Facial Emotion Recognition Using Deep Convolutional Neural Network

Abstract

Facial Emotion Recognition system aims at incorporating the people in understanding the emotion of person by their facial impression using deep convolutional neural network. One important task is to get features from the image set and the other important task is to get merge labels with the images to train the model or classifier to work. Our experimental results visualize the advantage of facial emotion recognition in terms of feature selection, classification performance, and interpretation. We trained deep convolutional neural network to classify facial expression images of different people in 6 categories which are Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral. The neural network consists of 9 different layers in which 6 are convolutional layers and 3 are dense or fully connected layers. We used TFLearn high level api for this purpose to that we don't have to create each convolutional layer's line by line. To make training faster, we use GPU in efficient way tensor board for visualization. To overcome overfitting in fully connected layer we use recently developed method called "dropout" that is very effective.

1 Introduction

Being able to analyze the images dataset of facial emotion recognition using deep learning is good approach for detection. The problem of entity detection and recognition in images can be seen as the analysis of a pixel array and, therefore, simplified to the analysis of single pixel. Unfortunately, there are no easy ways to identify correctly what a picture contains.

This report is part of kaggle dataset from where we take dataset to use it for our purpose and we modify the dataset according to that for our purpose. We used different technologies like TFLearn, tensorflow, tensorboard for this purpose. The goal of this project is to detect the expression of different people and so we can analyse or reply them efficiently.

Convolutional neural networks (CNNs) contributes to detection of emotion. Their capacity can be controlled by varying their layers and sizes and using different loss functions, and they also make strong and accurate assumptions about the nature of images. Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse.

To better understand the work process, we split the project in two parts: the first one is the data collections preparation, a large set of emotion images and their emotions in separate files for training and testing, and the second one is the best model generation for detecting facial expression. This part is resumed in the following steps:

- Features Extraction: Extract the key-points of all the images (and all the train images). These points are very descriptive for the contents of an image.

- Label Extraction: Extract the labels for each images from different files and convert them to different format for use and then take the labels for these images from other file and merge them so the emotion image and emotion status sticks together.

  The model generation part is resumed as follows:

- Create the Model: Creating model for this purpose is very important process and risk taking because in case of CNN the accuracy of model changes with the change or increase in layers. We are getting idea for this from the paper “ImageNet Classification Using Deep Convolutional Neural Network”

- Classify the features: Given the features of the image, we want to find the emotion image segment that match better each key-point and which shows best match gives him proper classification number.

So, in order to complete these tasks, we get the dataset from the Kaggle company website which is publicly available for analysis.
Our final network contains 6 convolutional and three fully-connected layers, and this depth seems to be important: we found that removing any convolutional layer (each of which contains no more than 1% of the model’s parameters) resulted in inferior performance.

Network takes more time in computing on CPU rather than on GPU.

2 The Dataset

The dataset taken is FER2013Train and FER2013Test dataset. This dataset consists of real-world images collected from Kaggle showing different person emotion in various circumstances. The dataset comes in two versions: The original FER2013 dataset and the FER2013plus dataset. Here we are using mixture of both for our purpose by creating new dataset from these datasets. This newly created dataset consists of around 28,000 images in which around 25,000 are used for training and other 3,000 for testing.

There are 6 emotion categories and we give numbers to them as follows:
0=Angry,
1=Disgust,
2=Fear,
3=Happy,
4=Sad,
5=Surprise,
6=Neutral

Some of the images are as follows:

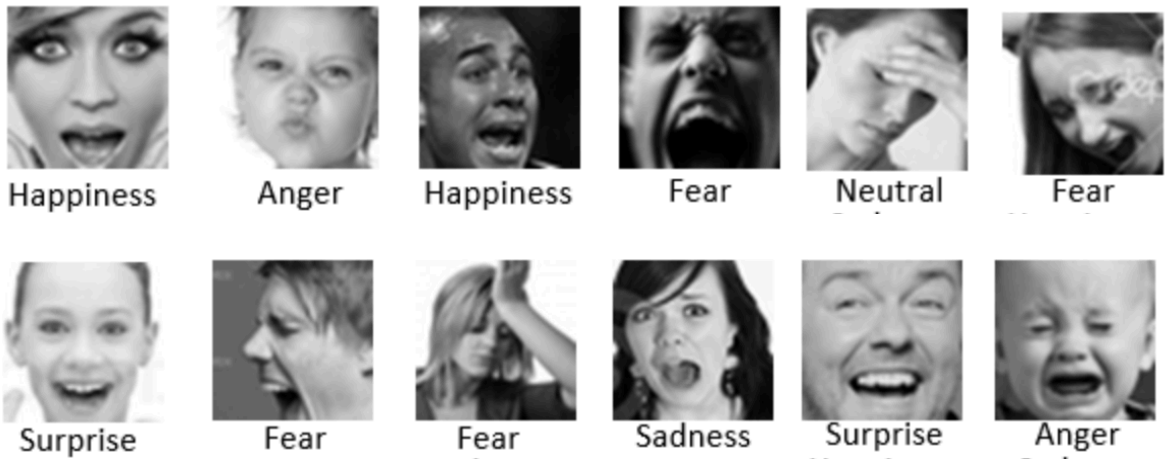

Figure 1

3 The Architecture

The architecture of the system is shown in figure 2 below and it consists of 5 convolutional layers and 2 fully connected layers. Below are the some of the functions details which are used in convolutional layers and fully connected layers.

3.1 Relu(Rectifier)

Relu is the rectifier activation function used in convolutional neural network in the convolutional layer. It is non linear function. The state of the art of non-linearity is to use

rectified linear units (ReLU) instead of sigmoid function in deep neural network. While training network it increases the training process. The Mathematical equation is: $h = \max(0, a)$ where $a = Wx + b$.

The other benefit of ReLUs is sparsity. Sparsity arises when $a > 0$. The more such units that exist in a layer the more sparse the resulting representation. Sigmoid on the other hand are always likely to generate some non-zero value resulting in dense representations.

3.2 Dropout Function

It is function used to automatically handles the scaling of output of neurons for reduction of overfitting in training.

3.3 Training on Multiple GPU

Training the dataset on multiple GPUs using CUDA installed takes less time to train the model rather on single GPU or more CPUs. CUDA is NVidia package for parallel processing. We used to do parallel processing and it takes less time to train the model and do predictions rather than training on CPUs. Also you can get multiple GPUs from different machine learning cloud platforms i.e. Google Cloud and Amazon ML.

3.4 The Overall Architecture

Now we are ready to describe the overall architecture of our CNN. As depicted in Figure 2, the net contains 9 layers with weights; the first six are convolutional and the remaining three are fully connected. The output of the last fully-connected layer is fed to a 6-way softmax which produces a distribution over the 6 class labels.

The neurons in the fully connected layers are connected to all neurons in the previous layer. Max pooling layers follow the first and second convolutional layers. Max-pooling layers, of the kind described in Section 3.4, follow both response-normalization layers as well as the fifth convolutional layer.

The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer. The first convolutional layer filters the 48×48×3 input image with 64 kernels of size 5×5×3 with a stride of 5 pixels (this is the distance between the receptive field canters of neighbouring neurons in the kernel).

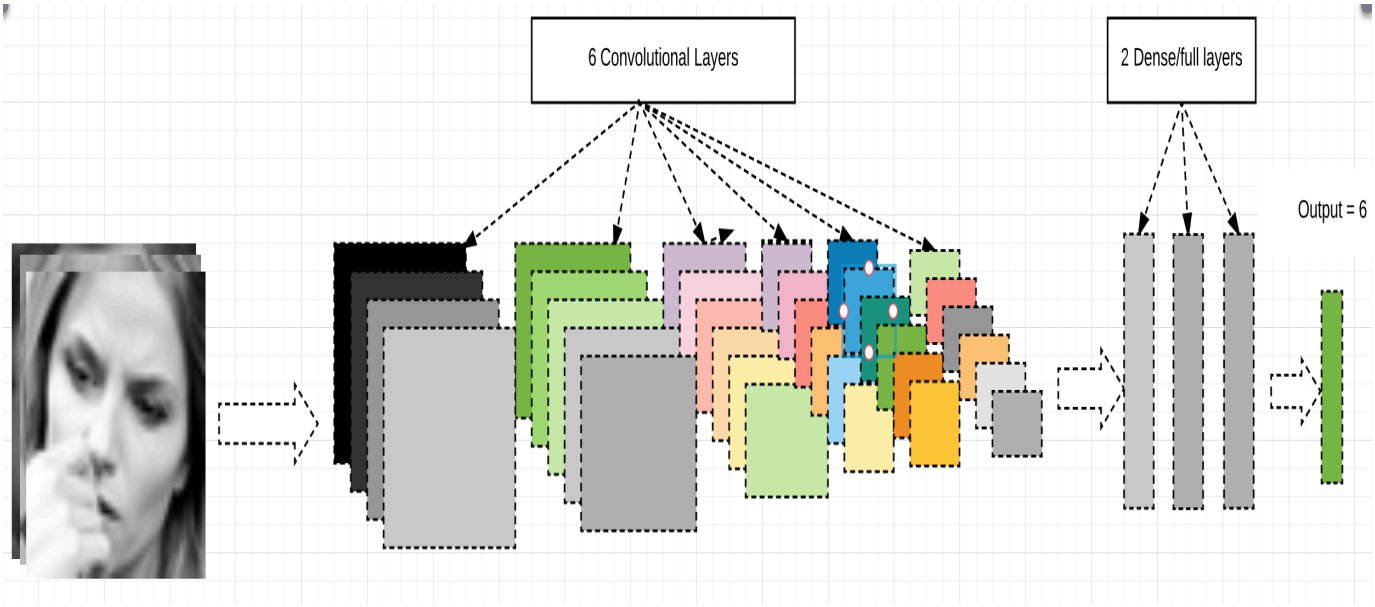

Figure 2

4 Details of Learning

We trained our model using stochastic gradient descent with batch size of 48 by 48 and learning rate of 0.03. Dropout with rate of 0.8 decreases the model's training error and increases accuracy.

4 Technologies

We used TFLearn high level api and tensorflow and tensorboard for this kind of work. Here is the snapshot of tensorboard with adam validation on the training set.

6 Test and Conclusion

We take test on the newly image downloaded from the internet and the algorithm classifies it with accuracy of 75%. We test the new image of face and give good prediction and give good results.

From our results, we show that deep convolutional neural network is capable of achieving record breaking results on a highly challenging dataset using purely supervised learning.

7 References

1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (NIP 2012). ImageNet Classification with Deep Convolutional Neural Networks
2. FER2013Train. (2013). Challenges in Representation Learning: Facial Expression Recognition Challenge [Data file]. Retrieved from https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data