iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.34863

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Emotion Detection using Deep Learning (Classification of FER 2013)

Tridib Chakraborty[1], Niladri Sen[2], Puja Mukherjee[3], Sandipan Kar[4], Sohom Chowdhury[5], Ritwik Kumar Naskar[6], Anushka Debnath[7]
[1]*Assistant Professor,* [2, 3, 4, 5, 6, 7]*Student, Department of Information Technology, Guru Nanak Institute of Technology, Kolkata-700114*

Abstract: ***With the rise of Deep Learning Technique, the accuracy of classification problems have improved in a drastic manner. In this project we have classified images of emotions belonging to 7 different classes (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral). The main objective of this project is getting a high accuracy.***
Keywords: Deep Learning, Machine Learning, Neural Network, Convolutional Neural Network (CNN)

I. INTRODUCTION

Deep Learning is a sub-domain of Machine Learning. It is a concept of Neural Network, it is also known as Deep Neural Network because it has deep layers or multiple layers of neurons connected with each other. The main purpose of classification of FER 2013 is to determine the emotions of a person by training the deep learning model

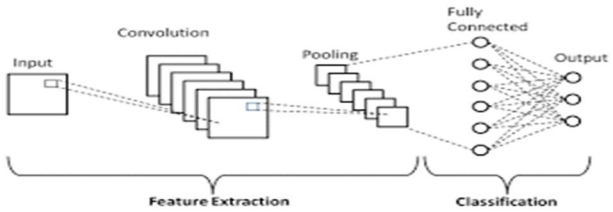
As we are dealing with images we have used Convolutional Neural Network (CNN), an algorithm of Deep Learning which works with images, for classification of the images. In this model we have been able to get an accuracy of 66.25% and training accuracy of 87.9%. We have used *keras* API of python for purpose of making the model and implementing it.

II. CONVOLUTIONAL NEURAL NETWORK (CNN)-

It is a class of deep learning used to used for analyzing images and visual data. It is a version of multilayer perceptron.
It is also known as ConvNet. A CNN layer consists of an input layer, an output layer and multiple layers in between.

III. MODEL ARCHITECTURE

The basic structure of any CNN model is Convolutional layer then Flatten layer and then Dense layer, which is a Fully connected neural network.



A. Convolutional Layer

The first layer of the deep learning model is *Conv2D* layer which takes in the data, it is also considered to be the input layer, as parameters it has an input shape, an activation function, in our case it is *ReLu*, it has a kernel size which is the size of kernel, a matrix used to extract the main features of an image by matrix multiplication of the kernel matrix and the pixel matrix of the image, the first layer has 32 filters which means the image pixels has to go through 32 layers of perceptrons. The numbers of filters increases with the increase of *Conv2D* layers, the standard increment procedure is doubling it. Then comes Pooling layer, there are many kinds of pooling layers, the main purpose of pooling layer is to pool the major features of the outcome of the convolutional layer (*Conv2D* layer). In our case we have used *MaxPooling* layer which has pool_size as parameter, pool_size means the layers

Normalization of the data is needed to enable the model generalize the image about to be classified. In our case we have used *BatchNormalization* which normalizes data row-wise.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

B. Regularization

It is a process to overcome Overfitting, a scenario in which the difference between the accuracy on training dataset and testing dataset differs drastically which happens when the model cannot generalize the data it has been trained on. For that we have used Dropout and regularizers.

C. Flatten Layer

This layer transforms the multidimensional matrix into a single-dimensional vector as the Dense layer or fully connected neural network cannot work on multidimensional data.

D. Dense Layer

This forms the fully connected neural network which evaluates the single dimensional vector and gives output. It takes units as parameter which means number of perceptron in a single layer.

IV. DATASET (FER 2013)

FER 2013 is an open source free dataset found in kaggle.com which 28,709 training images and 3,589 testing images. All the images are 48x48 pixels and are in grayscale. The images are shows emotions of 7 kinds (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral)

A. Data Augmentation

Deep learning is a concept which needs large number of data to train the model which would help to give a high accuracy. But what to do when less number of data is available? The solution to this problem is *Data Augmentation.* It is the process in which the existing data is increased in number by reshaping them. In our project we have used *ImageDataGenerator* for this purpose

V. MODEL COMPILATION, TRAINING AND EVALUTION-

For compilation of the model the optimizer used was *Adam* and the learning rate was 0.0001 the loss function used was *categorical_crossentropy* which is generally used for a dataset having more than 2 classes.

For the purpose of training data we had to model,fit() function and the number of epochs was decided to be 80. The validation of the data and the model's evaluation was done on validation dataset.

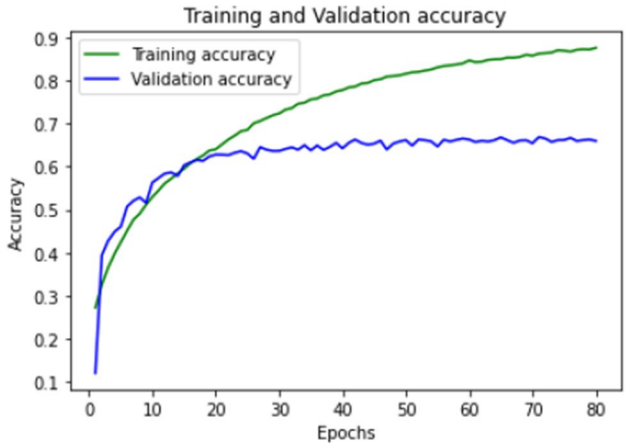
```
Epoch 70/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4680 - accuracy:
0.8589 - val_loss: 1.3156 - val_accuracy: 0.6540
Epoch 71/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4697 - accuracy:
0.8614 - val_loss: 1.3124 - val_accuracy: 0.6685
Epoch 72/80
448/448 [==============================] - 33s 74ms/step - loss: 0.4648 - accuracy:
0.8640 - val_loss: 1.3534 - val_accuracy: 0.6652
Epoch 73/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4644 - accuracy:
0.8649 - val_loss: 1.4425 - val_accuracy: 0.6569
Epoch 74/80
448/448 [==============================] - 33s 75ms/step - loss: 0.4454 - accuracy:
0.8696 - val_loss: 1.3886 - val_accuracy: 0.6613
Epoch 75/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4456 - accuracy:
0.8717 - val_loss: 1.4244 - val_accuracy: 0.6616
Epoch 76/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4563 - accuracy:
0.8651 - val_loss: 1.4273 - val_accuracy: 0.6666
Epoch 77/80
448/448 [==============================] - 33s 74ms/step - loss: 0.4428 - accuracy:
0.8713 - val_loss: 1.3384 - val_accuracy: 0.6592
Epoch 78/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4357 - accuracy:
0.8745 - val_loss: 1.4272 - val_accuracy: 0.6614
Epoch 79/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4385 - accuracy:
0.8740 - val_loss: 1.3368 - val_accuracy: 0.6625
Epoch 80/80
448/448 [==============================] - 34s 75ms/step - loss: 0.4253 - accuracy:
0.8790 - val_loss: 1.3683 - val_accuracy: 0.6595
```

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

VI. TRAINING ACCURACY VS VALIDATION ACCURACY

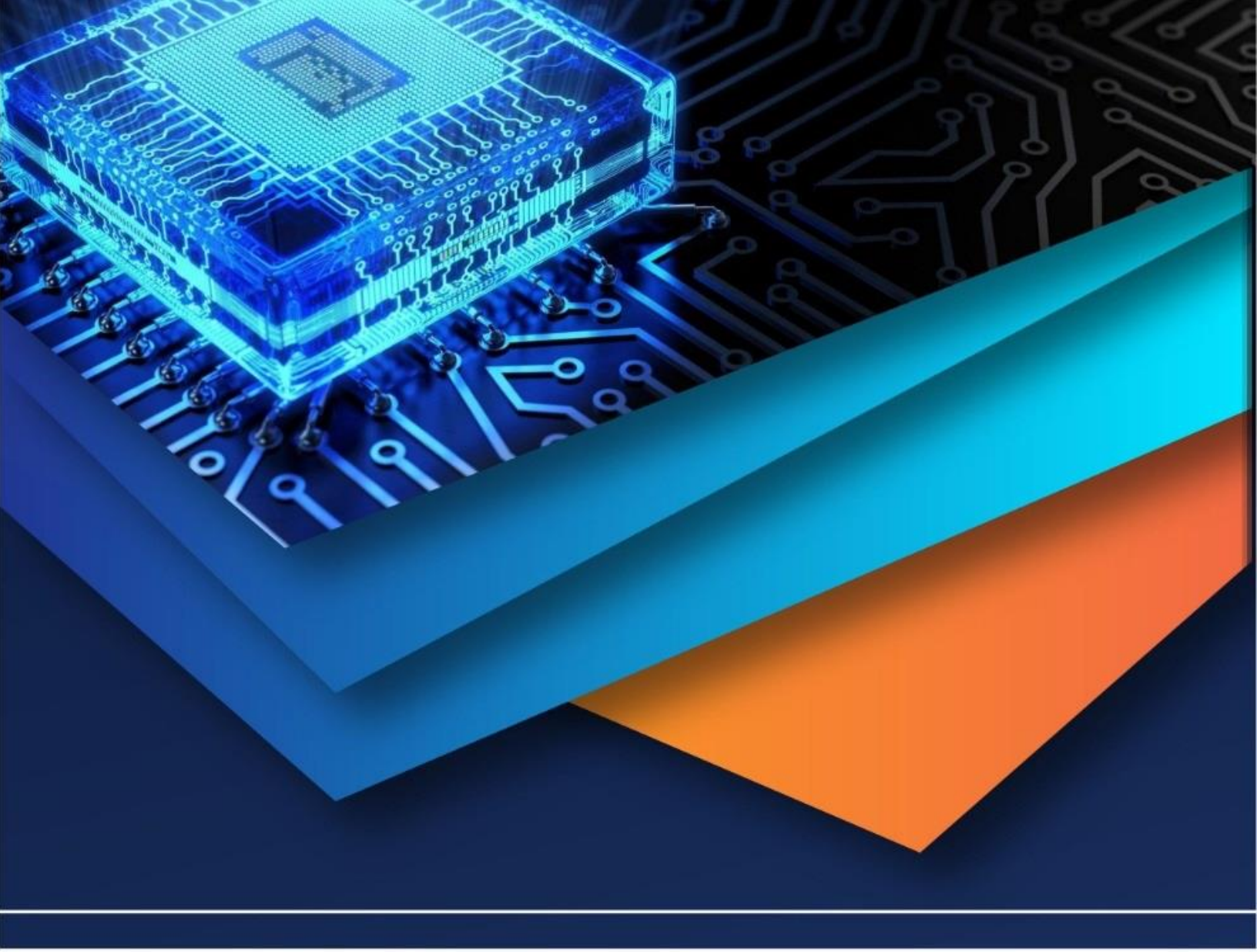
The performance of the model on the two different accuracy grounds is graphically represented as follows-



REFERENCES

[1] The 2nd International Workshop on the Future of Internet of Everything (FIoE) August 9-12, 2020, Leuven, Belgium Facial emotion recognition using deep learning: review and insights Wafa Mellouka*, Wahida Handouzia
[2] Overfitting Mechanism and Avoidance in Deep Neural Networks Shaeke Salman and Xiuwen Liu
[3] Emotion Recognition from Facial Expression using Deep Learning Nithya Roopa. S
[4] www.google.com

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)