सी-डैक | प्रगत संगणन विकास केंद्र, नोएडा
CDAC | Centre for Development of Advanced Computing, Noida

A PROJECT REPORT

ON

A Deep Learning & Computer Vision Based Image Processing System for Depression Detection

Submitted In Partial Fulfillment of the Requirement for the Award of

Post Graduate Diploma in Artificial Intelligence (PG-DAI)

Under the Guidance of

MR. NIMESH DAGUR

(Project Guide)

Submitted By

CHETAN DESHMUKH	**DIXANT DUTT**	**LOKESH PATIL**
PRN: 210920528016	**PRN: 210920528018**	**PRN: 210920528023**

CONTENTS

CDAC, B-30, Institutional Area, Sector-62

Noida (Uttar Pradesh)-201307

CERTIFICATE

CDAC, NOIDA

This is to certify that Report entitled **"A Deep Learning & Computer Vision Based Image Processing System for Depression Detection"** which is submitted by Chetan Deshmukh, Dixant Dutt and Lokesh Patil in partial fulfillment of the requirement for the award of **Post Graduate Diploma in Artificial Intelligence** (PG-DAI) to **CDAC, Noida** is a record of the candidates own work carried out by them under my supervision.

The documentation embodies results of original work, and studies are carried out by the student themselves and the contents of the report do not from the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

MR. NIMESH DAGUR

(Project Guide)

ACKNOWLEDGEMENT

We would like to express our best sense of gratitude & endeavor with respect to **Mr. Nimesh Dagur (Project Guide)** CDAC, Noida for suggesting the problems scholarly guidance and expert supervision during the course of this project. Special thanks to **Mr. Ravi Payal (Program Coordinator).**

We are very thankful to **Mr. Nimesh Dagur (Project Guide)** the project guide for constant simulated discussion, encouraging new ideas about this project.

CHETAN DESHMUKH	**DIXANT DUTT**	**LOKESH PATIL**
PRN: 210920528016	**PRN: 210920528018**	**PRN: 210920528023**

ABSTRACT

Psychological problems in college students like depression, pessimism, eccentricity, anxiety etc. are caused principally due to the neglect of continuous monitoring of students' psychological well-being. Identification of depression at college level is desirable so that it can be controlled by giving better counseling at the starting stage itself. If a counselor identifies depression in a student in the initial stages itself, he can effectively help that student to overcome depression. But among large number of students, it becomes a difficult task for the counselor to keep track of the significant changes that occur in students as a result of depression. But advances in the Image-Processing field have led to the development of effective systems, which prove capable of detecting emotions from facial images, in a much simpler way.

Thus, we need an automated system that captures facial images of students and analyze them, for effective detection of depression. In the proposed system, an attempt is being made to make use of the Image processing techniques, to study the frontal face features of college students and predict depression. This system will be trained with facial features of positive and negative facial emotions. To predict depression, a video of the student is captured, from which the face of the student is extracted. The level of depression is identified by calculating the amount of negative emotions present in the entire video.

INTRODUCTION TO THE PROBLEM STATEMENT AND THE POSSIBLE SOLUTION

In college students, depression is the result of the social change due to emergence of the internet, smart phones and different social media sites. Majority of students tend to conceal their psychological problems due to the social stigmas related to depression and also due to peer pressure. Some students remain totally unaware of their psychological problems and thus remain deprived of any help that may prove vital to their mental health. It becomes a difficult task for the counselor to keep track of the significant changes that occur in students as a result of depression in a large number of students.

Thus we need and automated system that captures images of students and analyze them for effective depression detection. Facial expressions are the most important form of non-verbal communications to express a persons' emotional or mental state. A large number of studies are currently undergoing on 'Facial feature analyses' for emotion recognition from images which effectively help in prediction of mental health condition of human beings. This study proposes an automated system that detects depression levels in students by analyzing frontal face images of college students. To predict depression, a video of the student is captured, from which the face of the student is extracted. The level of depression is identified by calculating the amount of negative emotions present in the entire video.

Data Pre-processing

The image files were collected from Kaggle. The dataset is divided into two parts training and validation data.

For the pre-trained model, ImageDataGenerator is used in which the image is rescaled, its height and width are shifted, creates a mirror image, zoomed images, etc. The process works with both training images. In validation images, the rescaling of images occurred.

In train generator, make one single image into many images and converts the image's color mode into grayscale, the batch size divides images into batches and stores them in categorical mode, and in the last shuffle the order of images that are being yielded.

For testing or prediction, a video of a person is taken whose frontal face is fully visible and make a "data" directory in which we extract the images from the videos store them in the "data" directory if the directory already exists then it stores in an already existing directory and stores the extracted images in the "data" if an error comes when there is an issue.

Images are written frame by frame, after that, it will capture images and store them in .jpg image format, it will create the images and store them in form of consecutive numerical names.

It will generate images till the video ends, after that it will stop or break, and the cam will release and it will destroy all the images which are still working in the backend.

CODING

Training

```
from __future__ import print_function

import keras

from keras.preprocessing.image import ImageDataGenerator

from keras.models import Sequential

from keras.layers import
Dense,Dropout,Activation,Flatten,BatchNormalization

from keras.layers import Conv2D,MaxPooling2D

import os


num_classes = 2

img_rows,img_cols = 48,48

batch_size = 32


train_data_dir = r'D:\Dixant\CDAC\Project\face-expression-recognition-
dataset\train'

validation_data_dir = r'D:\Dixant\CDAC\Project\face-expression-recognition-
dataset\validation'


train_datagen = ImageDataGenerator(rescale=1./255,

                                   rotation_range=30,

                                   shear_range=0.3,

                                   zoom_range=0.3,

                                   width_shift_range=0.4,
```

```
                        height_shift_range=0.4,
                        horizontal_flip=True,
                        fill_mode='nearest')

validation_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(train_data_dir,
                                    color_mode='grayscale',
                                    target_size=(img_rows,img_cols),
                                    batch_size=batch_size,
                                    class_mode='categorical',
                                    shuffle=True)

validation_generator =
validation_datagen.flow_from_directory(validation_data_dir,
                                          color_mode='grayscale',
                                          target_size=(img_rows,img_cols),
                                          batch_size=batch_size,
                                          class_mode='categorical',
                                          shuffle=True)

model = Sequential()

# Block-1
```

```
model.add(Conv2D(32,(3,3),padding='same',kernel_initializer='he_normal',input_shape=(img_rows,img_cols,1)))

model.add(Activation('elu'))

model.add(BatchNormalization())

model.add(Conv2D(32,(3,3),padding='same',kernel_initializer='he_normal',input_shape=(img_rows,img_cols,1)))

model.add(Activation('elu'))

model.add(BatchNormalization())

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Dropout(0.2))


# Block-2


model.add(Conv2D(64,(3,3),padding='same',kernel_initializer='he_normal'))

model.add(Activation('elu'))

model.add(BatchNormalization())

model.add(Conv2D(64,(3,3),padding='same',kernel_initializer='he_normal'))

model.add(Activation('elu'))

model.add(BatchNormalization())

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Dropout(0.2))

# Block-3
```

```
model.add(Conv2D(128,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(128,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-4

model.add(Conv2D(256,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(256,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))
# Block-5

model.add(Flatten())
model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
```

```
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-6

model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-7

model.add(Dense(num_classes,kernel_initializer='he_normal'))
model.add(Activation('softmax'))

print(model.summary())

from tensorflow.keras.optimizers import RMSprop,SGD,Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping,
ReduceLROnPlateau

checkpoint = ModelCheckpoint('Emotion_training.h5',
                             monitor='val_loss',
                             mode='min',
```

```
                    save_best_only=True,
                    verbose=1)

earlystop = EarlyStopping(monitor='val_loss',
                    min_delta=0,
                    patience=3,
                    verbose=1,
                    restore_best_weights=True
                    )

reduce_lr = ReduceLROnPlateau(monitor='val_loss',
                    factor=0.2,
                    patience=3,
                    verbose=1,
                    min_delta=0.0001)

callbacks = [earlystop,checkpoint,reduce_lr]

model.compile(loss='categorical_crossentropy',
          optimizer = Adam(learning_rate=0.001),
          metrics=['accuracy'])

nb_train_samples = 16095
nb_validation_samples = 3924
```

```
epochs=40

history=model.fit(train_generator,
                  steps_per_epoch=nb_train_samples//batch_size,
                  epochs=epochs,
                  callbacks=callbacks,
                  validation_data=validation_generator,
                  validation_steps=nb_validation_samples//batch_size)
```

Testing

```
from keras.models import load_model
from time import sleep
from keras.preprocessing.image import img_to_array
from keras.preprocessing import image
import cv2
import os
import numpy as np

cam = cv2.VideoCapture(r'Rajpal.mp4')

try:
    if not os.path.exists('data'):
        os.makedirs('data')
```

```
except OSError:
    print ('Error: Creating directory of data')


currentframe = 0


while(True):
    ret,frame = cam.read()


    if ret:
        name = './data/frame' + str(currentframe) + '.jpg'
        print ('Creating...' + name)


        cv2.imwrite(name, frame)


        currentframe += 1
    else:
        break


cam.release()
cv2.destroyAllWindows()


face_classifier = cv2.CascadeClassifier(r'D:\Dixant\CDAC\Project\Facial-Expressions-Recognition\Facial-Expressions-Recognition-master\haarcascade_frontalface_default.xml')
```

```
classifier =load_model('Emotion_training.h5')


class_labels = ['Happy','Sad']


emotions=[]
i=0


while True:
    try:
        print(i)


        file=(r"C:\Users\divya\Documents\data\frame"+str(i)+".jpg")
        img=cv2.imread(file)
        gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        i+=1


        faces=face_classifier.detectMultiScale(gray,1.1,4)
        for(x,y,w,h) in faces:
            cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),3)
            roi_gray = gray[y:y+h,x:x+w]
            roi_gray =
cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)


            if np.sum([roi_gray])!=0:
```

```
            roi = roi_gray.astype('float')/255.0
            roi = img_to_array(roi)
            roi = np.expand_dims(roi,axis=0)

            preds = classifier.predict(roi)[0]
            label=class_labels[preds.argmax()]
            label_position = (x,y)
            emotions.append(label)
cv2.putText(img,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,2,(0,2
55,0),3)
          else:
            cv2.putText(img,'No Face
Found',(20,60),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
            continue

        cv2.waitKey(0)
        cv2.destroyAllWindows()
    except:
        "error in image"
        break

import shutil
path='C:/Users/divya/Documents/data'
shutil.rmtree(path)
```

```
len(emotions)
from collections import Counter
d = Counter(emotions)
Sad = d['Sad']
Happy = d['Happy']
Sad
Total=Sad + Happy
emotion_1 = Sad/Total
emotion_1
if emotion_1 > 0.70:
    print("Severe Depression : consult with psychology immediately")
elif emotion_1 > 0.40:
    print("Mild Depression ")
elif emotion_1 > 0.10:
    print("Low Depression ")
else:
    print("You don't have  Depression")
```

RESULTS

This the result table that shows the comparison of different activation functions are used classification of emotions.

Model	Loss	Accuracy	Val Loss	Val Accuracy
Sigmoid Activation function	0.3674	0.8423	0.2292	0.9101
Softmax Activation function	0.3224	0.8614	0.1964	0.9214

From that pre-trained model has been created and extraction and prediction are done. Those predictions determine whether the person's emotion is happy or sad.

$$\text{Emotion}= \frac{Sad}{Total}\text{ , here Total=sad + happy}$$

If Emotion is greater than 70, then the level of depression is severe.

If Emotion is greater than 40 and less than 70, then the level of depression is mild.

If Emotion is greater than 10 and less than 40, then the level of depression is low.

And if the emotion is less than 10, then the person is having no depression.

CONCLUSION AND FUTURE SCOPE

This study was undertaken for finding out the level of depression in five different videos. The presence of ‘Happy’ - (positive emotion) and ‘Sad’- (Negative emotion) facial features, which are found prominent in depression videos were found out and analyzed. The dataset for training and testing was captured separately and the facial features of the same were classified using a Convolutional neural network (sigmoid) classifier. The amount of the positive and negative emotions in each video was analyzed and the videos were predicted as videos with ‘Severe Depression’, ‘Mild Depression’ or ‘Low Depression’. The classifier predicted the outcomes with a maximum accuracy of 84.4% accuracy.

The more the number of training samples, the more accurate will be the classifier prediction. The pre-trained model is created from the training and validation data. The videos captured are of more than thousand frames, out of which each frames were considered here for prediction purpose. And by calculating the sad, happy, and total emotions in frames and predict the depression level. This process can be done for the entire video, by finding out the key frames of the video, by using a key frame extraction technique in the future work. However, for more accurate depression detection, the history of the person should also to be taken into consideration. Therefore, in the future work, more videos of the same person, taken at different time duration can be considered. This may help to analyze and compare the past and the present mental state of the person and provide more information to the process of depression level identification.

Depression detection from videos alone forms only a part of the whole process of identifying depression. Those persons, who may be classified as not depressed, may be victims to depression in the future. For this reason, their other activities have to be continuously monitored. This includes the continuous monitoring of their academic activities, their extra-curricular activities and also their social activities. Monitoring academic activities include monitoring the person’s performance and attendance. Decreasing in performance or attendance may also be due to a person’s extra-curricular activities, like engaging in sports or arts. If a person's performance or attendance are poor and they are not active in other mediums like arts or sports also, then they may be at a high risk of falling into depression. Hence persons' extra-curricular activities have also to be continuously monitored for identification of depression. In addition to this, there should also be a way of monitoring a person's social media content because if the persons' social media content show a negative attitude towards life, then such a person may be a victim of stress and depression. The future work to this study is to form an elaborate model of depression identification process, by taking all the above mentioned factors into consideration and combining it with the current work of identifying depression with images.

REFERENCES & BIBLIOGRAPHY

- Venkataraman, D., Parameswaran, N. S. "Extraction of Facial Features for Depression Detection among Students." International Journal of Pure and Applied Mathematics, International Conference on Advances in Computer Science, Engineering and Technology, pp. 455-462, 2018.
- Owayjan, Michel, Roger Achkar, and MoussaIskandar. "Face Detection with Expression Recognition using Artificial Neural Networks." In Biomedical Engineering (MECBME), 3rd Middle East Conference on, pp. 115-119. IEEE, 2016

Articles:

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6749518/
- https://www.frontiersin.org/articles/10.3389/fpsyg.2021.688376/full