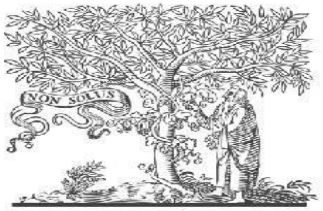
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 21st Feb 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 03

10.48047/IJIEMR/V12/ISSUE 03/28

Title **EMOTION DETECTION USING DEEP LEARNING**

Volume 12, ISSUE 03, Pages: 202-207

Paper Authors
Dr. Shanmukha Rao V, Suneeth Kumar B, Sri Karthik M, Venu M

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Emotion Detection Using Deep Learning

Dr. Shanmukha Rao V[1], Associate Professor, Department of Information Technology, Andhra Loyola Institute of Engineering and Technology, Vijayawada
shanmukharao.v@gmail.com
Suneeth Kumar B[2], Department of Information Technology, Andhra Loyola Institute of Engineering and Technology, Vijayawada
Sri Karthik M[3], Department of Information Technology, Andhra Loyola Institute of Engineering and Technology, Vijayawada
Venu M[4], Department of Information Technology, Andhra Loyola Institute of Engineering and Technology, Vijayawada

Abstract
These days, deep learning techniques have seen a lot of success in many different areas, computer vision being just one of them. Convolutional neural network (CNN) models can be trained to perform such analysis and detect emotions in human faces. Using facial features, we train a system to identify how a student is feeling. All together, our system consists of three stages: face detection with Haar Cascades, normalisation, and emotion recognition with CNN on the FER 2013 database with seven different expressions. The results obtained demonstrate the viability of face emotion recognition in the classroom, providing teachers with the means to tailor their presentation to the moods of their students.

Keywords: CNN, FER2013, Haar Cascade

Introduction

Human activity and character traits are reflected in a person's facial expressions, which can be identified using facial expression recognition systems. Ekman and Friesen of the United States established a set of six universal emotions (anger, fear, disgust, sadness, surprise, and happiness) in the twentieth century. In recent years, facial expression recognition has gained a lot of attention due to its potential applications in clinical practice, social robotics, and education. Multiple studies have pointed to the significance of emotional intelligence. Consider, a teacher's feedback options consist primarily of exams, questionnaires, and classroom observations; however, these traditional methods have a low efficiency rate. To better encourage student learning, teachers can use students' facial expressions to modify their teaching approach and course materials. The goal is to apply Convolutional Neural Network (CNN), a popular deep learning algorithm for image classification, to the task of implementing emotion recognition. The system will automatically analyse facial expressions. The process involves several steps of image processing to derive feature representations. Each of the seven basic emotions—neutral, anger, fear, sadness, happiness, surprise, and disgust—are covered across our system's three stages: face detection, normalisation, and emotion recognition.

Motivation

There has been a lot of interest in using Facial Expression Reading (FER) technology. A system capable of analysing facial expressions was proposed by Tang et al. [4]. The system has five stages: gathering data, finding faces, identifying people, identifying emotions, and processing the results. This method analyses patterns using K-nearest neighbour (KNN) classification and ULBGPHS (Uniform Local Gabor Binary Pattern Histogram Sequence). It was proposed by Savva et al. [1] that a web application be developed to analyse students' emotions while they are actively engaged in face-to-face instruction. Webcams were set up to capture live footage, which was then analysed using machine learning algorithms.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Existing System & Limitations

To enhance the e-learning setting for more effective content delivery, the authors proposed a system that detects and tracks emotional states and provides instantaneous feedback. The system detects a person's emotional state by analysing their eye, nose, mouth and head movements.

Facial emotion recognition system (FERS) was developed by Ayvaz et al. to identify feelings via video. The system employs four different machineearning algorithms (Support Vector Machines, K-Nearest Neighbors, Random Forest, and Classification & Regression Trees), with K-Nearest Neighbors and Support Vector Machines yielding the highest accuracy rates. Based on facial emotion recognition using the Haar Cascades method to identify mouth and eyes on the JAFF database in order to detect emotions, the authors proposed a model that recognises emotion in a virtual environment. An intelligent classroom management system was developed by Chiou et al. using wireless sensor network technology, and it helps teachers quickly switch between different modes of instruction to save time.

Limitations:

- Accuracy is not good [1]
- Works only with images [1]

Proposed System & Merits

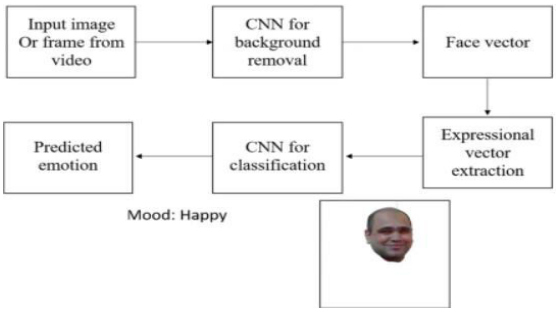
Using a Convolutional Neural Network (CNN) architecture, we describe our proposed system to analyse students' facial expressions. The system first extracts faces from the input image, which are then cropped and normalised to a 48x48 pixel resolution. The facial photos are then fed into a convolutional neural network. After all of this processing, the results of facial expression recognition are output (anger, happiness, sadness, disgust, surprise or neutral).

Advantages:

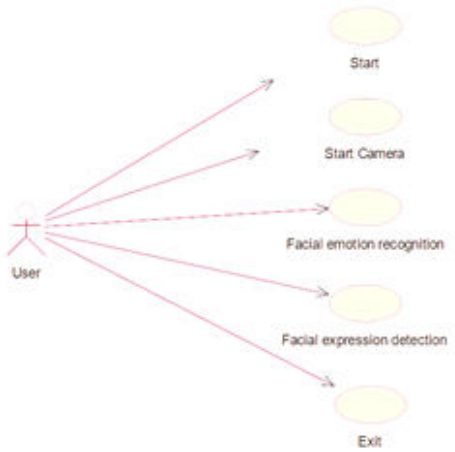
- Works in real time
- Displays emojis with probabilities of emotions

Design

Architecture



Use-Case Diagram



The main purpose of a use case diagram is to indicate what system functions are performed by that actor. Roles of the actors within the system are represented.

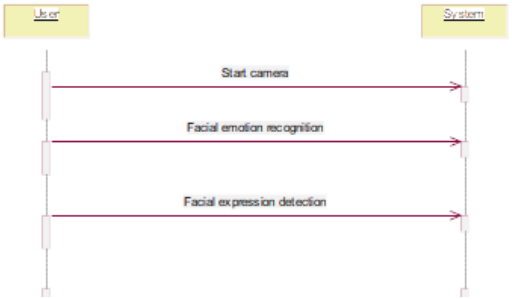
Class Diagram

Describes the structure of a system by showing the system's categories, their attributes, operations (or methods), and also the relationships among the categories.

Sequence Diagram

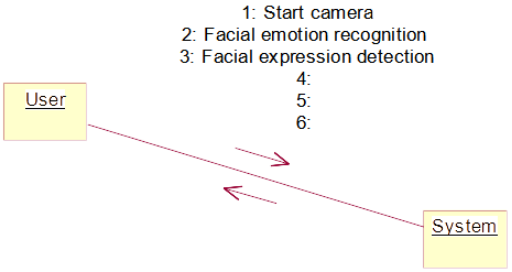
Sequence diagrams are generally known as event diagrams, event situations, and temporal order diagrams.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org



Communication Diagram

A Communication diagram is a diagram that shows interactions between elements at run-time in much the same manner as a sequence diagram.



Modules

Data Collection: Collect sufficient data samples and legitimate software samples.

Data Preprocessing: Data Augmented techniques will be used for better performance.

Train and Test Modelling: Split the data into train and test data Train will be used for training the model and Test data to check the performance.

Modelling: CNN model build and model is saved.

Predict: Select a single image and do basic image processing and predict using CNN model.

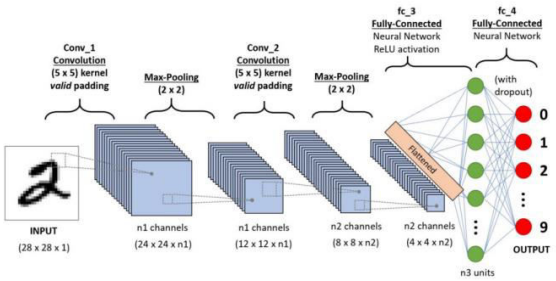
DATASET

The dataset is downloaded from Kaggle. (https://www.kaggle.com/deadskull7/fer2013). Data contains images of human emotions for 7 different classes: Angry, Happy, Neutral, Disgust, Surprise, Sad and Fear. It contains 16900 images and for each class as below:

Emotion	No. of Images
Angry	4593
Happy	8989
Neutral	6198
Disgust	547
Fear	5121
Sad	6077
Surprise	4002

Algorithms

Convolutional Neural Networks



One of the primary categories used in neural networks for image recognition and classification is the convolutional neural network.

Convolutional neural networks are frequently utilised in domains like scene labelling, object detection, face recognition, and other things.

CNN receives a picture as input and categorises and processes it using terms like "dog," "cat," "lion," "tiger," etc. The resolution of the image affects how the computer interprets it as an array of pixels. It will perceive as h * w * d, where h = height, w = width, and d = dimension, depending on the image resolution. For instance, a grayscale image is a matrix array of 4 * 4 * 1, while an RGB image is a matrix array of 6 * 6 * 3.

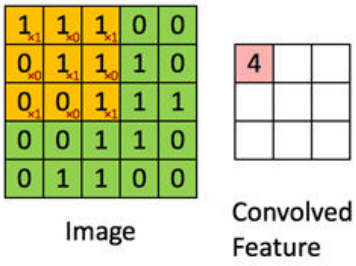
Each input image in CNN is processed by a series of convolutional layers, pooling, fully connected layers, and filters (Also

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

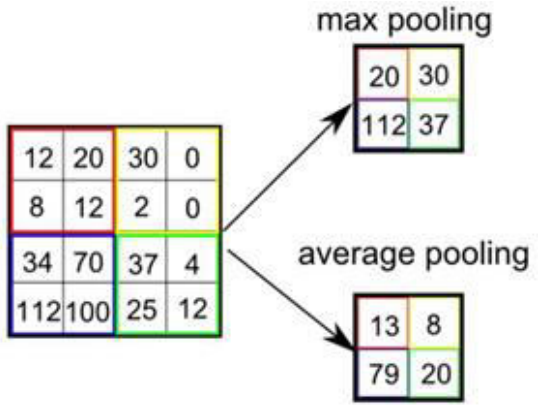
known as kernels). The Soft-max function will then be used to categorise an object using probabilistic values between 0 and 1.

Convolution Layer:

The first layer to extract features from an input image is the convolution layer. The convolutional layer maintains the link between pixels by learning visual properties using a tiny square of input data. Using an image matrix and a kernel or filter as two inputs, it performs a mathematical action.



Pooling Layer: When pre-processing an image, the pooling layer is crucial. When the photos are too big, the pooling layer minimises the number of parameters. The image obtained from the prior layers is "downscaled" in the pooling process. It is comparable to reducing the size of an image to lessen its pixel density.



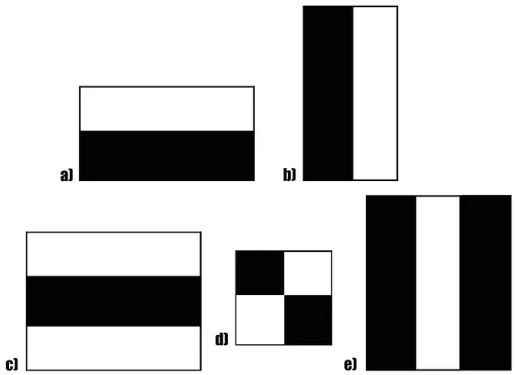
Fully Connected Layer:

The input from the other levels will be flattened into a vector and sent to the completely linked layer, which is a layer. The output will be changed by the network into the desired number of classes.

Haar Cascade Algorithm

It is an Object Detection Algorithm that is used to find faces in still photos or moving videos. The edge or line detection features Viola and Jones suggested in their 2001 study "Rapid Object Detection using a Boosted Cascade of Basic Features" are used by the technique. To train, the algorithm is given a large number of positive photos with faces and a large number of negative images without any faces. The OpenCV GitHub repository at https://github.com/opencv/opencv/tree/master/data/haarcascades has the model produced by this training.

The models are kept in the repository as XML files and can be read using OpenCV techniques. They include algorithms for detecting faces, eyes, upper and lower bodies, licence plates, and more.



CODE REPORT

The System is divided in to two woking modules

model.py

This code implements a convolutional neural network (CNN) for emotion detection on facial images.

The EMR (Emotion Recognition) class contains methods for building the network, predicting emotions from images, and loading a pre-trained model. The CNN architecture used is as follows:

- Input layer: 48x48 grayscale images
- Convolutional layer with 64 filters of size 5x5 and **`ReLU`** activation function
- Max pooling layer with a pool size of 3x3 and stride of 2
- Another convolutional layer with 64 filters of size 5x5 and **`ReLU**` activation function

- Another max pooling layer with a pool size of 3x3 and stride of 2
- A third convolutional layer with 128 filters of size 4x4 and **`ReLU**` activation function
- Dropout layer with a dropout rate of 0.3
- Fully connected layer with 3072 nodes and **`ReLU**` activation function
- Output layer with a **`SoftMax**` activation function and the number of nodes equal to the number of target classes (which are ['angry', 'disgusted', 'fearful', 'happy', 'sad', 'surprised', 'neutral']).

The network is trained using categorical cross-entropy loss and the momentum optimizer.

The accuracy metric is used to evaluate the model during training.

The program is run in **`singleface`** mode. The mode is implemented in separate Python file (`**singleface.py**`), and is selected using command-line arguments. The main function creates an instance of the EMR class and selects the appropriate mode based on the command-line argument.

The `load_model` method checks for the existence of a pre-trained model and loads it if it exists. The predict method takes an image as input, resizes it to 48x48, and returns the predicted emotion probabilities for each target class.

singleface.py

This code is an implementation of a real-time facial expression recognition system using the OpenCV library and a pre-trained Convolutional Neural Network (CNN) model. The system uses a webcam to capture frames of the user's face and then detects the face using the Haar Cascade Classifier algorithm. The detected face is then passed through the CNN model to predict the user's facial expression.

Here is a brief explanation of the main components of the code:

- Import necessary libraries: The first few lines of the code import the necessary libraries, including OpenCV, NumPy, and the pre-trained CNN model.
- Define emotions: The **`EMOTIONS`** list contains the different facial expressions that the model can recognize.
- Define helper functions: The **`format_image`** function is used to preprocess the input image before passing it through the CNN model. It converts the image to grayscale, detects the face using the Haar Cascade Classifier, crops the image to only include the face region, and resizes the image to 48x48 pixels.

- Initialise the CNN model: The **`EMR`** class is defined in the **`model.py`** file, and it contains the CNN model architecture and methods for loading and saving the model weights. The **`network`** object is initialised using the **`EMR`** class and its **`build_network`** method is called to create the CNN model.
- Load emotion images: The **`feelings_faces`** list contains the emoji images corresponding to each emotion in **`EMOTIONS`**.

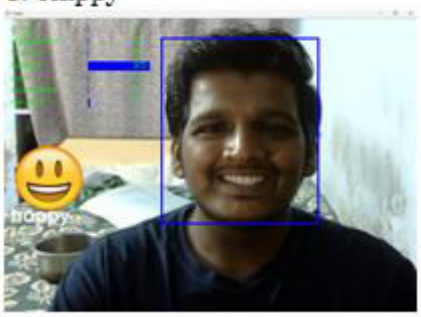
- Capture frames from webcam: The **`cv2.VideoCapture`** function is used to open a connection to the user's webcam and capture frames in real-time.
- Detect faces and predict emotions: The **`cv2.CascadeClassifier`** function is used to detect faces in each frame using the Haar Cascade Classifier. If a face is detected, the **`format_image`** function is called to preprocess the face image, and the preprocessed image is passed through the CNN model using its **`predict`** method to predict the user's facial expression.
- Display emoticons and emojis: The predicted emotion and its corresponding emoji image are overlaid onto the video frame using OpenCV **`cv2.putText`**, **`cv2.rectangle`**, and **`cv2.imread`** functions.
- Display video frame: The final video frame with the predicted emotion and emoji is displayed on the screen using the **`cv2.imshow`** function. The loop continues until the user presses the 'q' key, at which point the program ends and

the webcam connection is released using the **`cap.release()`** function.

Result

Run the single face.py file in the location where all code files exist. It opens the camera and produces the following results.

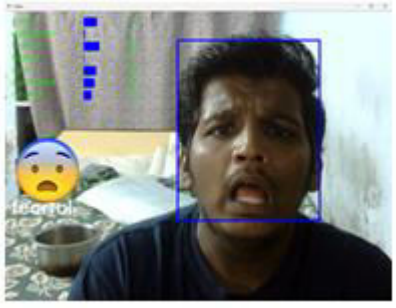
1. Happy



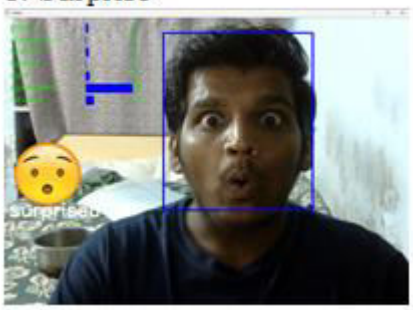
2.Angry

3. Neutral

4. Fearful



5. Surprise



References

[1] Andreas Savva, Vasso Stylianou, Kyriacos Kyriacou, Department of Computer Science, University of Nicosia Nicosia, Cyprus savva.a/stylianou.v/kyriacos.k@unic.ac.cy Florent Domenach, Basic Education, Akita International University Akita, Japan fdomenach@aiu.ac.jp

[2] EDPS TechDispatch on Facial Emotion Recognition, Konstantina VEMOU, Anna HORVATH, Thomas ZERDICK, : techdispatch@edps.europa.eu

[3] Towards Emotionally-Aware AI Smart Classroom: Current Issues and Directions for Engineering and Education Yelin Kim, Member, IEEE, Tolga Soyata, Senior Member, IEEE, Reza Feyzi Behnagh

[4] FACIAL EMOTION DETECTION USING CONVOLUTIONAL NEURAL NETWORKS by Mohammed Adnan Adil Bachelor of Engineering, Osmania University, 2016.

[5] Biometric Recognition, 10th Chinese Conference, CCBR 2015 Tianjin, China, November 13–15, 2015 Proceedings