Biosc.Biotech.Res.Comm. Special Issue Vol 14 No 05 (2021) Pp-47-50

BBRC
Bioscience Biotechnology Research Communications

Facial Expression Recognition Using CNN with Keras

Apeksha Khopkar[1] and Ashish Adholiya Saxena[2]
[1]*DES's Navinchandra Mehta Institute of Technology and Development & Research Scholar Pacific University, Udaipur, India*
[2]*Pacific Institute of Management, Udaipur, India*

ABSTRACT

Facial Expressions are an essential feature of non- verbal communication, as we move towards the digitization the human computer interactions plays a vital role. The emotional changes results change in the expressions. This paper elaborates development of Deep Convolutional Neural Network Model using tf.keras for building and training Deep Learning Model. The aim is to classify facial image into one of the seven face detection classifiers using open CV and one of its classifiers for drawing the boundary box around the face to detect the correct expression. For training the CNN models we have used 48x48 grey–scale images from Kaggle's ICMP 2013-Fecial Expression Recognition (FER) dataset The FER dataset is divided into two folders called test and train, further divided into separate folder each containing one of the seven types of FER dataset. To understand the spread of the distribution of the class data augmentation method is used to generate minority classes. To reduce over fitting of the models, dropout and batch normalization is used. We are using atom optimizer and softmax activation function as it is a multiclass classification problem. It is a categorical cross entropy and matrix that we are training this for accuracy based on the parameters to evaluate the performance of the developed CNN model by looking at the training epoch history.

KEY WORDS: FACIAL EXPRESSION RECOGNITION (FER), CONVOLUTIONAL NEURAL NETWORK (CNN), DEEP LEARNING.

INTRODUCTION

Facial Expression Recognition (FER) can be seen as a second step to face detection mechanism. Humans can express emotions through facial expressions which is a part of nonverbal communication. When a machine communicates with people, FER can give more affinities and personalized service to people depending on their emotions, which eventually increase the confidence and trust in people. Various ways in which we can express the emotions, such as facial expressions, voices, physiological signals, and text('1.https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data'). Machines capture the expressions through camera, videos. Facial Expressions can be classified as: surprise, happy, neutral, angry, sad, disgust, and fear.

For same emotions, expressions of same or different people may vary because emotions are highly situation dependent (K. M. Rajesh and M. Naveenkumar, 2016) (Padgett and Cottrell, 1996). This model focuses only on the facial area specifically around mouth and eyes. Over the period various techniques are used for facial expression recognition: Bayesian Networks, Neural Networks and the multilevel Hidden Markov Model (HMM) (Cohen, Ira and et al., 2003).

This paper build and train CNN in Keras to recognize facial expressions. For training the CNN models we have used 48x48 grey–scale images from Kaggle's ICMP 2013-

Biosc Biotech Res Comm P-ISSN: 0974-6455 E-ISSN: 2321-4007

Identifiers and Pagination
Year: 2021 Vol: 14 No (5) Special Issue
Pages: 47-50

Article Information
Received: 13th Jan 2021
Accepted after revision: 25th Mar 2021

DOI: *http://dx.doi.org/10.21786/bbrc/14.5/10*

FER dataset ('https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data').

We will accomplish it by completing following tasks:
Task 1: Explore the Dataset
Task 2: Generate Training and Validation Batches
Task 3: Create a CNN Model
Task 4: Train and Evaluate Model
We focus on the face detection technique and expression detection based on the extracted features.

Literature Review: In recent years, researchers have made considerable progress in developing automatic expression classifiers(Shima Alizadeh and Azar Fazel, no date). In 19th century, Charles Darwin has contributed important aspect in facial expression analysis which is directly linked with current science of automatic facial expression recognition. Charles Darwin's worked on the general principals of expressions in humans and animals (SRIMANI P.K. and HEGDE R., 2012). In his work he has grouped several kinds of expressions in groups like: shame, shyness, modesty, anxiety, grief, low spirits, dejection, despair joy, tender feelings, high spirits, love, devotion, reflection, meditation, ill-temper, anger disdain, contempt, sulkiness, determination, hatred, disgust, astonishment, fear, guilt, pride, surprise, horror, self-attention. The observation of the study says the enlargement of the muscles around eyes and mouth varies as per the changes in the emotions e.g.: depression, surprise, happiness, etc. (Darwin C., no date).

Various range of CNN, modelled and trained for facial emotion recognition are evaluated in (A. M. Badshaah, J. Ahmed and S. W. Baek, 2017). Facial emotion Recognition is drawing its own importance in the research field. Facial emotion recognition is inspected and analyzed on all research areas (A. Routrey, M. Swaen and P Kabisetpathy, 2018). Emotion is recognized from facial images using filter banks and Deep CNN (K.-Y. Hueng et al., 2016), this leads to high accuracy rate which implies that deep learning can also be used for facial expression detection.

Two important objectives of Facial Expression Recognition and Analysis (FERA) are feature extraction and expression classification. Ming et. al. defined three phases of FER: facial image registration, feature extraction, and FER (Nithya Roopa. S, 2019). The facial expression recognition has two methods: classification and regression methods. Deep learning is showing promising results, including in FER (Nithya Roopa. S, 2019). This paper contributes a Deep Learning and CNN approaches with multiclass classification problem so it is categorical cross entropy loss function to track highest accuracy.

Convolutional Neural Networks (CNN) are neural network architecture which has multilayers (D Y Liliana, 2018). CNN input and output are array vectors known as feature map. The type of input decides the array dimension value. As an example, audio input has one dimensional array as well as text input; image has 2D array. The output feature map describes the feature extracted from the input. CNN has three key layers: convolutional filter layer, pooling/subsampling layer, and classification layer (Nithya Roopa. S, 2019).

RESEARCH METHODOLOGY

Facial Expression and Emotions: According to the basic definition of emotion by Ekman and Friesen, emotions are divided into six classes, namely happy, sad, surprise, fear, disgust, angry [18]. In this paper talks about seven face detection classifiers using open CV and one of its classifiers for drawing the boundary box around the face to detect the correct expression. For training the CNN models we have used 48x48 grey–scale images from Kaggle's ICMP 2013-Fecial Expression Recognition (FER) dataset (https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data'). The FER dataset is divided into two folders called test and train, further divided into separate folder each containing one of the seven types of FER dataset.

Figure 1: Importing the library

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import utils
import os
%matplotlib inline

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense, Input, Dropout,Flatten, Conv2D
from tensorflow.keras.layers import BatchNormalization, Activation, MaxPooling2D
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from tensorflow.keras.utils import plot_model

from IPython.display import SVG, Image
from livelossplot import PlotLossesKerasTF
import tensorflow as tf
print("Tensorflow version:", tf.__version__)
```

Figure 2: Plotting the sample images

```
utils.datasets.fer.plot_example_images(plt).show()
```

Figure 3: Train folder each containing one of the seven types of FER dataset

```
for expression in os.listdir("train/"):
    print(str(len(os.listdir("train/" + expression))) + " " + expression + " images")
```

```
3171 surprise images
7214 happy images
4965 neutral images
3995 angry images
4830 sad images
436 disgust images
4097 fear images
```

To build the training model, training and validation batches are generated with the FER dataset image size 48x48 and batch size as 64 as per the memory size of CPU/GPU to speed up the training process. ImageDataGenerator() class is used to accept values or images to treat camera captured image as a horizontal mirror image. These images are used to generate the training set. Test and training sets are generated by keeping the various parameters same.

Figure 4: Generating training and validation batches

```
img_size = 48
batch_size = 64

datagen_train = ImageDataGenerator(horizontal_flip=True)

train_generator = datagen_train.flow_from_directory("train/",
                                                    target_size=(img_size,img_size),
                                                    color_mode="grayscale",
                                                    batch_size=batch_size,
                                                    class_mode='categorical',
                                                    shuffle=True)

datagen_validation = ImageDataGenerator(horizontal_flip=True)
validation_generator = datagen_validation.flow_from_directory("test/",
                                                    target_size=(img_size,img_size),
                                                    color_mode="grayscale",
                                                    batch_size=batch_size,
                                                    class_mode='categorical',
                                                    shuffle=False)
Found 28708 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
```

Figure 5: CNN Architecture (Revanth Krishna)

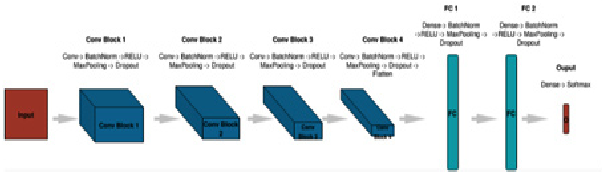


Figure 6: Four Convolution Layer

```
# 1 - Convolution
model.add(Conv2D(64,(3,3), padding='same', input_shape=(48, 48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
# 2nd Convolution layer
model.add(Conv2D(128,(5,5), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
# 3rd Convolution layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
# 4th Convolution layer
model.add(Conv2D(512,(3,3), padding='same'))
```

Figure 7: Two controlled layers

```
# Fully connected layer 1st layer
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

# Fully connected layer 2nd layer
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(7, activation='softmax'))

opt = Adam(lr=0.0005)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Designing the CNN Model: By following the above CNN architecture six activation layers are designed. Four convolution layer and 2 fully controlled layers.

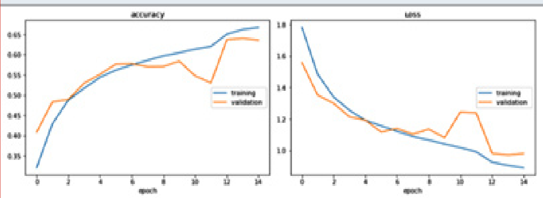
The ReLu function is used to increase the non-linearity in the images, maxpooling is used for dimensions reduction of the image, dropout function to avoid over fitting of the training data. Flatten to convert image to 1- dimensional array. 1- dimensional array becomes the input to the fully controlled layers. Output layer has two techniques dense and softmax.

To study the accuracy of the model, the metrics is set to accuracy. The loss is set to categorical_crossentropy as the data has to be classified into only the 7 defined categories and each image can belong to one classification type only.

Table 1. Structure of CNN Model and parameters

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	640
batch_normalization (BatchNo	(None, 48, 48, 64)	256
activation (Activation)	(None, 48, 48, 64)	0
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropout (Dropout)	(None, 24, 24, 64)	0
conv2d_1 (Conv2D)	(None, 24, 24, 128)	204928
batch_normalization_1 (Batch	(None, 24, 24, 128)	512
activation_1 (Activation)	(None, 24, 24, 128)	0
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 128)	0
dropout_1 (Dropout)	(None, 12, 12, 128)	0
conv2d_2 (Conv2D)	(None, 12, 12, 512)	590336
batch_normalization_2 (Batch	(None, 12, 12, 512)	2048
activation_2 (Activation)	(None, 12, 12, 512)	0
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 512)	0
dropout_2 (Dropout)	(None, 6, 6, 512)	0
conv2d_3 (Conv2D)	(None, 6, 6, 512)	2359808
batch_normalization_3 (Batch	(None, 6, 6, 512)	2048
activation_3 (Activation)	(None, 6, 6, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 3, 3, 512)	0
dropout_3 (Dropout)	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1179904
batch_normalization_4 (Batch	(None, 256)	1024
activation_4 (Activation)	(None, 256)	0
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
batch_normalization_5 (Batch	(None, 512)	2048
activation_5 (Activation)	(None, 512)	0
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 7)	3591
Total params: 4,478,727		
Trainable params: 4,474,759		
Non-trainable params: 3,968		

Figure 8: Accuracy for 15 epochs




```
accuracy
        training       (min:    0.320, max:    0.667, cur:    0.667)
        validation     (min:    0.408, max:    0.640, cur:    0.635)
Loss
        training       (min:    0.890, max:    1.781, cur:    0.890)
        validation     (min:    0.972, max:    1.557, cur:    0.981)

Epoch 00015: saving model to model_weights.h5
448/448 [==============================] - 26s 58ms/step - loss: 0.8904 - accuracy: 0.6672 - val_loss: 0.9813 - val_accuracy:
0.6353 - lr: 5.0000e-05
CPU times: user 6min 34s, sys: 56.7 s, total: 7min 30s
Wall time: 6min 51s
```

RESULTS AND DISCUSSION

CNN model is set to 15 epochs when trained gives accuracy 66.7%

It is been observed that the designed CNN model can flawlessly detect facial expressions such as: Happy, Sad, Surprise subject to 48x48 gray scale images only.

Figure 9: Kathak Facial Expressions Recognition (Navras)

CONCLUSION AND FUTURE SCOPE

Using 48x48 grey–scale images from Kaggle's ICMP 2013-Fecial Expression Recognition (FER) dataset, test accuracy is 66.7% with the above designed CNN model. The same CNN model will be used for recognizing the classical Kathak dancer's Facial Expressions. These expressions are called Navras. Trained the same model on few converted grey scale 48x48 images and the results are satisfactory. The same model can be trained and tested for KathakNavras facial expressions dataset and checked for its accuracy.

REFERENCES

M. Badshaah, J. Ahmed and S. W. Baek, "Emotion Recognition from Spectrograms with Deep Convolutional Neural Network," 2017 Int. Conf. Platf. Technol. Serv., 2017.

Ali, M.; Mosa, A.H.; Al Machot, F.; Kyamakya, K. Emotion recognition involving physiological and speech signals: A comprehensive review. In Recent Advances in Nonlinear Dynamics and Synchronization; Springer: Berlin/Heidelberg, Germany, 2018; pp. 287–302.

Cohen, Ira, et al. "Evaluation of expression recognition techniques." Image and Video Retrieval. Springer Berlin Heidelberg, 2003. 184- 195.

D Y Liliana State Polytechnic of Jakarta, Indonesia, Emotion recognition from facial expression using deep convolutional neural network 2018 International Conference of Computer and Informatics Engineering (IC2IE) IOP Publishing doi:10.1088/1742-6596/1193/1/012004

Darwin C. (1904) The Expression of the Emotions in Man and Animals, 2nd ed., London.

El Ayadi, M.; Kamel, M.S.; Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognit. 2011, 44, 572–587.

Facial Expression Recognition with Keras, coursera project network Certificate

Facial Expression Recognition with Keras, coursera project network Certificate

Harms, M.B.; Martin, A.; Wallace, G.L. Facial emotion recognition in autism spectrum disorders: A review of behavioral and neuroimaging studies. Neuropsychol. Rev. 2010, 20, 290–322.

https://www.coursera.org/account/accomplishments/verify/2J6TRRL674MU

https://www.coursera.org/account/accomplishments/verify/2J6TRRL674MU

https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

K. M. Rajesh and M. Naveenkumar, "A robust method for face recognition and face emotion detection system using support vector machines," 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, 2016

K.-Y. Hueng, C.-H. Wiu, T.-H. Yieng, M.-H. Sha and J.-H. Chiu, "Emotion recognition using autoencoder bottleneck features and LSTM," in 2016 International Conference on Orange Technologies (ICOT), 2016.

Lin, Y.P.; Wang, C.H.; Jung, T.P.; Wu, T.L.; Jeng, S.K.; Duann, J.R.; Chen, J.H. EEG-based emotion recognition in music listening. IEEE Trans. Biomed. Eng. 2010, 57, 1798–1806.

MinSeopLee , Yun Kyu Lee, Myo-Taeg Lim and Tae-Koo Kang Emotion Recognition Using Convolutional Neural Network with Selected Statistical Photoplethysmogram Features. Applied Sciences (ISSN 2076-3417; CODEN: ASPCC7)

NithyaRoopa. S, Emotion Recognition from Facial Expression using Deep Learning International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-6S, August 2019

P. Ekman and W. Friessen, "Facial action coding system.," Hum. Face, 2002.

Padgett, C., Cottrell, G.: Representing face images for emotion classification. In: Conf. Advances in Neural Information Processing Systems. (1996) 894900.

Revanth Krishna, Real-time facial expression recognition using CNN ISSN: 2454-132X Impact factor: 6.078 (Volume 6, Issue 3)

Routrey, M. Swaen and P Kabisetpathy, "Database, features and classifiers for emotion recognition: a review," Int. J. Speech Technol., 2018.

ShimaAlizadeh Stanford University, AzarFazel Stanford University Convolutional Neural Networks for Facial Expression Recognition,

SRIMANI P.K. AND HEGDE R. Department of Master of Computer Applications and Mathematics, Bangalore University, ANALYSIS OF FACIAL EXPRESSIONS WITH RESPECT TO NAVARASAS IN BHARATHANATYM STYLES USING IMAGE PROCESSING, International Journal of Knowledge Engineering ISSN: 0976-5816 & E-ISSN: 0976-5824, Volume 3, Issue 2, 2012

Wu, C.H.; Chuang, Z.J.; Lin, Y.C. Emotion recognition from text using semantic labels and separable mixture models. ACM Trans. Asian Lang. Inf. Process. TALIP 2006, 5, 165–183.

Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," ISCAS 2010 - 2010 IEEE Int. Symp. Circuits Syst. Nano-Bio Circuit Fabr. Syst., pp. 253–256, 2010.