Facial Expression Recognition Using Machine Learning Approach

P. Muthukumar[1], M. Sugan[2], A.K. Neeraj Kumar[3]
[1, 2, 3] Dept of Information Technology
[1, 2, 3] Dr.Mahalingam College of Engineering and Technology, Pollachi, Coimbatore, Tamil Nadu, India

Abstract- *Effective emotion recognition algorithms can help machines better understand people and promote the development of human-computer interaction applications. In recent years, many research efforts have used benchmark expression data to train deep neural network models to achieve state-of-art results. These high-accuracy models usually contain hundreds of layers, so they require complex calculations and may not be suitable for real-world scenarios. Facial expression recognition plays a very important role in communicating the emotions and intentions of people in general. The objective of this work is to develop a facial expression recognition system based on convolutional neural networks with data augmentation. This approach enables classifying seven basic emotions consisting of angry, disgust, fear, happy, neutral, sad and surprise from image data. This paper proposes to create a model with Dense layer and several convolutional layer to extract the feature of the expression from the image.*

Keywords- Emotion Recognition, Facial Expression, Convolutional Neural Network, Lightweight.

I. INTRODUCTION

Facial emotion recognition is the process of recognizing human emotions based on facial expressions. The human brain automatically recognizes emotions and software has been developed to recognize them. AI can recognize emotions by learning what each facial expression means and applying that knowledge to the new information presented to it. This technology is becoming more and more precise and will eventually be able to read emotions just like human brains. Artificial Emotional intelligence, or emotion AI, is a technology that can read, imitate, interpret, and respond to facial expressions and human emotions.

Facial expressions and other gestures transmit non-verbal communication signals that play an important role in human relationships. Therefore, when extracting and analyzing information from an image or video, facial expression recognition can provide unfiltered and unbiased emotional responses in the form of data.

Facial expression recognition(FER) has emerged as one of the important research areas over the past decade. Facial expression is one of the indirect mediums to communicate emotion between humans. Human facial emotion recognition(FER) has many use cases. A few of them are we can use them in virtual reality, video conferencing, and interviews, etc.

Mostly human facial expressions can be classified into sad, happy, disgust, fear, surprise, anger, and neutral. Because of the extensive use of digital devices, the ability of young people to read the feeling and emotion of other people is reduced which is shown in recent research. Therefore, there is a necessity to develop an accurate facial expression recognition system which detects the emotions in real time.

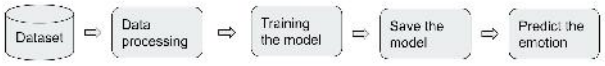


Fig. 1

II. LITERATURE WORK

1. An End-to-End Deep Model With Discriminative Facial Features for Facial Expression Recognition :

Instead of focusing more on the model they focused more on getting the feature data from the dataset itself. In this article, they introduced an end-to-end deep model to enhance the feature of the dataset which locate the range of the face target and enhance the image contrast . Next, to obtain further discriminative features, a hybrid feature representation method is proposed, in which four typical feature extraction methods are combined. The only problem with this approach is we have to process the dataset every time when we use new data.

2. Facial Expression Recognition Using Frequency Neural Network

- Different from convolutional neural networks in the spatial domain, FreNet inherits the advantages of processing images in the frequency domain, such as

efficient computation and spatial redundancy elimination.
- The experimental results show that the Block-FreNet not only achieves superior performance, but also greatly reduces the computational cost.
- We propose the LMK and multiplication layer that inherit the advantages of frequency domain analysis.
- There are numerous advantages to process image in frequency domain
- our Block-FreNet inherits the advantages of frequency domain analysis, which can learn frequency based features
- Our Block-FreNet obtains 64.41% that is lower than the performance on CK+ and Oulu-CASIA, due to the occlusions and pose variation in FER2013.
- So we decide to propose a dimension reduction technique following the multiplication layers for performance improvement
- FreNets perform well and the recognition accuracies on each dataset are greater than 80%

3. Face Recognition Algorithm Based on Correlation Coefficient and Ensemble-Augmented Sparsity

- ensemble-enhanced sparse classification algorithm based on the correlation coefficient can improve recognition accuracy.
- This method effectively enhances the classification performance and reduces the computational complexity
- advantages of avoiding "dimensional disaster"
- information fusion technology has been widely considered to combine multiple pieces of information to improve performance
- used nonnegative sparse coding and sparse matrix decomposition techniques to propose the CCLR-SCSPM algorithm
- performance of the Pm_SCRC algorithm is mainly affected by parameters such as CRC regularization parameter λ, SRC sparsity level S, and nearest neighbour number F

4. Fast and Efficient Facial Expression Recognition Using a Gabor Convolutional Network

- light Gabor convolutional network (GCN) consisting of only four Gabor convolutional layers and two linear layers for FER tasks
- Low computational cost, but not great level of accuracy
- GCN3 is faster than GCN4 10, its accuracy is 2.48% lower than that of GCN4 10
- Although DenseNet-BC-100 requires the fewest storage parameters, its recognition performance is lower than that of the other models.
- RTCRelief-F and alignment-mapping networks (AMNs) achieve better recognition performances, but they are ensemble approaches and very complicated.

III. ALGORITHMS USED

CONVOLUTIONAL NEURAL NETWORK

The convolutional neural network, or CNN for brief, could be a specialized sort of neural network model designed for operating with two-dimensional image information, though they will be used with one-dimensional and three-dimensional information also.

The convolutional neural network has a convolution layer in the center. Because of that, it gets that name. This layer performs AN operation referred to as a "convolution".

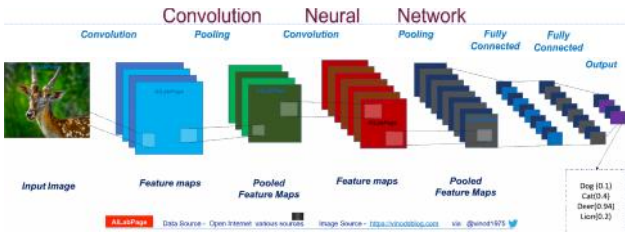
In the context of a convolutional neural network, a convolution could be a linear operation that involves the multiplication of a collection of weights with the input, much like a traditional neural network. This technique was designed mostly for two-dimensional input and the multiplication of the values is performed between an array of input data and a two-dimensional array of weights, called a filter or a kernel.

The filter is smaller than the input file and also the sort of multiplication applied between a filter-sized patch of the input and also the filter could be real. A dot product is an element-wise multiplication between the between to matrix. In this case, the matrix is filter-sized patch input and filter, which in return always results in a single value. The result is always a single value so it is often called a "*scalar product*".

Using a filter smaller than the input is intentional because it permits a constant filter (set of weights) to be increased by the input array multiple times at totally different points on the input. Specifically, the filter is applied consistently to every overlapping half or filter-sized patch of the input file, left to right, high to bottom.

This systematic application of constant filters across a picture could be a powerful plan. If the filter is intended to observe a selected variety of features within the input, then the appliance of that filter consistently across the whole input image permits the filter a chance to find that feature in a place within the image. This is commonly referred to as translation invariance.

These above images show the basic working principle of the Convolutional neural network.



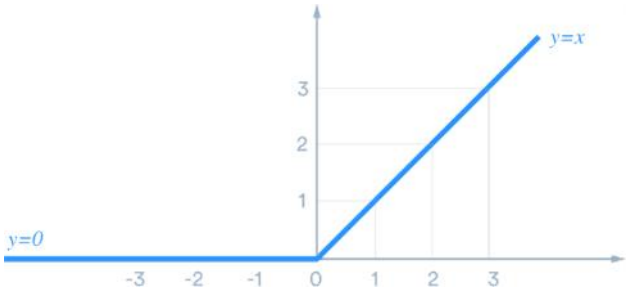
ReLU activation function

ReLU can be abbreviated as rectified linear activation unit. And is considered as one of the revolutions in the deep learning industry. The predecessor of ReLU is sigmoid and tanh. ReLU is better than Sigmoid and tanh because of its sparsity in nature and simplicity.

ReLu activation function formula:
f(x) = max(0, x)
where

- x - is the input value.

ReLU function is its derivative; both are monotonic. ReLU returns 0 if the input value is negative, but if the input value is positive, then it will return that value. Because of that, the output range is between 0 and infinity



As shown above, the ReLU is simple and it consists of no heavy computational complicated math. Therefore, the model can take less time to run or train. There is one more important property that we can consider as an advantage of the ReLU activation function is its sparsity.

Softmax function

The softmax function turns a vector of N values into a new vector of N real values that is a sum to one. The possible input values can be positive, negative, zero, or even greater than one. But in the end, softmax will transform those values between 0 and 1, so that they can be interpreted as probabilities. If one of the softmax function inputs is small or negative, then the softmax will turn it into a small probability and if the input is large, then the values are turned into large probability. But the return value always remain between 0 and 1

The softmax function is also called a soft argmax function or multi-class logistic regression. This is because of the generalization of softmax of the logistic regression it can be used for multi-class classification and its formula also very similar to the sigmoid function which is also used for logistic regression.

We can only use the softmax function only if the classes are mutually exclusive. Many multi-layer in the neural network end up in a penultimate layer that outputs real-valued scores which are not conveniently scaled and it is also difficult to work with. Here the softmax function is very useful as it converts the score to a normalized value(probability distribution), which can be displayed to a user or it can also be used as input to other systems. For this very reason, the softmax function is added as the activation function at the final layer of the neural network

Softmax formula :

The softmax formula is as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

IV. DATASETS

FER13 dataset

In our project we are using the FER13 dataset . It contains 34034 rows and 3 columns. In that we have taken pixel attributes and all records for testing and training based on the usage column.. This dataset states the FER with the attributes given below,

- image pixel value in zero and one
- usage - whether the rows is used for testing or training
- emotion - emotion of person in the image

Where it has an total images of 34034, from this we have to train and test the model

Table 1 - FER23 dataset attribute description

Attributes	Description
pixel	each pixel of the image
emotion	emotion of person in the image
usage	whether the row can be used for training the model or for testing

Table 2 - FER13 Dataset sample

emotion	pixel	usage
1	00 80...	1750
0	70 30...	1920
3	10 30...	535

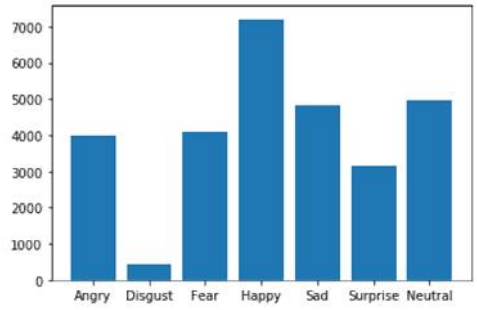


Fig 2 FER13 dataset

Preprocessing the dataset:

The dataset consists of a no of images which are represented as strings of 2404 space-separated numbers. Then the values are converted into a 48 * 48 matrix. Each number in a row represent a pixel value of an image

The total images 35,887 dataset images are split into a training set of 28, 709 and for testing it is 7,178 images - an 80:20 split ratio. When it comes to machine learning and mostly in deep learning, the dataset is the biggest factor. For deep learning, if the training data is less, then there will be variance in the final output due to the lack of a training dataset.

With keeping that in mind, having a test dataset of 20% of the total images available in the dataset may be seen as excessive. But it is not.

This is because we don't want our model to overfit. So it is necessary to have a valid sizable testing dataset. There is this 60-20-20 split variable that is used as training, testing & validation respectively. For example, dividing the 200k image dataset into 60% for training, 20% for testing, and 20% for validation. Here, the validation set is not considered as in the favor of retraining the entire model every single time when the hyperparameters are tuned. While this has a disadvantage of more time and more computational power than the usual one, it provides a bigger training dataset in the end. Instead of classifying the emotions with the numbers from 0-6, a one-hot encoding scheme is used for the labels. Later when live testing the model, Haar Cascades were used to identify the face from the camera. The images are preprocessed as the identified face was taken as an image and then converted into gray-scale and then the image is downscaled to 48 * 48 sized image.

Simple convolutional neural network model

Simple Convolutional Network: In this paper, we used a convolutional neural network as a model. This model layer system starts with two two-dimensional convolutional layers and it is followed by a two-dimensional max-pooling layer which is used to downsample the output from the convolutional layer which is followed by two dense layers. The output from the max-pooling was flattened before entering the dense layers. To reduce the overfitting of the model dropout was applied to the max pooling and the dense layer. As the model is more complicated, it ended up having issues as instead of predicting angry, it predicted happy for most of the inputs. This makes sense because a quarter of the input images are of happy expression. To train and make the model actually learn, we made the model even deeper. This model's architecture can be seen in the below Diagram.

ARCHITECTURE OF THE PROPOSED CNN
Proposed convolutional neural network
CONV2D-32 RELU
CONV2D-64 RELU
MAXPOOL2D DROPOUT
CONV2D-128 RELU
MAXPOOL2D DROPOUT
CONV2D-128 RELU
MAXPOOL2D DROPOUT
FLATTEN
DENSE-1024 RELU DROPOUT
DENSE-7 SOFTMAX

Now, the model network consists of 4 two-dimensional convolutional layers, three max-pooling layers, and two dense layers. Maximum values from each cluster of neurons are used by the Max pooling which is at the prior layer. Max pooling reduces the dimensionality of the output array. The initial input to the model is a preprocessed face of 48 x 48 pixels. The model was constantly observed with different combinations of fully connected and dense layers and in the end, we chose dense as it gave us some better results than the other layer. So instead of choosing a wide network we went with a deeper network for better results and accuracy. The one advantage of using more layers is that it can prevent memorization. Multi-layer networks learn features at different levels of abstractions and it will allow them to generalize well. We added more layers to the model to maintain a high level of accuracy while still having a fast model for real-time purposes. Also, the model utilized max pooling and dropout more effectively to minimize overfitting.

The network consists of two convolutional layers with a first layer filter size is 32 and for the second layer, the filter size is 64. And ReLU is used as the activation function for this layer. Then, this is followed by a max-pooling layer. In order to reduce the overfitting of the model, the dropout rate of 0.25 is applied. This is followed by a sequence of two convolutional layers. These two layers have a filter size of 128 each. Each of these layers is followed by a max-pooling layer to down sample the output. And the activation function used in this model is ReLU. And in this two-layer, each layer is followed by a dropout of 0.25 to reduce the model from overfitting. In order to convert the output for this layer into a single-dimensional vector the output of the previous layers was flattened. Then it is followed by a very dense layer of 1024 input neurons. And the activation function used in this model is ReLU And this layer is followed by a dropout of 0.5. Finally, a dense layer with softmax as an activation function is used as the output layer. The kernel size of all the convolutional layers is set to 3 x 3 which is the width and height of the 2D convolutional window. Each of the max-pooling layers is two-dimensional and the pool size is 2 x 2. Except for the last layer, all the layers used ReLU as the activation function. Here we used ReLU as our activation function which is used because it has the benefits of sparsity and it reduces the likelihood of vanishing gradients. The softmax activation function was used as the activation function for the final output layer of the model which predicts the probability of each emotion. This model provided a base accuracy of around 50% range on the testing dataset. The hyperparameters were then tuned, like batch size, the optimizer, and the number of epochs. The model was set to run for 50 epochs. This model detects emotions on all faces in the webcam feed. With a 4-layer CNN, the test accuracy of this trained model reached 63.2% for 50 epochs.

Testing

Initially, the dataset is split into an 80% training set and a 20% testing set.

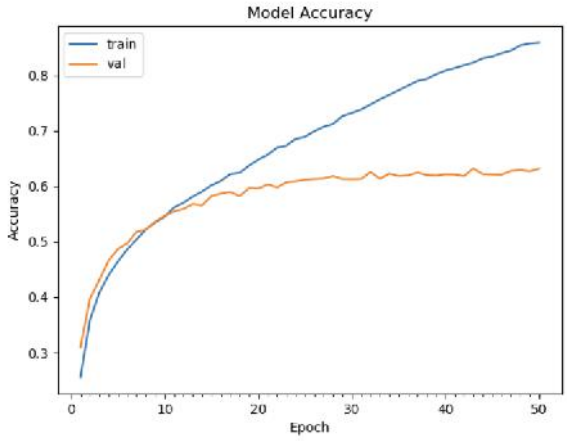
During the testing phase, the trained model is fed with the entire testing set one image at a time. The testing set has images that are not seen before by the model. The image that was fed to the model was preprocessed as detailed above.

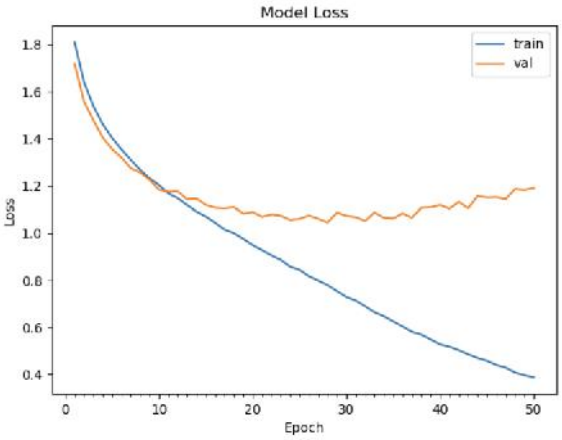
Thus the model has to predict what is the emotion of the person in the image based on the image it is trained with. The model attempted to classify the facial expression of the person in the image to determine the emotion of the person simply based on what it had already learned along with the characteristic of the image itself. In the end, it will predict the emotion of the image and spit out the probability of each emotion that the person in the image can be. The highest probability emotion. Then the emotion with the highest probability for each image is compared with the actual emotions associated with that particular image to count the number of accurate predictions. The accuracy calculation formula is detailed below. It counts the number of times the model predicted the emotion of the person in the image correctly and divides it by the total number of sample images in the testing set. In our case, the testing dataset contains 3,589 images

Accuracy = Total number of correct emotion predicted / Total number of samples

Result

Upon tuning the hyperparameters, the highest accuracy was achieved for each optimizer. When using the Adam optimizer with the default settings, with a batch size of 64 and epochs as low as 15 it leads to a very low accuracy of 0.20. After increasing the epochs to as high as 50, the model attained the highest accuracy of 62%.





In our manual testing, the model predicted happiness almost every time when the happiness image is given which is because of the huge sample for the happiness image. Interestingly enough the surprise emotion reached nearly the same accuracy as the happiness. The other emotions like anger and neutral reached nearly the same accuracy. The other emotions also had lower but reasonable accuracies. Another interesting point is that the model manages to determine the other emotions also. When the model is given an image (or a frame from a video feed) to predict the emotion, it does not simply give one final prediction. Instead, it predicts a list of probabilities of each emotion individually. Now, we'll take the emotion with has the highest probability as the final prediction. Thus we classified the emotion of the person based on the facial reaction which is predicted by the model. It is also to be noted that in some cases neutral and disgust as the sample data for this emotion is low. But in the second try, the model was able to predict the neutral or disgust. That means the top-2 predicted emotions will be much more accurate. Finally, to conclude, in the live testing of the trained model, the model was able to predict the emotions of the person in the webcam instantly with no delays.

V. CONCLUSION & FUTURE WORK

In this paper, our aim was to classify the facial expression of a person into one of seven emotions by using various models on the FER13 dataset. The model was experimented with small and lightweight convolutional networks before arriving at the proposed model. The changes and tuning of different hyperparameters on the final model were then investigated to get better accuracy. The final accuracy of the model was 62% and it was achieved using the Adam optimizer with modified hyperparameters. It also is noted that this nearly state-of-the-art accuracy was achieved with the use of a single dataset as opposed to a combination of many datasets. While the other related work model managed

to obtain higher accuracies, it also needs to note that they used a combination of many datasets and a lightweight model to increase their accuracy. It also to be noted that in our case we used only the FER-2013 dataset and we didn't use other datasets for training the model and an accuracy of 62% is admirable as it demonstrates the efficiency and the accuracy of the model. In other words, the trained model demonstrated that it used significantly less data for training and a deep but simple network to attain this near-state-of-the-art result. At the same time, it also has several shortcomings.

While the model did attain an admirable result, it also means that it did not achieve state-of-the-art accuracy. Additionally, the amount of data for emotions like "disgust" makes the model have difficulty in predicting as it has fewer sample data for training. But this model still has room for improvements. If the model is provided with more training data to train while still having the same network structure, then the efficiency of the proposed system will be enhanced considerably. The ability of the trained model to make the facial expression of the person in real-time efficiently indicates that there is a use case for the model in the real world. In the future, with more datasets, the model may predict the emotion accurately and reliably with fewer errors. The real-time capability of the model and in addition to that it also can be trained quickly and its near-state-of-the-art accuracy allows the model to be adapted and used in nearly any use-case. And this also implies that with some work and fine-tuning, the model could be deployed in the real-life application for effective utilization in domains such as in healthcare, the gaming industry, and also in interviews.

VI. ACKNOWLEDGEMENT

Apart from the efforts of us, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to price the almighty and express our gratitude to the people who have been instrumental in the successful completion of our project.

We wish to acknowledge with thanks for excellent encouragement given by the college management and we thank our project coordinators and project guide.

REFERENCES

[1] V. Mavani, S. Raman, and K. P. Miyapuram, “Facial expression recognition using visual saliency and deep learning,” *arXiv preprint arXiv:1708.08016*, 2017.

[2] L. Chen, M. Zhou, W. Su, M. Wu, J. She, and K. Hirota, “Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction,” *Info. Sci.*, vol. 428, pp. 49–61, 2018.

[3] W. Sun, H. Zhao, and Z. Jin, “A visual attention based ROI detection method for facial expression recognition,” *Neurocomputing*, vol. 296, pp. 12–22, June 2018.

[4] Z. Fan, J. Jiang, S. Weng, Z. He, and Z. Liu, “Human gait recognition based on discrete cosine transform and linear discriminant analysis,” in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2016, pp. 1–6.

[5] J. Lin and Y. Yao, “A fast algorithm for convolutional neural networks using tile-based fast Fourier transforms,” *Neural Process. Lett.*, vol. 50, no. 2, pp. 1951–1967, Oct. 2019.

[6] Y. Gan, “Facial expression recognition using convolutional neural net- work,” in *Proc. 2nd Int. Conf. Vis., Image Signal Process. (ICVISP)*, Aug. 2018, pp. 1–5.

[7] S. Elaiwat, M. Bennamoun, and F. Boussaid, “A spatio-temporal RBM- based model for facial expression recognition,” *Pattern Recognit.*, vol. 49, pp. 152–161, Jan. 2016.

[8] T. Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, “Spatial–temporal recurrent neural network for emotion recognition,” *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 839–847, Mar. 2019.

[9] X. Liu, B. V. K. V. Kumar, Y. Ge, C. Yang, J. You, and P. Jia, “Nor- malized face image generation with perceptron generative adversarial networks,” in *Proc. IEEE 4th Int. Conf. Identity, Secur., Behav. Anal. (ISBA)*, Jan. 2018, pp. 1–8.

[10] S. Hosseini and N. I. Cho, “GF-CapsNet: Using Gabor jet and capsule networks for facial age, gender, and expression recognition,” in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2019, pp. 1–8.

[11] H. Yang, U. Ciftci, and L. Yin, “Facial expression recognition by de- expression residue learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2168–2177.

[12] X. Liu, B. V. K. V. Kumar, P. Jia, and J. You, “Hard negative gen- eration for identity-disentangled facial expression recognition,” *Pattern Recognit.*, vol. 88, pp. 1–12, Apr. 2019.

[13] T. Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, “Spatial–temporal recurrent neural network for emotion recognition,” *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 839–847, Mar. 2019.

[14] M. H. Siddiqi, R. Ali, A. M. Khan, Y.-T. Park, and S. Lee, “Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields,” *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1386–1398, Apr. 2015.

[15] A. Majumder, L. Behera, and V. K. Subramanian, "Automatic facial expression recognition system using deep network-based data fusion," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 103–114, Jan. 2018.

[16] J. Chen, Z. Chen, Z. Chi, and H. Fu, "Facial expression recognition in video with multiple feature fusion," *IEEE Trans. Affect. Comput.*, vol. 9, no. 1, pp. 38–50, Jan. 2018.

[17] C. Huang, "Combining convolutional neural networks for emotion recognition," in *IEEE MIT Undergraduate Research Technology Con- ference (URTC)*, 2017, pp. 1–4.