

**MANIPAL SCHOOL OF INFORMATION SCIENCES**

**(A Constituent unit of MAHE, Manipal)**

**Facial Expression Recognition Using Convolutional Neural Network**

|  |  |  |
| --- | --- | --- |
| **Reg. Number** | **Name** | **Branch** |
| **201046019** | **Swathi M Huttada** | **Big Data and Data Analytics** |
| **201046020** | **Rakshita Raghava Shetty** | **Big Data and Data Analytics** |
| **201046037** | **Bonagiri Sri Pranathi** | **Big Data and Data Analytics** |

**Under the guidance of**

**Dr. Harishchandra Hebbar**

Manipal School of Information Sciences,

MAHE, MANIPAL

**10/12/2020**



**DECLARATION**

We declare that this mini project, submitted for the evaluation of course work of Mini Project to Manipal School of Information Sciences, is extending an existing idea available at (**https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9031283**)), conducted under the supervision of my guide **Dr. Harishchandra Hebbar** and panel members, **Prof. Samarendranath Bhattacharya, Prof. Raghudatesh G P**. References, help and material obtained from other sources have been duly acknowledged.

**ABSTRACT**

Emojis or avatars are ways to indicate nonverbal cues. These cues have become an important part of online chatting, product review, brand emotion, and many more. Human emotions and intentions are expressed through facial expressions, which play a communicative role in interpersonal relations.

The objective is to implement Convolutional Neural Networks for classification of facial expressions. Facial images are classified into seven facial expression categories namely Happy, Sad, Surprise, Anger, Fear, Disgust, and Neutral. Kaggle dataset is used to train and test the classifier.

**Keywords**

Facial Expression Recognition, Convolutional Neural Network

**Contents**

[1.](#_2et92p0) Introduction 1

[2.](#_1t3h5sf) Material and methods 2

[3.](#_4d34og8) Results 5

[4.](#_2s8eyo1) Discussion 5

[5.](#_17dp8vu) Conclusions 5

[6.](#_3rdcrjn) Scope for further work 5

[7. References](#_26in1rg)  6

**LIST OF FIGURES**

Figure A: Sample images from Kaggle Dataset 2

Figure 1(a): Training Phase 3

Figure 1(b): Testing Phase 3

Figure 1(c): General Architecture of CNN 4

**ABBREVIATIONS**

|  |  |
| --- | --- |
| CNN | Convolutional Neural Network |
| FER | Facial Expression Recognition |
| ReLu | Rectified Linear Units |

# **Introduction**

Being able to recognize facial expressions is key to nonverbal communication between humans, and the production, perception, and interpretation of facial expressions have been

widely studied. Due to the important role of facial expressions in human interaction, the ability to perform Facial Expression Recognition (FER) automatically via computer vision enables a range of novel applications in fields such as human-computer interaction and data analytics .Consequently, FER has been widely studied and significant progress has been made in this field.

The objective is to implement Convolutional Neural Networks (CNNs) for classification of facial expressions. It is mainly used for feature extraction and inference.Facial images are classified into seven facial expression categories namely Happy, Sad, Surprise, Anger, Fear, Disgust, and Neutral. Kaggle dataset is used to train and test the classifier.

# 

# **Material and methods**

**Implementation**

1. Install dependencies.

Tensorflow, Tflearn, Numpy, Keras, opencv 3,scipy, os, pandas, skimage, Anaconda Environment.

2. Download and prepare the data.

The dataset from a Kaggle Facial Expression Recognition Challenge (FER2013) is used for the training and testing. It comprises pre-cropped, 48-by-48-pixel grayscale images of faces each labelled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. Dataset has a training set of 35887 facial images with facial expression labels. The dataset has class imbalance issues, since some classes have a large number of examples while some have few. The dataset is balanced using oversampling, by increasing numbers in minority classes.



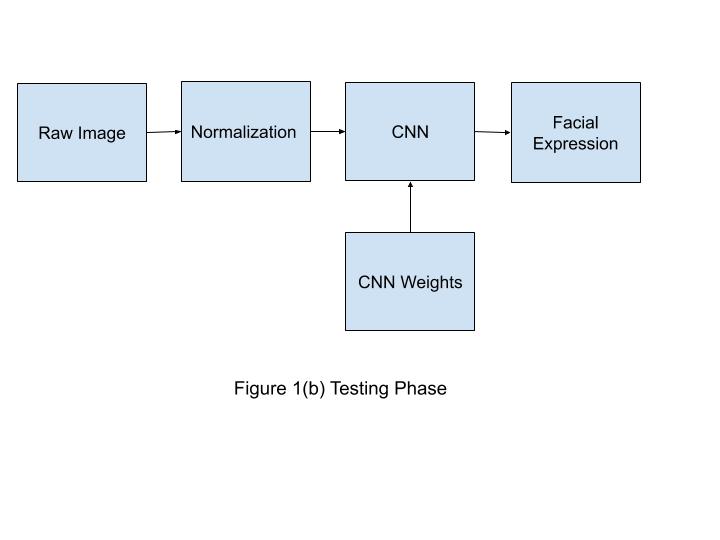
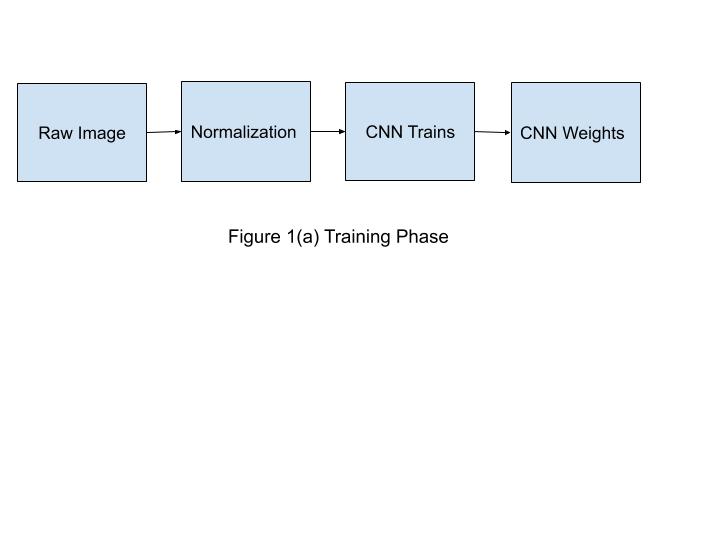
Fig A :Example images from the FER2013. Images in the same column depict identical expressions, namely anger, disgust, fear, happiness, sadness, surprise, as well as neutral.

3. Train the data.

We use many pre-trained models to train the data. This will be implemented by next phase of this project

4. Evaluate a trained model.

5. Recognizing facial expressions from an image file.

****

**General Architecture of CNN**

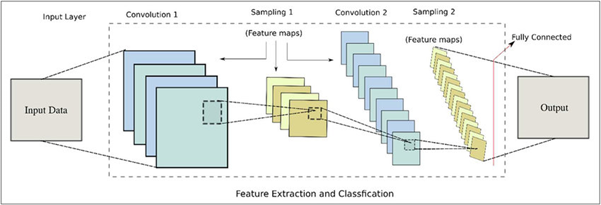
****

Fig 1(c):General Architecture of CNN

Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way. In particular, unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. (Note that the word depth here refers to the third dimension of an activation volume, not to the depth of a full Neural Network, which can refer to the total number of layers in a network.) The neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner.

The Different layers of a CNN:

There are four types of layers for a convolutional neural network: the convolutional layer, the pooling layer, the ReLU correction layer and the fully connected layer.

* The convolutional layer - The Convolutional layer is the core building block of a Convolutional Network that does most of the computational heavy lifting. Its purpose is to detect the presence of a set of features in the images received as input.
* The pooling layer - The pooling operation consists in reducing the size of the images while preserving their important characteristics.
* The ReLU correction layer - ReLU (Rectified Linear Units) refers to the real non-linear function defined by ReLU(x)=max(0,x).
* The fully connected layer - This type of layer receives an input vector and produces a new output vector.

# 

# 

**Results**

We have downloaded and pre-processed the Fer2013 dataset. We have split our data into training and testing data subsets and started to process the training set consisting of several images with various facial expressions. We have to start training the data using various pre- trained models. We plan to train the data several more times and test to seek accuracy.

# **Discussion**

We create our Anaconda Environment and install python, Tensorflow and OpenCV and related dependencies (modules related to computer vision) and import our dataset and process it. We split our available dataset into two halves, one for training set and other for evaluation/test dataset to prove if our algorithm works properly. We use pre-trained models to further start to train our data accordingly to the specific parameters and evaluate them.

We desire to obtain the facial expression of the person in the image scene using CNN learning algorithms

# **Conclusions**

Fer2013 dataset is downloaded from Kaggle. Dataset is pre-processed and split into train dataset and test dataset. Training dataset is used to train the program by giving the desired output whereas test dataset is used for testing, to check whether the program gives desired output.

# **Scope for further work**

As the data is pre-processed, we will train the program with few pre-trained models in further phases.

# 

# 

# **References**

[1] Shan, C., Gong, S., & McOwan, P. W. (2005, September). Robust facial expression recognition using local binary patterns. In Image Processing, 2005. ICIP 2005. IEEE International Conference on (Vol. 2, pp. II-370). IEEE.

[2] Chibelushi, C. C., & Bourel, F. (2003). Facial expression recognition: A brief tutorial overview. CVonline: On-Line Compendium of Computer Vision, 9.

[3] "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.

[4] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[5][Fer2013] Dataset from Kaggle [[https://www.kaggle.com/c/challenges-in-representation- learning-facial-expression-](https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-) recognition-challenge/data]

[6] <https://en.wikipedia.org/wiki/Convolutional_neural_network>

[7] <https://en.wikipedia.org/wiki/Deep_learning>