Empirical Evaluation of Residual CNN in Emotion Recognition

Rongxin Liu
Stanford University
450 Serra Mall, Stanford, CA 94305
rongxinl@stanford.edu

Abstract

Emotion plays a vital role in people's daily lives. Humans are capable of detecting a person's emotions. However, in some circumstances, machines must be able to detect and respond to people's emotions. We performed a literature review on what is the state-of-the-art algorithms for solving the emotion recognition problem. Then we focus on performing an empirical evaluation of the residual CNN in emotion recognition.

1. Introduction

1.1. Overview

Neural networks are used in most computer vision applications for learning, analyzing, processing, and training data. A neural network helps in the understanding of data relationships through processing in a way analogous to that of a human brain but in a simpler form.

The emotion recognition system is one of many computer vision applications, including face recognition, objection recognition, etc. A driver drowsiness system, for example, is a type of emotion recognition system that is a very important application for detecting whether or not an automobile driver is sleepy, which is a critical application in the long-distance trucking situation. The algorithm relies mostly on eye motions and positions to assess if the driver is falling asleep.

It's worth noting that emotion recognition isn't solely a computer vision task; sentiment analysis can also be used to detect it. The emotions can be recognized via sentimental analysis by examining the text, such as tweets on Twitter. However, the focus of this study will be solely on the computer vision domain.

1.2. Application Area

Since COVID-19, people spend more time working in front of the computer (e.g., zoom meetings). It could be an interesting application of computer vision in monitoring human emotion. For example, a built-in emotion recognition system in webinar software might help the presenter quickly gauge how the audience is responding. For personal use, emotion recognition can be used to summarize the overall happiness of a person in performing a certain task and generate a report for the user to reflect upon. There could be a psychology resilience use case with the help of recognition.

Furthermore, healthcare personnel can utilize an emotion recognition system to prioritize their patients by monitoring facial expressions in the waiting room, especially in urgent care centers where individuals do not make appointments. Those who are in the most pain may be prioritized, while those with minor ailments may have to wait longer.

Additionally, as previously indicated, car manufacturers can benefit from emotion recognition technologies, especially in auto-driving. Cars that warn drivers when they are drifting off or becoming tired can help avoid dangerous collisions, or the alarm could be triggered by road rage or other intense emotions. The autopilot can take control of the car if the human driver becomes extremely emotional or tired.

1.3. Problem Statement

This paper proposes a computer vision-based emotion recognition system that takes an image input and outputs the corresponding emotion from a class of emotions. The baseline model for this recognition system could be framed as a supervised learning process using a deep convolution neural network. The baseline CNN model should obtain at least an emotion classification accuracy of 60% or above.

In practical application, we can deploy this emotion recognition system to monitor human emotions, and the model gets image input and outputs the classified emotion continuously. For example, we can pipe an image from a webcam video feed to this model and get the predicted result.

1.4. Outline

We first survey the literature on related work about the emotion recognition system, describing various algorithms and methods used to recognize a person's facial expression,

such as happy, angry, sad, etc. The literature review identifies what is considered the state-of-the-art algorithms in the emotion recognition domain. Then, we introduce the proposed algorithms, the dataset used, and the conducted experiments on solving the emotion recognition problem in detail. We end the paper with a conclusion in which we will summarize the report based on the experiment results of the proposed algorithm. We will also briefly address future work related to emotion recognition.

2. Related Work

People's facial expressions can be recognized by systems that have been trained using machine learning techniques. Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) are two well-known models used for categorization.

Feature extraction, classifiers, and neural networks were the major components of facial expression recognition.

2.1. Feature Extraction

In the facial expression recognition domain, the Gabor filter and local binary pattern operator seem to be used frequently in feature extraction.

The Gabor filter is regarded as one of the best feature extractors for data like photos, and it will extract the data's characteristics and patterns and send them on to the training model. The local and small pieces of data are generally extracted from the images. All these pieces are then combined to make a neighborhood that will be recognized. Sometimes the whole frontal face image can be processed in order to end up with the classifications of facial expression.

The Local Binary Pattern Operator (sometimes known as a texture descriptor or texture classifier) can also be used to extract features and classify data from an image. Initially, it will label the pixels of the images locally. And after a whole array, it combines all the neighboring pixels to create a global pixel value.

Facial Expression Recognition was proposed by Ketki R. Kulkarni et al., and they used Gabor filters in their proposed emotion recognition system. Zahir M. Hussain suggested using the LBP and Gabor filters to create an emotion detecting algorithm. One thousand tiny pieces make up the face regions. Each of these parts is then combined into a single histogram. These histograms carry important information about edges, corners, and spots.

2.2. Classifiers

Classifiers are the core of a supervised learning model, and it analyzes the features of the images to categorize the objects according to the labels. Support Vector Machines (SVM), KNN, and other classifiers are utilized in emotion recognition.

SVM is a classifier that can classify emotions based on their weights. Tuhin Kundu et al. proposed conventional emotion recognition techniques that measure the five primary emotions or moods recorded on a human face containing photographs: anger, joy, normality, somnolence, and automatic machinery surprise. The SVM classifier receives data from a feed-forward neural network and completes the classification. Likewise, Ma Xiaoxi also suggested multiple algorithms like support vector machines.

KNN is another kind of basic classifier used in machine learning models. The KNN can categorize unlabeled data with the labeled data getting the nearest majority vote. M. Murugappan used KNN as the classifier for facial expression identification, and Dr. Poonam Tanwar assessed emotions using KNN and Hidden Markov models. The KNN is used as the initial classifier where the obtained value is passed to the Hidden Markov Model for further deep classification.

2.3. Neural Networks

Neural Networks are the set of neurons interconnected to identify specific patterns in the data. The neural networks can also be used to recognize people's facial expressions, and Convolutional Neural Networks (CNN) is used most throughout the literature.

D Y Liliana utilized CNN to recognize emotions. She claims that occurrences of Facial Action Units can be used to process the detection (FAU). The FAU is one of the Facial Action Coding System's sub-categories (FACS). FACS is a coding system developed by Paul Ekman and Wallace Friesen in 1976 that uses degrees of intensity to assess the contraction and relaxation of facial muscles. There are other FACS-based feature extraction attempts. In general, research that attempts to detect face emotion using FACS would use a specialized dataset to train neural networks with facial images cataloged by FACS experts, implying that the feature extraction process is not computational but human.

Rohit Pathar et al. compared two neural networks for emotion recognition. A shallow CNN is compared to a deep CNN, and only one convolutional layer and three fully connected layers comprise the shallow CNN. Deep CNN, on the other hand, has eight convolutional layers. Deep CNN has a better accuracy of 89 percent on testing results, while shallow CNN only has a 45 percent accuracy. The higher the number of convolutional and fully connected layers, the higher the accuracy rate of the model.

Akash Saravanan et al. proposed facial emotion recognition using algorithms from the Convolutional Neural Networks. He categorized facial expressions into seven categories using several models on the FER-2013 dataset. Before arriving at the suggested model, feed neural networks, decision trees, and smaller convolutional networks were

tested with various hyper-parameters combinations. The ultimate accuracy was 60% using the Adam optimizer with updated hyperparameters.

2.4. Summary

Facial expression recognition is a complex task in machine learning, and many methods and techniques have been developed to achieve this recognition task.

From all the above discussions, the convolutional neural network is used by most of them since it gives high accuracy. Classical approaches, while effective, may be very dependent on the execution parameters and environmental conditions to achieve efficient results. Neural network-based algorithms can be generalized to solve a series of problems and have been used in the past few years to clarify the issue of facial emotion recognition as well.

Similarly, the classification process is the final stage of any supervised learning model. I have surveyed a few classification algorithms, such as SVM and KNN. SVM has a higher precision rate than the K-Nearest Neighbor.

3. Methods

From the literature review, I know that the state-of-art approach to building an emotion recognition system is to use the Convolutional Neural Networks. In addition, the literature tends to agree that the deeper the layer, the higher the predicted testing accuracy. Therefore, we will build our emotion recognition system using the CNN approach.

3.1. Traditional CNN models

Convolutional Neural Networks are regarded as the go-to solution for prediction problems involving image data as an input because they operate efficiently with data with a spatial relationship. The input images in our scenario would be photographs of human facial expressions.

The ability of CNNs to construct an internal representation of a two-dimensional image is one of their advantages. This enables the model to learn location and scale in various data forms, which is critical when working with images. These additional layers have been credited with a great portion of Deep Neural Networks' success. The idea underlying their function is that these layers learn more complicated features over time. Edges are learned by the first layer, shapes by the second layer, objects by the third layer, eyes by the fourth layer, etc.

When deeper networks start converging, however, a degradation problem emerges. In other words, as the network depth grows, accuracy becomes saturated and consequently rapidly declines. However, overfitting does not cause this degradation because adding more layers to a sufficiently deep model increases training error. To address this degradation issue, the deeper layers must directly propagate information from the shallow layers.

3.2. Residual CNNs

If a shallow model can attain accuracy, its deeper counterparts should also be able to do so. However, as the model becomes more complex, the layers' ability to propagate information from shallow layers becomes increasingly difficult, and the information is lost.

To overcome this issue, one of the most critical features of the proposed CNN model is the residual blocks, which add the original input back to the output feature map produced by passing the input through one or more convolutional layers.

3.2.1 Residual Learning

The function learned by the layers is $g(x) = f(x) - x$. Consider the layers $f(x) = g(x) + x$, which have skip connections. The skip connection is denoted by the $+x$ word.

The $+x$ term in $f(x) = g(x) + x$ returns the original value. The $g(x)$ layer must learn to recognize changes in the value, commonly known as the residual. Whatever is learned in $g(x)$ is simply the residual, positive or negative, used to change x to the desired value.

To make $f(x)$ an identity function, the residue $g(x)$ must be made a zero function, which is fairly simple to understand, that is, all weights must be set to zero. The required identity function is then $f(x) = 0 + x = x$. This will help in the remedy of the degradation issue.

Without skip connections, the weights and bias values have to be modified to correspond to the identity function. The degradation problem is caused by the difficulty of learning identity functions from scratch, exacerbated by the layers' non-linearity. If identity mappings are optimal, solvers can drive the weights of the many nonlinear layers toward zero to approach identity mappings with residual learning.

3.2.2 Residual Blocks

A typical residual block looks like the following:

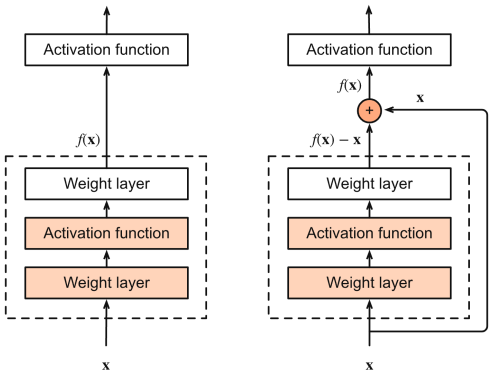

Figure 1. A regular block (left) and a residual block (right).

The skip connection helps in bringing the identity function to deeper layers. We're interested in why the activation function is performed after adding the skip connection in the residual block and why there are two weight layers in one residual block.

The residues will all be positives or zero if we perform ReLU before the addition. Only positive identity increments are learned, resulting in a substantial reduction in learning ability. We want to add an unconstrained response from the weight layer to the skip layer (covering any numerical range), then use activation to achieve non-linearity. This helps the model's ability to learn any function.

$F(x) = Wx + x$ is a simple linear function if we used a single weight layer and added skip connection before ReLU. There's no use in adding a skip connection because this is identical to simply a single weight layer. Before adding a skip connection, we require a minimum of one non-linearity, which is achieved by stacking two layers.

As we've seen previously, the weight layers in these blocks are learning residuals. The performance of these blocks will not degrade as they are stacked deeper and deeper.

3.3. Face Recognition

Since we are building an emotion recognition system, and our proposed CNN model is only responsible for emotion classification, we would inevitably need to first detect faces from an input image in our processing pipeline.

3.3.1 Haar Cascades for Object Detection

Haar Cascade classifiers are an effective way for object detection. This method is proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection using a Boosted Cascade of Simple Features. Haar Cascade is a machine learning-based approach where lots of positive and negative images were used to train the classifier.

4. Experiments and Results

4.1. Environment Setup

I built a docker image based on the Nvidia Deep Learning image (PyTorch) to standardize the experiment environment. This will allow us to quickly deploy the entire experiment environment to AWS if I need to utilize the cloud computing capability. In addition, I thought using the docker container solution could resolve many environment configuration issues such as missing/incompatible dependencies/packages. Since the docker image was based on the Nvidia docker image, it should utilize the CUDA framework correctly.

4.2. Dataset

The dataset used in this paper was the FER-2013 dataset. It is a large, publicly available FER dataset consisting of 35,887 face crops. The dataset is challenging as the depicted faces vary significantly in terms of person age, face pose, and other factors, reflecting realistic conditions.

Figure 2. Selected examples of the FER-2013 dataset.

The dataset is split into training and test sets with 28,709 and 3,589 samples, respectively. Basic expression labels are provided for all samples, and all images are grayscale and have a resolution of 48 by 48 pixels. There are seven categories in this dataset and the mapping is the following: 0: Angry, 1: Disgust, 2" Fear, 3: Happy, 4: Sad, 5: Surprise, and 6: Neutral. The training set consists of 28,709 examples and the public test set consists of 3,589 examples. The human accuracy on this dataset is around 65.5%

4.3. Data Pre-processing

Perhaps for better storage efficiency, the dataset I obtained is in a CSV format of around 100 MB in file size (when compressed). Each row in the CSV file represents an image (flattened pixel) and is labeled with emotion and its usage (training/test). Therefore, I would need to convert this CSV file into a dataset of images for training/testing.

The restored dataset consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image.

4.4. Training and Optimization

4.4.1 Baseline Models

I started with a simple shallow CNN model with only three hidden layers and train it for 20 epochs. From Figure 1 (left), we can see that the model achieves a validation accuracy of around 47%.

To validate the idea that a deeper CNN could improve accuracy, I then trained a 4-layer deep CNN network for 20 epochs. In Figure 1 (right), I can observe that the model achieves a validation accuracy of about 55%, a 17% improvement compared to the 3-layer CNN model in terms of

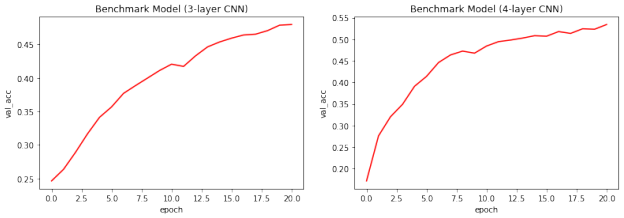

Figure 3. Validation accuracy vs. epochs of baseline CNN models with 3 and 4 Conv-layers respectively.

validation accuracy. I did not experiment with 5-layer CNN model (or CNNs with more than 4 layers) because the output size will be too small.

Therefore, subsequent experiments and optimization will be conducted based on a 4-layer CNN model architecture.

4.4.2 Spacial Batch Normalization

We will apply batch normalization for all the convolutional layers because it reduces internal covariant shift and reduces the dependence of gradients on the scale of the parameters or their initial values.

4.4.3 Activation Functions

For completeness, I first experimented with what activation functions worked best for our CNN model. I experimented with ReLU, ELU, and Tanh activation functions for the 4-layer CNN model.

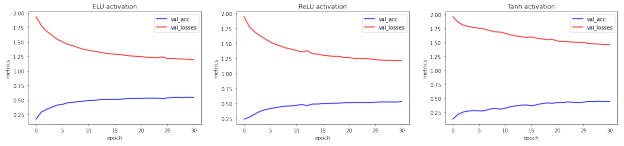

Figure 4. Cross comparisons between ELU, ReLU, and Tanh activation functions.

Both ReLU and ELU achieved similar validation accuracy, and the corresponding validation loss also decreased to lower values steadily compared to the model that uses the Tanh activation function.

ReLU function is considered the most widely used activation function in training deep neural networks in research and empirical studies. One of the greatest advantages ReLU has over other activation functions is that it does not activate all neurons at the same time. ReLU converts all negative inputs to zero, and the neuron does not get activated. This makes it very computational efficient as few neurons are activated per time, and it does not saturate at the positive region. In practice, ReLU converges six times faster than Tanh.

Therefore, we chose ReLU as our activation function for our CNN model.

4.4.4 Optimization Functions

Many optimization functions are available for training a deep neural network, such as Adam, RMSProp, SGD, etc. To find out the optimal optimization function for our CNN model, I experimented with 3 popular optimization functions and plotted the loss curve for each of them, respectively.

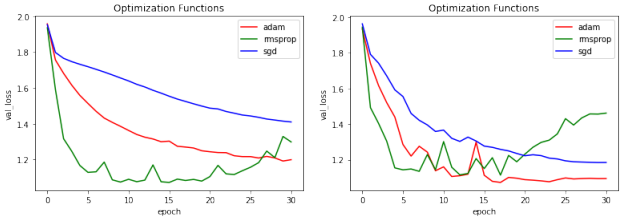

Figure 5. Cross comparisons between Adam, RMSProp, and SGD optimization functions. Before (left) and after (right) applying learning rate scheduling and gradient clipping.

With higher learning rates, we might be moving too much in the direction opposite to the gradient and may move away from the local minima, which could increase the loss. Learning rate scheduling and gradient clipping could help with this issue.

From Figure 4, we can observe that RMSProp helps the CNN model converge quickly, but then the loss rate increases gradually even with gradient clipping and learning rate schedule. Adam optimization seemed to be the best fit for our CNN model from both experiments on optimization functions.

4.4.5 Regularization

To make our CNN model more generalized during test time, we need to apply regularization. I experimented with applying weight decay and dropout regularization techniques to the CNN.

Weight Decay Weight decay adds a small penalty (1e-4), usually the L2 norm of the weights (all the weights of the model), to the loss function. Applying weight decay can help prevent overfitting, and keeping the weights small can avoid exploding gradient. Since the L2 norm of the weights is added to the loss, each iteration of your network will try to optimize/minimize the model weights in addition to the loss. This will help keep the weights as small as possible, preventing the weights from growing out of control and thus avoiding exploding gradient.

Dropout Dropout is a regularization method that approximates training many neural networks with different architectures in parallel. Concretely, applying dropout during training can be interpreted as learning an ensemble of an exponential number of subnetworks. During training, some number of layer outputs are randomly ignored or "dropped out". In effect, each update to a layer during training is performed with a different "view" of the configured layer.

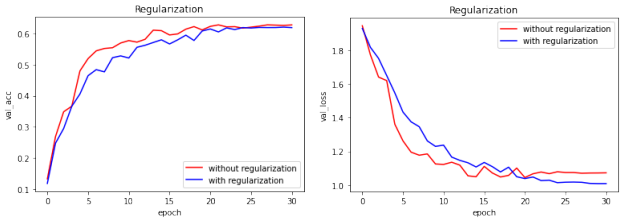

Figure 6. Before vs. after applying regularization.

After applying regularization techniques, our CNN model achieved a lower loss while maintaining similar accuracy to the CNN model trained without applying any regularization. This is a good sign that the model with regularization is more generalized.

4.4.6 CNN with Residual Blocks

Lastly, we added residual blocks to our CNN model, and we ran experiments on training a CNN model with or without adding residual blocks and observing their obtained validation accuracy.

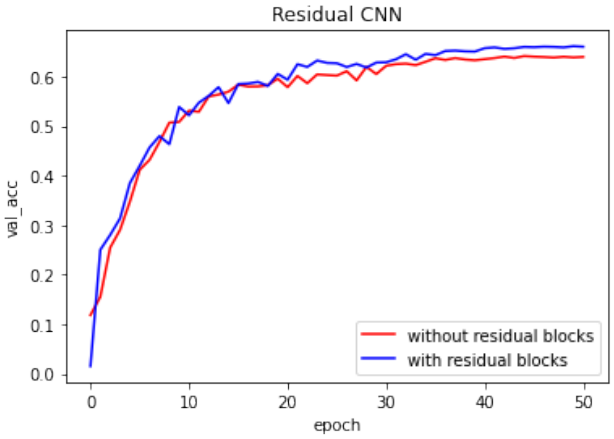

Figure 7. Added residual blocks help improve validation accuracy.

From Figure 6, we can see that the CNN model with residual blocks achieves a higher validation accuracy than the CNN model without residual blocks. This shows that a residual CNN is more robust in emotion recognition than traditional CNN models.

4.5. Face Recognition with OpenCV

To bring the model into production, we utilized OpenCV to capture video feeds from laptop video devices and convert them to frames. I found that using OBS' virtual camera as a camera feed was easier. This will allow you to capture one of your displays as a video stream, allowing you to explore images of facial expressions and test the model quickly.

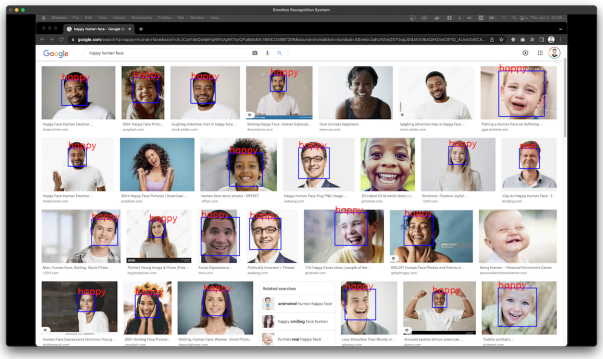

Figure 8. Demo - Running emotion recognition on a video frame captured from OBS's virtual camera feed. The author was browsing human happy face images.

Haar cascades were used to identify and draw a bounding box around faces as the region of interest (ROI). We then fed the ROI to our prediction model and obtained the classified emotion.

4.6. Results

The experimented results showed that the CNN approach was appropriate for solving the facial expression recognition problem, and the trained model could be applied in our proposed emotion recognition system.

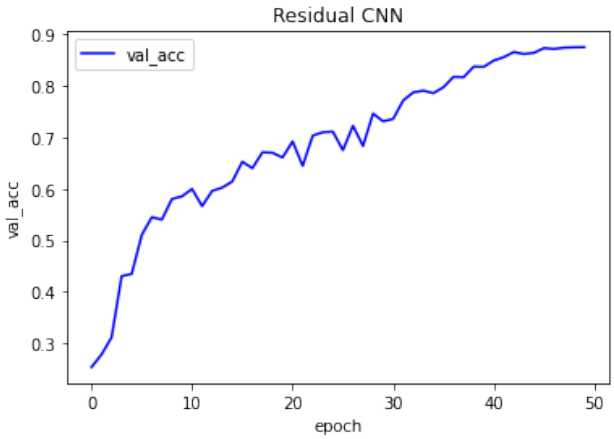

Figure 9. Validation accuracy of a 4-layer residual CNN.

Compared to our baseline CNN model with an accuracy of 47%, the final tuned residual CNN achieved an accuracy of around 85%, a 85% increase in prediction accuracy. It also surpassed the human-level accuracy of 65% for the FER-2013 dataset.

5. Conclusion

This paper conducted an empirical evaluation of training a residual CNN for emotion recognition and put the trained residual CNN model into production to demonstrate a potential use case in a real word setting.

In neural network architectures, network depth is critical, yet deeper networks are more difficult to train. Despite the fact that our model has four layers, the input image size of 48x48 is deemed quite small. The residual learning framework simplifies deep network training and overcomes the degradation problem by short-circuiting shallow layers to deep layers, resulting in better performance in computer vision tasks. These residual networks are more complex than their typical counterparts (a simple CNN), but they use the same number of parameters (weights).

Designing and creating algorithmic solutions capable of interpreting facial emotions from human faces opens up new avenues for human–computer interaction in areas such as robotics, gaming, digital marketing, and intelligent tutoring systems, among others. Human expression is consistent across individuals, and how biological and social factors may interfere with human communication over time are intriguing problems that should be investigated and computationally modeled.

6. Future Work

There are limitations of emotion recognition based solely on computer vision, and emotion could also be detected from audio, text, social interaction, and many other possible vectors. Therefore, a more robust emotion recognition system utilizing image, audio, and text might help improve the overall emotion recognition accuracy.

This paper only uses the FER-2013 dataset for empirical evaluation, and its input size is small, which limits the CNN architecture we could use. Using a larger dataset (e.g., CK+, AffectNet) and training a deeper residual CNN is worth experimenting with.

References

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[2] D. V. Sang, N. Van Dat and D. P. Thuan, "Facial expression recognition using deep convolutional neural networks," 2017 9th International Conference on Knowledge and Systems Engineering (KSE), 2017, pp. 130-135, doi: 10.1109/KSE.2017.8119447.

[3] L. Pham, T. H. Vu and T. A. Tran, "Facial Expression Recognition Using Residual Masking Network," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 4513-4519, doi: 10.1109/ICPR48806.2021.9411919.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[5] Wang, Heng Kläser, Alexander Schmid, Cordelia Liu, Cheng-Lin. (2013). Dense Trajectories and Motion Boundary Descriptors for Action Recognition. International Journal of Computer Vision. 103. 10.1007/s11263-012-0594-8.

[6] Simonyan, Karen Zisserman, Andrew. (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. Advances in Neural Information Processing Systems. 1.

[7] Tanwar, Dr. Poonam. "Facial Expression Detection using Hidden Markov model." (2017).

[8] Bao, Wentao et al. "Evidential Deep Learning for Open Set Action Recognition." 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021): 13329-13338.

[9] Mollahosseini, Ali et al. "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild." IEEE Transactions on Affective Computing 10 (2019): 18-31.

[10] Donahue, Jeff Hendricks, Lisa Guadarrama, Sergio Rohrbach, Marcus Venugopalan, Subhashini Darrell, Trevor Saenko, Kate. (2015). Long-term recurrent convolutional networks for visual recognition and description. 2625-2634. 10.1109/CVPR.2015.7298878.

[11] Kundu, Tuhin Chandran, Saravanan. (2017). Advancements and recent trends in emotion recognition using facial image analysis and machine learning models. 1-6. 10.1109/ICEECCOT.2017.8284512.

[12] Liliana, Dewi Yanti. (2019). Emotion recognition from facial expression using deep convolutional neural network. Journal of Physics: Conference Series. 1193. 012004. 10.1088/1742-6596/1193/1/012004.