Improving Facial Emotion Recognition Systems with Crucial Feature Extractors

Ram Krishna Pandey(✉), Souvik Karmakar, A. G. Ramakrishnan, and N. Saha

Department of Electrical Engineering, Indian Institute of Science, Bangalore, India
{ramp,souvikk,agr}@iisc.ac.in, snabagata@nitw.ac.in

Abstract. In this work, we have proposed enhancements that improve the performance of state-of-the-art facial emotion recognition (FER) systems. We believe that the changes in the positions of the fiducial points and the intensities capture the crucial information regarding the emotion of a face image. We propose the inputting of the *gradient* and the *Laplacian* of the input image together with the original into a convolutional neural network (CNN). *These modifications help the network learn additional information from the gradient and Laplacian of the images. However, as shown by our results, the CNN in the existing state-of-the-art models is not able to extract this information from the raw images.* In addition, we employ spatial transformer network to add robustness to the system against rotation and scaling. We have performed a number of experiments on two well known datasets, namely KDEF and FERplus. Our approach enhances the already high performance of the state-of-the-art FER systems by 3 to 5%.

In another contribution, we have proposed an efficient architecture that performs better than the state-of-the-art system on FERplus dataset, with the number of parameters reduced by a factor of about 24. Here also, the fusion of gradient or Laplacian image with the original image improves the recognition performance of the proposed model.

Keywords: Laplacian · Gradient · Convolutional neural network · Facial emotion recognition

1 Introduction

Machine recognition of human emotions is an important and challenging artificial intelligence problem. Human emotions can be recognized from voice [1], body language, facial expression and electroencephalography [2]. However, facial expression forms a simpler and more powerful way of recognizing emotions. Excluding neutral, there are seven types of human emotions that are recognized universally: anger, disgust, fear, happiness, sadness, surprise and contempt. In certain situations, humans are known to simultaneously express more than one emotion. Developing systems for facial emotion recognition (FER) has application

E. Ricci et al. (Eds.): ICIAP 2019, LNCS 11751, pp. 268–279, 2019.
https://doi.org/10.1007/978-3-030-30642-7_24

in areas such as clinical practice, human-computer interaction, behavioural science, virtual reality, augmented reality, entertainment and advanced driver assistant systems. Traditional techniques for FER mainly consist of four successive steps: (i) pre-processing (ii) face and landmark detection (iii) feature extraction and (iv) emotion classification. These approaches heavily depend on the algorithms used for face detection, landmark detection, the handcrafted features and the classifiers used. Recent developments in deep learning reduce the burden of handcrafting the features. Deep learning approaches perform well for all the above-mentioned tasks by learning an end-to-end mapping from the input data to the output classes. Out of all the learning based techniques, convolutional neural network (CNN) based techniques are preferred.

There is a tendency in researchers to design deep neural networks (DNN's) as end to end systems, where every kind of processing is accomplished by the network, including the feature extraction, by learning from the data. Some opine that there is no need for any hard-coded feature extraction in any machine learning system. However, the deep neural networks have been designed to simulate the biological neural network in the brain. It is well known that there are many hard-coded feature extractors in the human brain, and even animal sensory systems, in addition to the natural neural network, that also learns from data (exposure and experience). One might argue that it is possible for the visual neural pathway or cortex to extract the gray image from the colour image obtained by the cones in the retina. However, nature has chosen to have many more rods than cones to directly obtain the gray images also in parallel. Further, the work of Hubel and Wiezel [3] showed the existence of orientation selective cells in the lateral geniculate nucleus and visual cortex of kitten. Also, different regions of the basilar membrane in the cochlea respond to different frequencies [4] in both man and animals and this processing is akin to sub-band decomposition of the input audio signal. Thus, there are many examples of hard-coded feature extraction in the brain, which enhance the classification potential of the biological neural network; our work reported here, is inspired from this aspect of nature's processing.

2 Related Work

Darwin and Phillip suggested that human and animal facial emotions are evolutionary [5]. Motivated by Darwin's work, Ekman et al. [6,7] found that the seven expressions, namely happiness, anger, fear, surprise, disgust, sadness and contempt remain the same across different cultures. Facial action coding system (FACS) is proposed in [8] to investigate the facial expressions and the corresponding emotions described by the activity of the atomic action units (cluster of facial muscles). Facial expression can be analyzed by mapping facial action units for each part of the face (eyes, nose, mouth corners) into codes.

2.1 Traditional Approaches

Features are desired that possess maximal inter-class and minimal intra-class variabilities for each of the expressions. Traditional systems for facial emotion recognition depend mainly on what and how the features are extracted from the facial expression. The features extracted can be categorized into (i) geometric features, (ii) appearance based features or (iii) their combination. In the work reported by Myunghoon et al. [9], facial features are extracted by active shape model, whereas Ghimire and Lee [10] extract geometric features from the sequences of facial expression images and multi-class Ada-boost and SVM classifiers are used for classification. Global face region or regions containing different facial information are used to extract appearance-based features. Gabor wavelets, Haar features, local binary pattern [11] or its variants such as [12] are used to extract appearance-based features. Ghimire et al. [13] proposed a single frame classification of emotion using geometric as well as appearance based features and SVM classifier. In [14], features are extracted using pyramid histogram of orientation gradients. Here, the facial edge contours are constructed using Canny edge detector. Histograms are calculated by dividing the edge maps into different pyramid resolution levels. The histogram vectors are concatenated to generate the final feature to be used for classification using SVM or AdaBoost classifier.

2.2 Deep Learning Based Approaches

The above techniques in the literature depend heavily on handcrafted features. However, deep learning algorithms have shown promising results in the recent years. CNN based models have shown significant performance gain in various computer vision and image processing tasks, such as image segmentation, denoising, super-resolution, object recognition, face recognition, scene understanding and facial emotion recognition. Unlike the traditional techniques, deep learning based techniques learn ("end-to-end") to extract features from the data. For FER, the network generally uses four different kinds of layers, namely convolution, max-pool, dense layer and soft-max. Batch normalization with skip connection is also used to ease the training process. The features extracted have information about local spatial relation as well as global information. The max-pool layer makes the model robust to small geometrical distortion. The dense and soft-max layers help in assigning the class score.

Breuer and Kimmel [15] demonstrate the capability of the CNN network trained on various FER datasets by visualizing the feature maps of the trained model, and their corresponding FACS action unit. Motivated by Xception architecture proposed in [16], Arriaga et al. [17] proposed mini-Xception. Jung et al. [18] proposed two different deep network models for recognising facial expressions. The first network extracts temporal appearance features, whereas the second extracts temporal geometric features and these networks are combined and fine tuned in the best possible way to obtain better accuracy from the model. Motivated by these two techniques, we have trained and obtained multiple models, the details of which are explained in Sect. 5.

For the task of FER, the current state of the art model [24] proposed a miniature version of VGG net, called VGG13. The network has 8.75 million parameters. The dataset used is the FERplus dataset [24], which has 8 classes, adding neutral to the existing seven classes. The reported test accuracy is ≈84%. In 2014, Levi et al. [19] obtained improved performance of emotion recognition using CNN. They convert images to local binary patterns (LBP). These patterns are mapped to a 3D metric space and used as input to the existing CNN architectures, thus addressing the problem of appearance variation due to illumination. They trained the existing VGG network [20], on CASIA Webface dataset [33], and then used transfer learning to train the static facial expressions in the wild (SFEW), to address the problem of the small size of SFEW dataset [34].

Ouellet [21] used a CNN based architecture for realtime detection of emotions. The author uses transfer learning to train the Cohn-Kanade [22] dataset on AlexNet. The author used the model to capture the emotions of gamers, while they are playing games.

3 Datasets Used for the Study

We have used the KDEF [23] and FERplus [24,25] datasets for our experiments. The FERplus dataset contains nearly 35000 images divided into 8 classes, including contempt. The FERplus dataset improves upon the FER dataset by crowdsourcing the tagging operation. Ten taggers were asked to choose one emotion per image, which resulted in a distribution of emotions for each image. The training set contains around 28000 images. The remaining are divided equally into validation and test sets. The original image size is 48×48 pixels. Figure 1 shows some sample face images from the FERplus dataset, with multiple emotion labels for each image. KDEF dataset contains a total of 4900 images (divided into the 7 classes of neutral, anger, disgust, fear, happiness, sadness, and surprise), with equal number of male and female expressions. Figure 2(a), (b) and (c) show, respectively, a sample input image from KDEF dataset, its derivative image obtained by the Sobel operator (gradient) and its second derivative obtained by the Laplacian operator.

Fig. 1. Face image samples from FERplus dataset, with multiple emotion labels for each image [24]

Fig. 2. (a) A sample input image from the KDEF dataset [23]. (b) Its derivative image obtained by the Sobel operator (Gradient). (c) Its second derivative obtained by the Laplacian operator. Zoom to see the details in the Laplacian image.

4 The Proposed Models and Our Contributions

Rather than proposing a totally new architecture, which performs marginally better than the state-of-the-art model, one approach could be to work on good existing models and propose enhancements to significantly improve their performance. Another approach could be to come out with a computationally efficient model that performs as good as the state of the art models. We propose that the performance of a classifier for facial emotion recognition can be improved by making it robust to transformations such as scaling and rotation. The fiducial points of a face change predictably, depending upon the specific emotion and these changes are the important features for emotion recognition. Such changes in the image landmark points and their intensities can be effectively captured by the gradient and Laplacian of an image. Thus, in our first approach, we have significantly improved the emotion recognition performance of two state-of-the-art architectures by adding the following enhancements [35].

Spatial Transformer Layer (STL): CNN is a very powerful model, invariant to some transformations like in-plane rotation and scaling. To obtain such invariance, CNN requires a huge amount of training data. To achieve such invariance in a computationally efficient manner, spatial transformer network [26] is used as the input layer, called here as the spatial transformer layer. This allows spatial manipulation of the data within the network. This differentiable module, when combined with the CNN, infuses invariance to rotation, scaling, and translation, with less training data than that needed by the normal CNN.

Sobel and Laplacian Operators: The gradient captures information such as the direction of the maximum change and the Laplacian identifies regions of rapid changes in the intensity. Thus, by adding the gradient and Laplacian images as additional inputs, we can largely obviate the need for extracting the fiducial or the landmark points. The gradient and Laplacian of an image $f(x, y)$, denoted by $\Delta f(x, y)$ and $\Delta^2 f(x, y)$, can be approximated by applying Sobel [27] and Laplacian [28] operators on an image. We have taken the input images from the dataset and applied Sobel and Laplacian operators on them to obtain their first and second derivatives, respectively. They detect the intensity discontinuities as

contours. These images are fed as inputs, in series or parallel with the original image, into the state of art models for FER.

Global Average Pooling and DepthSep Layers: The real time convolutional neural network (RTNN) architecture selected by us [17] uses global average pooling (GAP) [29]. The GAP layer has multiple advantages over the dense layer: (i) it reduces over-fitting to a large extent; (ii) huge reduction in the number of parameters compared to dense layer; (iii) the spatial average of feature maps is fed directly to the soft-max layer. The latter enforces better correspondence between the feature maps and the categories.

The RTNN model also employs depthwise separable convolution (DepSep) layers [16]. The advantage of using depthwise separable convolution layer is that it greatly reduces the number of parameters compared to the convolution layer. At a particular layer, let the total number of filters be N, the depth of the feature maps be D, and the size of the filter (spatial extent) used be S_e. In such a case, the total number of parameters in normal convolution is $S_e \times S_e \times D \times N$. DepSep is a two-step process: (i) filters of size $S_e \times S_e \times 1$ are applied to each feature; therefore, the total number parameters at this step is $S_e \times S_e \times D$; (ii) then, N filters of size $1 \times 1 \times D$ are applied. So, the number of parameters required at this step are $D \times N$. Combining steps (i) and (ii), the total number of parameters in DepSep layer are $S_e \times S_e \times D + D \times N$. Hence, the reduction in the number of parameters compared to normal convolution at each layer, where convolution is replaced by DepSep convolution, is: $\frac{S_e \times S_e \times D + D \times N}{S_e \times S_e \times D \times N} = (1/N) + (1/S_e^2)$

Our Contributions: In this work, our main contributions are:

- By adding the spatial transformer layer as the input processing block, we have introduced robustness to scaling, rotation and translation.
- By adding the gradient and/or Laplacian image(s) as additional inputs to the system, we have improved the recognition accuracies of three different FER architectures by a good margin (see Tables 1, 2 and 4).
- We have trained multiple models to validate the performance gain obtained due to the addition of gradient and Laplacian, on KDEF [23] and FER-Plus [24] datasets.
- We have proposed an efficient architecture (refer Table 3) that performs better than the state-of-the-art system on FERplus dataset (refer Table 4), while reducing the number of parameters by a factor of 24.
- Our proposed architecture (reported in Table 3) has model size of around 9.7 MB compared to the original VGG-13 [24], which has the model size of 107.4 MB. Thus, our model can be run on a mobile phone more efficiently.

5 Experiments and Results

Three sets of different experiments have been carried out.

Experiment 1: In the first set of experiments, the *Real-time neural network (RTNN)* model, with all the modifications proposed by us, has been tested on

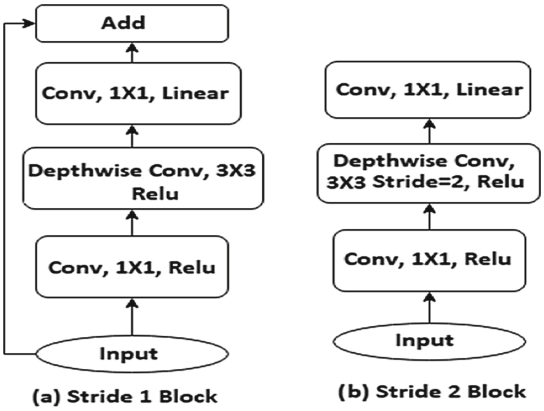

Fig. 3. Inverted bottleneck module used in MobileNetV2 [31].

Table 1. FER results of RTNN and its various modifications proposed by us (parallel networks) on the KDEF dataset (4900 images with 7 classes).

Architecture details	Accuracy %
Orig. RTNN by Arriaga et al. [17]	**83.16**
STL + RTNN	84.08
RTNN + Lap. RTNN	84.39
STL with RTNN + Grad RTNN	85.10
STL with RTNN + Lap RTNN	85.51
STL with Orig., Grad and Lap. RTNN	**88.16**

the KDEF dataset. RTNN is the model proposed by Arriaga et al. [17], trained on the KDEF dataset and validated. Table 1 reports the results of the experiments conducted. *STL + RTNN* is the RTNN model trained with the addition of STL at the input. *RTNN + Lap. RTNN* is the architecture, where the input image and its Laplacian are fed in parallel. The outputs of these parallel networks are combined and passed to a soft-max layer for classification. *STL with RTNN + Grad RTNN* is the case when the input image and its gradient are first fed to a STL, followed by the parallel subnetworks. The parallel networks extract more useful features in the beginning layer, which are combined to obtain better accuracy. *STL with RTNN + Lap. RTNN* is the architecture, where the model is trained in parallel with the input image and its Laplacian. *STL with Orig., Grad & Lap. RTNN* is the case, where the model is trained in parallel with the original, the gradient, and the Laplacian images. These input streams are first fed independently to a STL, before being fed to the subnetworks in parallel.

Experiment 2: We have reimplemented the VGG13 network, used in [24], in Tensorflow. We use the majority voting technique, as described in [24], for labelling each image. The only modification we have made to the original model

is the use of Adam optimizer [30] instead of momentum optimizer. Table 2 compares the results of the original network with those after our enhancements. In our setup, we get an average accuracy of 83.56% instead of 83.85% as reported in the original paper. Next we propose two experimental setups. First, we modify the input by taking the Laplacian of the original image and channel wise concatenating it with the original image. The resultant image is a 2 channel 64*64 input. In this setup, without modifying the learning rate, we get an average accuracy improvement of close to 3% on an average, compared to our VGG13 implementation. In the second setup, we use Sobel operator instead of Laplacian, and get gradients in x and y directions. The resultant gradients are again concatenated to the original image channelwise to get 3 channel input. This setup again gives an improvement of close to 3%.

Table 2. FER accuracies on FERPlus dataset and the number of parameters (in millions) of our models vs. VGG13 [24]. Each type of model has been trained 4 times and its maximum, minimum and average accuracies are reported. Training set: 28000 images; validation and test sets: 3500 images each; number of classes: 8.

Models	Avg	Min	Max	Parameters
VGG13 (reported)	83.85	83.15	84.89	8.75
VGG13 (our implementation)	83.56	82.99	84.08	8.75
VGG13 + Laplacian (input concatenated)	86.22	85.94	86.56	8.75
VGG13 + Sobel (input concatenated)	86.42	86.08	86.55	8.75

Experiment 3: We propose our own architecture (details listed in Table 3, having (1/24)-th the number of parameters compared to all the architectures reported in Table 2), developed using inverted bottleneck module (refer Fig. 3) reported in [31]. Even in this model, fusion of the Laplacian or the gradient image to the input image (by concatenation) enhances the recognition performance by 2.3 and 2.5 %, respectively. The results with this proposed model and its feature-fusion enhancements are listed in Table 4. The *base+Sobel* model performs better than the original VGG13 model listed in Table 2 by 0.62%.

6 Conclusion

We have shown that feeding the gradient and/or Laplacian of the image, in addition to the input image, improves the performance of any FER system. We have performed many experiments on KDEF and FERplus datasets and enhanced the recognition accuracies of state-of-the-art techniques [17,24]. We believe that our proposed approach will largely impact the community working on similar area. The advantages of our proposal are many folds: (i) improves the recognition accuracy of any classifier (ii) the dataset size increases by two or three times (depending on the Laplacian or/and gradient used together with the

Table 3. Details of the architecture proposed by us, with *inverted bottleneck* 3 as the core module. c, s and t denote the number of output channels from each layer, stride and expansion factor used in the bottleneck module, respectively.

Layer	Parameters	Input ($H \times W \times C$)	c	s	t
*conv*2*d* 3×3	64	$64 \times 64 \times 1$	64	2	–
bottleneck	6720	$64 \times 64 \times 32$	32	1	1
bottleneck	12480	$32 \times 32 \times 32$	24	2	6
bottleneck	8208	$16 \times 16 \times 24$	24	1	6
bottleneck	9360	$16 \times 16 \times 24$	32	2	6
bottleneck	14016	$8 \times 8 \times 32$	32	1	6
bottleneck	14016	$8 \times 8 \times 32$	32	1	6
bottleneck	20160	$8 \times 8 \times 32$	64	1	6
bottleneck	52608	$8 \times 8 \times 64$	64	1	6
bottleneck	52608	$8 \times 8 \times 64$	64	1	6
bottleneck	52608	$8 \times 8 \times 64$	64	1	6
bottleneck	77184	$8 \times 8 \times 64$	128	1	6
*conv*2*d* 1×1	40960	$8 \times 8 \times 128$	320	1	–
avg_pool	0	$8 \times 8 \times 320$	320	–	–
*conv*2*d* 1×1	2048	$1 \times 1 \times 320$	8	1	–
Total	363616	–	–	–	–

input), which is desirable in most deep learning tasks; (iii) the variability in the input image space increases (iv) DepSep, inverted bottleneck module and GAP layers help in reducing the computational complexity of the model. The proposed enhancements result in absolute performance improvements, as listed in Tables 1, 2 and 4, over those of the original models. Researchers working on similar areas can use our proposed features to add performance gain to any existing DNN based classifier, thus obviating the need for designing a new classifier to achieve similar performance gain. We have also proposed an efficient architecture that performs better than the state of the art algorithm proposed in [24] with (1/24)-th the number of parameters. Thus, if there is any need for designing any new

Table 4. FER performance of our models, with less complexity than VGG13, on FERPlus [24] dataset. Models are trained 4 times with the same hyper-parameter settings and their average, maximum and minimum accuracies are reported.

Models	Avg	Min	Max	Parameters
Base model (given in Table 3)	81.95	81.79	82.15	0.36 million
Base + Laplacian (input concatenated)	84.26	83.84	84.87	0.36 million
Base + Sobel (input concatenated)	84.47	84.21	84.69	0.36 million

classifier, it can be made computationally efficient to a good extent with our proposed approaches.

One might argue that the gradient and Laplacian of the input image can very well be computed by the CNN. However, there are strong evidences for the need for appropriate representations in accomplishing certain vision and motor control tasks [32]. It is also clear from the results that at least the networks proposed by Arriaga et al. [17] and VGG13 [24] are not able to compute these derived images internally. On the other hand, pre-computing these features and feeding them to the same network in parallel or in series, is clearly able to improve the performance of the network. Thus, our experiments show that there is a clear case for optimally combining appropriate feature extractors with learning neural networks, to obtain better performance for specific pattern recognition tasks.

References

1. El Ayadi, M., Kamel, M.S., Karray, F.: Survey on speech emotion recognition: features, classification schemes, and databases. Pattern Recogn. **44**(3), 572–587 (2011)
2. Davidson, R.J., Ekman, P., Saron, C.D., Senulis, J.A., Friesen, W.V.: Approach-withdrawal and cerebral asymmetry: emotional expression and brain physiology: I. J. Pers. Soc. Psychol. **58**(2), 330 (1990)
3. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. **160**(1), 106–154 (1982)
4. Greenwood, D.D.: A cochlear frequencyposition function for several species-29 years later. J. Acoust. Soc. Am. **87**(6), 2592–2605 (1990)
5. Darwin, C., Prodger, P.: The Expression of the Emotions in Man and Animals. Oxford University Press, Oxford (1998)
6. Ekman, P., Friesen, W.V.: Nonverbal leakage and clues to deception. Psychiatry **32**(1), 88–106 (1969)
7. Ekman, P., Dacher, K.: Universal facial expressions of emotion. Calif. Ment. Health Res. Digest **8**(4), 151–158 (1970)
8. Friesen, E., Ekman, P.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto (1978)
9. Suk, M., Prabhakaran, B.: Real-time mobile facial expression recognition system-a case study. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition Workshops (2014)
10. Ghimire, D., Lee, J.: Geometric feature-based facial expression recognition in image sequences using multi-class adaboost and support vector machines. Sensors **13**(6), 7714–7734 (2013)
11. Happy, S.L., George, A., Routray, A.: A real time facial expression classification system using local binary patterns. In: 4th International Conference on Intelligent Human Computer Interaction (IHCI), pp. 1–5. IEEE (2012)
12. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 915–928 (2007)
13. Ghimire, D., Jeong, S., Lee, J., Park, S.H.: Facial expression recognition based on local region specific features and support vector machines. Multimed. Tools Appl. **76**(6), 7803–7821 (2017)

14. Bai, Y., Guo, L., Jin, L., Huang, Q.: A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In: International Conference on Image processing (ICIP) (2009)
15. Breuer, R., Kimmel, R.: A deep learning perspective on the origin of facial expressions. arXiv preprint arXiv:1705.01842 (2017)
16. Chollet, F.: Xception: deep learning with separable convolutions. arXiv Preprint arXiv1610 2357 (2016)
17. Arriaga, O., Valdenegro-Toro, M., Ploger, P.: Real-time convolutional neural networks for emotion and gender classification. arXiv preprint arXiv:1710.07557 (2017)
18. Jung, H., Lee, S., Yim, J., Park, S., Kim, J.: Joint fine-tuning in deep neural networks for facial expression recognition. In: Proceedings IEEE International Conference on Computer Vision (2015)
19. Levi, G., Hassner, T.: Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In: Proceedings ACM International Conference on Multimodal Interaction (ICMI), November 2015
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. Ouellet, S.: Real-time emotion recognition for gaming using deep convolutional network features. arXiv preprint arXiv:1408.3750 (2014)
22. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Workshops, San Francisco, CA (2010)
23. Lundqvist, D., Flykt, A., Öhman, A.: The Karolinska directed emotional faces (KDEF). CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, pp. 91–630 (1998)
24. Barsoum, E., Zhang, C., Ferrer, C.C., Zhang, Z.: Training deep networks for facial expression recognition with crowd-sourced label distribution. In: Proceedings of the 18th ACM International Conference on Multimodal Interaction. ACM (2016)
25. https://github.com/Microsoft/FERPlus/
26. Jaderberg, M., Simonyan, K., Zisserman, A.: Spatial transformer networks. In: Advances in Neural Information Processing Systems (2015)
27. Sobel, I., Feldman, G.: A 3×3 isotropic gradient operator for image processing. A talk at the Stanford Artificial Project in 271–272 (1968)
28. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision. Addison-Wesley, Boston (1992)
29. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
30. Kingma, D.P., Ba, L.J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
31. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv 2: inverted residuals and linear bottlenecks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)
32. Schomaker, L.: Anticipation in cybernetic systems: a case against mindless anti-representationalism. In: International Conference on Systems, Man and Cybernetics, vol. 2. IEEE (2004)
33. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)

34. Dhall, A., Ramana Murthy, O.V., Goecke, R., Joshi, J., Gedeon, T.: Video and image based emotion recognition challenges in the wild: EmotiW 2015. In: Proceedings International Conference on Multimodal Interaction, pp. 423–426. ACM (2015)
35. Krishna Pandey, R., Karmakar, S., Ramakrishnan, A.G., Saha, N.: Improving facial emotion recognition systems using gradient and Laplacian images. arXiv preprint arXiv:1902.05411 (2019)