FACIAL EXPRESSION RECOGNITION WITH DEEP AGE

Zhaojie Luo, Jinhui Chen, Tetsuya Takiguchi, Yasuo Ariki

Graduate School of System Informatics, Kobe University, Kobe, Hyogo, Japan
{luozhaojie, ianchen}@me.cs.scitec.kobe-u.ac.jp, {takigu, ariki}@kobe-u.ac.jp

ABSTRACT

This paper presents a novel deep learning framework for facial expression recognition (FER). Our framework is derived from Convolutional Neural Networks (CNNs), adopting outline, texture, and angle raw data to generate 3 different convolutional feature maps for deep learning. In so doing, the proposed method is cable of robustly classifying expressions, by emphasizing the facial deformation, muscle movement, and outline feature descriptions. Therefore, our method makes the facial expression task not only more tractable, but also more reliable in the case of expression images, which leads to an overall improvement of classification performance compared to conventional methods. The proposed method is valid for the Static Facial Expression Recognition (SFEW) database, improving the baseline results by 6.98%.

Index Terms— facial expression recognition, Convolutional Neural Networks, AGE features

1. INTRODUCTION

Facial expression recognition (FER) is one of the most significant technologies for auto-analyzing human behavior. It can be widely applied to various application domains. Therefore, the need for this kind of technology in various different fields continues to propel related research forward every year.

In this paper, we propose a novel framework adopting the angle, outline, and texture feature maps to describe the facial features. To obtain a discriminative expression classification model, these initial feature data are fed into a CNN for deep learning. There are two main contributions in this study. The first one is that we have designed a deep CNN that is suitable for training small amounts of available labeled data. Second, we found that combining the angle, outline, and texture descriptor together can describe the expression features better than other raw data. Therefore, using this feature repression mechanism for deep learning, we can gain an overall significant improvement of facial expression recognition performance compared to the conventional raw data.

The proposed CNN architecture is derived from VGG net [1], which is a deep visual recognition architecture and the winner in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC). However, due to its deep structure, the original VGG Visual Recognition framework requires massive amounts of labeled training data [2]. Since the existing referenced datasets do not contain large expression data, these approaches are not yet available for expression recognition tasks. Consequently, we need to design a more suitable deep CNN architecture for facial expression recognition.

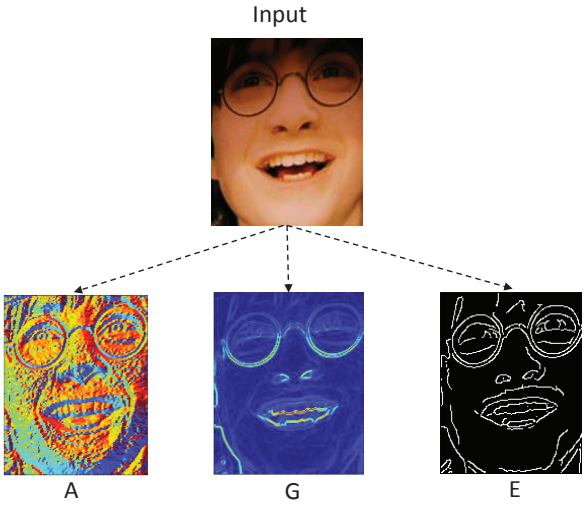

Fig. 1. Input image is converted to 3D AGE features, 'A' means the matrix containing the angle values, 'G' represents the matrix containing the gradient values and 'E' means the edge image.

Within this decade, deep CNN has obtained excellent performance in a wide variety of image classification tasks, and these CNN-based models usually use the RGB color space as input features or perform PCA on the set of RGB pixel values. In contrast, we propose novel transformations of input images to 3D features that adopt the edge to describe outline features, angle to describe the facial deformation features and gradient to indicate the texture for muscle movement representation. In this way, the proposed framework can robustly classify the facial expression classes. Fig. 1 shows the details of 3D feature maps implementation. We call the 3D features AGE features. In our experiments, the results confirmed that using AGE features with a CNN learning platform outperforms other raw data feature maps for FER.

978-1-5386-0560-8/17/$31.00 ©2017 IEEE

In the remainder of this paper, we describe the related work in Section 2. Our proposed method is described in Section 3, and Section 4 gives the experimental results and analysis. Concluding remarks are made in Section 5.

2. RELATED WORK

Facial expression recognition (FER) is a typical multi-class classification problem in Affective Computing and has received increasing interest in the last two decades. Many attempts have been made to recognize facial expressions. Most of the existing work utilizes various human-crafted features, including Gabor wavelet coefficients, Haar features, histograms of Local Binary Patterns (LBP) [3], Histograms of Oriented Gradients (HOG) [4], and scale-invariant feature transform (SIFT) descriptors [5]. Images represented using these features were then classified for different emotions using a Support Vector Machine (SVM) [6] or AdaBoost [7]. Recently, unsupervised feature learning approaches have been employed to extract features from facial images. and these have shown promise in facial expression analysis. To become more adaptable to the unsupervised features, deep learning networks have been employed for FER applications. Tang [8] reported on a deep CNN jointly learned with a linear support vector machine (SVM) output. Liu *et al.* [9] proposed a facial expression recognition framework with a 3D CNN and deformable action parts constraints in order to jointly localize facial action parts and learn part-based representations for expression recognition. The work by Yu *et al.* [10] utilized multiple deep network learning for static facial expression recognition. Similar to some of the models listed above, we propose a deep CNN, and use it to train pre-processed image features, which are the AGE features. AGE features are similar to HOG features, which are rather sensitive to object deformations, but, unlike HOG features, AGE features are 3D features that contain edge images, angle values, and gradient values, which can be trained well by our proposed deep CNN.

2.1. Deep CNNs

Though CNNs were introduced more than three decades ago, it is only recently that they have become a predominant method in image classification tasks. With the increase in computation power, algorithmic improvements, and the fact that current GPUs, when paired with highly-optimized implementation of 2D convolution, are powerful enough to facilitate the training of deep architecture CNNs, the huge number of model parameters is no longer a limiting factor in making use of CNNs in practical settings. Thus, recent CNNs are getting deeper than ever before, (*e.g.*, the VGG-net [1], which uses over 16 layers to build deep CNNs and proved to be capable of achieving record breaking results on a highly challenging database).

Before introducing the architecture of the proposed CNN models, we will describe some important features of the networks architecture.

2.1.1. Max pooling

Pooling layers in CNNs summarize the outputs of neighboring groups of neurons in the same kernel map. In our CNN model, we used max pooling for pooling layers which partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum value. There are two reasons for choosing the max pooling layer. First, it can eliminate non-maximal values to reduce computation for upper layers. Second, max pooling provides additional robustness to position, and it is an effective way of reducing the dimensionality of intermediate representations. In our proposed networks, the window sizes of the pooling layers are set to 3×3 and the strides are both set to 2. This reduces the sizes of the response maps to half after each pooling layer.

2.1.2. ReLU

ReLU is the abbreviation ofRectified Linear Units [11]. This is a layer of neurons that applies the non-saturatingactivation function $f(x) = max(0, x)$. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. It has been proven to be trained much faster in networks than the saturating nonlinearities, such as the hyperbolic tangent $f(x) = \tanh(x)$, and thesigmoid function: $f(x) = (1 + e^{-x})^{-1}$.

2.1.3. Batch-size normalization

Batch-size normalization (BN) was proposed by GOOGLE in 2015 [12].It can avoid the exploding or vanishing caused by a too-high learning rate or by getting stuck in poor local minima. BN normalizes each (scalar) feature independently with respect to the mean and variance of the mini batch. The normalized values are scaled and shifted with two new parameters (per activation) that will be learned. BN makes normalization part of the model architecture. By normalizing activations throughout the network, it prevents small changes in layer parameters from being amplified as the data propagates through a deep network. BN greatly reduces the time required to train CNNs. In particular, the gains are greater when it is combined with higher learning rates. In addition, BN works as a regularizer for a model that allows the use of less dropout or less L2 normalization.

2.2. Feature Representation

There are many researchers who have focused on image representation methods. In the MPEG-7 standard [13], color descriptors consist of a number of histogram descriptors, such

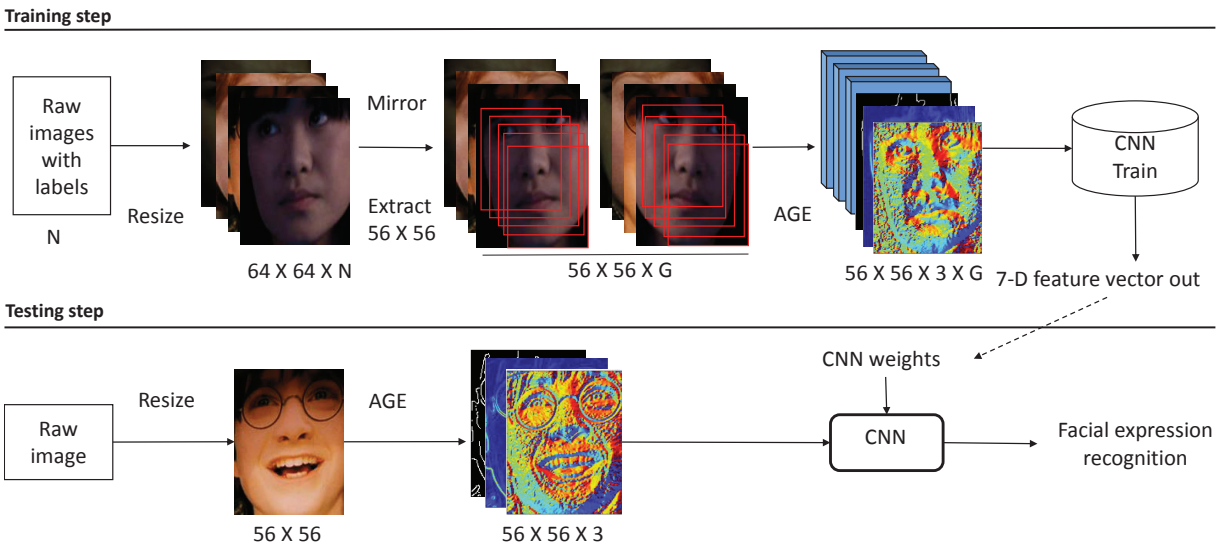

Fig. 2. Overview of the proposed facial expression recognition system. The system is divided in two main steps: training and testing. The training step takes raw images as input and resizes them to 64×64 images. N represents the number of input images for training. To increase the number of input images from N to G, we extract 56×56 patches from 64×64 images and do their horizontal reflections. Here, G equals 8×8×2×N. Then, we process AGE features for each 56×56 image. The transformed 56×56×3×G features preprocessed from the G pictures are used to train the Convolutional Neural Network. The testing step uses the similar methodology as the training step. We resize the input image to 56×56 and transform it to 56×56×3 AGE features for testing. The convolutional networks calculate the scores for the AGE features and the one with the highest score is chosen from seven expressions.

as the dominant color descriptor, the color layout descriptor, and a scalable color descriptor [13, 14]. Researchers have also made wide use of texture descriptors, which provide the important information of the smoothness, coarseness, and regularity of many real-world objects, such as fruit, skin, clouds, trees, bricks, and fabric, *etc.*, including Gabor filtering [15], and local binary pattern (LBP) [16] are two examples of this. Generally, texture descriptors consist of the homogeneous texture descriptor, the texture browsing descriptor, and the edge histogram descriptor. More recently, researchers have begun to combine color descriptors and texture descriptors, such as the multi-texton histogram (MTH) [17] and the micro-structure descriptor (MSD) [18]. They use Gabor features to separately compute the color channels. This is done so that they can combine the color channels with classical texture descriptors to improve the feature describing ability. Inspired by these feature-combination methods, we adopt angle (A), gradient (G) and edge (E), which we refer to as AGE, to describe the outline information, texture information, and geometrical information for faces. For CNN learning models, we have found experimentally that the above input data performance is better than the other raw data (*e.g.*, color data R-B-G) for FER.

3. PROPOSED METHOD

We train the deep CNN models based on cuDNN implementation of a 7-hidden-layers CNN. As shown in Fig. 2, before training the images from SFEW, we resized all images to 64×64. Then, we extract random 56×56 patches and their horizontal reflections from the 64×64 images to increase the size of our training set by a factor of 128. By doing so,it reduces the possibility of over-fitting when using a deep CNN. After that, we transform them to gray-scale, and then preprocess the gray-scale images to 3D features AGE, which is compatible with being trained in our proposed CNN models. The proposed CNNs were trained to predict 7D vectors of emotion class probabilities using the labeled training data that consist of seven different emotion classes. In order to predict the emotion labels, the selected class is the one with the highest probability in the resulting 7D average prediction vector. In the remainder of this section, we will describe the processing of AGE features and our proposed CNN model.

3.1. AGE Features Extraction

Implementing the network shown in Fig. 3, the input images applied recursively to down-sampling layers reduce the computational complexity for upper layers and reduce the dimension of the input, also the network has a 3×3 receptive field that processes the sub-sampled input and output the expres-

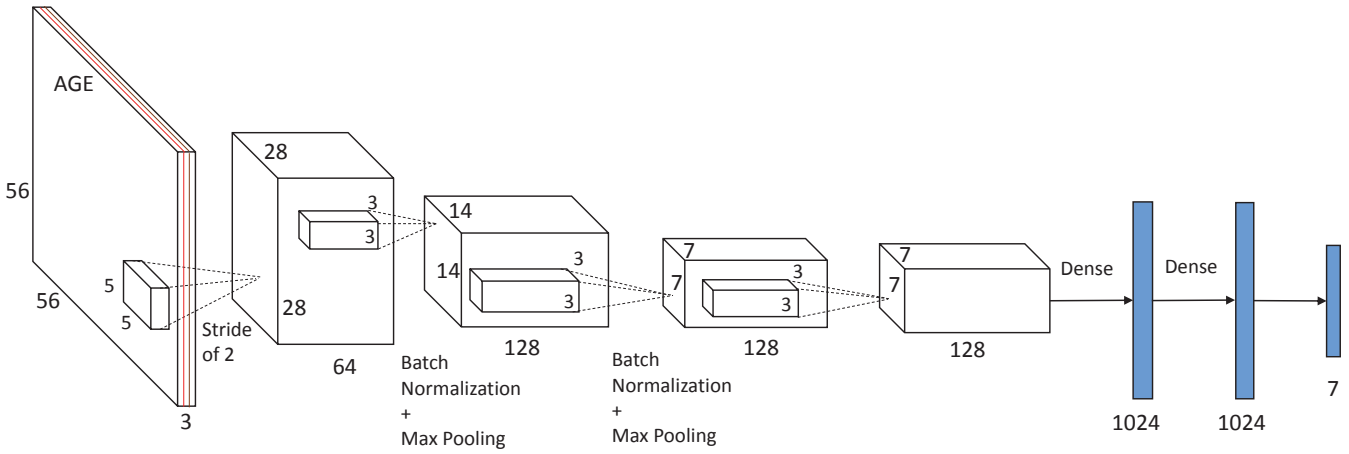

Fig. 3. An illustration of the architeture of our convolutional neural network

sion recognized images.

In this study, the initialized sample images are calculated using an angle filter, gradient filter, and edge filter to obtain the initial convolutional layers:

$$\begin{aligned} v_1 &= I \bigotimes D_a, \\ v_2 &= I \bigotimes D_g, \\ v_3 &= I \bigotimes D_e, \end{aligned} \tag{1}$$

where v is the initial map layer, I donates the sample image, and D_a, D_g, and D_e indicate the angle filter, gradient filter, and edge filter, respectively. In practice, the receptive field size of 5×5 is shown better than the others. These pre-processed data are fed to the following CNN layers for learning.

3.2. Proposed CNN models

A CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume, and the convolutional layer is the core building block of a CNN. The layers of a CNN have neurons arranged in 3 dimensions: width, height, and depth. As shown in Fig. 3, the 3 dimensions of the input layer are 56, 56 and 3. The neurons inside a layer are only connected to a small region of the layer before it, called a receptive field. Distinct types of layers, both locally and completely connected, are stacked to form the CNN architecture. The spatial size of the output volume can be computed as the function below.

$$W_2 = \frac{(W_1 - K + 2P)}{S} + 1 \tag{2}$$

where W_1 is the input volume size, K represents the kernel field size of the Convolutional Layer neurons, S is the stride (the distance between the receptive field centers of neighboring neurons in a kernel map), and P is the amount of zero padding used on the border, respectively. As shown in Fig. 3, $W = 56$, $K = 5$, and $S = 2$, and we set P at 2. So the spatial size of the output volume, which is the input of the second layer, can be calculated with the result being $W_2 = 28$.

An overview of our CNN architecture is depicted in Fig. 3. The net contains seven layers with weights; the first four are convolutional and the remaining three are fully connected layers containing dropout [19]. The output of the last fully-connected layer is fed to a 7-way softmax, which produces a distribution over the 7 class labels. The kernels of all convolutional layers are connected to the previous layer, and neurons in the fully connected layers are connected to all neurons in the previous layer. Batch-normalization [12] layers follow the first and second convolutional layers. the max pooling layers described above follow both batch-normalization layers. The nonlinear mapping functions for all convolutional layers and fully connected layers are set as a rectified linear unit (ReLU) [11]. The first convolutional layer filters the $56 \times 56 \times 3$ image with the 64 kernels of size $5 \times 5 \times 3$ with a stride of 2 pixels. The stride is the distance between the receptive field centers of neighboring neurons in a kernel map, and we set the stride of the filters to 1 pixel for all the other convolutional layers. The input of the second convolutional layer is the output of the first convolutional layer, which is batch-normalized and max pooled. And the second convolutional layer filters the input with 128 kernels of size $3 \times 3 \times 64$. The third convolutional layer has 128 kernels of size $3 \times 3 \times 64$ connected to the outputs of the second layer (batch-normalized and max pooled). The forth convolutional layer has 128 kernels of size $3 \times 3 \times 128$. The fully-connected layers have $1,028$ neurons each.

3.3. Details of Learning

We trained our deep CNNs using a NVIDIA GTX745 4GB GPU. In the CNN learning model, there are some important

parameter settings, including weight decay, momentum, batch size, learning rate, and the cycles of learning. We show the detailed settings of the learning models below. Our training used asynchronous stochastic gradient descent with 0.9 momentum [20], and a weight decay of 0.0005. The update rule for weight w was represented as followed:

$$m_{i+1} = 0.9 \cdot m_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}|_{w_i} \right\rangle_{D_i} \quad (3)$$

$$w_{i+1} = w_i + m_{i+1} \quad (4)$$

where i is the iteration index, m is the momentum variable, ε is the learning rate, and $\left\langle \frac{\partial L}{\partial w}|w_i \right\rangle_{D_i}$ is the average over the ith batch D_i of the derivative of objective with respect to w, evaluated at w_i. Increasing the batch size leads to a more reliable gradient estimate and can reduce the training time, but it does not increase the maximum stable learning rate. So, we need to choose a suitable batch size for our models. We trained our models with batch sizes of 64, 128, 256, and 512, respectively. Then, we find the most suitable batch size for the models. We used an equal learning rate for all layers, which we adjusted manually throughout training. The learning rate was initialized at 0.1, and the heuristic that we followed was to divide the learning rate by 10 when the validation error rate stopped improving with the current learning rate. We trained the network for roughly 20 cycles using the training set images.

4. EXPERIMENTS

The experiments were performed using the Static Facial Expression in the Wild (SFEW) database [11], and using the training and testing methodology described in Fig. 3. The SFEW was assembled by selecting frames from different videos of the Acted Facial Expressions in the Wild (AFEW), and then assigning them one of the following seven emotion labels: angry, disgusted, happy, sad, surprised, fearful, and neutral.

To find the most suitable batch size for the proposed CNN models, we compared the training effectiveness of the different batch sizes. As shown in Fig. 4, the batch size of 256 got best results. Then, we compared effective of AGE features and the original RGB features without any intervention or image pre-processing training by the same models. Fig. 5 compares our method with the color feature descriptors that were implemented for facial expression recognition classifiers with deep CNN models. Real-world images usually do not contain homogenous textures or regular textures; thus, texture filters are usually used to enhance the other feature descriptors for image representation. In many related works, researchers combine color and texture; thus, their describing ability is powerful. But they have ignored local outline representation. This limits their discrimination power. The proposed method adopts data maps, which connect outline, texture, angle, and color data with the CNN learning model. In so doing, the proposed framework has significantly enhanced the representation ability and robustness of features. Therefore, its performance outperforms the conventional raw data and single-type feature descriptors.

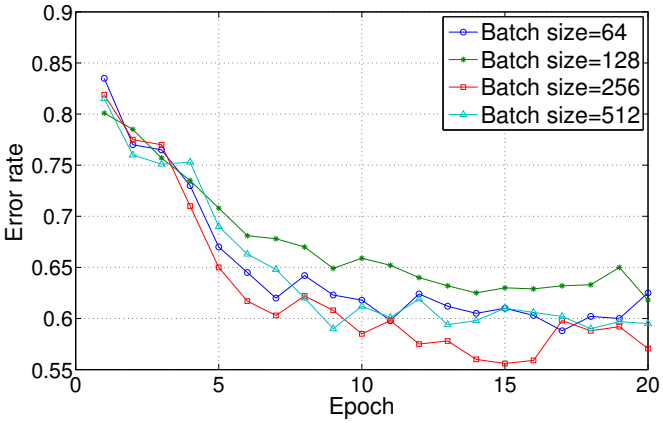

Fig. 4. Training the AGE features with different batchsize 64, 128, 256 and the 512. The horizontal axis means the number of cycles and the ordinate represents the error rates of at each cycles by the training models

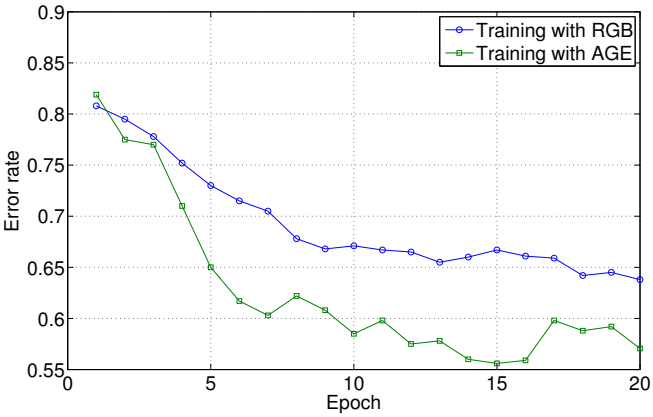

Fig. 5. Training AGE features and RGB features in the proposed CNN models with the 256 batchsize.

5. CONCLUSIONS AND FUTURE WORK

In this study, we proposed a deep classification framework for facial expressions recognition. The proposed framework combines outline, texture, angle, and color data to generate 3 different feature maps for CNN model learning. The proposed classification framework was evaluated with the referenced database (SFEW) to experimentally confirm the validity of the proposed method. The results show these ap-

proaches enhance the discriminative power of the deep classification framework and gain an overall significant improvement in facial expression recognition performance compared to the conventional raw data. These issues are important to those with related research interests.

In future work, we will attempt to apply the proposed framework to the other classification tasks, such as handwriting image recognition.

6. REFERENCES

[1] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[3] Guoying Zhao and Matti Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 915–928, 2007.

[4] Mohamed Dahmane and Jean Meunier, "Emotion recognition using dynamic grid-based hog features," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 2011, pp. 884–888.

[5] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] J. Chen, Y. Ariki, and T. Takiguchi, "Robust Facial Expressions Recognition Using 3 D Average Face and Ameliorated Adaboost," in *Proc. ACM Multimedia Conf. (MM)*, 2013, pp. 661–664.

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

[9] Mengyi Liu, Shaoxin Li, Shiguang Shan, Ruiping Wang, and Xilin Chen, "Deeply learning deformable facial action parts model for dynamic expression analysis," in *Computer Vision–ACCV 2014*, pp. 143–157. Springer, 2014.

[10] Zhiding Yu and Cha Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 435–442.

[11] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[12] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[13] Bangalore S Manjunath, Philippe Salembier, and Thomas Sikora, *Introduction to MPEG-7: multimedia content description interface*, John Wiley & Sons, 2002.

[14] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circ. Syst . Video Tech.*, vol. 11, no. 6, pp. 703–715, Jun. 2001.

[15] B.S. Manjunath and W.Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. PAMI*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. PAMI*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[17] Guang-Hai Liu, Lei Zhang, Ying-Kun Hou, Zuo-Yong Li, and Jing-Yu Yang, "Image retrieval based on multitexton histogram," *Pattern Recognition*, vol. 43, no. 7, pp. 2380 – 2389, 2010.

[18] Guang-Hai Liu, Zuo-Yong Li, Lei Zhang, and Yong Xu, "Image retrieval based on micro-structure descriptor," *Pattern Recognition*, vol. 44, no. 9, pp. 2123 – 2133, 2011.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th international conference on machine learning (ICML-13)*, 2013, pp. 1139–1147.