International Journal of Research in Engineering, Science and Management
Volume 4, Issue 11, November 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Real Time Facial Emotions Detection Using Convolutional Neural Network

Battula Purna Praveen[1*], Asritha Pidikiti[2], Veda Gayatri[3], Aparna Gurram[4]
[1,2,3,4]*Student, Department of Computer Science and Engineering, SRM University, Mangalagiri, India*

***Abstract*: Facial emotions or expressions are recognized by computers and enhancing modern-day machines to understand human emotions from their reality time. Through this project, i'd wish to provide solution for real face expressions or emotions by video capture from emotion detector frame by Open-CV it'll capture video by camera which is built-in to the machine or ADPS. The countenance are identified by different operations provided by OpenCV and also the region consisting of parts of the face are made to surround or enclose by a contour. This region, enclosed by the contour is used as an input to the Convolutional Neural Network (CNN). The CNN model created consists of six activation layers, of which four are convolution layers and two are fully controlled layers. The scope of the project is to demonstrate the accuracy and validation of Convolution Neural Network (CNN).**

***Keywords*: Face detection, feature extraction, emotion classification, CNN.**

1. Introduction

Human beings will communicate with one another by speech, actions and expressions (or) emotions. Facial expression recognition has its branches spread across various applications like computer games, webinar technologies, online surveys and plenty of other fields. While there are several advantages that have been witnessed in this field, but there are several di-advantages that exist. The traditional features extraction methods show very low response and lack in performance. For these traditional features extraction, it's very difficult to extract the desired features and it's hazardous to us in real world. These emotions are often detected by machine or computer with regard to computing and etc. In this project we are deducting expressions by Convolution neural networks(CNN). The facial expressions recognition is finished by keras. The CNN will have accustomed train the model. The trained model is deployed to an emotion detector by tensor flow frame. This developed model is employed for real time faces, images and videos and its accuracy and validation is additionally analyzed.

2. Dataset

In this project, the database accustomed train the mode is taken from the keggal website it's facial expression recognition dataset developed by author Jonathan Oheix. In this dataset it's two directories one is train and second is validation. In test directory it's 7 directories they're happy, sad, fear, anger, surprise, disgust, neutral. And in validation directory it's 7 directories they're happy, sad, fear, anger, surprise, disgust, neutral. In this data set the pictures are grayscale and its dimensions of 48*48 pixels. This data set was created by gathering from google images to look for emotions. Every image of every emotion type is returned by the function OS module in python. The number of images in two directories with each emotion will display below and sample images are attached below. Our aim is to style CNN models with better accuracy and validation.

Table 1

S. No.	Type of Emotion	No. of images in the Train dataset	No. of images in the Validation dataset
1.	Angry	3993	960
2.	Disgust	436	111
3.	Fear	4103	1018
4.	Happy	7164	1825
5.	Neutral	4982	1216
6.	Sad	4938	1139
7.	Surprise	3205	797

3. Data Prepossessing

A. Preprocessing the data

In this we will give Batch Size it will tell how many training examples should take our model for one iteration. We will give two variables one is datagen_train and second one is datagen_val. Both are defining ImageDataGenerator. After this we are defining two variables one is train_set and second is test-set. In train-set it will directory of datagen_train. In datagen_train it will contain folder path and train. The images in folder path will get trained and with specific size, colour, class_mode, shuffle.same in test_set also it will contain directory of datagen_val. In datagen-val it contains folder_path and train. Then the images which are in folder path will be trained with specific size, colour, class mode, shuffle. Then it will found '7' classes in both datagen_train and datagen_val.

B. Model Building

Here we are going to use a Convolutional neural network to acknowledge different expressions. Convolutional neural network, here we'll define the quantity of classes as 7 and outputs are going to be 7 expressions and also the model we are defining in sequential order. We are defining seven layers

*Corresponding author: battula_purna@srmap.edu.in

which are a part of CNN. These all 7 layers will make a man-made neural network or deep neural network. In those four are convolutional layers and three are fully connected layers. In each and every layer we will define Conv2d filter and kernel size, padding, input_shape, Batch Normalization layer, Activation layer (we will take liner), Maxpooling 2D (we must always define size then it'll extract important information from the purpose where we kept size), Dropout (to forestall our model to induce overfitted). The input_shape should be defined in 1st layer no have to be define in every layer. We will define these layers in every CNN layer expect input_shape. Then we'll crete flatten layer. The function of flatten layer is to collapse the input size to 1-dimension array which easily fed into the system. Then we are going to fully connected to the first layer, 2nd layer and 3rd proposed layer with the assistance of "Dense". Then we'll define adam optimizer (learning rate=0.001) next we'll define model compiler in this optimizer will fed into it with categorical cross entropy and that we will define metrics with accuracy. Finally, we'll we will print model summary.

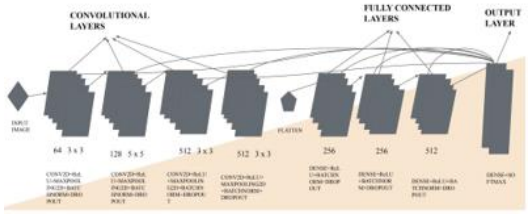

Fig. 1. Proposed CNN's architecture

C. Model Training

For training and validation, we should always import the Model Checkpoint, early Stopping, Reduce R On Plateau from keras. Now checkpoint (it will check every point in our model) it'll save your model. we've saved the model in our output keggle within the type of model.h5, monitor will monitor my validation accuracy, mode are maximum, verbose are going to be one. Now Earlystopping (if my model isn't increasing accuracy then epoch is continuous then we are going to use early stopping) it'll monitor the validation accuracy and that we will use some parameters like min_delta, patience.verbose.restore_best_weight (it will restore the simplest weights or best model).Then reduce_learningrate (if my model isn't cope up with certain learning rate then it'll reduce it) during this we'll use parameters like factor, patience, verbose, min_delta. Then we'll define call back list it'll contain checkpoint, reduce learning rate, early stopping and that we will define epochs. Then we are going to fit my model with training set and test set from training and validation data. Then we'll consider generator as training set,steps_per_epoch, validation data are as test_set, validation_steps and call back list.

D. Opencv Model Building

Import all wanted files from keras model class like load model(to load model and haar cascade files),preprocessor image to array, preprocessor load image from kerras, cv2, numpy. Then define two classifiers one for haar cascade and another one for model.h5 load them by folder url. After define emotion label like angry, sad, happy, fear, surprise, disgust, neutral. After define cap for video capture from machine or computer. Then define frame which might read by opencv.convert frame into gray scale and every one the photographs are gray. Faces variable contains all faces and returns in four parameters for every face it'll draw an oblong box round the face. The region interest is face only; image size are 48*48 pixels because my model is trained on 48*48-pixel size only. Then the image will convert into an array within the region of interest. By using classifiers it'll predict the face emotion by emotion labels. If it doesn't find any faces it'll appear NO faces. Then it'll show Emotion detector frame and if we press "q" it'll destroy all.

4. Results

Fig. 2. Surprise

Fig. 3. Happy

Fig. 4. Sad

Fig. 5. Neutral

Fig. 6. Angry

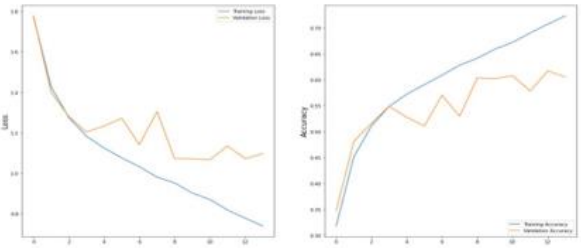

Fig. 7. Graphs

Plotting for training and validation loss, training and validation accuracy.

Before adding 3rd fully connected layer an accuracy in between 68% to 69%. After adding the 3rd layer the accuracy improved to 72%.

5. Conclusion

Using the kaggle facial features data set, a test accuracy is 72% is attained with this designed proposed CNN model. The achieved results are satisfactory because the average accuracies and so, this CNN model is accurate. For an improvement during this model and its outcome, it's recommended to alter the parameters wherever useful in CNN model and removing unwanted parameters. Resize the training rate and adapting with within the location may helpful to enhance the accuracy and model. The number of epochs will be set to higher number to attaining the accuracy as output. But by increasing the number of epoch may result in overfitting. This similar CNN model are often wont to different datasets to be trained and tested and check for its accuracy.

References

[1] E. Sariyanidi, H. Gunes, et A. Cavallaro, “Automatic Analysis of Facial Affect: A Survey of Registration, Representation, and Recognition,” IEEE Trans. Pattern Anal. Mach. Intell., Oct. 2014

[2] M. Lyons, M. Kamachi, et J. Gyoba, “The Japanese Female Facial Expression (JAFFE) Database,” Zenodo, 1998.

[3] Sajid M, Ali N, Dar SH, Iqbal Ratyal N, Butt AR, Zafar B, Shafique T, Baig MJA, Riaz I, Baig S (2018) Data augmentation-assisted makeup-invariant face recognition. Math Probl. Eng. 2018:1–10.

[4] Cowie, Roddy, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G. Taylor, “Emotion recognition in human-computer interaction,” IEEE Signal processing magazine 18, no. 1: 32-80, 2001.

[5] Giannopoulos, Panagiotis, Isidoros Perikos, and Ioannis Hatzilygeroudis. “Deep Learning Approaches for Facial Emotion Recognition: A Case Study on FER-2013,” Advances in Hybridization of Intelligent Methods. Springer, Cham, 1-16, 2018.