WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IMPLEMENTATION OF FACIAL EMOTION RECOGNITION SYSTEM USING CONVOLUTIONAL NEURAL NETWORK

[1]Y.Mounika, [2] G. Narmada, [3]S. Mohan Kumar

[1,2,3]Department of ECE, Madanapalle Institute of Technology and Science, Andhra Pradesh, India.
E-mail: [1]mounikareddymouni91199@gmail.com, [2]gaduputinarmada@gmail.com, [3]mohansanda520@gmail.com.

ABSTRACT

Facial emotion recognition finds its application in devising Human Computer Interaction (HCI) systems for disabled persons. In this paper, Haar Cascade Classifier is implemented to detect the face in the real time image or video. Furthermore, convolution and max pooling layers of Convolutional Neural Network (CNN) are employed to extract the facial features from the detected face. Finally, the extracted facial features are classified into the critical seven human feelings: Angry, Disgust, Fear, Happy, Neutral, Sadness and Surprise by using SoftMax function of fully connected layer in CNN. The effectiveness of the implemented system is assessed on our faces representing different emotions in real time. Simulation results presented in this paper illustrates the accuracy of the implemented facial emotion recognition system in recognizing the emotions.

1. INTRODUCTION

In the field of pattern recognition, facial expression recognition (FER) has a significant impact, and researchers are working hard to develop a FER system for human–computer interaction applications. The facial expression provides sensitive information cues for constructing a FER system and is widely regarded as the most effective instrument for quickly recognising human emotions and intentions. Ekman and Friesen [1] named six basic emotions in 1971 (happy, sad, anger, surprise, fear, and disgust), and each emotion is connected with a specific facial expression that is easily recognised across cultures. A study on human information communication was proposed by the psychologist Mehrabian [2]. According to the study, facial expressions carry 55 percent of information, supporting language (sound, voice, etc.) 38 percent, and spoken language only 7 percent.

Currently, a FER system is a key component of artificial intelligence, with potential real-world applications in areas such as psychological studies [3], driver fatigue monitoring, interactive game design, portable mobile application to automatically insert emotions in chat and assistance systems for autistic people, facial nerve grading in the medical field [4], emotion detection system used by disabled to assist a caretaker, and socially intelligent robot with emotional intelligence [5].

Most of the research work in FER system follows the framework of pattern recognition [**6**]. It consists of three phases: face detection, facial feature extraction, expression classification. It is quite substantial and noteworthy to research these phases. In this current survey, various phases of facial expression analysis are discussed with distinct algorithms to classify seven basic expressions. Face detection is performed by Haar classifier and feature extraction and classification is performed by CNN.

2. FACIAL EMOTION RECOGNITION SYSTEM MODEL

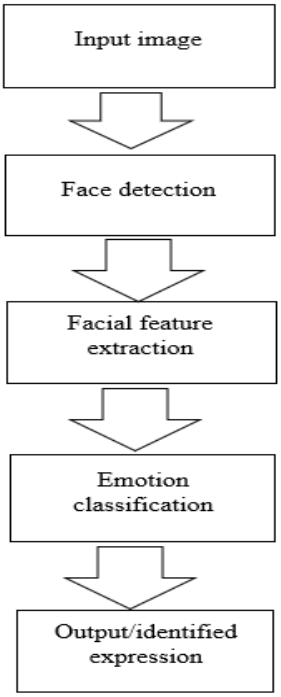


Figure 1: Block schematic of the Model

Face Detection:

Face detection is an artificial intelligence-based computer system that can recognise and locate human faces in digital pictures and videos. It may be thought of as a subset of object-class detection, in which the goal is to locate and quantify all objects belonging to a certain class. In this example, faces – inside a given image or images.

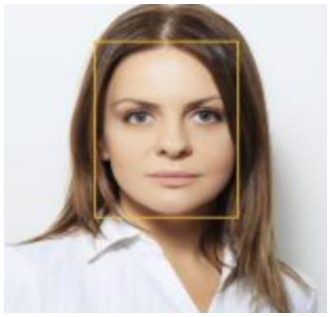


Figure 2: Face Detection

Facial Feature Extraction:

The technique of extracting face component parts such as eyes, nose, and mouth from a human face picture is known as facial feature extraction. Face feature extraction is critical for the setup of processing techniques including face tracking, facial expression detection, and face recognition.

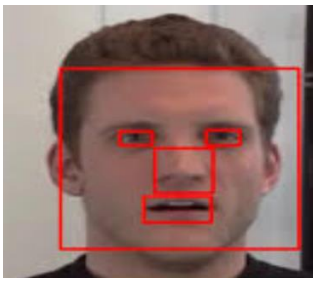


Figure 3: Facial Feature Extraction

Emotion Classification:

Emotion classification refers to the task of recognizing an individual's emotions and classifies an emotion from their reactions and responses.

Figure 4: Emotion Classification

Face Detection Using Haar Cascade Classifier:

In the real-time face detector, classifiers were used. Haar classifiers, also known as Haar cascade classifiers[7], are machine learning object identification programmes that identify objects in images and videos.

Making a Haar Cascade Classifier:

The algorithm can be explained in four stages:

- Calculating Haar Features
- Creating Integral Images
- Using Adaboost
- Implementing Cascading Classifiers

It's vital to remember that, like other machine learning models, this approach requires a large number of positive images of faces and negative images of non-faces to train the classifier.

Calculating Haar Features:

Haar is comparable to Kernels, a feature commonly employed to detect edges. The eye region is darker than the upper check region, and the nose region is brighter than the eye region in all human faces. The location and size of these matchable traits will aid us in detecting a face[7].

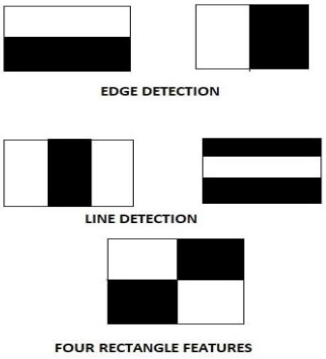


Figure 5: Types of Haar features

Here are some Haar features that can be used to determine whether or not there is a face. The Haar feature indicates that the black region is represented by +1 and the white region by -1. An image is displayed in a 24×24 window. Each feature is a single value produced by subtracting the total of pixels in the white and black rectangles. Many features are now calculated using all possible sizes and positions of each kernel. We must find the total of pixels under white and black rectangles for each feature computation. There will be 160000+ Haar features for a 24×24 window, which is a significant quantity. They used integral pictures to address the problem. It reduces the sum of pixels calculation, regardless of how many pixels there are, to a four-pixel process.

Integral Images:

The primary goal of an integral picture is to compute the area. So, instead of adding up all of the pixel values, we'll take the corner values and do a simple calculation. The integral image at location x , y contains the sum of the pixels above and to the left of x , y, inclusive:

$$ii(x,y)=\sum_{x'\leq x,y'\leq y} i(x',y')$$

The integrated picture for this input image will be calculated by adding all of the above and left pixels together.

1	4
3	5

Input image

0	0	0
0	1	5
0	4	13

Integal image

Figure 6: Integrated image

The sum of the pixels within rectangle D can be computed with four array references:

- The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D.
- The sum within D can be computed as 4 + 1 − (2 + 3).

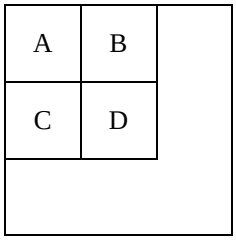


Figure 7: Integrated image

This one is less difficult than the last one. Converting an image into an integrated image has this advantage..

Adaboost:

Adaboost is used to remove Haar's superfluous feature. A classifier can be created by combining a small number of these features. The most difficult part is locating these characteristics. Both the features and the classifier are trained using a variation of AdaBoost.

Figure 8: Adaboost

We can use adaboost to figure out which of the 160000+ features are relevant. After finding all of the features, a weighted value is applied, which is used to determine whether or not a specific window is a face.

$F(x) = a1f1(x) + a2f2(x) + a3f3(x) + a4f4(x) + a5f5(x) + \ldots.$

F(x) denotes a strong classifier, while f(x) denotes a weak classifier. Weak classifiers always return a binary result, such as 0 or 1. The value will be 1 if the feature is present, otherwise it will be 0. A strong classifier is usually made up of 2500 classifiers. Selected features are deemed to be acceptable if they outperform random guessing, i.e. they must detect more than half of the cases.

Cascading:

Assume we have a 640×480 resolution input image. Then we must move 24×24 windows throughout the image, evaluating 2500 features for each window. Using all 2500 features in a linear fashion, it determines whether there is a threshold and, if so, whether it is a face or not. We will use cascade instead of using all 2500 features for 24×24 times. Out of 2500 features, the first ten are classified in one classifier, the next 20-30 in another, and the remaining 100 in yet another. As a result, the complexity will increase. Instead of going through all 2500 features for a 24×24 window, we may exclude non-face from the first step. Assume we have a picture. It could be a face if the image passes the first step, which stores 10 classifiers. The image will then be checked for the second time. If the image fails the first stage, we may just discard it. Cascading is the most efficient and smallest classifier. Cascading is a simple way to create non-face zones.

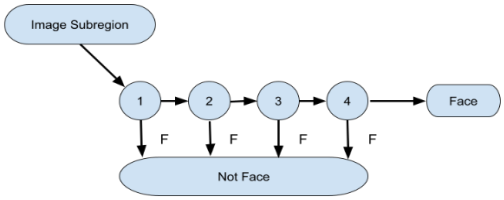


Figure 9: Cascade work flow

FER Dataset:

The dataset used to train the model comes from a previous Kaggle Facial Expression Recognition Challenge (FER2013) [8]. The data consists of grayscale images of faces at a resolution of 48×48 pixels. The faces have been automatically registered such that they are more or less centred in each image and take up around the same amount of area. The goal is to categorise each face into one of seven categories based on the emotion expressed in the facial expression (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

3. DETAILED DISCUSSION ON THE FACIAL EMOTION RECOGNITION SYSTEM

Deep learning [9] is a common computer vision approach. Convolutional Neural Network (CNN) [10] layers were chosen as the building blocks for our model architecture. When processing pictures, CNNs are known to mimic how the human brain functions. A convolutional neural network's usual architecture includes an input layer, some convolutional layers, some dense layers (also known as fully-connected layers), and an output layer. These are stacked layers that are organised in a linear fashion. The model is constructed as Sequential() in Keras, and further layers are added to build the architecture.

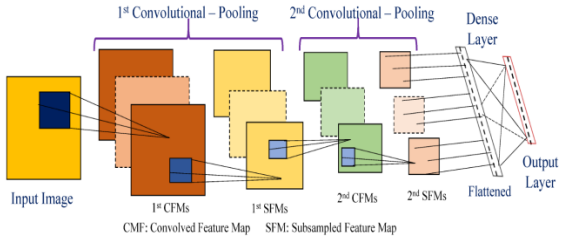


Figure 10: FER CNN Architecture

Input Layer:

Because the input layer's dimensions are pre-determined and fixed, the image must be pre-processed before being fed into it. For face detection in the image, we utilised OpenCV, a computer vision package. The OpenCV file haar_cascade_frontal face _default.xml includes pre-trained filters and employs Adaboost to swiftly locate and crop the face. cv2.cvtColor is used to convert the cropped face to grayscale, and cv2.shrink is used to resize it to 48-by-48 pixels. When compared to the original RGB format with three colour dimensions, this step considerably reduces the dimensions (3, 48, 48). Every image may be supplied into the input layer as a (1, 48, 48) numpy array thanks to the pipeline.

Convolutional Layers:

The numpy array is handed to the Convolution2D layer, where one of the hyperparameters is the number of filters. With randomly generated weights, the set of filters (also known as the kernel) is unique. To construct a feature map, each filter, (3, 3) receptive field, glides across the source image with shared weights. For example, edge and pattern detection, convolution provides feature maps that represent how pixel values are enhanced. Filter 1 is applied to the entire image to build a feature map. Other filters are applied one by one, resulting in a collection of feature maps.

Pooling:

Pooling is a technique for reducing dimension that is commonly used after one or more convolutional layers. When developing CNNs, this is a critical stage because adding more convolutional layers can significantly increase processing time. We utilised the MaxPooling2D pooling approach, which uses (2, 2) windows across the feature map to maintain only the maximum pixel value. The pooled pixels form a picture with reduced dimensions of 4.

Dense Layers:

The dense layer (also known as fully connected layers) is modelled after the way neurons transmit impulses in the brain. It accepts a large number of input features and transforms them using trainable weights and layers. Forward propagation of training data and backward propagation of errors are used to learn these weights. Back propagation begins by calculating the weight adjustment required for each layer before analysing the difference between prediction and true value. By adjusting hyper-parameters like learning rate and network density, we can regulate the training speed and complexity of the architecture. The network can gradually make adjustments as more data is fed in, until errors are minimised. The more layers/nodes we add to the network, the better it will be able to take up signals. The model grows progressively prone to overfitting the training data, as nice as it seems. Dropout is one way to prevent overfitting and generalise on unknown data. During training, Dropout randomly picks a subset of nodes (typically less than 50%) and sets their weights to zero.

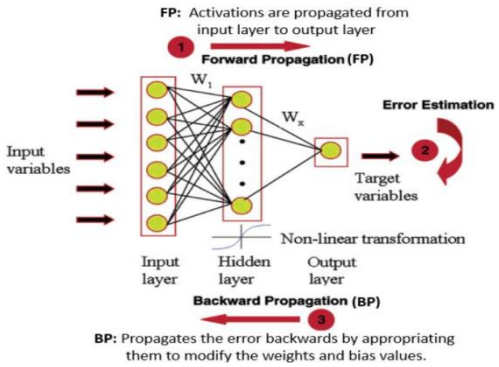


Figure 11: CNN Forward & Backward Propagation

Output Layer:

At the output layer, we employed SoftMax instead of the sigmoid activation function. For each emotion class, this output appears as a likelihood. As a result, the model can display the detailed probability composition of facial expressions. As you shall see later, classifying human face expressions into a single emotion is inefficient. Our facial expressions are typically far more complex, containing a variety of emotions that might be used to effectively characterise a specific expression. To begin with, we created a simple CNN with an input, four convolution layers, one dense layer, and an output layer. Our final convolutional net architecture has 12 layers, with one max-pooling layer after every three convolution layers.

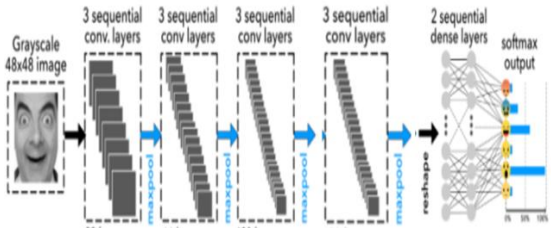
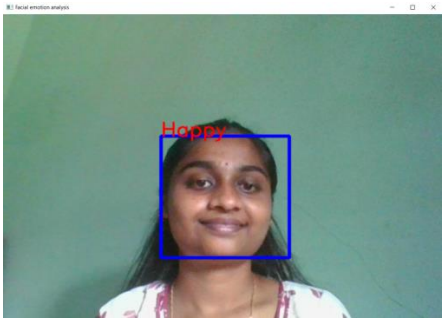
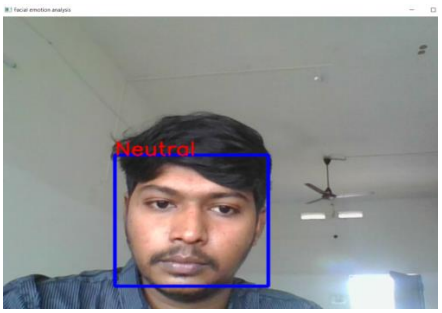


Figure 12: Final Model CNN Architecture

Qualitative Assessment Of The Implemented System:

The effectiveness of the implemented system was assessed in real time representing different emotions of our batchmates. In Blue color box ,face was detected and red color letters are emotion of the detected face.





4. CONCLUSION

In this paper, Haar Cascade Classifier was implemented to detect the face in the real time image and video. Furthermore, convolution and max pooling layers of Convolutional Neural Network (CNN) were employed to extract the facial features from the detected face. Finally, the extracted facial features were classified into the critical seven human feelings: Angry, Disgust, Fear, Happy, Neutral, Sadness and Surprise by using softmax function of fully connected layer in CNN.

REFERENCES

[1] Ekman P., and Friesen W.V.: "Constants across cultures in the face and emotion", *J. Pers. Soc. Psychol.*, 1971, 17, (2), pp. 124– 129

[2] Mehrabian A.: "Communication without words", *Psychol. Today.*, 1968, 2, (4), pp. 53– 56

[3] Lau B.T.: 'Portable real-time emotion detection system for disabled', *Expert Syst.*, 2010, 37, (9), pp. 6561– 6566

[4] Dulguerov P., Marchal F., and Wang D. *et al*: "Review of objective topographic facial nerve evaluation methods", *AM. J. Otol.*, 1999, 20, (5), pp. 672– 678

[5] Alazrai R., and George Lee C.: "Real-time emotion identification for socially intelligent robot", *IEEE International Conference Robotics and Automation*, USA, May 2012, pp. 14– 18.

[6] Tian Y.L., Kanade T., and Cohn J.F." Facial expression analysis", in ' *Handbook of face recognition*' Springer, New York, NY, 2005, pp. 247– 275

[7] Viola P, Jones M. Robust, "Real Time Face Detection," *International Journal of Computer Vision*, pp: 137-154, 2004.

[8] [Online]. Available: https://www. kaggle.com/msambare/fer2013

[9] [Online].Available:https://www.youtube.com/watchv=dafuAz_CV7Q&list=PL9ooVrP1hQOEX8BKD plfG86ky8s7Oxbzg

[10] [Online].Available:https://www.researchgate.net/publication/339347740_Facial_emotion_recognition_usingconvolutional_neural_networks_FERC