JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# A Deep Learning Model for Facial Expression to Emoji Generation and Statistical Representation

**[1]Prof. Balaji A. Chaugule, [2]** Bhakti Sambal**, [3]Janakee Shelke, [4]Aachal Chimankar, [5]Unnati Shinde**

[1]Head of the Dept., Department of Information Technology, Zeal College of Engineering and Research, Pune, Maharashtra, India.
[2]Department of Information Technology, Zeal College of Engineering and Research, Pune, Maharashtra, India.
[3]Department of Information Technology, Zeal College of Engineering and Research, Pune, Maharashtra, India.
[4]Department of Information Technology, Zeal College of Engineering and Research, Pune, Maharashtra, India.
[5]Department of Information Technology, Zeal College of Engineering and Research, Pune, Maharashtra, India.

***Abstract :*** ***The "A Deep Learning Model for Facial Expression to Emoji Generation and Statistical Representation" project aims to recognize and categories facial expressions in real time using a convolutional neural network (CNN) architecture. Using computer vision approaches and powerful deep learning algorithms, the system seeks to recognize and analyze emotions expressed in face images taken with a webcam. The detected face parts are then clipped and standardized in size. These processed face photos are then fed into a pre-trained CNN model that has been thoroughly trained on a varied dataset of facial emotions. Leveraging renowned libraries such as TensorFlow, OpenCV, and Tkinter for deep learning, image processing, and user interface components, respectively, this project amalgamates computer vision techniques and advanced deep learning algorithms to deliver an interactive and captivating platform for real-time emotion recognition. Its versatility extends across various domains encompassing user sentiment analysis, market research, and human- computer interaction.***

## I. INTRODUCTION

The “A Deep Learning Model for Facial Expression to Emoji Generation and Statistical Representation” can quickly recognise and categorise facial expressions and associate them with appropriate emoji portrayals. It examines face images collected from webcams or video feeds, using a combination of computer vision methods and powerful deep learning algorithms, to provide real-time predictions about the emotions represented. This endeavour aims to provide an immersive user interface experience by overlaying recognised emotions with corresponding emojis on the user's face. At its heart is a convolutional neural network (CNN) architecture that has been rigorously trained on a large dataset (FER-2013) of annotated face photos representing emotions such as happiness, anger, fear, neutrality, sadness and surprise. Throughout the training phase, the model adeptly learns to effectively extract relevant features from input photos.

The "A Deep Learning Model for Facial Expression to Emoji Generation and Statistical Representation" project combines several components to improve user engagement. A user-friendly graphical interface displays live video feeds that are enhanced with real-time emotion recognition and emoji overlays. This interface allows users to interact with the application, take snapshots, create reports, and view emotion data. Furthermore, the project simplifies report preparation and emotional dispersion research by allowing users to preserve snapshots together with relevant metadata such as usernames, dates, and timestamps. Users may gain insights into the distribution of detected emotions over time via pie charts or bar plots, allowing for a more in-depth knowledge of their emotional displays. In summary, the A Deep Learning-Powered System for Facial Expression-To-Emoji Conversion project smoothly amalgamates computer vision and deep learning.

## II. METHODOLOGY

### A. Convolutional Neural Network :

A typical convolutional neural network (CNN) structure involves an input layer, numerous convolutional layers, fully-connected layers, and an output layer at the very tip. The CNN architecture is constructed with six levels in total, ignoring the layers for input and output. The following figure illustrates the schematic that represents the framework of the Convolutional Neural Network that is being used for the project. Excluding the input and output layers, the CNN architecture consists of a total of six layers. The diagram depicting the architecture of the Convolutional Neural Network utilized in the project is depicted in the preceding illustration.[3]

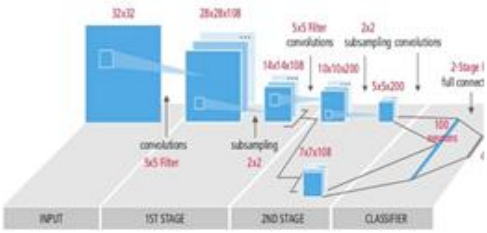


Fig. 1: CNN Architecture

**i. Input Layer:**

The input layer of the network has set dimensions, necessitating preprocessing of the image before feeding it into the layer. Additionally, for testing, images captured via laptop webcams are utilized. These images undergo preprocessing, involving face detection and cropping using the OpenCV HaarCascade Classifier, followed by normalization.

**ii. Convolution and Pooling (Conv-Pool) Layer:**

Convolution and pooling operations are conducted through batch processing, where each batch consists of N images, and the CNN filter weights are updated accordingly. Each convolutional layer receives input batches of images with four dimensions: N x Color-Channel x Width x Height. Similarly, the feature maps or filters for convolution are also four-dimensional, consisting of the number of feature maps in, the number of feature maps out, filter width, and filter height. Within each convolutional layer, a four-dimensional convolution is performed between the image batch and the feature maps. Post- convolution, the only parameter that undergoes alteration is the width and height of the images. After each convolutional layer, down sampling or subsampling is employed for dimensionality reduction, a process known as pooling. Max pooling and average pooling are two widely recognized pooling methods. In this project, max pooling is applied following convolution. A pool size of (2 x 2) is selected, dividing the image into a grid of blocks each measuring (2 x 2) pixels and extracting the maximum value from each block. Following pooling, only the height and width dimensions of the image are affected. The architecture incorporates two convolutional layers and a pooling layer. The initial convolutional layer operates on an input image batch of dimensions (N x I x 48 x 48), where N represents the batch size, i denotes the number of color channels, and both the height and width of the image are 48 pixels. Convolution with a feature map of (1 x 20 x 5 x 5) yields an image batch of size (N x 20 x 44 x 44). Subsequently, pooling is conducted with a pool size of (2 x 2), resulting in an image batch of dimensions (N x 20 x 22 x 22). This is followed by a second convolutional layer employing a feature map of (20 x 20 x 5 x 5), generating an image batch of size (N x 20 x 18 x 18). Subsequent pooling with a pool size of (2 x 2) produces an image batch measuring (N x 20 x 9 x 9).

**iii. Fully Connected Layer:**

In this architecture, two fully-connected hidden layers are employed, each comprising 500 and 300 units, respectively. During training, these layers' weights are adjusted iteratively through forward and backward propagation of training data errors. Back propagation involves evaluating the variance between predictions and actual values and then adjusting the weights of each layer accordingly. To optimize training efficiency and control the complexity of the architecture, hyperparameters such as learning rate and network density are adjusted. These hyperparameters include learning rate, momentum, regularization parameter, and decay. The output from the second pooling layer is structured as Nx20x9x9, while the input to the first fully-connected hidden layer is Nx500. Consequently, the output of the pooling layer is flattened to Nx1620 size and passed to the first hidden layer. Subsequently, the output of the first hidden layer is forwarded to the second hidden layer, which contains Nx300 units. The output of the second hidden layer is then directed to the output layer, the size of which corresponds to the number of facial expression classes being considered.

**iv. Output Layer:**

The output of the second hidden layer is linked to the output layer, which comprises seven distinct classes. Employing the SoftMax activation function, the output is derived based on the probabilities associated with each of the seven classes. The predicted class corresponds to the one with the highest probability among them.

**B. HAAR Cascade:**

The API-supplied image undergoes processing through the HAAR cascade, leveraging a dataset for training purposes. To construct an effective model, we'll utilize the Fer2013 dataset. HAAR-Like features demonstrate notable accuracy in detecting faces from varying perspectives. The integral image is calculated from the input image. Haar-like features are rectangular patterns used for detecting objects. These features are simple and efficient to compute. The integral image allows for rapid computation of Haar-like features, which significantly speeds up the detection process. Instead of recalculating the sum of pixel intensities for each feature, the integral image enables these sums to be computed efficiently using just four values. After classifying the emotions that each face in a picture is conveying, it recognizes every face in the frame.

**C. Dataset:**

For training and testing, the Kaggle Facial Expression Recognition Challenge (FER2013) dataset is used. 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. The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories. The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

## III. PROPOSED SYSTEM

The proposed method uses the HAAR cascade approach to detect faces so that additional image processing can extract facial features. Subsequently, six distinct categories of emotions are created using an SVM classifier. The OpenCV package's HAAR feature is used to superimpose emojis that correspond with the recognized emotions over the individuals' faces. Different facial characteristics are utilized to convey different emotions. These traits include high cheekbones, mouth openings, eyebrows, wrinkles around the nose, wide-open eyes, and many more. Next, an SVM Classifier is used to categorize emotions into seven different kinds. Emojis that match to the identified emotions are superimposed over the subjects' faces using the OpenCV package's HAAR feature. Emojis can be used as filters in these applications because they have inbuilt face detection algorithms

that detect faces with ease. When a face is identified, the associated emoji appears on screen to provide a picture of the emotion that was identified. It produces statisticsdescribing the emotions identified using a pie chart.

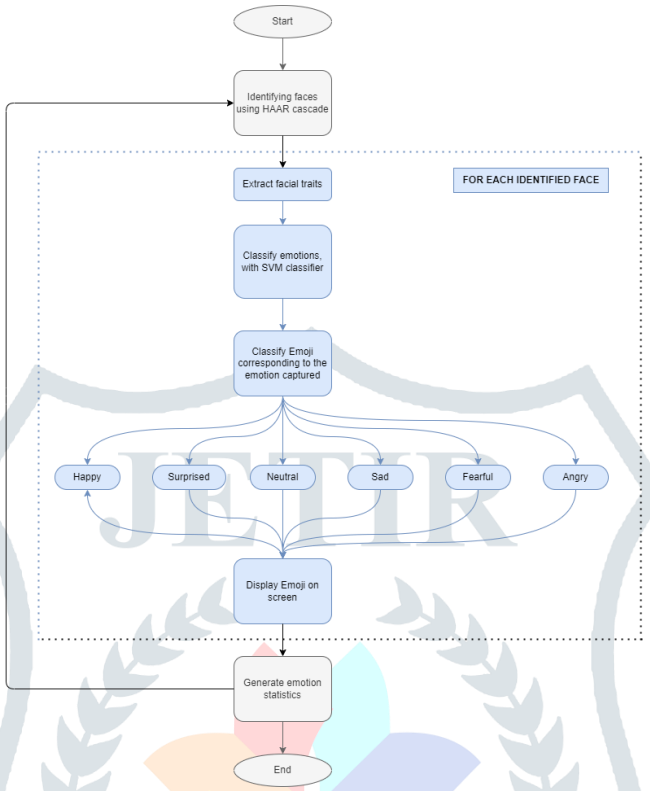


Fig. 2: System Architecture

## IV. EXPERIMENTAL ANALYSIS

Building a deep learning model for facial expression to emoji generation involves training a neural network to predict emojis based on facial expressions extracted from images. This model also includes a module of statistical representation

**Accuracy Improvement Experiments:**

The accuracy of the model declines and the assortment of loss parameters rises proportional to the accuracy accomplished when it is trained on fewer epochs. In a similar vein, the model demonstrates substantially higher accuracy—up to 95%—while trained on an increasing number of exemplars. The model initially underwent training over a single epoch, and wore an accuracy score of 24.30%. The accuracy upped to 26.75% while the epochs were boosted to 2. As the epochs were boosted further to 5, the model's accuracy went up to 32.26%. The accuracy climbed up to 45% post training on 10 epochs, and it eventually touched 75% accuracy after training on 50 epochs. In final form, we utilised 70 epochs to attain a 92% accuracy performance. Below is the chart representation of the accuracy of the model achieved successfully.

| Number of Epochs | Accuracy (in %) |
|---|---|
| 1 | 24.30 |
| 2 | 26.75 |
| 5 | 32.26 |
| 10 | 75.50 |
| 70 | 92.25 |

Table 1: Increment observed in accuracy

## V. RESULT AND DISCUSSION

The output pane, displayed in the above graphics, is where the webcam captures user expressions and identifies the corresponding emotions. The equivalent emoji appears along the left portion of the display when the emotion is detected. When an individual present in front of the webcam changes their facial expressions, so does this emoticon. Consequently, it modifies in tandem with the subject's shifting facial expression when facing the webcam. Below are the captured results.

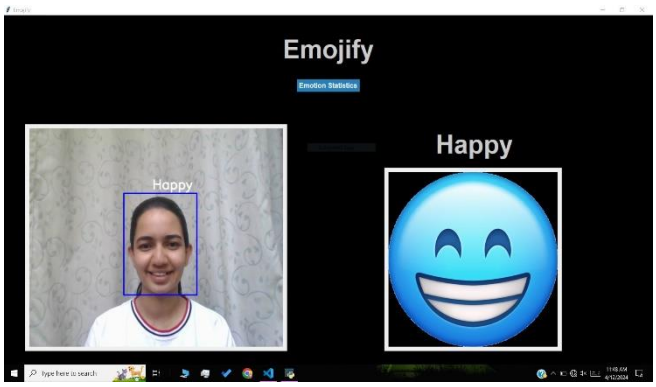


Fig. 3: 'Happy' Emoticon generated

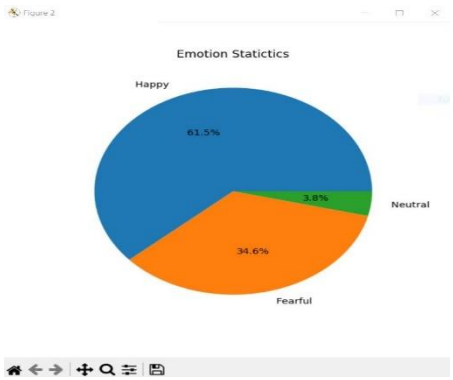


Fig. 4: Statistical representation of 'Happy' Emotion

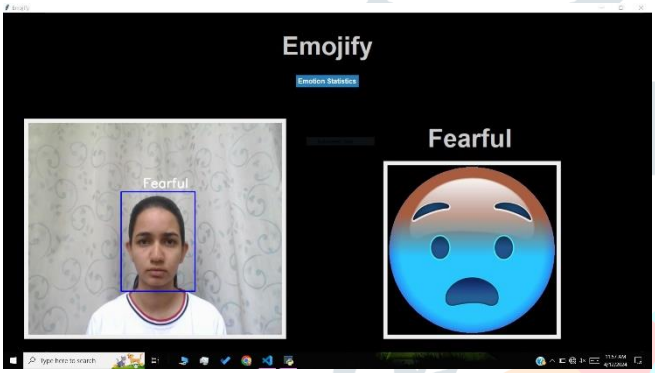


Fig. 5: 'Fearful' Emoticon generated

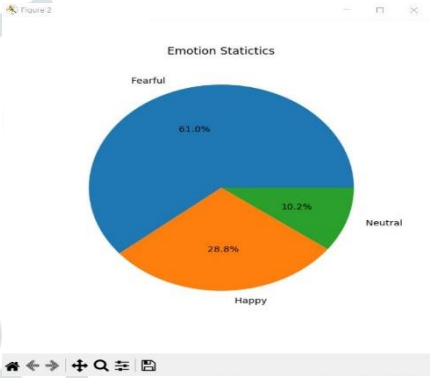


Fig. 6: Statistical representation of 'Fearful' Emotion

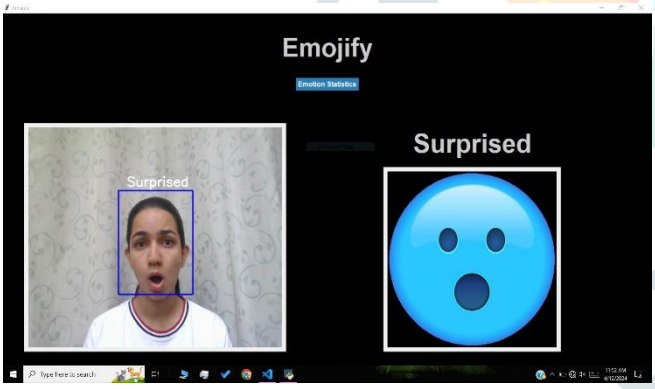


Fig. 7: 'Surprised' Emoticon generated

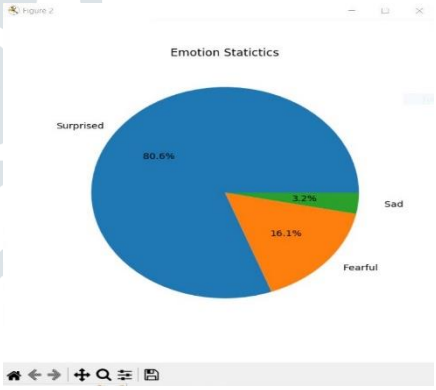


Fig. 8: Statistical representation of 'Surprised' Emotion

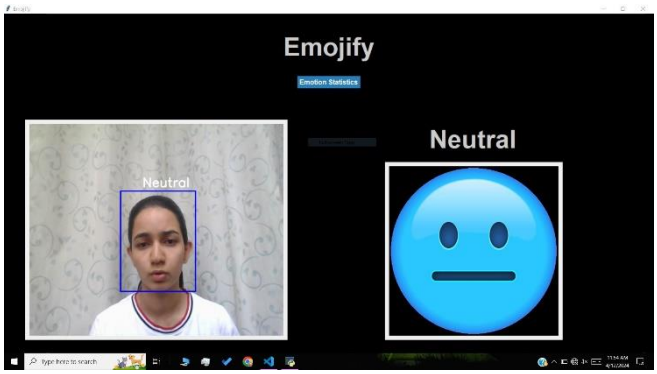


Fig. 9: 'Neutral' Emoticon generated

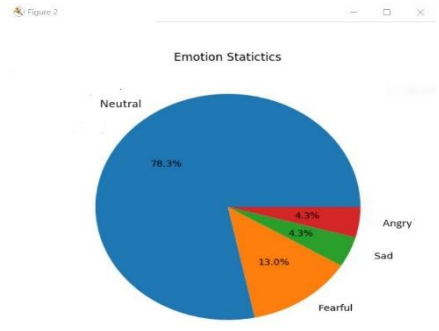


Fig. 10: Statistical representation of 'Neutral' Emotion

## VI. CONCLUSION

The present research has discussed a CNN-based methodology for 'A Deep Learning Model for Facial Expression to Emoji Generation and Statistical Representation'. Utilising the FER2013 dataset, a CNN model was developed and assessments of the architecture were carried out to obtain test 92.25% using 70 epochs. Using a webcam, this leading-edge algorithm has been implemented to classify people' emotions in real time. The webcam captures a series of visuals, it analyses then classifies into different emotions and provides an emoji response. It anticipates people's emotions and uses emoticons to represent them. These include capturing images, preprocessing them, identifying faces in them, extracting features, and classifying them.

## REFERENCES

**[1]** EMOJIFY-CREATE YOUR OWN EMOJIS WITH DEEP LEARNING Sagar Chilivery, Sandeep Pukale, Yashraj Sonawane in February2022, IRJETS.
**[2]** Deep Learning Models for Facial ExpressionRecognition: Atul Sajjanhar, ZhaoQi Wu, Quan Wen, 2018, IEEE.
**[3]** Survey Paper On: "From Faces to Emojis: A Deep Learning Approach to Convert Facial Expressions into Emoticons", Prof. Balaji Chaugule, Bhakti Sambal, Janakee Shelke, Aachal Chimankar, Unnati Shinde, Nov 2023, IJIRCCE.
**[4]** Deep Learning Models for Facial ExpressionRecognition: Atul Sajjanhar, ZhaoQi Wu, Quan Wen, 2018, IEEE.
**[5]** EMOJI CLASSIFICATION USING CNN: Pradeep Bharati, Omkar Singh, April 2023, IJCRT.
**[6]** Facial Expression Recognition via Deep Leaming: Abir Fathallah, Lotti Abdi, Ali Douik, March 2018, IEEE.
**[7]** C. Marechal et al., « Survey on Al-Based Multimodal Methods for Emotion Detection, in High-Performance Modelling and Simulation for Big Data Applications: Selected Results of the COST Action IC1406 cHiPSet, J. Kolodziej et H. Gonzalez-Velez, Ed. Cham: Springer International Publishing, 2019, p. 307-324.
**[8]** N. Morgan, Deep and wide: Multiple layers inautomatic speech recognition, Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 1, pp. 7–13, 2012.
**[9]** The Extended Cohn–Kanade Database. Availableonline: http://www.consortium.ri.cmu.edu/ckagree/.
**[10]** D. H. Kim, W. J. Baddar, J. Jget, Y. M. Ro, Multi-Objective Based Spatio-Temporal Feature Representation Learning Robust to ExpressionIntensity Variations for Facial Expression Recognition», IEEE Trans. Affect. Compute, 2019
**[11]** Z. Hao, "The development of emoji in the intelligent era," 2021 International Conference on Intelligent Design (ICID), 2020, pp. 55-59, DOI: 10.1109/ICID52250.2020.00020.
**[12]** T. Cao and M. Li, "Facial Expression Recognition Algorithm Based on the Combination of CNN and K-Means," presented at the Proceedings of the 2019 11th International Conference on Machine Learning and Computing, Zhuhai, China, 2019.
**[13]** 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: I EEE.
**[14]** K. Clawson, L. Delicato, and C. Bowerman, "Human Centric Facial Expression Recognition," 2018
**[15]** J. M. B. Fugate, A. J. O'Hare, W. S. Emmanuel, "Emotion words: Facing change," Journal of Experimental Social Psychology, vol. 79, pp. 264-274, 2018.
**[16]** D. Meena and R. Sharan, "Facial Recognition and Approaches to Recognition", International Conference 2016 (ICRAIE), pp. 1–6, 2016.
**[17]** Barsoum, Emad, et al, Training deep networks for facial expression recognition with crowd- sourced label distribution, ACM International Conference on Multimodal Interaction ACM, 2016.
**[18]** EMOJIFY- USING DEEP LEARNING Onkar Mahindrakar, Parth Chaudhari, Kunal Chaudhari, Sairaj Kambale, IRJMETS 2023.