Facial Emotion Detection Using CNN

Sarthak Jain - 191IT145
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: sarthak94511@gmail.com

Gaurav Kumar - 191IT114
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: gaurav.191it114@nitk.edu.in

***Abstract*—Emotion detection models are of great importance since they are important in increasing interaction of machines and humans. Systems can adapt their beahavioral patterns on basis of emotions detected. In this project we tried building a Convolution Neural Network that can detect seven major human emotions (Angry,Disgust, Fear, Happy,Sad, Surprise,Neutral)**

I. Introduction

The problem trying to solve

- With this intermediate-level project, we have developed a CNN model that predicts Human Facial Emotions.Facial Emotions helps us to detect someone's intentions similarly machines too can use these emotions for various purposes like the facial expression is a nonverbal way of emotional expression, and it can be considered as concrete evidence to uncover whether an individual is speaking the truth or not

Existing solutions are:

- Most Of the current models for determining facial expressions are based on Ekman and Friesen 's model.They proposed a Facial Action Coding System (FACS) for determining facial expressions, which is based on a fact that expressions result from the change of facial parts.
- Popular databases are : Jaffe, CK+, and FER2013

Core idea of our project is

- Classification of facial emotion out of seven major human emotions (Angry,Disgust, Fear, Happy,Sad, Surprise,Neutral)
- Image of any dimension can be given as input
- Images (both RGB and grayscale can be given as input)

This problem has grabbed the attention of many researchers around the world, leading to innumerable projects in this field. These projects aim at detection of human emotions and perform analysis of various types.

II. Literature Survey

References we used for our project:

For learning about Convolution Neural Networks we referred to Coursera course on CNN by Andrew ng and also many articles on popular websites like medium,towards datascience

For learning more on faster and easier model building we referred to tensorflow official documentation

To learn more on existing models we referred to "Challenges in Representation Learning: A report on three machine learning contests." I Goodfellow, D Erhan, PL Carrier, A Courville, M Mirza, B Hamner, W Cukierski, Y Tang, DH Lee, Y Zhou, C Ramaiah, F Feng, R Li, X Wang, D Athanasakis, J Shawe-Taylor, M Milakov, J Park, R Ionescu, M Popescu, C Grozea, J Bergstra, J Xie, L Romaszko, B Xu, Z Chuang, and Y. Bengio. arXiv 2013.

III. Problem Statement

"Facial Emotion Detection using Convolutional Neural Networks"

A. Objectives

- Developing a CNN model
- Training and testing on very large dataset (FER-2013 used for this project)
- Taking image of any dimension from user.
- Predicting emotion of input image

B. Solved using:

- Dataset used is FER2013 that we took from kaggle
- We did GPU enabled training of our model (since it has 35000 + images)
- for creating model we used python libraries tensorflow ,keras.
- other python libraries used are numpy,scipy,pandas etc.

IV. Methodology

The dataset is loaded using pandas,images are reshaped to 48*48,Standardisation of all images is done.CNN model is built using keras library, optimiser used is adam and loss function used is categorical.Various dropout layers are added to prevent the overfitting of model.The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting.

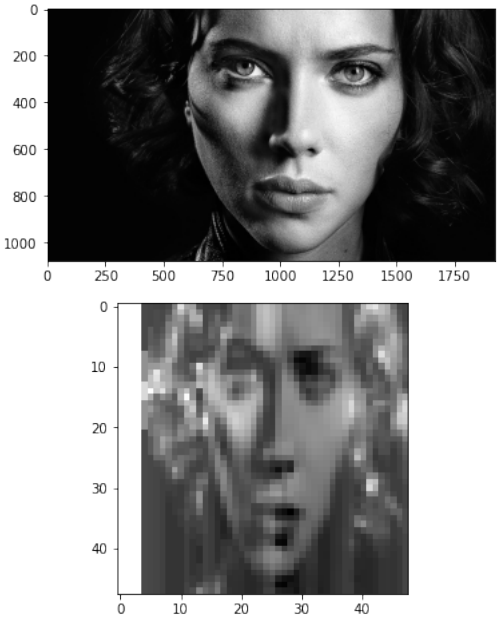
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 46, 46, 64)        640
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 46, 46, 64)        36928
_________________________________________________________________
batch_normalization (BatchNo (None, 46, 46, 64)        256
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 23, 23, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 23, 23, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 23, 23, 128)       73856
_________________________________________________________________
batch_normalization_1 (Batch (None, 23, 23, 128)       512
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 23, 23, 128)       147584
_________________________________________________________________
batch_normalization_2 (Batch (None, 23, 23, 128)       512
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 11, 11, 256)       295168
_________________________________________________________________
batch_normalization_3 (Batch (None, 11, 11, 256)       1024
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 11, 11, 256)       590080
_________________________________________________________________
batch_normalization_4 (Batch (None, 11, 11, 256)       1024
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 256)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 5, 5, 256)         0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 5, 5, 512)         1180160
_________________________________________________________________
batch_normalization_5 (Batch (None, 5, 5, 512)         2048
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 5, 5, 512)         2359808
_________________________________________________________________
batch_normalization_6 (Batch (None, 5, 5, 512)         2048
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 512)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 2, 2, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 512)               1049088
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_6 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 7)                 903
=================================================================
Total params: 5,905,863
Trainable params: 5,902,151
Non-trainable params: 3,712
_________________________________________________________________
```

Fig. 1. Model

GPU enabled training is done to improve time of training .Kaggle is used for the purpose. Using PIL library images are taken as input from user and facial emotion detection result is show in form of code assigned to various emotions. (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral) block.

And in this model, the output size N of each convolutional layer can be formulated as : N=(I-F+2P)/S+1 where I,F,P,S, and denote the input size, kernel size, padding size, and stride size, respectively. In each max pooling layer, the padding size is 0. Rectified Linear Units (ReLUs) are adopted as the activation function in the convolutional layers and max-pooling layers to avoid gradient explosion and ensure faster convergence speed during the back-propagation operation. Softmax is used as the activation function in the output layer, the input of which is the matrix output from the fully connected layers.

V. Results and Analysis



```
arr=model.predict(np.array([image]))

c=0
index=0
max=0.00
for x in arr[0]:
    if x > max:
        max=x
        index=c
    c+=1

print(index)
# (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)
```

```
0
```

Fig. 2. Output 1

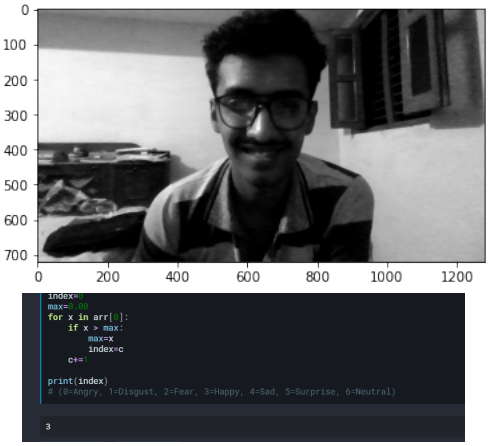


Fig. 3. Output 2

The outputs shown has number in range 0 to 6.Each number corresponds to a human emotion. The accuracy that we achieved through our model is around 66 percent on our test data.

Our program takes image of any dimension as well as of any type RGB,grayscale as input.

We resize it to 48*48 for classification.

In a nutshell, the project accurately detects the human emotions.The drawback that our project has is during downgrading of image given as input , image quality sometimes becomes too bad that wrong results are displayed.

Scope for improvement: Trying various image compression techniques to reduce data loss during downgrading of image taken as input.

VI. Conclusion

Our project aims at detection of facial emotions. Our program classifies image given as input to one of the seven classes/emotions.(0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)

We have used keras library of python for building our model and the dataset we used is FER2013.Images are downgraded to 48 x 48 since the images present in our dataset were of this dimensions.Although this is lossy downgrading in future we would like to extend our work by using various image compression techniques to get more better results.

Acknowledgment

We would like to express our gratitude to Mr Jaidhar CD Sir for his assistance in making our project successful, along with the IT department, NITK, for providing us the opportunity to execute this project.

References

[1] https://docs.opencv.org/master/da/df6/tutorial_py_table_of_contents_setup.html

[2] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[3] Dennis Hamester et al., “Face Expression Recognition with a 2-Channel Convolutional Neural Network”, International Joint Conference on Neural Networks (IJCNN), 2015.

[4] lex Krizhevsky et al. “ImageNet Classification with Deep Convolutional Neural Networks”, Neural Information Processing Systems (NIPS), 2012.