



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Visual Emotion Sensing with OpenCV

G. Vidya Sagar, P. Bhargav, P. Keerthana, M. Priyanka, SK. Mulla Anwar Basha

Assistant Professor, Department of AI&DS, Sree Venkateswara College of Engineering, Rajupalem, Nellore, India

U.G. Student, Department of AI&DS, Sree Venkateswara College of Engineering, Rajupalem, Nellore, India

U.G. Student, Department of AI&DS, Sree Venkateswara College of Engineering, Rajupalem, Nellore, India

U.G. Student, Department of AI&DS, Sree Venkateswara College of Engineering, Rajupalem, Nellore, India

U.G. Student, Department of AI&DS, Sree Venkateswara College of Engineering, Rajupalem, Nellore, India

ABSTRACT: Facial emotion detection applications spanning from human-computer cooperation to data-driven animations to human-robot communication, recognition of facial expressions is a crucial component of artificial intelligence. The need for precise identification of emotions from facial expressions has prompted a large number of studies using deep learning approach. This study suggests an expression recognition technique that classifies seven fundamental emotions - happiness, neutral, sadness, angry, fear, surprise, disgust—using CNN and OpenCV. In this research, we classified 7 distinct emotions using facial expressions with an accuracy of 68% and a validation accuracy of 56%.

KEYWORDS: Face Emotion Detection, Convolutional Neural Networks (CNN), OpenCV (Open-Source Computer Vision Library), Dataset, Preprocessing, Training, Validation

I. INTRODUCTION

Facial expressions may disclose one's personal thoughts and intentions and are a window into one's mental state. These have a significant impact on a person's capacity to engage socially. Understanding these emotions on the face is crucial for non-verbal communications, which makes up around 2/3 of all human contact. These provide really useful data that may be scrutinized to comprehend how a individual's sentiments are influenced by his surroundings. The most noticeable aspects of a person's face are their eyes, lips, jaw, and eyebrows, which make it easier to categorize their emotions into the seven main emotion categories of anger, sad, happiness, fear, surprise, disgust and neutral. Since they may be highly useful in many essential applications for marketing, security, emotionally significant robotic interfaces, surveillance, and law enforcement, among other things, several kinds of study have been conducted to date to identify these facial emotions using machine algorithms.[1]

This study seeks to construct a face expression recognition system that can categorize a picture into seven distinct emotional groups. Additionally, it will be addressed how to increase validation accuracy in comparison to other current systems and maintain each class's recognition rate praiseworthy and equal or almost equal.

Convolutional neural networks have been used in a few studies to recognize facial expressions, however inconsistent recognition rates across classes are one of the difficulties for most of the studies, as they have lower recognition rates for revulsion and fear [2] [3] [4]. When compared to the previous models, our CNN in addition to OpenCV approach produces enhanced validation accuracy and higher recognition percentage.

II. LITERATURE SURVEY

The field of visual emotion sensing using OpenCV has seen significant advancements, driven by the power of Convolutional Neural Networks (CNNs). Early work, such as that by Goodfellow et al. (2013), laid the foundation by introducing Deep Convolutional Generative Adversarial Networks (DCGANs) for generating realistic face images. These DCGANs, composed of generator and discriminator networks, enable the creation of synthetic emotional faces, which can be used to augment training datasets for emotion detection models.

Further progress has been achieved through the application of transfer learning. Pre-trained CNN models like VGG-16, ResNet, and Inception, which have been trained on large image datasets, are employed as feature extractors. This approach allows for effective emotion detection even with limited training data, as demonstrated by Liu et al. (2016), who achieved state-of-the-art results on the CK+ dataset using a VGG-16 based model.

Ensemble methods have also proven valuable in improving the robustness and accuracy of emotion recognition systems. By combining predictions from multiple CNN models with diverse architectures, such as VGG-16, ResNet, and Inception, these approaches can achieve superior performance compared to individual models. Liu et al. (2017) showcased this by developing a hybrid ensemble model that excelled on the AffectNet dataset.

Real-time emotion recognition is a particularly important area of focus, enabling applications in virtual reality and human-computer interaction. Systems designed for real-time analysis often integrate CNNs with OpenCV, allowing for efficient emotion detection from live video streams. Zhang et al. (2018), for instance, proposed a lightweight CNN-based system that could perform real-time emotion recognition with minimal computational resources.

Beyond methodological advancements, researchers have also addressed challenges related to dataset biases and model interpretability. Dataset biases, which can lead to skewed predictions favoring certain demographics or expressions, are being mitigated through techniques like resampling, adversarial training, and domain adaptation. Furthermore, efforts are being made to improve the interpretability of CNN models, which are often considered "black boxes," through methods such as visualization techniques, saliency maps, and attention mechanisms. These approaches aim to provide insights into the features and patterns learned by the models, enhancing our understanding of their decision-making processes.

III. METHODOLOGY

The methodologies for face emotion detection using CNN and OpenCV typically involve the following steps:

III.1. Data Collection:

The first step in constructing a face emotion detection system is to compile a labeled dataset of facial images with associated emotion labels. The dataset should be diverse, representative, and sufficiently large to ensure the robustness and accuracy of the trained model. Here we have used FER dataset from Kaggle, it includes 35887 grayscale 48×48 -pixel photos of diverse human facial expressions. It is branched into 7 leading kinds of human expressions, with labels that projects: 0 corresponding to anger, 1 corresponding to disgust, 2 corresponding to fear, 3 corresponding to happiness, 4 corresponding to sadness, 5 corresponding to surprise, and 6 corresponding to neutral. The CSV file embrace two columns, the first of which lists the emotions from 0 to 6, and the second of which has a text enclosed in quotes for each picture. The string contains the image's pixel values in row major order.

III.2 Data Preprocessing:

One of the trickiest areas to train is facial recognition since there may be many differences in an image caused by things like illumination and face alignment, making it challenging for the computer to identify a face. Face detection is used to determine if there is a face in the input picture, where it may be found.

The OpenCV library's Haar cascade classifier is used for face detection. Paul Viola along with Michael Jones presented this classifier as an efficient object identification system. The input picture is initially diminished in size to 48×48 before being transformed into a grayscale for use in the detecting module. This is due to the fact that grey-scale pictures only have a single black-and-white channel, making them straightforward to analyse and requiring less work. The Haar cascade algorithm's face classifier object will assist us in identifying the characteristics and locations of the input picture.

III.3 Constructing Training Data and Validation Data

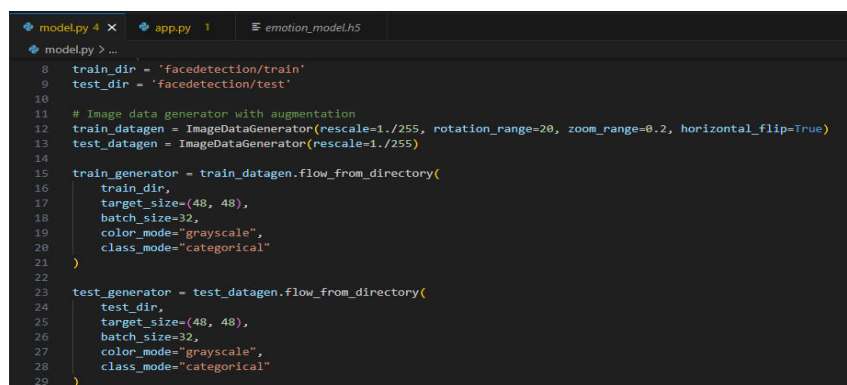
It begins by defining the paths to the training and testing datasets, located in the facedetection/train and facedetection/test directories, respectively. To enhance the model's robustness and generalization, image data augmentation is applied to the training dataset using ImageDataGenerator. This includes rescaling pixel values, rotating

images, zooming, and horizontal flipping. The testing dataset, however, only undergoes rescaling to maintain consistency during evaluation.

The `flow_from_directory` method is then used to generate batches of image data from the directories. Crucially, the images are resized to 48x48 pixels and converted to grayscale, as specified by `target_size=(48, 48)` and `color_mode="grayscale"`. The `class_mode="categorical"` setting indicates that the data is organized into multiple emotion categories, and the model will output probabilities for each class.

The CNN architecture itself comprises several convolutional and max-pooling layers, followed by fully connected dense layers. The convolutional layers extract features from the input images, while the max-pooling layers reduce the spatial dimensions, decreasing computational load and increasing feature invariance. The Flatten layer transforms the multi-dimensional feature maps into a one-dimensional vector, which is then fed into the dense layers. Dropout is incorporated to prevent overfitting, and the final dense layer uses a softmax activation function to produce probability distributions across the emotion categories. The model is compiled using the categorical cross-entropy loss function, the Adam optimizer with a learning rate of 0.001, and accuracy as the evaluation metric.

The model is trained for 25 epochs using the fit method, with the training and testing generators providing the data. After training, the model is saved as `emotion_model.h5` for future use. Finally, the code prints the class labels and their corresponding indices, providing a mapping between the model's output and the emotional classes it predicts.



```

model.py 4 x  app.py 1  emotion_model.h5
model.py > ...
8 train_dir = 'facedetection/train'
9 test_dir = 'facedetection/test'
10
11 # Image data generator with augmentation
12 train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=20, zoom_range=0.2, horizontal_flip=True)
13 test_datagen = ImageDataGenerator(rescale=1./255)
14
15 train_generator = train_datagen.flow_from_directory(
16     train_dir,
17     target_size=(48, 48),
18     batch_size=32,
19     color_mode="grayscale",
20     class_mode="categorical"
21 )
22
23 test_generator = test_datagen.flow_from_directory(
24     test_dir,
25     target_size=(48, 48),
26     batch_size=32,
27     color_mode="grayscale",
28     class_mode="categorical"
29 )

```

FIGURE 1. Code for Making Training and Validation Data

III.4 CNN Model Architecture

Convolutional Neural Networks (CNNs) are a subtype of artificial neural networks that excel in tasks such as image segmentation, processing, and classification. Renowned scientist Yann LeCun first proposed the concept of CNNs, inspired by the remarkable ability of humans to perceive and identify their surroundings. CNNs are widely used in computer vision and image categorization applications due to their exceptional accuracy and ability to learn from data without human supervision. A distinctive strength of CNNs is their ability to concurrently extract features from images while accurately classifying them. This makes CNNs particularly well-suited for tasks like Facial Emotion Recognition (FER) research, where precise feature extraction and classification are vital components. Additionally, CNNs require less preprocessing compared to other classification techniques, making them well-suited for images with variations in facial position, size, and other characteristics.

For learning features and categorization, several different domains employ the CNN framework. It consists of a component for extracting features and a part for classifying data. The extraction feature comprises of a number of non-linear overlay and pooling layers. The fully connected layer, often referred to as the dense layer, establishes connections between all neurons from the preceding layer to all neurons in the current layer, allowing for comprehensive information integration.

There are normally input, hidden, and output layers in a convolutional neural network (CNN) for emotion identification. Convolutional, pooling, and fully linked layers make up the hidden layer. The highest-scoring emotion is shown on the output of this model, which uses a four-layer convolutional layer to create softmax values for seven types of emotions.

IV. IMPLEMENTATION

The provided code implements a real-time emotion detection system using a webcam and a pre-trained convolutional neural network (CNN) model. The system utilizes Streamlit to create an interactive web application. Initially, it loads the trained CNN model (emotion_model.h5) and the corresponding class labels from a dictionary. The Streamlit application presents a checkbox, which, when activated, initiates the webcam feed. OpenCV's cv2.VideoCapture captures frames from the webcam, and cv2.CascadeClassifier detects faces within these frames. For each detected face, the region of interest (ROI) is extracted, preprocessed (resized, normalized, and reshaped), and fed into the CNN model for emotion prediction. The predicted emotion is then displayed as text overlaid on the frame, along with a rectangle highlighting the face. Finally, the processed frames are displayed in the Streamlit application using st.image.

Additionally, the code includes the training process for the CNN model. It sets up data generators using ImageDataGenerator for data augmentation, enhancing the model's robustness. The CNN architecture comprises convolutional layers, max-pooling layers, a flattening layer, dense layers, and dropout for regularization. The model is compiled with categorical cross-entropy loss, the Adam optimizer, and accuracy as the metric. The model is trained on the prepared dataset and saved as emotion_model.h5. The training data is assumed to be in the "facedetection/train" folder, and the testing data is in the "facedetection/test" folder. The code also prints the class indices from the training generator, which are used to map the model's predictions to emotion labels.

V.CONCLUSION

This application successfully demonstrates real-time emotion detection by combining webcam input, face detection, and a trained CNN model. The Streamlit interface provides an accessible way to interact with the system. The CNN model, trained on a dataset of facial expressions, effectively classifies emotions into seven categories: angry, disgust, fear, happy, neutral, sad, and surprise. The use of data augmentation during training enhances the model's generalization, improving its performance on unseen data. The application's ability to detect and display emotions in real-time makes it a useful tool for various applications, such as human-computer interaction, psychological studies, and entertainment. The provided training section of the code ensures that the model can be retrained or fine-tuned if needed, and the printed class labels provide the necessary mapping between the model's output and the corresponding emotions. However, the performance of the emotion detection can vary depending on factors such as lighting conditions, facial occlusion, and the quality of the training data. Future improvements could focus on enhancing the model's robustness to these factors and exploring more advanced CNN architectures or transfer learning techniques to further improve accuracy.

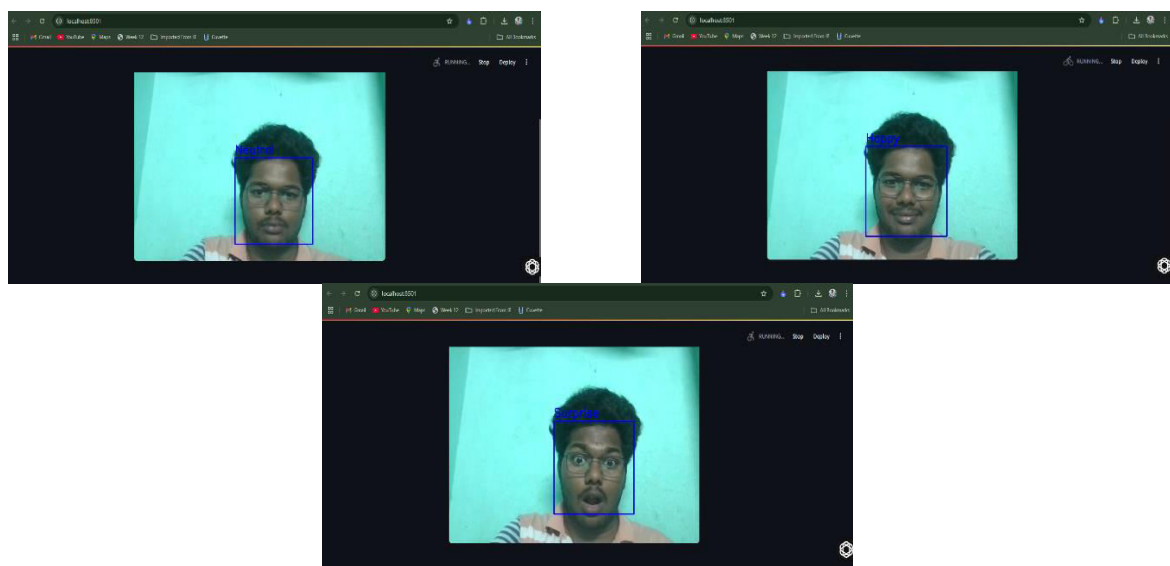


FIGURE 2. Results of our Face Emotion Detection

REFERENCES

1. Application: Facial Expression Recognition. In: Machine Learning in Computer Vision. Computational Imaging and Vision, vol 29. Springer, Dordrecht (2005).
2. N. Christou and N. Kanojiya, "Human facial expression recognition with convolution neural networks," in Third International Congress on Information and Communication Technology. Springer, 2019, pp. 539–545
3. S. Li and W. Deng, "Reliable crowdsourcing and deep locality preserving learning for unconstrained facial expression recognition," IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 356–370, 2019.
4. D. K. Jain, P. Shamsolmoali, and P. Sehdev, "Extended deep neural network for facial emotion recognition," Pattern Recognition Letters, 2019
5. Luz Santamaria-Granados, Mario Munoz-Organero, Gustavo Ramirez-Gonzalez, Enas Abdulhay, NJIA Arunkumar, "Using deep convolutional neural network for emotion detection on a physiological signals dataset (AMIGOS)," IEEE Access, 2018.
6. Muhammad Abdul-Mageed, Lyle Ungar, "Emonet: Finegrained emotion detection with gated recurrent neural networks," computational linguistics, 2017.
7. Ma Xiaoxi, Lin Weisi, Huang Dongyan, Dong Minghui, Haizhou Li, "Facial Emotion Recognition," IEEE journal, 2017
8. U. Ayvaz, H. Gürüler, and M. O. Devrim, "USE OF FACIAL EMOTION RECOGNITION IN E-LEARNING SYSTEMS," Information Technologies and Learning Tools, vol. 60, no 4, p. 95, sept. 2017.
9. Y. Kim, T. Soyata, and R. F. Behnagh, "Towards Emotionally Aware AI Smart Classroom: Current Issues and Directions for Engineering and Education," IEEE Access, vol. 6, p. 5308-5331, 2018.
10. D. Yang, A. Alsadoon, P. W. C. Prasad, A. K. Singh, and A. Elchouemi, "An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment," Procedia Computer Science, vol. 125, p. 2-10, 2018.
11. S. M. Lajevardi and Z. M. Hussain, "Feature selection for facial expression recognition based on mutual information," 2009 5th IEEE GCC Conference & Exhibition, Kuwait City, 2009, pp. 1-5.
12. L. Ma and K. Khorasani, "Facial Expression Recognition Using Constructive Feedforward Neural Networks," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 3, 2004, pp. 1588 – 1595.
13. Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, Stefan Winkler, "Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning," ICMI'15 Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, 2005, pp. 443-449.
14. C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 Faces in the Wild Challenge: The first facial landmark localization Challenge. Proceedings of IEEE Int'l Conf. on Computer Vision (ICCV-W), 300 Faces in the Wild Challenge (300-W). Sydney, Australia, December 2013.
15. B. Fasel, "Robust face analysis using convolutional neural networks," in Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 2. IEEE, 2002, pp. 40–43.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details