CS766 - PROJECT MID-TERM REPORT

Emotion detection from a given image using Convolutional Neural Networks

Naman Agrawal (nagrawal8@wisc.edu), Harsh Singhal (hsinghal@wisc.edu)

Problem that we are trying to solve ?

This project detects the emotion (Angry, Happy, Neural, Fear, Sad and Surprise) of each face detected in a given image using deep learning techniques such as CNN.

Summary of current progress

a. Completed until now

A Short Literature survey

A lot of work is being done currently, both academically and industrially, in this field of emotion detection such as sentiment detection and consumer reaction on advertisements.

Last year, Apple bought Emotient, ‘a startup company that utilises artificial intelligence to analyze facial expressions and read emotions'. Companies such as Affectiva and Emotient in the USA, and CrowdEmotion in the UK are doing similar work.

Emotient's acquisition by Apple coupled with the statement made by Andrew Moore, the dean of computer science at Carnegie Mellon, that 2016 is the year when machines learn to grasp human emotions, excites all people that have been working in this challenging field for some time.

Data set collection

The dataset we are using for training the model is from a Kaggle Facial Expression Recognition Challenge a few years back (FER2013). It comprises a total of 35887 pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 6 emotion classes: anger, fear, happiness, sadness, surprise, and neutral.

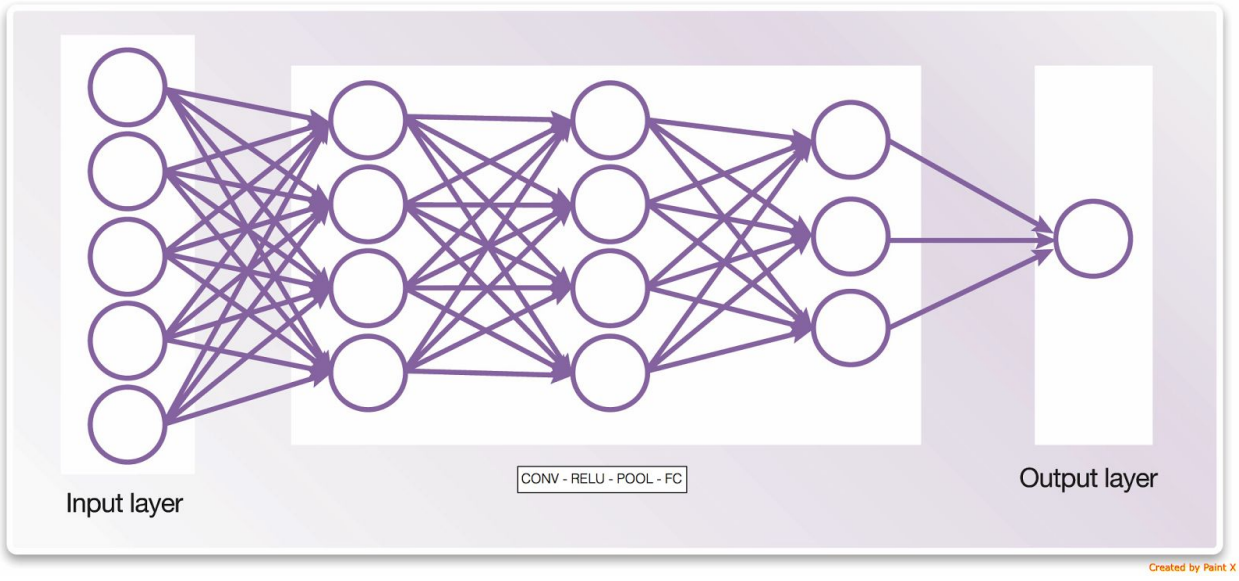
Experimental setup

CNNs are known to imitate how the human brain works when analyzing visuals. So we are training our emotion detector using CNN on Theano and Keras Deep Learning models.

A typical architecture of a convolutional neural network will contain an input layer, some convolutional layers, some dense layers (aka. fully-connected layers), and an output layer (as shown below). These are linearly stacked layers ordered in sequence. In Keras, the model is created as Sequential() and more layers are added to build architecture.

HAAR filter in OpenCV is being used for face detection.

CNN



b. Current results

Accuracy that we are getting is around 40% so we are trying to improve the accuracy by training the model further to 60%

c. Difficulties that arose during implementation

Initially we built a simple CNN with an input, three convolution layers, one dense layer, and an output layer to start with. As it turned out, the simple model performed poorly with low accuracy, so we realised we needed to go deeper and implement more layers and then train again.

While training a Convolutional Neural Network (CNN) for face detection and emotion detection using labelled database of face images, one dilemma was to choose between using backend- using Theano or Tensor-Flow. Theano provides low-level support while programming, so we decided to use that.

Furthermore, the API call syntax of the Theano and Keras library changes much with every new release (which are very frequent) since these libraries are still being developed. Being consistent across all updates and versions is another issue we faced.

Also, without GPU support, Theano was running slow. So any new changes took a lot of time to compile and test.

Changes in proposal

We are focussing now on training the basic expressions using CNN and we may decide not to use micro-expressions. Instead, after training the model for a sufficient accuracy, we will try to make it resistant to edge cases of images - images taken in dark background, faces having beard, spectacles, etc which make feature detection from an image difficult.