IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HUMAN EMOTION DETECTION USING HAAR-CASCADE AND CNN

T. Nalini prasad[1], J.N.M. Krishna[2], K. Chandini[2], M. Aakaash Chowdary[2], M.V.S.S. Abhinav[2]

Assistant Professor[1], B.Tech Students[2]
Electronics and Communication Engineering
SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE(A), Chinna Amiram, Bhimavaram, India

Abstract: In this research, we present a method for teaching convolutional neural networks to recognize facial expressions in a computer (CNN) and Haar cascade method is used to detect the face before emotion recognition. The datasets from FER-2013 were utilized to train the CNN algorithm. This achieves a decent level of accuracy for training and testing points. This work makes use of the FER-2013 expression dataset, which includes seven facial expression labels (happy, sad, surprised, fear, anger, disgust, and neutral). This system managed to achieve an acceptable level of accuracy.

Index Terms - — Face Expression Recognition, Convolutional Neural Networks, Haar-cascade, FER-2013 data set

INTRODUCTION

Computer vision is an interdisciplinary scientific topic that examines how machines can comprehend digital image or video processing at a high level . Using techniques for gathering, processing, analysing, and extracting high-dimensional data from photos, video, text, and other sources, computer vision can solve problems. Scene reconstruction, picture compression, image restoration, vehicle identification, and face expression recognition are examples of sub-domains of computer vision. The primary indicator of human expressions is the identification of facial emotions. The many different ways that emotions can be represented are through word, voice, and facial expressions.

The classifier method using the Haar-Cascade and the deep learning method utilising the Convolution Neural Network are currently the main focus of this research. An efficient method for object detection is facial detection using a Cascade classifier based on Haar features . In terms of high network depth and algorithmic approach, the deep neural network is similar to the CNN. Although the fundamental idea behind CNN is nearly identical to that of the multilayer perceptron, each neuron in CNN will be constructed using a two-dimensional structure. Only two-dimensional data from voice recorders and images can be used with CNN. In order to read an input and extract a feature depending on the characteristics of the input, CNN has numerous layers of learning. The layers are represented as vectors of numbers.

A convolution layer plus a pooling layer make up this feature extraction layer. Before the images with small sizes are joined into the input volume, each neuron's output in the convolution layer will calculate its weight Units.

The manuscript will concentrate on seven facial expressions—angry, disgusted, fearful, pleased, sad, surprised, and neutral—based on the training dataset. The FER2013 dataset is used to train the suggested model. The related work or literature review, methodology, the Haar-Cascade Classifier, and CNN will all be covered in the next section. The results and conclusions will be given in the final section.

Methodology

A. HAAR CASCADE:

In their 2020 publication " Facial expression recognition using feature fusion” based on the deep learning[1] object identification approach known as Haar Cascade[2] using machine learning, a cascade function is trained using a large number of both positive and negative images (where positive images are those where the object to be detected is present, negative are those where it is not).The next step is to utilize it to find items in other pictures. Fortunately, OpenCV provides pre-trained Haar cascade algorithms that are categorized (e.g., into faces, eyes, and so on) based on the images they were trained on.

Similar to the principle of the convolutional, the Haar cascade method entails extracting features from images using a form of "filter". These filters, which go by the name Haar features, appear as follows:

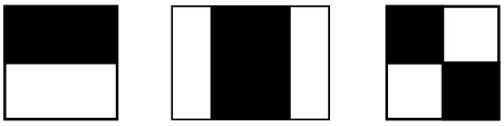


Fig 1: Edge feature Line feature 4-Rectangle feature

The concept is to apply various filters to the image and examine it window by window. After that, all of the pixel intensities for the white and black areas of each window are added up. The value of the feature extracted is finally determined by deducting those two summations. A feature's high value should ideally imply that it is important. Specifically, if we use the Edge feature on the B&W image above

As a result of the big value we will get, the algorithm will almost certainly return an edge feature. Naturally, the actual intensities of pixels are never equal to white or black, and we frequently see the following scenario:

The basic principle is still the same: the greater the result (i.e., the distinction between the black and white summations), the more likely it is that the window has a meaningful feature. Imagine the vast number of features that this calculation has returned at this point. The concept of the Summed-area table, commonly known as the Integral Image, was the answer. It is an algorithm and data structure for producing the total of values in a grid's rectangular subset. In order to acquire the summations of pixel intensities within a window, fewer computations must be performed.

The goal is to convert an input image into a summed-area table, where any point (x, y value's) is equal to the total of all pixels to its left and above, inclusive:

$$I(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$

I (x, y) is the intensity of the corresponding pixel in the original image, and I (x, y) is the value of the integral image pixel at the position (x, y).

B. *Convolution Neural Network (CNN)*

Convolutional Neural Network [3][4] is a Machine Learning algorithm which is used for supervised learning to analyses data. It is also known ConvNet.

CNN is primarily to classify images, cluster images by similarity and perform object recognition.

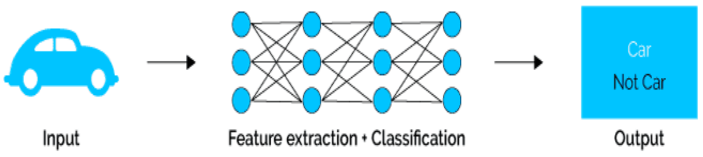


Fig 2: Out sketch of CNN

Here the feature learning is achieved by using Convolutional layer and pooling layer.

Then the fully connected layer acts as a classifier on the features to detect the valid emotion.

The four primary ConvNet operating phases as follows:

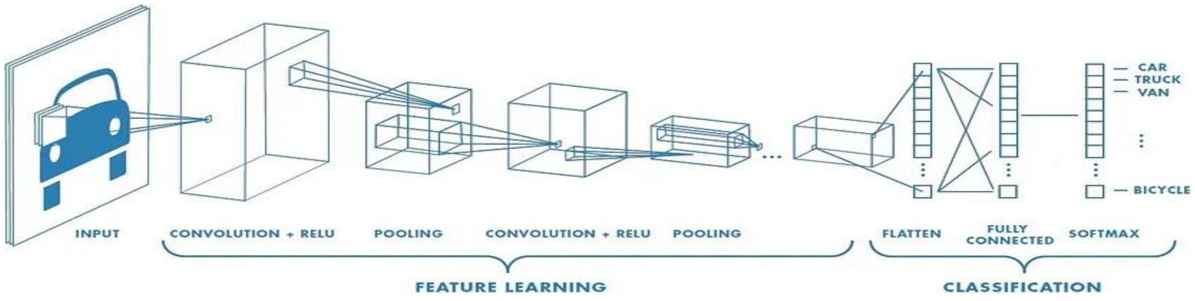
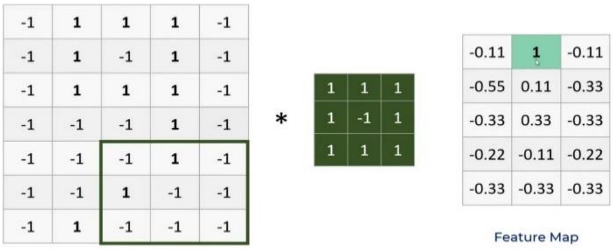


Fig 3: Combination of Convolution+Relu

❖ Convolution step

The word "ConvNets" is derived from the "convolution" operation. In the case of a ConvNet, the main goal of convolution is to extract features from the input image. Convolution uses small squares of input data to learn visual attributes, preserving the spatial relationship between pixels. We won't dive into the specifics of convolution's mathematics here; instead, we'll focus on how it applies to photographs.

Think about a 5 × 5 image with only 0 and 1 as the pixel values (note that for a grayscale image, pixel values range from 0 to 255,the green matrix below is an exception, where the only valid values for pixels are 0 and 1.



-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

*

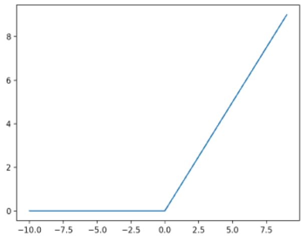
1	1	1
1	-1	1
1	1	1

-0.11	1	-0.11
-0.55	0.11	-0.33
-0.33	0.33	-0.33
-0.22	-0.11	-0.22
-0.33	-0.33	-0.33

Fig 4: Convolution step by using mask

❖ **Non linearity:**

Rectified Linear Unit, or ReLU, is a non-linear operation. Its output is given by



Output = max (Zero, Input)

Fig 5: ReLU graph

ReLU is an element wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero. The purpose of ReLU is to introduce non-linearity in our ConvNet, since most of the real-world data we would want our ConvNet to learn would be non-linear Convolution is a linear operation element wise matrix multiplication and addition, Hence, by including a nonlinear function like ReLU, we account for nonlinearity.

Below provides a thorough understanding of the ReLU operation. It shows the ReLU operation applied to one of the feature maps obtained in Figure 5 above. The output feature map here is also referred to as the 'Rectified' feature map.

❖ **Pooling**

Each feature map's dimensionality is decreased but the most crucial data is retained by spatial pooling, also known as sub sampling or down sampling. Spatial pooling comes in a variety of forms: Maximum, Average, Sum, etc.

In the Max Pooling scenario, we specify a spatial neighborhood (for instance, a 2 by 2 window) and select the largest element from the rectified feature map inside that window. We might choose to take the average (Average Pooling) or total of all the items in that window rather than just the largest one. Max Pooling has been demonstrated to function better in practice.

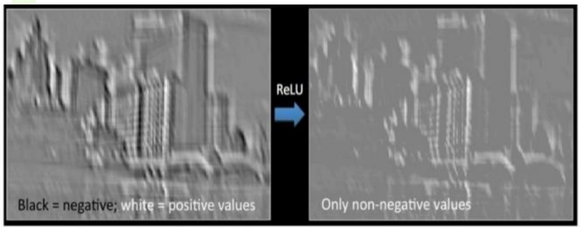
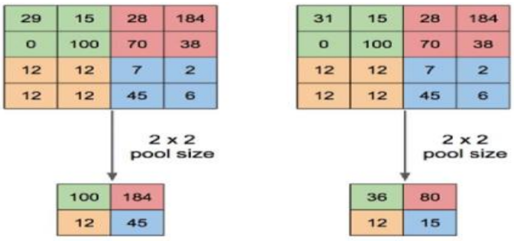


Fig 6: Before Input Image and After Output Image

Figure below shows an example of Max Pooling operation on a Rectified Feature map (obtained after convolution +ReLU operation) by using a 2×2 window.



29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

100	184
12	45

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

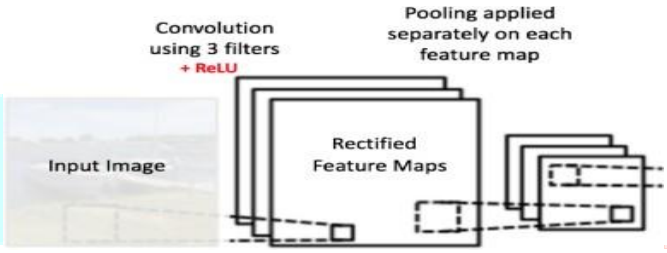
36	80
12	15

Fig 7: Max pooling and average pooling

We slide our 2x2 window by2 cells (also called 'stride') and take the maximum value in each region. As shown in above Figure, this reduces the dimensionality of our feature map. In the network shown in below figure, pooling operation is applied separately to each feature map (notice that, due to this, we get three output maps from three input maps).

Fig 8 : Shows convolution + RELU and pooling step

The purpose of pooling is to gradually reduce the input representation's spatial size. In particular, pooling reduces and streamlines the input representations (featured-dimension) and controls over fitting by reducing the number of parameters and computations in the network enables the network to adapt to subtle visual distortions, translations, and transformations. As we take the maximum/average value in a local neighborhood, a slight distortion in the input will not affect the outcome of pooling.
It enables the network to adapt to subtle visual distortions, translations, and transformations. As we take the maximum/average value in a local neighborhood, a slight distortion in the input will not affect the outcome of pooling.
lets us arrive at a nearly scale-equivariant representation of our image, which is particularly effective because it allows us to recognize objects in an image regardless of their location.
It's critical to realize that these layers serve as the foundation of any CNN. As seen in the image, there are two sets of Convolution, ReLU, and Pooling layers. The second Convolution layer uses six filters to conduct convolution on the output of the first Pooling Layer, resulting in the production of six feature maps in total. Afterwards, ReLU is individually applied to each of these six feature maps. The six rectified feature maps are then subjected to distinct maximum pooling operations.
Collectively, these layers try to make the features somewhat equivariant to scale and translation while extracting the useful characteristics from the images, introducing non-linearity into our network, and reducing feature dimension.


Convolution using 3 filters + ReLU
Pooling applied separately on each feature map
Input Image
Rectified Feature Maps

❖ **Classification Part (Fully Connected layer):**

Other classifiers like SVM canal may be used, but will stick to the Fully Connected layer, a typical Multi-Layer Perception that uses a SoftMax activation function in the output layer.
According to the definition of "Fully Linked," every neuron in the layer below is linked to every neuron in the layer above.
High-level characteristics of the input image are represented in the output from the convolution and pooling layers. The Fully Connected layer's objective is to categories the input image into several classes using these attributes and the training dataset. The image classification problem, for instance, offers four potential results, as indicated in the following figure:

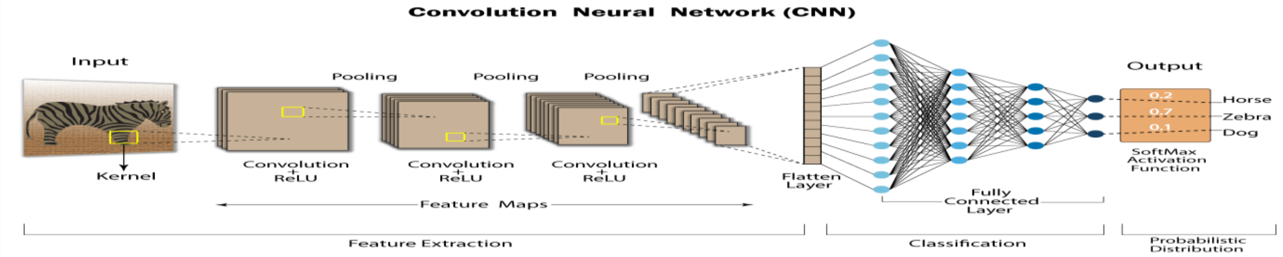


Fig 9: Fully Connected Layer

A cheap method of learning non-linear combination soft he se characteristics, in addition to classification, is by adding a fully linked layer. For the classification job, the majority of the features from convolution and pooling layers may be useful, but combinations of those features may be even more effective.

one. The Fully Connected Layers output probabilities ass up to uses the SoftMax as its activation function to ensure this. A vector of arbitrary real-valued scores is compressed by the SoftMax function to a vector of values between zero and one that add to one.

FER 2013 Data Set:

- This dataset [5] is used for training Convolutional Neural Network.
- It has training dataset of 35000 grey-scale facial images.
- The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

- The data 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.

TABLE 1: Number of data in the FER- 2013 data set

Micro-expression(Classification)	Validation Data		Training Data	Dataset Total
	Public	Private		
Angry	467	491	3995	4953
Disgust	56	55	436	547
Fear	496	528	4097	5121
Happy	895	879	7215	8989
Sadness	653	594	4830	6077
Surprise	415	416	3171	4002
Contempt	607	626	4965	6198
	3589	3589	28709	35887

Results:

Finally, we got the face emotion through image. In this image we can analyse two boxes, in these boxes indicates of two methods (Haar Cascade and CNN). Blue colour box indicates Haar Cascade method this method helps in detecting the faces and after detecting it will crop the face. In this second method, helps in detecting the emotion in the face, before detecting emotion the algorithm gets trained with the data base (FER 2013). In this data base we have 1000+ images in one emotion this all images will get trained with algorithm, not only one emotion all 7 emotion each has 1000+ images all are trained to the algorithm. At last pink colour box indicates CNN and it shows emotion name on the pink box.

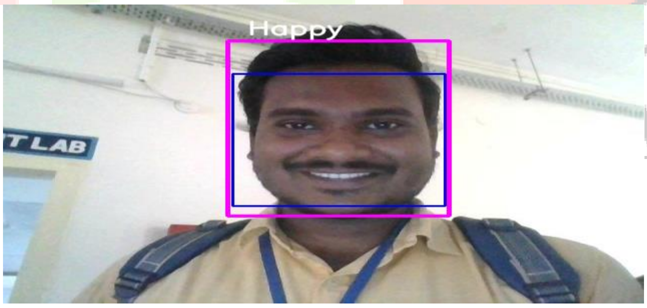


Fig 10 : The above figure shows **Happy** image by using CNN and Haar- cascade.

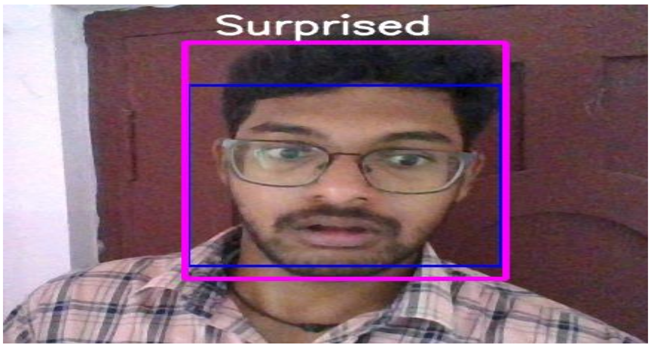


Fig 11 : The above figure shows **Surprised** image by using CNN and Haar- cascade

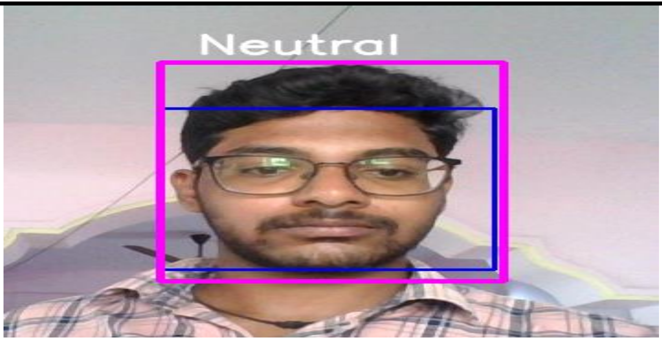


Fig 12 : The above figure shows **Neutral** image by using CNN and Haar-cascade

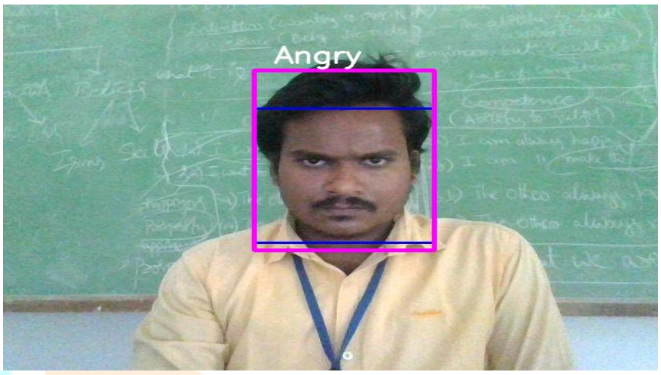


Fig 13 : The above figure shows **Angry** image by using CNN and Haar-cascade

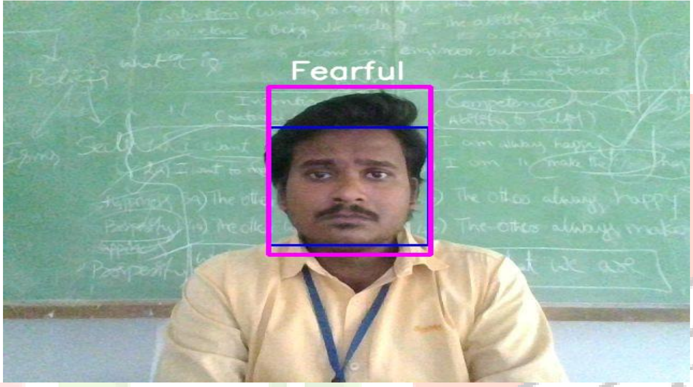


Fig 14 : The above figure shows **Fearful** image by using CNN and Haar-cascade

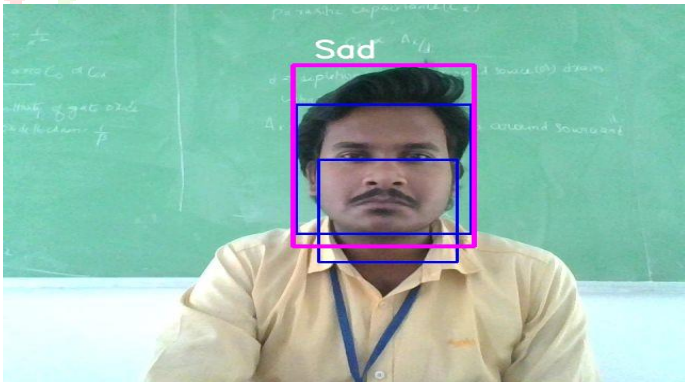


Fig 15 : The above figure shows **Sad** image by using CNN and Haar-cascade

Conclusion:

An emotional facial expression reveals a person's status, mood, and current feeling through nonverbal communication. If we examine an emotion at different levels, we can comprehend the feeling of another. The percentages of emotions change dramatically among stages. In this study, we trained and classified seven different types of common emotions using a convolutional neural network with additional layers using data set FER-2013 (Training and Testing). Haar Cascade classifiers were used to recognize faces before recognizing emotion. Convolutional neural networks are used to design and propose an emotion recognition system that uses facial data. The system's effectiveness is around 65+.Given that only the FER-2013 dataset was used for training without the use of other datasets, an accuracy of 0.66 is admirable as it demonstrates the efficiency of the model. The proposed system's efficiency will be greatly increased if given more training data while keeping the same network layout.

References:

[1] A. Sajjanhar, Z. Wu, and Q. Wen, "Deep learning models for facial expression recognition,"in 2018 Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1-6: IEEE

[2] Li X L and Niu H T, “Facial expression recognition using feature fusion based on VGG-NET,” Computer Engineering and Science, Vol. 42(03), pp. 500-509, 2020.

[3] Yao M Z and Huang G W, “Facial expression recognition based on convolutional neural network,” Computer Knowledge and Technology, Vol. 16(16), pp. 19-23, 2020.

[4] J. Chen, Y. Lv, R. Xu, and C. Xu, "Automatic social signal analysis: Facial expression recognition using difference convolution neural network," Journal of Parallel and Distributed Computing, vol. 131, pp. 97-102, 2019.

[5] K. Clawson, L. Delicato, and C. Bowerman, "Human Centric Facial Expression Recognition," 2018.