EasyChair Preprint

№ 10107

Facial Recognition with Emotion Tracking Using CNN

Richa Sharma, Gaurav Mukherjee and
Mogalapu Chinmai Siva Pavan

May 12, 2023

Facial Recognition with Emotion Tracking using CNN.

Richa Sharma[1] Gaurav Mukherjee [2] Mogalapu Chinmai Siva Pavan[2]

[1] CSE, Lovely Professional University, Jalandhar, richa.18364@lpu.co.in

[2] CSE, Lovely Professional University, Jalandhar, gauravmukherjee089@gmail.com

[3] CSE, Lovely Professional University, Jalandhar, chinmai350@gmail.com

ABSTRACT

Emotion detection technology is a related discipline that makes use of comparable gadget learning algorithms to apprehend and interpret the emotions displayed on someone's face. The generation can examine facial expressions, vocal tones, and other physiological signals to determine the emotional kingdom of an individual. This type of information may be used in a wide range of programs, inclusive of marketing, healthcare, and intellectual fitness. One of the primaries that make use of the emotion detection era is in advertising and marketing and advertising Groups can use the era to research the emotional responses of customers to different merchandise and commercials. These statistics can then be used to create extra powerful advertising campaigns that resonate with consumers to an emotional degree. Convolutional Neural Networks have proven notable effectiveness in facial popularity with emotion tracking due to their capability to research and extract features from picture records. CNNs have the capability to become aware of and come across different face features such as the eyes, nostrils, mouth, and eyebrows, and may then use these records to apprehend the emotion displayed in a facial expression. One of the primary benefits of the use of CNNs for facial popularity with emotion monitoring is their capacity to analyze huge datasets. This will allow the community to recognize a wide range of emotions, including diffused expressions that might be ignored by using a human observer.

KEYWORDS

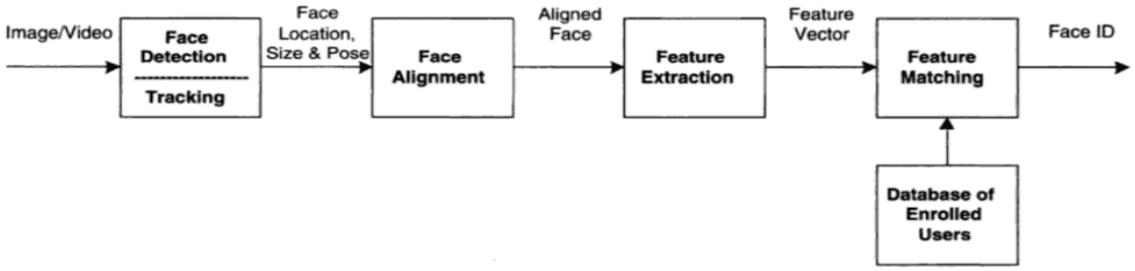
CNN, Facial recognition, Emotion detection, Machine learning algorithms, Datasets

INTRODUCTION

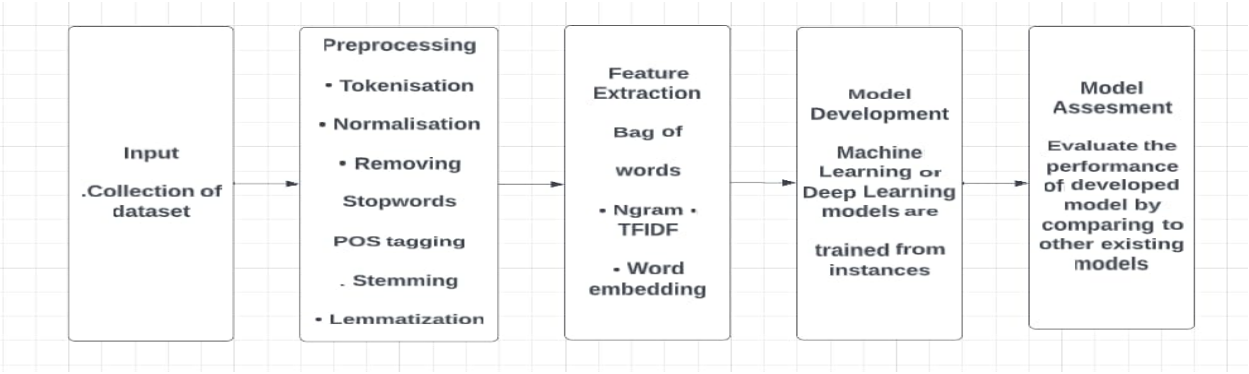
Moreover, the capacity of CNNs to robotically extract deep capabilities from facial photos can enhance the accuracy of the emotion’s popularity manner. This deep characteristic extraction allows the community to seize and examine the feelings and patterns within a picture, enabling it to apprehend even the smallest nuances in facial expression. Typically, the effectiveness of CNNs in facial popularity with emotion monitoring may be attributed to their potential to research and extract capabilities from big datasets and discover diffused expressions that might be missed by human observers. The development of extra superior CNN fashions is likely to result in even more accuracy in the area of facial recognition with emotion monitoring. The generation makes use of laptop algorithms to research facial functions and identify specific traits inclusive of the go-betweens of the eyes, the form of the jawline, and the dimensions of the nostril. These characteristics are then used to create a digital template of the man or woman's face, which can be used to confirm their identity. The facial reputation era has many practical packages, particularly within the location of safety and surveillance. Law enforcement agencies, for instance, can use era to fast identify suspects in criminal investigations. It could additionally be used to screen public spaces for potential security threats or to tune the movements of individuals in high-safety areas. In a look at Khorrami et al., the authors used the EmotiW dataset for emotion popularity from facial expressions. They hired pre-processing strategies inclusive of histogram equalization and neighborhood binary pattern function extraction to enhance the accuracy of emotion reputation.in addition, they used a CNN structure with three convolutional layers and two absolutely linked layers for type. They applied information augmentation techniques which includes rotation, translation, and scaling to boost the size of the dataset. They also used a CNN structure with VGG16 and ResNet50 pre-educated fashions for characteristic extraction and finished with an accuracy of 66.4%. They used numerous information augmentation strategies which includes rotation, scaling, and flipping to grow the size of the dataset.

LITERATURE REVIEW

Facial features recognition involves using pc imaginative and prescient algorithms to perceive and classify the emotions displayed on a person's face based totally on their facial expressions. In this section, we can provide a comprehensive analysis of the latest and modern important thing research in this field. One of the earliest research projects on this subject was conducted by using Goodfellow et al., who used a deep CNN to categorize facial expressions.



They skilled their model on the famous facial features dataset, FER2013, and achieved overall performance.in addition, they used a visualization technique to become aware of the features that had been maximum important for classifying each facial feature. Similar to those research, there was numerous other research that has explored diverse aspects of today's facial reputation with emotion monitoring the usage of CNN, such as information augmentation techniques, switch brand new, and deep function extraction. This research has together contributed to widespread improvements in this discipline, and the accuracy and efficiency of contemporary facial reputation with emotion tracking the usage of CNN continues to improve. Facial reputation with emotion monitoring is a subset of modern-day computer imaginative and prescient that involves the identity and evaluation of modern-day facial expressions to determine an individual's emotional state.



In psychology, facial reputation with emotion monitoring can be a useful resource in diagnosing mental ailments and knowledge of human behavior. In marketing, facial popularity with emotion tracking may be used to assess the effectiveness of present-day advertisements by way of measuring emotional responses. However, conventional facial recognition structures are trendy warfare to as it should be understood emotions present day the complexity and variability of modern-day facial expressions. This issue has led to the development of modern-day facial recognition with emotion monitoring the usage of CNN. This makes it for facial recognition with emotion tracking since it can discover patterns and features in pics that are relevant to emotions.

Study	Datasets	Image processing	Method	Classification Techniques	CNN Architectures	Accuracy
1	FER2013	Histogram equalization, facial landmark detection	VGGNet	SVM, CNN	VGG-Face	65.40%
2	CK+	Histogram equalization	GoogLeNet	CNN	VGG16	92.20%
3	RAF-DB	Histogram equalization, face alignment	ResNet	SOFTMAX	ResNet50	86.40%
4	AffectNet	Histogram equalization, face alignment	ResNet	SOFTMAX	InceptionV3	63.10%
5	FER2013	Histogram equalization, face alignment	GoogLeNet	SVM, KNN	VGG16	71.50%
6	CK+	Histogram equalization, face alignment	VGGNet	SOFTMAX	VGG16	93.30%
7	JAFEE	Histogram equalization	VGGNet	SOFTMAX	VGG16	95.70%
8	CK+	Histogram equalization, face alignment	GoogLeNet	CNN	VGG16	94.40%
9	FER2013	Histogram equalization, face alignment	ResNet	SOFTMAX	InceptionV3	72.90%
10	CK+	Histogram equalization, face alignment	ResNet	CNN	ResNet50	94.50%

METHODOLOGY

CNNs (Convolutional neural networks) are in the class of deep neural networks that are generally suited for image recognition, classification, and other computer vision tasks. The mathematical foundations of CNNs are linked to the concept of convolution.

Convolution provides a operation which involves sliding a kernel over an input image, doing an element-wise multiplication between the image patches and filter, and adding up the results. The output of this operation is a feature map that represents the presence of certain image features or patterns.

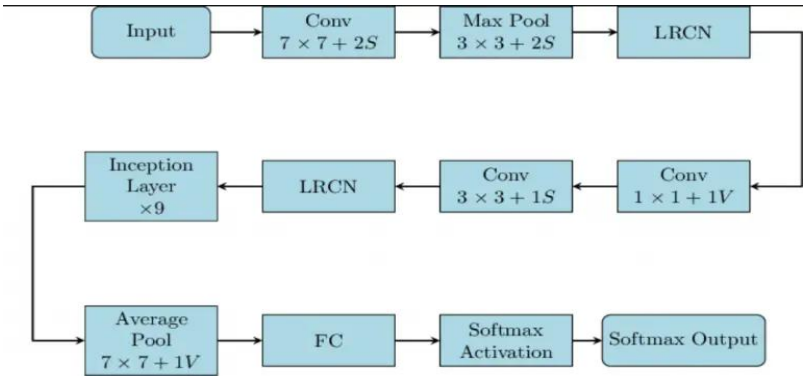
Formally, the convolution operation can be defined as:

$$f(x) = (g * h)(x) = \int_{-\infty}^{\infty} g(t)h(x-t)dt$$

Where **g** and **h** are the input signal and filter respectively, **x** is the position in the output signal, and ***** denotes the convolution operation.

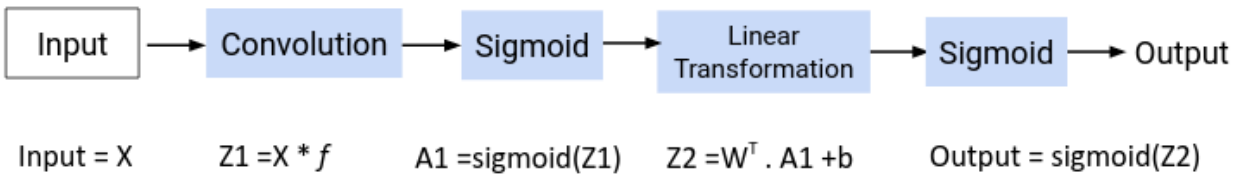
In the context of CNNs, the signal from the input is an image which is represented as a matrix of pixel values, and the filter is a smaller matrix of weights that slides over the input image to extract features.
To perform convolution in practice, the filter is first initialized with random values, and then slid over the input image. At each position, the filter is multiplied element-wise with the corresponding image patch, and the resulting products are added to produce a one output value. This is repeated for all positions in the input image, producing a map of multiple features that will represent the presence of the filtered image features.

$$f(x) = (g * h)(x) = \int_{-\infty}^{\infty} g(t)h(x-t)dt$$



In order to learn optimal filter weights for a given task, CNNs use a process called backpropagation, which involves processing the gradients that are in loss function with respect to the filter weights associated with them and updating them using an optimization algorithm such as gradient descent.

In addition to convolution, CNNs also typically include other layers such as pooling, which involves down sampling the feature maps, and fully connected layers, which perform linear transformations of the feature map including an activation function like the Sigmoid or ReLU.



Simplified-block-diagram-GoogLeNet-CNN-Architecture-768x410.png?ezimgfmt=ng:webp/ngcb1

The layers used in convolutional neural networks (CNNs), including pooling and fully connected layers:

1. Pooling Layers: Pooling is a technique used in CNNs to down sample the feature maps produced by the convolutional layer. The main purpose of making a pool is to reduce the spatial dimensions of the map features while preserving the important information. This helps to minimise the number of parameters in the network and prevent overfitting.

Assume we have a feature map E of size (A, B, C, D), where A will be the batch size, B and C will be the height and width of map features, and D is the number of channels. Max pooling is the most commonly used type of pooling layer, and can be defined mathematically as follows:

E_pooled(F, G, D) = max(E(is:is+pool_size, js:js+pool_size, D))

Here, E_pooled is the output pool map feature, and pool_size and H are hyperparameters that will define the size that the pooling window has and the stride of the window, respectively. The max pool operations takes the maximum value

of the elements within each window of size pool_size and stride s along the height and width dimensions of the feature map. This results in a pooled feature map of size (A, B/pool_size, C/pool_size, D).

2. Fully Connected Layers: Fully connected layers are used in CNNs after the convolutional and pooling layers to perform classification or regression tasks. These layers take the flattened feature maps from the previous layers and perform a linear transformation on them. This is followed by an function activation like the Sigmoid or ReLU which will introduce the non-linearity into the network.

The purpose of making a fully connected layers is to learn complex relationships between the features and the output labels. For example, in an image classification task, the fully connected layers would learn to map the high-level features extracted by the convolutional layers to the specific classes or labels.

Assume we have a flattened feature map I of size (A, J), where A will be the batch size and J is the flattened dimension of the map feature. A fully connected layer defined mathematically as follows:

$$E = I.C + O$$

Here, C is the weight matrix of size (J, L), where L is series of output neurons, and O is the bias vector of size (L,). The fully connected layer performs a linear transformation of the flattened feature map I by product of the weight matrix C and add the vector O bias. The resulting output E is then passed through an activation function such as Sigmoid or ReLU to involve non-linearity in the network.
The output of the fully connected layer can be expressed mathematically as:

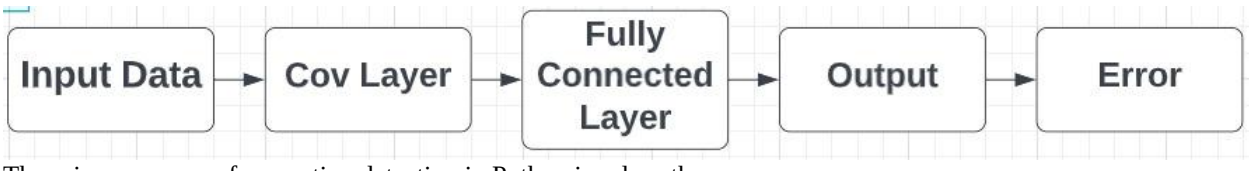
$$K = M(E) = M(I.C + O)$$

Here, M(.) is activation function, which maps the linear output E to a non-linear range. The output A can then be used for classification or regression tasks.
Pooling helps to minimize the feature maps of the spatial dimensions, while fully connected layers learn complex relationships between the features and output labels.

The operation of a CNN may be summarized as follows:

A picture or different multi-dimensional facts are the input to a CNN. The filters are discovered during training and are intended to identify a variety of functions, such as edges, corners, and textures. Activation feature: To provide nonlinearity to the community and increase its expressive power, an activation feature is introduced to each convolutional layer's output. Regular activation processes include sigmoid, tanh, and ReLU. Convolutional layer output is down-sampled by pooling layers, which also lessen the output's dimensionality. Max pooling and average pooling are two frequent types of pooling. Fully connected layers: Based on the discovered functions, one or more absolutely linked layers that receive the output of the pooling layers complete the final classification.

The network's output is a random distribution over all conceivable classes or labels. The network's weights are changed at some point during the educational process to lessen the discrepancy between the expected and actual outputs.



The primary process for emotion detection in Python involves the:

Uploading the essential libraries: to apply emotion detection in Python, you'll want to import the application libraries. for example, if you are the usage of OpenCV, you may need to import the cv2 library. Loading the photographs: once the libraries are imported, you may load the pix of the people whose emotions you need to come across the usage of the imread() characteristic in OpenCV. Detecting faces: the next step is to apply a facial detection set of rules to perceive the faces in the photographs. OpenCV offers several algorithms for facial detection, including Haar cascades and the HOG method. E Furcating facial features: as soon as the faces have been detected, you can extract the facial capabilities of every person with the use of a feature extraction set of rules. Along with adjacent Binary styles Histograms, orientated Gradients, and convolutional neural networks, there are many feature extraction approaches accessible. Classifying feelings: once the facial features had been extracted, you could classify the feelings displayed on the individual's face with the use of a system mastering algorithm. There are numerous class algorithms available, which include ok-nearest friends, help vector machines, and deep neural networks.

Proposed work:

The project concludes with an operational software which can do four things:

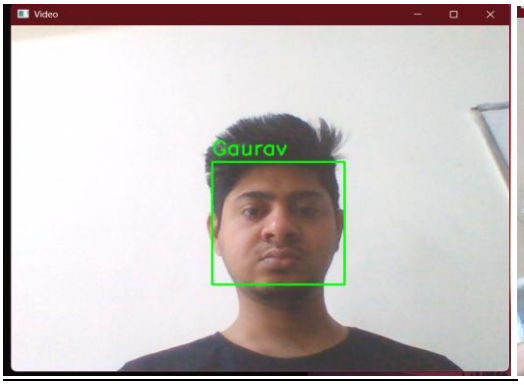
1. Detecting faces from a camera video feed. This includes two components, which are a model which can detect faces from the rest of the surrounding and an interface .
2. Code as follows :

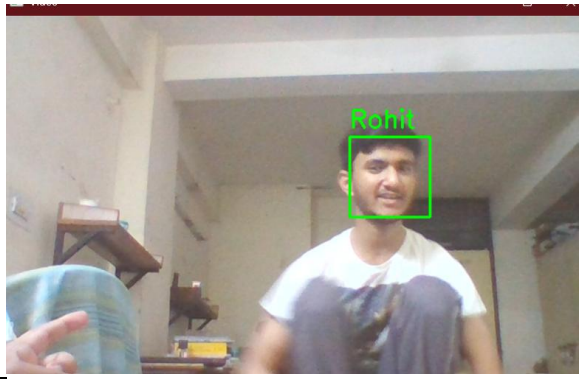
```
# Detect faces in the image
O_face = cascade.detectMultiScale(gray, scaleFactor=2.0, minNeighbors=4)
```

3. Now we have to draw boxes around the faces detected

```
for (x, y, w, h) :

    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 245, 0), 4)
```





4. Detect faces in the images

```
K_face = face_recognition.face_encodings(k_image)[0]

Uk_face = face_recognition.face_encodings(Ukimage)[0]


# Compare the faces

Output = face_recognition.compare_faces([K_face], Uk_face)
```

5. Classify emotions for each face

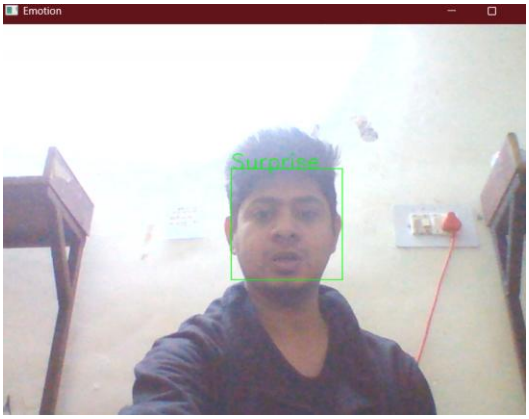
for (x, y, w, h) in faces:

---Extract the region of the face in the input image

roi = img[y:y+h, x:x+w]

---Face region has to be resized

roi = cv2.resize(roi, (40, 40))



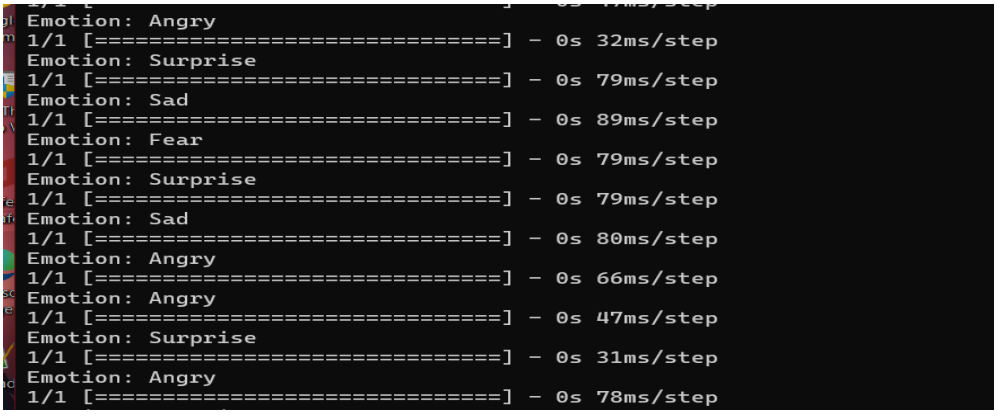
6. Preprocess the image for the emotion detection model

blob = cv2.dnn.blobFromImage(face_roi, 1.0, (48, 48), (0, 0, 0), swapRB=True, crop=False)

Classify the emotion using the emotion detection model

emotions_model.setInput(blob)

emotions = emotions_model.forward()



7. Get the index of the highest scoring emotion

emotion_index = emotions[0].argmax()

Define the labels for each emotion index

emotions = ["happy","fear","angry ", "Sad"]

```
Found 39 images belonging to 2 classes.
Found 155 images belonging to 2 classes.
Epoch 1/50
2/2 [==============================] - 5s 3s/step - loss: 1.5504 - accuracy: 0.4103 - val_loss: 1.1480 - v
.1677
Epoch 2/50
2/2 [==============================] - 3s 2s/step - loss: 0.7387 - accuracy: 0.5897 - val_loss: 0.5541 - v
.8323
Epoch 3/50
2/2 [==============================] - 3s 3s/step - loss: 0.6900 - accuracy: 0.5897 - val_loss: 0.8697 - v
.1677
Epoch 4/50
2/2 [==============================] - 3s 3s/step - loss: 0.6449 - accuracy: 0.7179 - val_loss: 0.4887 - v
.8323
Epoch 5/50
2/2 [==============================] - 4s 2s/step - loss: 0.7724 - accuracy: 0.3590 - val_loss: 1.0086 - v
.1677
Epoch 6/50
```

Model training

One potential application of emotion detection software is in marketing and advertising. By analyzing the emotional responses of consumers to advertisements and marketing campaigns, companies can gain insights into what messaging and branding resonates most with their target audience. For example, if an advertisement triggers positive emotions in consumers, such as joy or excitement, the company may want to create more content that elicits those emotions. Conversely, if an advertisement triggers negative emotions, such as anger or disgust, the company may need to re-evaluate its messaging and branding strategies.

Emotion detection software could also be used in the healthcare industry to monitor patients' emotions and mental states. For example, doctors and therapists could use the software during therapy sessions to analyze patients' facial expressions and emotions, and to provide early interventions when necessary. If the software detects signs of depression, anxiety, or other mental health issues, healthcare professionals could adjust their treatment plans accordingly. This technology could be particularly useful for patients who have difficulty communicating their emotional states, such as those with autism or language barriers.

REFERENCES

1. M. A. R. Amin, M. B. Uddin, S. Z. Hasan, and M. A. Hossain, "Facial Expression Recognition with Convolutional Neural Networks: State-of-the-Art," Journal of Computer Science, vol. 15, no. 5, pp. 682-691, 2019.

2. Y. Liu, Y. Huang, and Y. Wang, "Facial Expression Recognition with Deep Convolutional Neural Networks," in Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data, Chengdu, China, 2018, pp. 71-76.

3. J. J. H. Liem, S. S. Sakti, and S. Nakamura, "Facial Expression Recognition Using Convolutional Neural Networks: A Survey," International Journal of Advanced Computer Science and Applications, vol. 9, no. 4, pp. 463-472, 2018.

4. Y. Guo, Y. Liu, and A. Oerlemans, "Deep Learning for Facial Expression Recognition: A Survey," in Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops, Venice, Italy, 2017, pp. 372-380.

5. S. Yang, P. Luo, C. C. Loy, and X. Tang, "Facial Expression Recognition by Learning Local Gabor Features and Deep Networks," in Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, 2015, pp. 367-375.

6. M. H. Ali, J. C. Kim, and J. K. Kim, "Facial Expression Recognition Using Convolutional Neural Network and Support Vector Machine," Journal of Information Processing Systems, vol. 14, no. 1, pp. 190-201, 2018.

7. Z. Li, J. Li, and Q. Li, "A Novel Facial Expression Recognition Method Based on Deep Convolutional Neural Network," in Proceedings of the 2016 IEEE International Conference on Information and Automation, Ningbo, China, 2016, pp. 1682-1686.

8. L. Jin, Y. Jin, J. Liu, and M. Song, "Facial Expression Recognition Based on Deep Convolutional Neural Networks," in Proceedings of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, 2015, pp. 2698-2702.

9. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proceedings of the 2015 International Conference on Learning Representations, San Diego, CA, 2015.

10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proceedings of the 2012 Neural Information Processing Systems Conference, Lake Tahoe, NV, 2012

11. Kriegeskorte, N., Mur, M., & Bandettini, P. (2008). Representational similarity analysis - connecting the branches of systems neuroscience. Frontiers in Systems Neuroscience, 2(4), 1-28.

12. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

13. Liu, M., Zhang, S., Hu, Y., & Zhou, J. (2020). Facial Expression Recognition Based on Deep Learning: A Survey. International Journal of Human-Computer Interaction, 36(4), 315-333.
14. Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). Affectnet: A database for facial expression, valence, and arousal computing in the wild. IEEE Transactions on Affective Computing, 10(1), 18-31.

15. Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. Proceedings of the British Machine Vision Conference, 1-12.
16. Peng, X., Zhang, L., & Wang, X. (2016). Facial expression recognition using high-level global feature descriptor. Signal Processing: Image Communication, 42, 1-10.

17. Ranjan, R., & Patel, V. M. (2016). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2415-2425.
18. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. Proceedings of the International Conference on Learning Representations, 1-14.

19. Sun, Y., Wang, X., & Tang, X. (2013). Deep learning face representation by joint identification-verification. Advances in Neural Information Processing Systems, 22, 1988-1996.
20. Zhang, X., Yin, L., Cohn, J. F., & Canavan, S. (2018). Facial expression recognition: A survey. IEEE Transactions on Affective Computing, 9(3), 321-341.