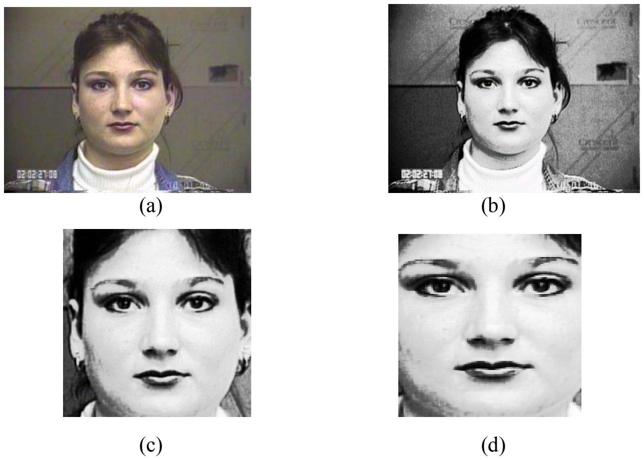
(a) (b) (c) (d)

Fig. 2. Preprocessing of Face Image (a) Original Image (b) Grayscale and Contrast Enhancement (c) Crop by Face ROI (d) Crop by Fixed Rectangle and Resize

ROI images eliminate regions of the raw image which are not relevant for FER. Hence, ROI images are created by cropping the original image to remove regions such as hair

978-1-5386-6602-9/18/$31.00 ©2018 IEEE

and ears. We use object detection using Haar feature-based cascade classifiers proposed by [5] to obtain ROI images. An example of preprocessing a CK+ face image to obtain a ROI image is shown in Fig. 2. Fig. 2(a) is the original image, Fig. 2(b) is the image after conversion to grayscale and contrast enhancement. In Fig. 2(c), cropping is applied based on face region-of-interest (ROI). Fig. 2(d) is the final ROI image which is obtained by fixed size cropping and scaling to 128x128 pixels.

Another approach for FER is by using difference images, as shown in Fig. 1. Difference images are obtained by further processing ROI images. Each difference image is obtained from two ROI images, namely, the ROI images for neutral and peak expressions. A difference image is obtained by computing the difference between gray level intensities of corresponding pixels in ROI images of neutral and peak expressions, as shown in Fig. 3.

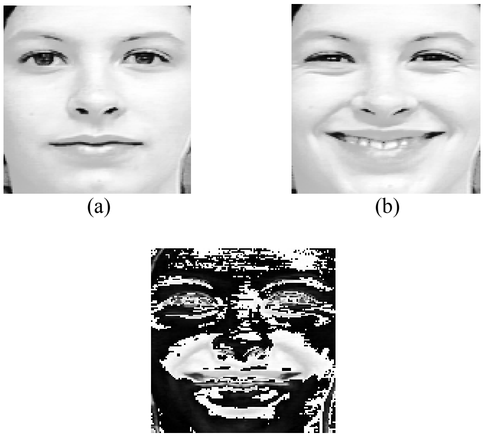
(a) (b) (c)

Fig. 3. (a) Neutral Expression (b) Peak Expression (c) Difference Image

Besides ROI images and Difference images, we consider LBP images for FER. Here, we give an overview of LBP [6]. LBP considers the neighborhood of each pixel in the image. It thresholds grayscale values of P neighbors with the center pixel and conducting the code in a binary form. If the grayscale value of the neighbor pixel is greater than or equal to the grayscale value of the center pixel, it is labeled as 1, otherwise it is labeled as 0. The resulting binary values are then converted to decimal and assigned to the center pixel. Each decimal value represents a label to every pixel of the image. The histogram of the labels can be utilized as a texture descriptor. The decimal value of the LBP code of a pixel *c* at position (x_c, y_c) is mathematically described as follows.

$$LBP_{P,R}(x_c, y_c) = \sum_{m=0}^{P-1} s(k_P - k_c) 2^m \tag{1}$$

$$s(x) = \begin{cases} 1, if\ x \geq 0 \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

where, P is the number of pixels in the neighborhood, R is the radius around the center pixel, kc is the intensity of the pixel *c*, and k*p* is the intensity of the neighboring pixels. The basic LBP operator considers 8 neighbors. The operator can be extended to cover different number of neighbors (P) and allow for different radius pixels (R). Hence, $LBP_{8,1}$, $LBP_{8,2}$, $LBP_{16,2}$ are some examples of LBP operators. An illustration is given in Fig. 4. We considered 8 neighbors (P=8) and one pixel radius (R=1) for the LBP operator.

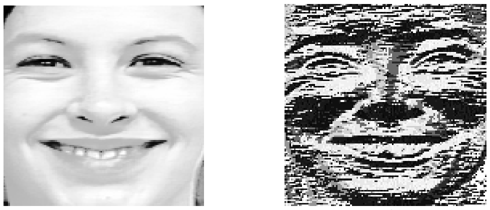
(a) (b)

Fig. 4. (a) Raw Image (b) LBP Image

B. CNN Architecture

CNN extracts high level image features by applying a series of convolution filters to the raw pixels of an image; the model can then use these features for classification. A typical CNN architecture is composed of a stack of convolutional modules; each module consists of a convolutional layer followed by a pooling layer. The convolutional layer is the key building block of a CNN. It extracts features from all locations of the image. The pooling layers down-sample the data extracted by the convolutional layer to reduce the computation in the network, and control overfitting. The last convolutional module is followed by one or more dense layers that perform classification. We built a CNN classifier using the following architecture.

- Convolutional Layer #1: Applies 32 5x5 filters (extracting 5x5-pixel sub-regions), with rectified linear units (ReLU) activation function
- Pooling Layer #1: Performs max pooling with a 2x2 filter and stride of 2 (which specifies that pooled regions do not overlap)
- Convolutional Layer #2: Applies 64 5x5 filters, with ReLU activation function
- Pooling Layer #2: Again, performs max pooling with a 2x2 filter and stride of 2
- Dense Layer #1: 1,024 neurons, with dropout regularization rate of 0.4 (probability of 0.4 that any given element will be dropped during training)
- Dense Layer #2 (Logits Layer): 8 neurons, one for each expression (0-7).

III. Pre-trained Models for Classification

In this section, we describe pre-trained models for image classification. The approach described in Section II requires significant amounts of labelled data and resources for training when applied to a deep CNN. Further, training a deep CNN is complicated by overfitting and convergence issues [7]. These problems are overcome in the transfer learning approach in which a deep CNN is pre-trained on a very large database. The pre-trained model can be used for new domains. Transfer learning models are divided into four categories: instance-based transfer learning, parameter transfer learning, feature-representation transfer learning, and relational-knowledge transfer learning [8]. In instance-based-transfer learning, certain parts of the data in the source domain can be reused in the target domain by reweighting. Parameter-transfer learning assumes that individual models for source domain and target domain share some parameters or prior distributions of hyper-parameters. Feature-representation-transfer learning aims at finding good feature representation for the target domain. The relational-knowledge-transfer learning deals with transfer learning problems in relational domains. We describe Inception-v3, VGG, and VGG-Face which are pre-trained models. We evaluate these models for FER. In each of these models, we remove the last fully-connected layer, and then use the rest of the CNN to extract features from face databases. Each image is reused multiple times during training. Calculating each bottleneck takes a significant amount of time; these features are used as input to train the last fully-connected layer for the new database. The advantage is that the training time is greatly reduced, and the disadvantage is that training accuracy and generalization ability cannot be guaranteed.

A. Inception

Inception [9] is pre-trained on the image database used in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [10]. Inception-v3 provides high accuracy results. The improvement of Inception-v3 and Inception-v2 compared with Inception-v1 is achieved from factorization of convolutions into smaller convolutions. This accelerates the computations, and the extra computing power can be used to deepen the network. The architecture of Inception-v3 is shown in Table I.

TABLE I. ARCHITECTURE OF INCEPTION-V3

Input Image width × height × channels	Architecture C = Convolutional layer, P = Maxpool layer, M = Mixed layer, F = Fully connected layer	No. of parameters	Patch Fusion	Feature Length	Training set
299x299x3	C0:32x3x3, C1:32x3x3, C2:64x3x3, P0:3x3, C3:80x1x1, C4:192x3x3, P1:3x3, M0:35x35x256, M1:35x35x288, M2:35x35x288, M3:17x17x768, M4:17x17x768, M5:17x17x768, M6:17x17x768, M7:17x17x768, M8:8x8x1280, M9:8x8x2048, M10:8x8x2048, P3:8x8, F0	Less than 25M	yes	2048	ImageNet 126M Images 1K subjects

B. VGG

VGG is a pre-trained model for object recognition. Similarly to Inception-v3, VGG is pre-trained on ImageNet database. VGG has a two variants, namely, VGG16 and VGG19. VGG16 VGG19 is trained using the ImageNet database for 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [10]. The architecture of VGG16 is shown in Table II.

TABLE II. ARCHITECTURE OF VGG16

Input Image width × height × channels	Architecture C = Convolutional layer, P = Maxpool layer, F = Fully connected layer	No. of parameters	Patch Fusion	Feature Length	Training set
224x224x3	C0:64x3x3, C1:64x3x3, P0:3x3, C2:128x3x3, C3:128x3x3, P1:3x3, C4:256x3x3, C5:256x3x3, C6:256x3x3, P2:3x3, C7:512x3x3, C8:512x3x3, C9:512x3x3, P3:3x3, C10:512x3x3, C11:512x3x3, C12:512x3x3, F0:4096, F1:4096, F2:1000	138M	No	4096	ImageNet 126M Images 1K subjects

VGG19 has more depth than VGG16, and it achieved better performance than VGG16 in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [10]. VGG19 is trained in ImageNet for 2014 ILSVRC. The architecture of VGG19 is shown in Table III.

TABLE III. ARCHITECTURE OF VGG19

Input Image width × height × channels	Architecture C = Convolutional layer, P = Maxpool layer, F = Fully connected layer	No. of parameters	Patch Fusion	Feature Length	Training set
224x224x3	C0:64x3x3, C1:64x3x3, P0:3x3, C2:128x3x3, C3:128x3x3, P1:3x3, C4:256x3x3, C5:256x3x3, C6:256x3x3, C7:256x3x3, P2:3x3, C8:512x3x3, C9:512x3x3, C10:512x3x3, C11:512x3x3, P3:3x3, C12:512x3x3, C13:512x3x3, C14:512x3x3, C15:512x3x3, F0:4096, F1:4096, F2:1000	144M	No	4096	ImageNet 126M Images 1K subjects

C. VGG-Face

Deep CNN architectures for face recognition are different from architectures for object recognition because feature extraction for face images typically needs to perform transformation for pose alignment [11]. Models such as DeepID [12], DeepFace [13] and Webface [14], have been proposed for face recognition. However, these are not publicly available which makes it difficult to systematically evaluate these models. Further, these models are trained on databases which are not publicly available. On the other hand, VGG-Face [22] is a publicly available pre-trained model for face recognition unlike Inception-v3 and VGG which are pre-trained for general object recognition. VGG-Face has been shown to achieve results comparable to the state-of-the-art. The architecture of VGG-Face is shown in Table IV.

TABLE IV. ARCHITECTURE OF VGG-FACE

Input Image width × height × channels	Architecture C = Convolutional layer, P = Maxpool layer, F = Fully connected layer	No. of parameters	Patch Fusion	Feature Length	Training set
224x224x3	C0:64x3x3, C1:64x3x3, P0:2x2, C2:128x3X3, C3:128x3x3, P1:2x2, C4:256x3x3, C5:256x3x3, C6:256x3x3, P2:2x2, C7:512x3x3, C8:512x3x3, C9:512x3x3, P3:2x2, C10:512x3x3, C11:512x3x3, C12:512x3x3, P4:2x2, F0:4096, F1:4096, F2:2622	138M+	No	4096	2.6M Images 2622 subjects

VGG-Face uses the approaches of [20] and [21] for training. The training aims to minimize the average prediction log-loss after the softmax layer [22].

IV. EXPERIMENTAL SETUP

We performed experiments for expression recognition of face images. In this section, we describe the databases used in the experiments. We also explain preprocessing of images which is required for FER based on image appearance.

A. Databases

Approaches for FER based on image appearance (see Section II) are extensively evaluated on three face expression databases, namely, Cohn-Kanade (CK+) [15], JAFFE [16], and FACES [17]. Each of these are described below.

The original CK database includes 65% females and 35% males with age ranges between 18-30 years old. The second version of this database which is referred to as CK+ database has an additional 27% of subjects. It comprises 123 subjects and 593 sequences. Each sequence has a series of facial images from neutral to a peak expression. The images display seven facial expressions: neutral, sad, disgust, contempt, fear, anger, and happiness.

JAFFE database contains 213 images of facial expressions posed by 10 Japanese females. The images display six facial expressions: neutral, sad, disgust, fear, anger, and happiness. The images have been rated on the 6 emotion adjectives by 60 Japanese subjects. The photos were taken at the Psychology Department in Kyushu University.

FACES database contains face images of 171 young (n = 58), middle-aged (n = 56), and old (n = 57) people displaying each of six facial expressions: neutrality, sadness, disgust, fear, anger, and happiness. It comprises two sets of pictures per person and per facial expression, resulting in a total of 2,052 images. It was developed at the Center for Lifespan Psychology, Max Planck Institute for Human Development, Berlin, Germany.

B. *Image Preprocessing*

Image preprocessing is required to evaluate FER based on the image appearance. The CNN classifier model is trained and tested on three sets of face images which are obtained by preprocessing the original images in the face databases. The three sets of images obtained by preprocessing are ROI images, difference images and LBP images, as described in Section II. The pre-trained models are tested for ROI images only. These models are not tested for difference images and LBP images because these images are not used during training either.

A significant step in the preprocessing described above is contrast enhancement. We use Histogram Equalization (HE) [18] for contrast enhancement. HE is a widely used approach for contrast enhancement because of its simplicity and effectiveness. By applying HE, the histogram of pixel intensities is remapped to achieve a better distribution. This improves the contrast in regions of low local contrast. HE accomplishes this by spreading out the most frequent intensity values.

V. Experimental Results

We performed expression recognition of face images in CK+, JAFFE and FACES databases (see Section IV). We conducted experiments to investigate classification accuracy based on image appearance. TensorFlow API [19] is used for implementing the experiments. Two sets of experiments were conducted. The first set of experiments is used to evaluate expression recognition using a CNN. The second set of experiments is used to evaluate expression recognition based on transfer learning approaches using deep CNN models (Inception-v3, VGG, and VGG-Face).

A. *Classification using a CNN*

The CNN described in Section IIA was applied to preprocessed face images. We performed three experiments in this category. In the first experiment, we used the raw pixel data of face images (as shown in Fig. 2) for training and testing a CNN. In the second experiment, we applied the same CNN to difference images (as shown in Fig. 3). In the third experiment, we applied the same CNN to LBP images (as shown in Fig. 4). The classification accuracy using 10-fold cross validation is shown in Table V.

TABLE V. FACIAL EXPRESSION RECOGNITION USING A CNN

	CK+	JAFFE	FACES
ROI	73.53	56.26	67.38
Difference	85.19	65.17	84.38
LBP	78.72	53.51	78.67

B. *Classification using Pre-trained Models*

The process of fully training the model used in the above experiments has high computational cost. Transfer learning is used to avoid fully training the model, as discussed in Section III. We used transfer learning technique by repeating the above experiments with pre-trained models. We retrained the final layer of Inception-v3, VGG19, and VGG-Face for FER using ROI images. The results of the experiments are shown in Table VI.

TABLE VI. FACIAL EXPRESSION RECOGNITION USING PRE-TRAINED MODELS

	CK+	JAFFE	FACES
Inception-v3	76.52	75.88	82.19
VGG19	86.20	94.71	97.16
VGG-Face	91.37	86.67	95.06

VI. Discussion

Experiments were performed using CK+, JAFFE and FACES face databases. The results in Table V are obtained by fully training a CNN model from scratch. Table V shows FER results for different approaches based on image appearance, including, ROI, Difference and LBP. The Difference images give the best results. The drawback of using Difference Images (and LBP images) is that these are not used for pre-training classification models. Typically, raw images are used as training data in pre-trained classification models. In Table VII and Table VIII, we see that Difference images and LBP images are not effective for pre-trained models.

TABLE VII. FACIAL EXPRESSION RECOGNITION FOR DIFFERENCE IMAGES USING PRE-TRAINED MODELS

	CK+	JAFFE	FACES
Inception-v3	51.65	51.20	63.24
VGG19	82.26	69.60	81.98

TABLE VIII. FACIAL EXPRESSION RECOGNITION FOR LBP IMAGES USING PRE-TRAINED MODELS

	CK+	JAFFE	FACES
Inception-v3	63.64	55.88	68.25
VGG19	88.75	71.76	88.65

In addition to fully training a CNN for FER, we performed experiments using pre-trained models, namely, Inception-v3, VGG19 and VGG-Face. The results for FER using pre-trained models are shown in Table VI. As expected, VGG-Face generally has better accuracy than Inception-v3 and VGG19. This is attributed to the training of VGG-Face, specifically for face classification. However, we note two results in Table VI which warrant further discussion. First, the improvement in the performance of

VGG-Face compared with VGG19 is not significant for JAFFE database. Second, the performance of VGG-Face is slightly lower than that of VGG19 for the FACES database. These results are attributed to the architecture of VGG-Face which is similar to VGG16 rather than VGG19. On the other hand, VGG19 has been shown to be more effective than VGG16 in the 2014 ILSVRC [10]. Therefore, we performed experiments using VGG16 which has comparable architecture to VGG-Face. Table IX shows that VGG-Face is better than VGG16 for FER.

TABLE IX. FACIAL EXPRESSION RECOGNITION USING VGG16 AND VGG-FACE

	CK+	JAFFE	FACES
VGG16	88.27	81.33	95.25
VGG-Face	91.37	86.67	95.06

VII. CONCLUSION

There is extensive literature about CNNs for face recognition; however, classification models based on CNNs have not been thoroughly evaluated for facial expression recognition. In this paper, we evaluated and compared state-of-the-art approaches based on CNNs for FER. The main limitation of training a deep CNN from scratch is that significant amounts of labelled data and resources are needed to minimize overfitting and convergence issues. On the other hand the advantage of training a deep CNN from scratch is that we can choose different appearance images as discussed in Section IIA.

We performed extensive experiments on publicly available face databases. First, we evaluated a CNN by training it from scratch on small databases. Second, the FER results obtained by the CNN were used as benchmark to evaluate pre-trained deep CNN models, including, Inception-v3, VGG and VGG-Face. In literature, VGG-Face has been shown to outperform VGG (VGG16 and VGG19) for face classification; however, in the context of FER, VGG-Face which is pre-trained for face recognition is comparable with VGG which is pre-trained for object recognition.

REFERENCES

[1] O'Toole, A. J., Phillips, P. J., An, X., Dunlop, J. Demographic effects on estimates of automatic face recognition performance, Image and Vision Computing, vol. 30, pp. 169-176, 2012.

[2] Demarkus, M., Gang, K., Guler, S. Automated person categorization for video surveillance using soft biometrics" In Proceedings of SPIE, pp. 76670P-76670P, 2010.

[3] Dantcheva, A., Velardo, C., D'angelo, A., Dugelay, J. L. Bag of soft biometrics for person identification, Multimedia Tools and Applications, vol. 51, pp. 739-777, 2011.

[4] Anil, J., Suresh, L. P. Literature survey on face and face expression recognition, Proc. Int. Conf. Circuit Power Computing Technol., pp. 1-6, 2016.

[5] Viola, P., Jones, M. Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, Vol. 1, pp. I-I, IEEE, 2001.

[6] Suruliandi, A., Meena, K., Rose, R. R. Local binary pattern and its derivatives for face recognition, IET computer vision, vol. 6, pp. 480-488, 2012.

[7] Lu, Z., Jiang, X., Kot, A. Enhance deep learning performance in face recognition, International Conference on Image, Vision and Computing, pp. 244-248, IEEE, 2017.

[8] Pan, S. J., Yang, Q. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, 2010.

[9] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the inception architecture for computer vision, IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818-2826, 2016.

[10] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[11] Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W. J., Li, S. Z., Hospedales, T. M. When Face Recognition Meets with Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition. International conference on computer vision, pp. 384-392, 2015.

[12] Sun, Y., Wang, X., Tang, X. Deep learning face representation, from predicting 10,000 classes. IEEE Conference on Computer Vision, and Pattern Recognition (CVPR), pp. 1891–1898. IEEE, 2014.

[13] Taigman, Y., Yang, M., Ranzato, M., Wolf., L. Deepface:, Closing the gap to human-level performance in face verification. IEEE Conference Computer Vision and Pattern Recognition (CVPR), on, pp. 1701–1708. IEEE, 2014.

[14] Yi, D., Lei, Z., Liao, S. Li, S. Z. Learning face representation from scratch. arXiv preprint arXiv:1411.7923, 2014.

[15] Kanade, T., Cohn, J. F., Tian, Y. Comprehensive database for facial expression analysis, In Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition, pp. 46-53. IEEE, 2000.

[16] Lyons, M. J., Akamatsu, S., Kamachi, M., Gyoba, J. Coding Facial Expressions with Gabor Wavelets. IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, pp. 200-205, 1998.

[17] Ebner, N. C., Riediger, M., Lindenberger, U. FACES—A database of facial expressions in young, middle-aged, and older women and men: Development and validation. Behavior research methods, 42(1), 351-362, 2010.

[18] Gonzalez, R. C., Woods, R. E. Digital Image Processing, 2nd ed., Prentice Hall, 2002.

[19] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016.

[20] Krizhevsky, A., Sutskever, I., Hinton, G. E. ImageNet classification with deep convolutional neural networks. In NIPS, pp. 1106–1114, 2012.

[21] Simonyan K., A. Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 2015.

[22] Parkhi, O. M., Vedaldi, A., & Zisserman, A. Deep Face Recognition. British machine vision conference, 2015.