Facial Expression Recognition Using Multi-deep Convolutional Neural Network Encoders with Support Vector Machines

Tran Ngoc Dong, Le Van, and Pham The Bao

Abstract—**Although there have been many breakthroughs in the use of convolutional neural networks (CNN) for image classification, facial expression recognition (FER) in real-life is still a challenge in this research area. This paper proposes a method to leverage state-of-the-art multi-deep CNN encoders with support vector machines (SVM) to classify facial expression. We conducted experiments to show that combining features from multi-deep CNN is better than using a single deep CNN model. As well as combining multiple CNN models, we show that using rules to remove noise images from the training dataset improves the performance of the FER system. The FER2013 dataset was used to evaluate the proposed approach, which achieved 73.78% accuracy.**

Index Terms—**Convolutional neural networks, deep convolutional neural network features, facial expression recognition in the wild, FER2013.**

I. Introduction

Facial expression is one of the most useful sources of information in understanding human feelings. To take advantage of this information, many researchers in different fields are trying to analyze facial representation using a variety of approaches [1, 2]. In the real world, many commercial services or applications need to obtain feedback from customers to improve the service or application; customer facial expression is one of the most useful pieces of information in evaluating feed-back. With tremendous increases in customer numbers, an automatic system to extract expressions has become a more efficient method than manual processing.

Particularly in the computer vision field, many studies have proposed an approach to an automatic facial expression recognition (FER) system using methods from computer vision, machine learning, and deep learning [1], [2]. The use of FER system in real-world scenarios is still a significant challenge due to variations in illumination, occlusion, head pose, and unclear expressions. These issues make a solution based on handcrafted features unsuitable for the tremendous diversity of facial expressions in the wild.

Unlike some machine learning methods where features are designed by hand, a convolutional neural network (CNN) model can auto-capture features from input datasets, which means that this approach can effectively extract general features from various input data. Along with the growth of computational resources, such as graphics processing unit (GPU), many breakthroughs in deep CNN architecture have been emerged that could address the classification problem [3]-[9]. Also, combining a support vector machines (SVM) [10] with deep CNN could offer a potential solution to this problem [11]-[13].

This paper is structured as follows: we review related work in Section II, explain the proposed framework in Section III, describe experiments and detail the results in Section IV, and provide our conclusion in Section V.

II. Related Work

This section looks at methods using CNN and their variants in FER that achieve good results with the FER2013 dataset.

Goodfellow *et al*. introduced a completely new FER for real-word conditions at the ICML 2013 Workshop: Facial Expression Recognition 2013 (FER-2013) [14].

Tang proposed a method replacing the softmax layer in the CNN architecture with L2-SVM and using margin-based loss instead of cross-entropy loss; this method helped Tang to win first place in the ICML 2013 challenge [11].

Yu and Zhang proposed a face detection module containing three state-of-the-art face detectors: mixtures of trees (MoT), joint cascade detection and alignment (JDA) and deep-CNN-based (DCNN); it has an architecture similar to a "cascade" flow, which can reduce false positives over multiple-stages. They used a simple deep CNN as an emotion classifier [15].

Mollahosseini *et al*. proposed a deep CNN architecture based on an Inception module, which can give the CNN architecture additional convolutional layers without too great an increase in computational cost [16].

Connie *et al*. studied the impact of a scale invariant feature transform (SIFT) feature and dense SIFT as external features for CNN models. Unlike some previous methods using SIFT or dense SIFT of the face region as the input for the CNN model, they used a single fully connected layer to combine SIFT/dense SIFT features and CNN features. To improve the final result, they aggregated three outputs from CNN, CNN with SIFT, and CNN with dense SIFT by using average sums [17].

In 2018, Nguyen *et al*. enhanced the VGG [5] architecture by making it automatically select important mid- and high-level features based on their contribution and by using

Manuscript received April 8, 2020; revised December 20, 2020.

Tran Ngoc Dong is with Information Science Faculty, Univeristy of Information Technology, Vietnam National University, Hochiminh City, Vietnam (e-mail: dongtn.11@grad.uit.edu.vn).

Le Van and Pham The Bao is with Information Science Faculty, Sai Gon University, Hochiminh City, Vietnam (Corresponding author: Pham The Bao; e-mail: levan1461995@gmail.com, ptbao@sgu.edu.vn).

doi: 10.18178/ijmlc.2021.11.5.1059

the grad-CAM method to analyze information from the convolution filters [18].

Recently, Minaee and Abdolrashidi applied attention mechanisms in CNN to build models that could automatically localize and focus on important parts of the face. They also visualized which region the model should focus on and concluded that parts like eyes, lips, eyebrows, and forehead are more influential in facial expression than other regions of the human face [19].

Hua *et al*. proposed a method for integrating three weak deep CNN architectures to form a strong classifier. They showed that, when combined, three networks with accuracy of around 68% could achieve 71.91% on the FER2013 dataset [20].

In 2013, Ionescu *et al*. proposed a method using the FER2013 dataset based on dense SIFT descriptors for a bag-of-visual-words representation with a local SVM to classify emotion, achieving 67.48% accuracy [12]. In 2019, some of the same authors (Georgescu *et al*.) concatenated features from three deep CNN using VGG-face [21], VGG-f [22], and VGG-13 [5] for a bag-of-visual-words representation to train a local SVM in three datasets: FER2013, FER+ [23], and AffectNet [24]. They achieved 74.92% accuracy without augmentation and 75.42% with augmentation in the private test of FER2013. These results show that increased accuracy comes from using combined deep CNN and a local SVM. In addition, they showed that the highest accuracy that an individual deep CNN model can achieve is 72.11% [13].

As with the attention mechanisms, Wang *et al*. proposed an algorithm to extract features from the eyes, nose, and mouth regions and considered those three subregions as important for extracting information on emotion recognition. They designed an auxiliary model to capture this information automatically and combine it with full-face information from CNN models by weighting; evaluating feature information that ismore important for the final result [25].

Unlike previous methods, using a classifier model to address the FER problem, AlMarri proposed a detection approach based on fast R-CNN [26] with VGG-16 [5] as a backbone. However, they only achieved 30.19% accuracy in emotion recognition with FER2013 [27].

III. Methodology

A. Overview

Before the training stage, we manually removed un-wanted images from the training dataset, as this can help the CNN model to avoid learning features from noise.

To train a linear SVM to classify facial emotion, the facial images were fed into four different trained CNN models. The last pooling layer was extracted as a feature vector and concatenated to create a single feature vector to represent facial emotion information as input to the SVM model.

The deep CNN models for feature extraction were trained from scratch using four different architectures: VGG by Simonyan and Zisserman [5], ResNet by He *et al*. [6], pre-activation ResNet (PreAct ResNet) by He *et al*. [7], and ResNeXt by Xie *et al*., [9]. The full proposed pipeline is shown in Fig. 1.

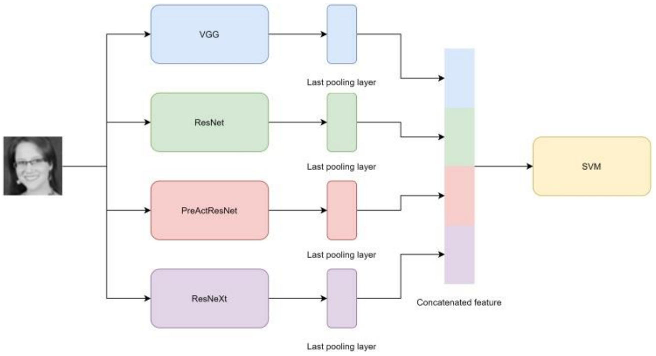

Fig. 1. Proposed pipeline for FER system using multi-deep CNN and SVM.

B. Fusion Deep CNN Feature

This section briefly describes the contribution of the four chosen CNN architectures:

- VGG: By using only 3×3 convolution filters instead of 5×5 or 7×7 convolution filters as in other methods, this model could increase the depth of the CNN up to 19 weight layers without too great an increase in computational parameters. By increasing the number of captured features via the number of filters in the model, the VGG model offered state-of-the-art performance in image classification when it was proposed [5].
- ResNet: With the increased depth of CNN models , we face a degradation problem, where the gradient in the backpropagation flow becomes too small to affect the low-level convolution filter, making the CNN model harder to train. To overcome this problem, He *et al*. reformulated the convolution block with identity mapping as a shortcut connection to present a new residual block. With a new residual block, the ResNet model with 152 weight layers achieved first place in ILSVRC 2015 [6].
- PreAct ResNet: Based on the ideal of residual layers, He *et al*. analyzed the information flow of the residual block. They used batch normalize and rectified linear unit (ReLU) for "pre-activation" before each residual block, forming "pre-activation residual units". With the new residual blocks, they created a 1001-layer ResNet model that is easier to optimize and reduces the overfitting problem. The result surpassed ResNet and other previous deep CNN architectures in image classification [7].
- ResNeXt: Recently, the deeper CNN models have gained in accuracy, but at increased computational cost, even with a small 3×3 convolution filter size. To overcome this problem, Szegedy proposed a method to decompose $n \times n$ filters to multiple smaller convolution filters, enabling a deeper model to run at low computational cost while maintaining high quality [4, 8]. In addition, Krizhevsky *et al*. showed that groups of convolution filters with different information flows could learn different features from the same input data, so that the CNN model is capable of fitting diverse input data [3]. To take advantage of these ideas, Xie *et al*. constructed a ResNeXt block that has a multi-branch with the nearly same computational cost as a ResNet block but with an increased overall convolution filter number and group convolution without a significant increase in computational cost [9].

For the deep CNN feature vector, we fed an input image to each trained model and extracted information from the last pooling layer as a feature vector. We concatenated the four different feature vectors from four deep CNN models in the sequence shown in Fig. 1 to obtain a multi-deep CNN feature vector to represent the single image.

There are four regions in the face that contribute information on facial emotions: the forehead, the two eyes, and the mouth. Like [19], [25], we needed different CNN models to extract different features to “see” different information. Previous methods, such as [13], [20], [28], used an ensemble of three similar deep CNN architectures. However, based on the idea of four major information regions for facial emotion, we combined four different deep CNN architectures to show the impact of the ensemble in a different deep CNN model than previously proposed. We also considered processing time and computation costs when increasing the number of deep CNN models

C. Emotion Classifier

A linear SVM [10] was trained with the multi-deep CNN feature vector to classify the seven facial expressions using a sequential minimal optimization method [29].

We also tried different classifiers in the machine learning field, including k-nearest neighbor (k-NN) [30], decision trees [31], random forests [32], and na ïve Bayes [33], as well as SVM with other kernels including poly, RBF, and sigmoid [34]. Some of these give a combined result that is better than linear SVM, but we found from our experiments that a linear SVM is the fastest at predicting one image, which is important for a real-time FER system.

IV. Experiment and Results

A. Dataset and Implementation Detail

We used the FER challenge dataset (FER2013) [14] to evaluate our method. This dataset consists of 28,709 images in the training set, 3,589 images in the public set and 3,589 images in the private set; all images are 48x48 pixel grayscale images. The dataset contains seven facial expressions: angry, disgust, fear, happy, sad, surprise, and neutral, as illustrated in Fig. 2. Table I provides details of the emotions in the FER-2013 dataset.

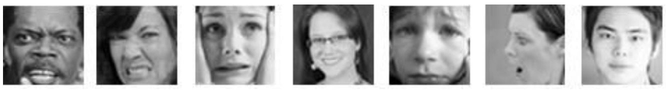

Fig. 2. FER2013 examples.

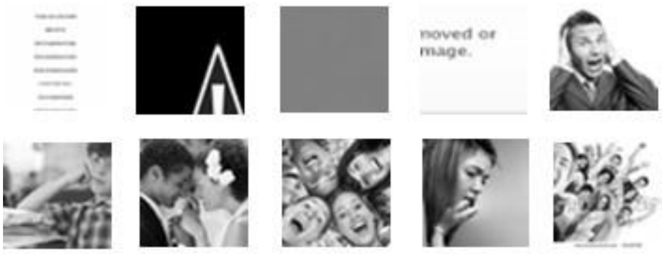

Fig. 3. Noise samples.

This dataset also contains some invalid images, including non-face images and face regions smaller than 48 pixels. We removed from the training set noise images and images that had face regions with a width and height smaller than half of the image (24 pixels). Also, an image with multiple face regions counted as a noise image because not only was the face region smaller than 24 pixels but also one facial expression label could not be correct, as illustrated Fig. 3. We did not remove noise images from the public and private sets so that the results of the proposed method would be comparable to other methods. After removing 605 noise images, the training set comprised 28,104 images, as detailed in Table II.

We trained from scratch the four models using the deep learning framework Pytorch [35] with the full training set and the cleaned set, evaluating the method with the public and private sets. As some face regions were not cropped clearly, a random crop of 40 pixels was used in the training stage. No other image augmentation was used. For the optimizer method, the stochastic gradient descent (SGD) algorithm was used with a 0.01 learning rate and learning decay every five epochs after the 80th epoch. One deep CNN model will be trained over 200 epochs.

TABLE I: Distribution of Facial Expression in the FER2013 Dataset

Emotion	Training	Public	Private	**Total**
Angry	3,995	467	491	**4,953**
Disgust	436	56	55	**547**
Fear	4,097	496	528	**5,121**
Happy	7,215	895	879	**8,989**
Sad	4,830	653	594	**6,077**
Surprise	3,171	415	416	**4,002**
Neutral	4,965	607	626	**6,198**
Total	**28,709**	**3,589**	**3,589**	**35,887**

B. Results

In this section, we explain the performance of each deep CNN model, then show the accuracy of the proposed method to reveal the impact of multi-deep CNN features on the final result. As shown in Tables III and IV, removing the noise images from the training set led to an improvement of 1-2% in accuracy on the public and private sets. To evaluate the impact of the combined multi-deep CNN models, we also trained four SVM models with features from each deep CNN single model to compare performance with our proposed method. As can be seen in Table IV, combining multi-deep CNN features enhances accuracy compared to four original deep CNN models.

TABLE II: Distribution of Training Set in FER2013 Dataset after Removed Noise Images

Emotion	Full dataset	Cleaned dataset
Angry	3,995	3,887
Disgust	436	431
Fear	4,097	3,999
Happy	7,215	7,123
Sad	4,830	4,685
Surprise	3,171	3,091
Neutral	4,965	4,888
Total	**28,709**	**28,104**

TABLE III: Accuracy of deep CNN Models in the Public and Private Sets (%)

Model	Full dataset		Cleaned dataset	
	Public	Private	Public	Private
VGG19	65.00	66.42	67.35	68.62
ResNet18	70.13	71.94	71.66	73.38
PreActResNet18	69.16	71.74	70.58	72.82
ResNeXt29	69.84	71.17	70.35	72.48

TABLE IV: PERFORMANCE OF EACH MODEL FEATURE WITH SVM AND THE PROPOSED METHOD ON BOTH TRAINING SET (%)

Model	Full dataset		Cleaned dataset	
	Public	Private	Public	Private
VGG19	67.27	68.94	68.94	70.27
ResNet18	69.84	71.74	71.46	73.18
PreActResNet18	68.31	71.60	70.44	72.81
ResNeXt29	69.84	70.32	69.82	71.51
Proposed method	**71.01**	**72.33**	**71.74**	**73.78**

We conducted experiments using four CNN features with different classifications in machine learning; most of the classifications with proper parameters improved accuracy when combining four CNN features. We selected a linear SVM since it gave the shortest time to predict one image, as shown in Table V.

TABLE V: PERFORMANCE OF MULTI-CNN FEATURES WITH SVM (LINEAR KERNEL, POLY KERNEL, RBF KERNEL, SIGMOID KERNEL), K-NN, DECISION TREES, RANDOM FORESTS, AND NA ÏVE BAYES

Classifier	Parameter	Accuracy in-public set (%)	Accuracy in private set (%)	Average time to predict one image (s)
SVM	linear kernel	71.74	73.78	0.0000087
	poly kernel	71.60	73.94	0.0000135
	RBF kernel	71.71	73.55	0.0000328
	Sigmoid kernel	71.60	73.92	0.0001047
k-NN	k = 1	71.55	73.39	0.0003230
	k = 3	71.41	73.47	0.0003894
	k = 5	71.60	73.86	0.0004182
	k = 7	71.66	73.86	0.0004321
	k = 9	71.69	73.80	0.0004559
Decision Trees		65.84	68.90	0.0000016
Random Forests	depth = 5	67.62	69.57	0.0000127
	depth = 10	70.96	73.80	0.0000143
	depth = 15	70.93	74.00	0.0000146
	depth = 20	71.16	73.80	0.0000145
Na ïve Bayes		71.27	73.14	0.0000478

We compared our proposed approach with state-of-the-art methods on the private set of FER2013, as shown in Table VI.

TABLE VI: RESULTS OF THE PROPOSED METHOD COMPARED TO SEVERAL STATE-OF-THE-ART METHODS

Method	Accuracy (%)
Fast R-CNN [27]	30.19
Auxiliary model [25]	67.70
DLSVM [11]	71.20
Multi-scale CNN [28]	71.80
HERO [20]	71.91
Multiple Deep CNN [15]	72.00
MLCNN [18]	73.03
Hybrid CNN-SIFT [17]	73.40
CNNs, BOW and global SVM [13]	73.25
Proposed method	**73.78**
CNNs, BOW and local SVM [13]	75.42

Table VI shows that combining four state-of-the-art deep CNN architectures for FER classification can surpass the results of previous methods [11], [13], [15], [17], [18], [20], [25], [27], [28]. However, a local SVM can achieve up to 75.42%, showing that the proposed method still improves on the results compared to using a global SVM, 73.25% [13]. In addition, we compared the result by using the classification approach (the proposed method) with the object detection approach [27].

We performed our proposed method on a system with one Tesla V100 GPU, with 480GB RAM and a CPU of 2.30GHz. Time for extracting deep CNN encodes of each deep CNN model for a single image was around 0.02 seconds; the time for the linear SVM to predict one image is shown in Table V. Thus, the time to predict one image by using the proposed method took under one second.

V. CONCLUSION

In this paper, we show that using simple rules to remove noise images in the training set can improve the performance of the deep CNN model. The proposed method leverages most of the state-of-the-art deep CNN models in the classification problem to represent facial expression as multi-deep CNN features. Using a basic discriminator model, a linear SVM, to combine the different features can improve the final result, which is competitive to other methods of FER in the wild.

For further work, by using the proposed schema, we can consider and combine any new state-of-the-art deep CNN model to extract facial features to achieve potentially better results on facial expression in the real-world and using specific CNN models to analyze different information on each part of the facial image.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest in the subject matter or materials discussed in this manuscript.

AUTHOR CONTRIBUTIONS

All authors discussed ideas and solutions for the proposed pipeline, Tran Ngoc Dong and Le Van conducted the experiment and wrote the paper, Pham The Bao gave advice and approved the final manuscript.

REFERENCES

[1] E. Sariyanidi, H. Gunes, and A. Cavallaro. "Automatic analysis of facial affect: A survey of registration, representation, and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, iss. 6, pp. 1113-1133, 2015.

[2] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, pp. 1-1, 2020.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, New York: Curran Associates, Inc., vol. 25, pp. 1097-1105, 2012.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations*, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Computer Vision – ECCV 2016*, Amsterdams: Springer, Cham, vol. 9908, pp. 630-645, 2016.

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *Proc. the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco: AAAI Press, 2017, pp. 7278-4284.

[9] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu: IEEE, 2017, pp. 5987-5995.

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proc. the Fifth Annual Workshop on Computational Learning Theory*, New York: Association, for Computing Machinery, 1992, pp. 144-152.

[11] Y. Tang, “Deep learning using linear support vector machines,” in *Proc. Workshop on Challenges in Representation Learning International Conference on Machine Learning*, 2013.

[12] R. T. Ionescu, M. Popescu, and C. Grozea, “Local learning to improve bag of visual words model for facial expression recognition,” in *Proc. Workshop on Challenges in Representation Learning International Conference on Machine Learning*, 2013.

[13] M. I. Georgescu, R. T. Ionescu, and M. Popescu, “Local learning with deep and handcrafted features for facial expression recognition,” *IEEE Access*, vol. 7, pp. 64827-64836, 2019.

[14] I. J. Goodfellow, D. Erhan, P. L. Carrier *et al*., “Challenges in Representation Learning: A report on three machine learning contests,” in *Proc. International Conference on Neural Information Processing*, Berlin: Springer, 2013, vol. 8228, pp. 117-124.

[15] Z. Yu and C. Zhang, “Image based static facial expression recognition with multiple deep network learning,” in *Proc. the 2015 ACM on International Conference on Multimodal Interaction*, New York: Association for Computing Machinery, 2015, pp. 435–442.

[16] A. Mollahosseini, D. Chan, and M. H. Mahoor, “Going deeper in facial expression recognition using deep neural networks,” in *Proc. IEEE Winter Conference on Applications of Computer Vision*, Lake Placid: IEEE, 2016, pp. 1-10.

[17] T. Connie, M. Al-Shabi, W. P. Cheah, and M. Goh, “Facial expression recognition using a hybrid CNN-SIFT aggregator,” *Multi-disciplinary Trends in Artificial Intelligence*, Gadong: Springer, Cham, vol. 10607, pp. 139-149, 2017.

[18] H. D. Nguyen, S. Yeom, G. S. Lee, H. J. Yang, I. S. Na, and S. H. Kim, “Facial emotion recognition using an ensemble of multi-level convolutional neural networks,” in *Proc. IEEE Transactions on Affective Computing*, IEEE, pp. 1-1, 2019.

[19] S. Minaee and A. Abdolrashidi, *Deep-emotion: Facial Expression Recognition Using Attentional Convolutional Network*, 2019.

[20] W. Hua, F. Dai, L. Huang, J. Xiong, and G. Gui, “HERO: Human emotions recognition for realizing intelligent internet of things,” *IEEE Access*, vol. 7, pp. 24321-24332, 2019.

[21] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *Proc. the British Machine Vision Conference*, Swansea: BMVA Press, 2015, vol. 1, no. 41, pp. 1-12.

[22] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *Proc. the British Machine Vision Conference*, Nottingham: BMVA Press, 2014, pp. 1-12.

[23] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, “Training deep networks for facial expression recognition with crowd-sourced label distribution,” in *Proc. the 18th ACM International Conference on Multimodal Interaction*, Tokyo: Association for Computing Machinery, 2016, pp. 279-283.

[24] A. Mollahosseini, B. Hasani, and M. H. Mahoor, “AffectNet: A database for facial expression, valence, and arousal computing in the wild,” in *Proc. IEEE Transactions on Affective Computing*, Washington: IEEE Computer Society Press, 2019, vol. 10, pp. 18-31.

[25] Y. Wang, Y. Li, Y. Song and X. Rong, “Facial expression recognition based on auxiliary models,” in *Proc. Algorithms MDPI AG*, 2019, vol. 12, iss. 11.

[26] R. Girshick. “Fast R-CNN,” in *Proc. 2015 IEEE International Conference on Computer Vision*, Chile: IEEE, 2015.

[27] S. B. S AlMarri, “Real-time facial emotion recognition using fast R-CNN,” Thesis, Rochester Institute of Technology, 2019.

[28] S. Zhou, Y. Liang, J. Wan, and S. Z. Li, “Facial expression recognition based on multi-scale CNNs,” in *Proc. Chinese Conference on Biometric Recognition*, Chengdu: Springer, Cham, 2016, vol. 9967, pp. 503-510.

[29] J. C. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” in *Proc. Advances in Kernel Methods-Support Vector Learning*, Cambridge: MIT Press, 1998, vol. 208.

[30] P. Cunningham and S. J. Delany, “k-Nearest neighbour classifiers,” *Multi Classif Syst.*, 2007.

[31] L. Rokach and O. Maimon, “Decision Trees,” in *Proc. The Data Mining and Knowledge Discovery Handbook*, Boston: Springer, vol. 6, pp. 165-192, 2005.

[32] L. Breiman, “Random forests,” in *Proc. Machine Learning*, Springer, vol. 45, pp. 5-32, 2001.

[33] G. I. Webb, “Naïve Bayes,” in *Proc. Encyclopedia of Machine Learning*, Boston: Springer, 2011.

[34] R. Amani, D. B. Ayed, and N. Ellouze, “Practical selection of SVM supervised parameters with different feature representations for vowel recognition,” 2015.

[35] A. Paszke, S. Gross, F. Massa *et al*., “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019, pp. 8024-8035.

Tran Ngoc Dong was born in Binh Thuan Province, Viet Nam, 1985. He graduated with the B.S. in mathematics and computer science from Ho Chi Minh City University of Education, Vietnam in 2010. He has currently been studying for a master's degree at Vietnam National University Hochiminh City-Hochiminh City University of Information Technology. His research interests is about image processing, machine learning and neural network.

Le Van was born in Hochiminh City, Viet Nam, 1995. He graduated with the B.S. in mathematics and computer science from Vietnam National University-University of Science, Vietnam in 2017. Currently he is a research assistant at IC-IP Lab. His research interests include image processing, machine learning, neural network and visualization.

Pham The Bao was born Hochiminh City, Viet Nam, 1972. He received with the B.S. in 1995, MSc. in 2000 and Ph.D. degree in 2009 in Vietnam National University-University of Science, Vietnam. He is a professor of the Department Computer Science, Faculty of Mathematics and Computer Science, University of Science from 2013 to 2018. Currently he is the Chair of IC-IP Lab, professor of the Department Computer Science, Faculty of Information Science, Sai Gon University. His research interests include image processing, pattern recognition, and intelligent computing.