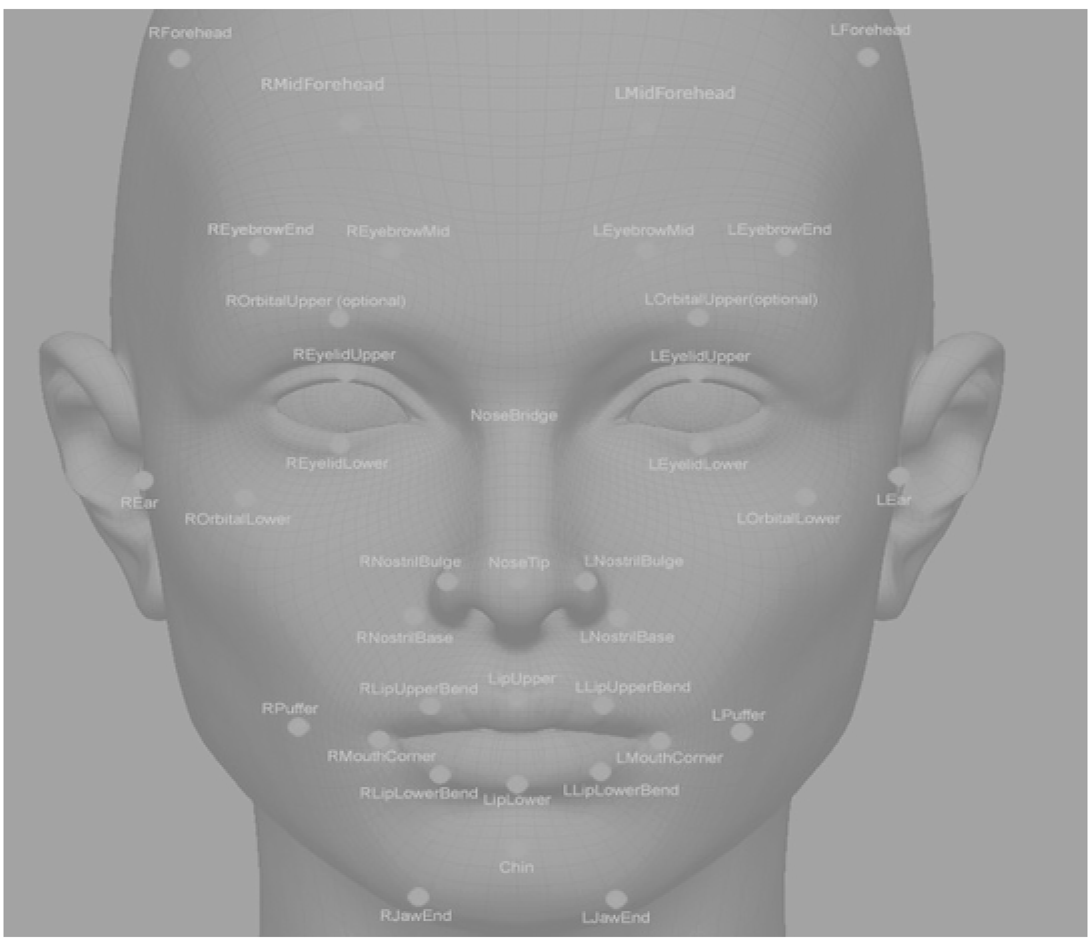
Machine Learning

FACIAL EMOTION RECOGNITION

Brains In Jars

- Hemachandra Kolisetty 14MA20015
- Sarath NVP 14ME10039
- Sri Keerthi 14IE10007
- Nayan Raju Vysyaraju 14MA20049
- Nikitha Kondapalli 14IE10017
- Sri Lekha D 14EE10014
- Manasa Gogineni 14EC10018
- Sharath Chandra Vemula 14EC10060



Problem Statement

The goal of this project is to predict, from the grayscale picture of a person's face, which emotion the facial expression conveys. Our evaluation metric will be the accuracy for each emotion (fraction of correctly classified images). In short,

Input : 48 x 48 grayscale image of a face

Output : Emotion conveyed by facial expression

Methodology

We chose Tensorflow as our framework to train neural networks on. Tensorflow is a scientific computing framework with wide support for machine learning algorithms including popular neural network and optimization libraries which are simple to use. It is easy to use, efficient and beginner friendly.

Dataset

We used a dataset provided by Kaggle website which comprises 48x48 pixel grayscale images of human faces each labeled with one of the 7 emotion categories: anger, disgust, fear, happiness, sadness, surprise and neutral. We used a training set of 28,709 examples, a validation set of 3,589 examples and a test set of 3,589 examples.

The class distribution of the dataset is shown in the figure:

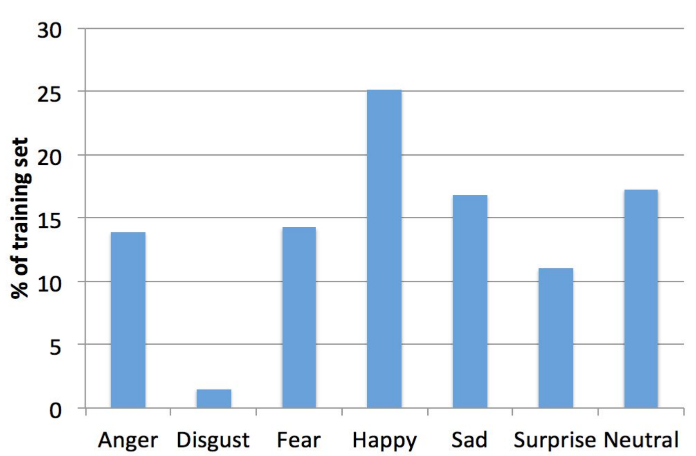


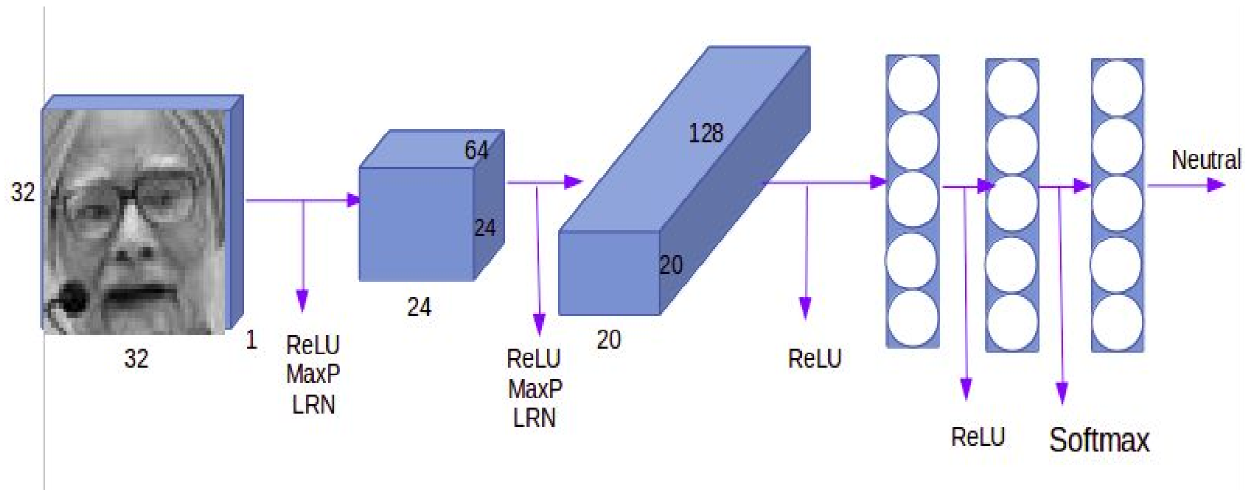
Image Pre-Processing

- We performed central cropping to the images to get the target width and height(32x32), so that they form a bounding box around the face region.
- We accounted for variation in lighting of the input images by adjusting brightness and contrast to fixed band values.
- We converted the image pixel values to zero mean and unit norm with the per image whitening function.

CNN Architecture

The final architecture retained can be described as follows :

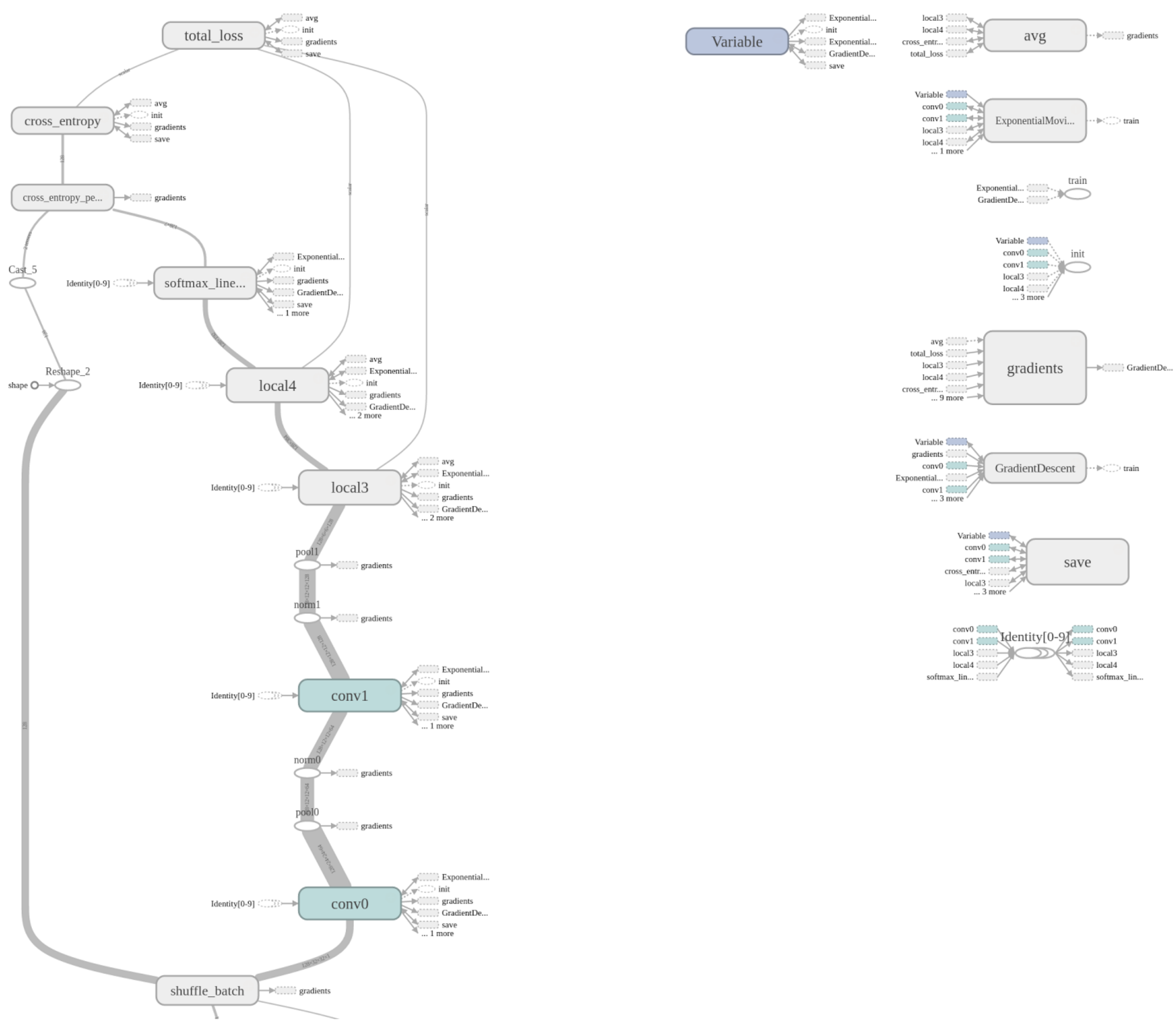
- 9 x 9 Conv (stride 1) - ReLU - 3 x 3 Max Pool (stride 2) - LR_Norm with 64 filters
- 5 x 5 Conv (stride 1) - ReLU - 3 x 3 Max Pool (stride 2) - LR_Norm with 128 filters
- 3 FC hidden layers with ReLU activation



We built a CNN with shallow architecture and fixed depth of two convolutional layers and two fully connected layers. The first part of the network refers to two convolutional layers that possess local response normalization, max pooling and ReLU non-linearity. When we deal with ReLU neurons, we have unbounded activations, LRN is used to normalize that. The second part of the network refers to two fully connected layers having ReLU non-linearity. Finally, the network is followed by softmax cross entropy loss function which computes the loss with labels. Along with dropout and local response normalization, we included L2 regularization in our implementation.

Architecture characteristics

- Exponentially decayed learning rate, L2 regularization
- Local Response normalization after each layer
- Gradient Descent optimiser
- ReLU non-linearity activation
- 9x9 convolution filter with stride 1 for first convolutional layer and 5x5 convolution filter with stride 1 for second convolutional layer
- 3x3 max pools with a stride of 2 for both the convolutional layers
- Softmax cross-entropy to compute cost



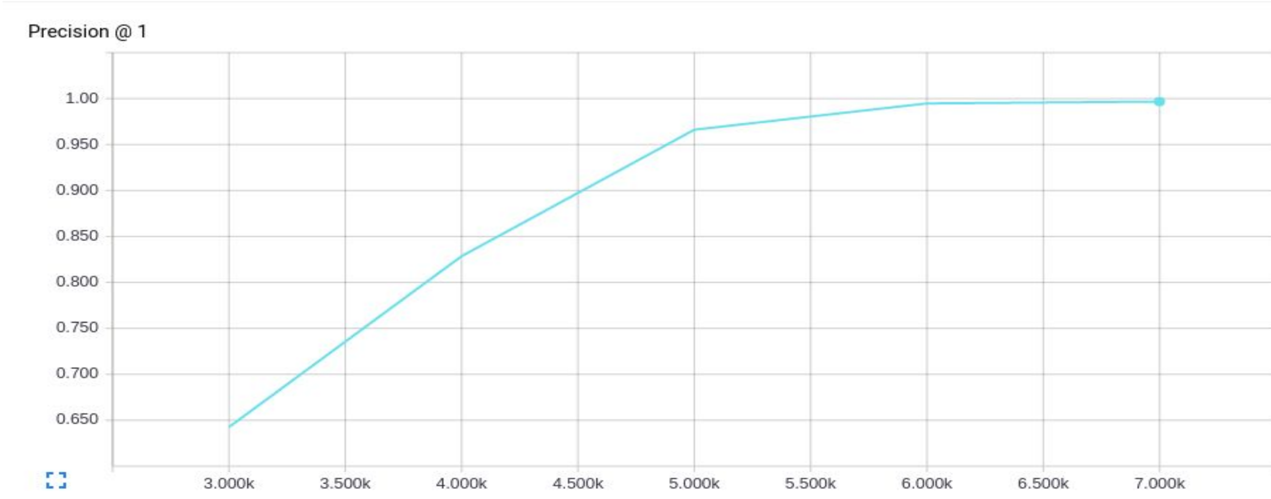
Experiments

Initially, we build the above mentioned network and obtained a train accuracy of 0.997 and test accuracy of 0.57. In order to reduce the over-fitting, we used dropout for the first and second convolutional layers and the last fully connected layer. Doing that, we obtained a train accuracy of 0.85 and a test accuracy of 0.53. This was undesirable as though the train accuracy decreased, the test accuracy also decreased than the previous value. We also added another convolutional layer to the mentioned architecture, but this also failed as we got a train accuracy of 0.92 and a test accuracy of 0.54. Hence, our first architecture was better than the other two.

Results

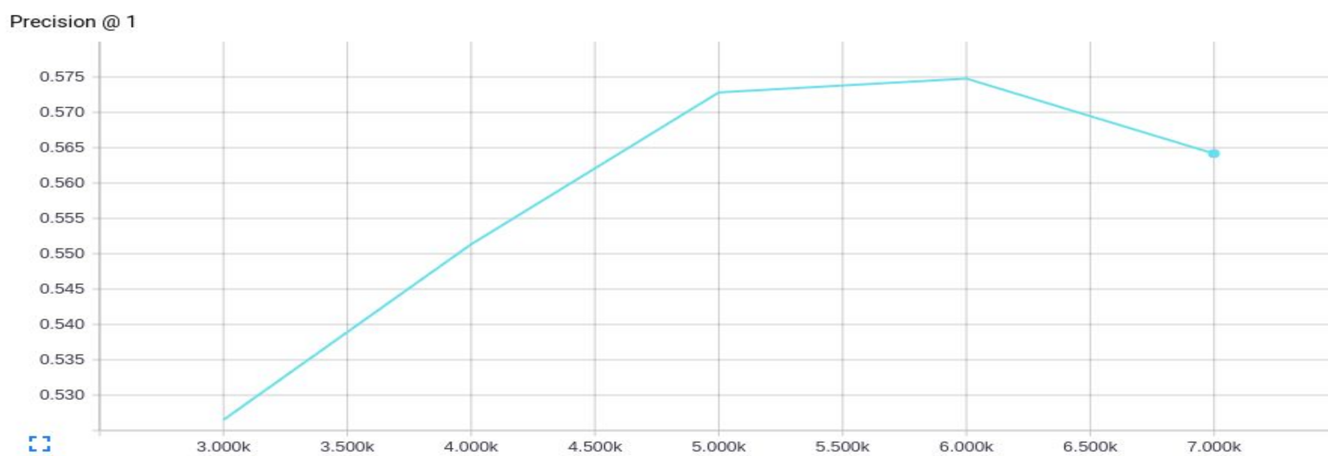
We achieved final validation accuracy of 57% while the state of the art test accuracy for 7 emotion categories using deep networks is 61%, and the top Kaggle implementation received an accuracy of 71%.

Train Accuracy : 0.997



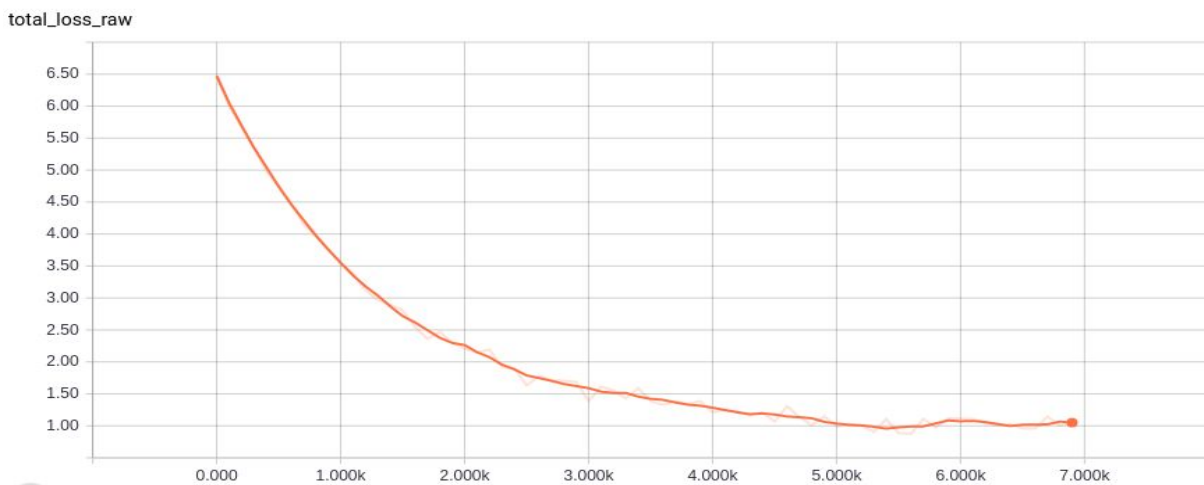
```
  File "eval.py", line 106, in evaluate
    graph_def=graph_def)
NameError: global name 'graph_def' is not defined
sarath@sarath-Inspiron-3537:~/ml_project$ python eval.py
False
evaluating model...
('Reading file:', ['/home/sarath/ml_project/fer_data/train_batch.csv'])
WARNING:tensorflow:Passing a `GraphDef` to the SummaryWriter is deprecated. Pass
 a `Graph` object instead, such as `sess.graph`.
2016-11-14 02:58:28.293357: precision @ 1 = 0.643
2016-11-14 03:01:01.864236: precision @ 1 = 0.829
2016-11-14 03:03:38.144185: precision @ 1 = 0.966
2016-11-14 03:06:12.963486: precision @ 1 = 0.995
2016-11-14 03:08:49.366027: precision @ 1 = 0.997
sarath@sarath-Inspiron-3537:~/ml_project$ tensorboard --logdir=/home/sarath/ml_p
roject
WARNING:tensorflow:IOError [Errno 2] No such file or directory: '/usr/local/lib/
```

Test Accuracy : 0.57




```
sarath@sarath-Inspiron-3537: ~/ml_project
sarath@sarath-Inspiron-3537:~$ cd ml_project/
sarath@sarath-Inspiron-3537:~/ml_project$ python eval.py
True
evaluating model...
('Reading file:', ['/home/sarath/ml_project/fer_data/test_batch.csv'])
WARNING:tensorflow:Passing a `GraphDef` to the SummaryWriter is deprecated. Pass
 a `Graph` object instead, such as `sess.graph`.
2016-11-14 03:34:57.018887: precision @ 1 = 0.527
2016-11-14 03:35:22.591494: precision @ 1 = 0.551
2016-11-14 03:35:50.113864: precision @ 1 = 0.573
2016-11-14 03:36:17.568392: precision @ 1 = 0.575
2016-11-14 03:36:45.191446: precision @ 1 = 0.564
sarath@sarath-Inspiron-3537:~/ml_project$ cd ml_project/
```

Loss Function



CONCLUSION

We developed various CNNs for a facial expression recognition problem and evaluated their performances using different post-processing and visualization techniques. The results demonstrated that deep CNNs are capable of learning facial characteristics and improving facial emotion detection.

References

[1] http://cs231n.stanford.edu/reports2016/022_Report.pdf

[2] http://cs231n.stanford.edu/reports2016/005_Report.pdf

[3] http://cs231n.stanford.edu/reports2016/023_Report.pdf

[4] http://cs231n.stanford.edu/reports2016/009_Report.pdf

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

[6] https://www.tensorflow.org/

[7] http://cs231n.github.io/convolutional-networks/