Machine Learning Engineer Nanodegree

Capstone Project

Swaraj Jena
29 May 2017

Emotion Recognition using Deep Learning

I. Definition

Project Overview

Currently a lot of research happening in the domain of Human Computer interaction(HCI) to increase user-friendliness of computers and make it as close as possible to that of Human-Human interaction. A key feature in human interaction is the universality of facial expression and body language.Recently one of the top application of Artificial intelligence is to recognise and understand human faces. One of the advanced development in this field is Emotion Recognition. In addition to identifying faces ,the computer is now able to determine the facial expression and hence the emotion of a person. There are various potential applications of emotion recognition. One possible application is in the area of surveillance and behavioural analysis of people for law enforcement. It can also be used in theme parks or shopping malls to understand customer behaviour ,so that the services can be offered accordingly. It can also be used for the robots to keep track of the mental state of the user so that it can behave appropriately. Emotion recognition therefore plays a key role in improving human machine interaction.

In nineteenth century ,Charles Darwin published that facial expression plays a greater role in nonverbal communication. In 1971,Ekman & Friesen declared that facial behaviours are universally associated with particular emotions[10]. Hence if properly modeled,this can be a very convenient feature in human machine interaction,

Due to the importance of facial expression in designing Human–computer interaction systems, various feature extraction and machine learning algorithms have been developed for Facial Expression Recognition. Most of these methods are hand-crafted features extractions followed by a classifier such as [1] who's used Local binary pattern feature extractor with SVM classification, Haar[2], SIFT[3], Gabor filters with fisher linear discriminant[4], and Local phase quantization (LPQ) [5]. The recent success of **convolutional neural networks (CNNs)** in tasks such as image classification has been extended to the problem of facial expression recognition[8]. Unlike traditional machine learning and computer vision approaches where features are defined by hand, CNN learns to extract the features directly from the training database using iterative algorithms like gradient descent.

Problem Statement

The majority of existing techniques focus on classifying 7 basic (prototypical) expressions, which have been found to be universal across cultures and subgroups, namely: neutral, happy, surprised, fear, angry, sad, and disgusted. Main objective of my research is to build a CNN based deep learning model capable of deriving these emotion of a person through pictures of his/her face.Given a picture of a person this model will automatically determine various probabilities of above mentioned emotions and find the best emotion that fits the picture. I analyzed the situations where the model might fail to recognise emotion correctly.Humans are well-trained in reading the emotions of others. So the main objective of this research is **Can a computer interpret facial expression close to human level accuracy ?**

Datasets and Inputs

For emotion recognition ,several datasets are available for research,varying from a few hundred high resolution photos to tens of thousands of smaller images. Most famous publicly available datasets are the **Facial Expression Recognition Challenge (FERC -2013)** [8],Extended Cohn-Kanade (CK+)[9]. In CK+ the facial expressions are posed (i.e 'clean') as all images are captured in controlled environment in a lab, while in FERC-2013 set shows emotions 'in the wild'. This makes the pictures from the FERC-2013 set harder to interpret,but given larger size of the dataset,the diversity can be beneficial for the robustness of the model. Because of this reason for my work I used FERC-2013 dataset.

FERC-2013 dataset is available in kaggle for public research purpose. This dataset has around 35000 low resolution images and corresponding emotion. All images are grayscale and have a resolution of 48 by 48 pixels.For my model I used these image values matrix as the inputs to my model.

Metrics

The most important metrics I considered for this research is **accuracy** of the model.

Here is the formula to calculate the accuracy from my model.
Given ,

$p_{ij} \in (0,1) : \sum_j p_{ij} = 1 \;\forall\; i,j$ is the estimated prediction by our model.

y : sample label (one-hot vector)
I will calculate,
$p\_best_i = argmax_j\, p_{ij}$

Accuracy $= \sum_i \left\{ 1 : p\_best_i = y_i \,, 0 : otherwise \right\}$

Where, i indexes samples/observations and j indexes classes

This is basically the number of correct predictions with respect to actual labeled emotion.We can only considered a model as good if it gives high level of accuracy . I used separate training and test sets for my research. My model is trained on training dataset and final reported accuracy was calculated on test dataset,to check if our model is generalized well enough.The human accuracy on FERC-2013 dataset is around 65.5% [9]. So using accuracy as an metrics we can have a very good quantitative comparison ,if my model is acceptable.

In addition to that, **Confusion matrix** for true emotions and prediction emotion counts will be generated . This matrix will be analysed to get idea of how the model is performing on different emotions. Confusion matrix will show us how the model is performing with respect to individual labels,so that we can have a broader idea about the performance of the model.

II. Analysis

Data Exploration

As mentioned earlier,I used FERC-2013 dataset for the purpose of my research. This dataset contains 28709 Training samples, 3589 samples categorized as PublicTest and 3589 samples as PrivateTest .For my work ,I merged PublicTest and PrivateTest samples to create a single testing dataset containing 7178 samples.

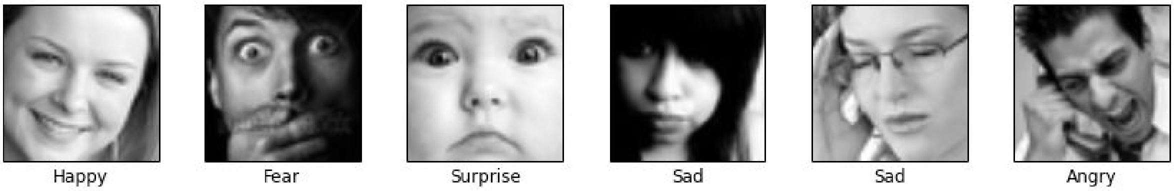


Figure 1:sample images

Each sample contains a 48*48 pixels image in grayscale as shown in Figure 1 above and a label containing corresponding emotion. As we can observe the images are not posed/lab controlled. So this dataset will help us in generalising the model well.

Exploratory Visualization

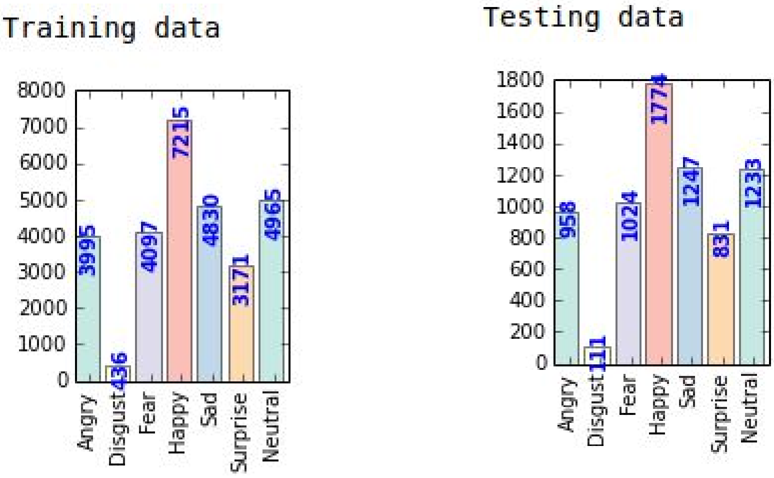


Figure 2: emotion distribution

I created bar chart to visualize number of samples available for each individual emotions as shown in figure 2 above. As we can see samples available for emotion **Disgust** is very less as compared to all other emotions. So my model was not able to predict this class well enough. Because of this reason I merged Disgust class with Angry class to create a common class **Angry**.

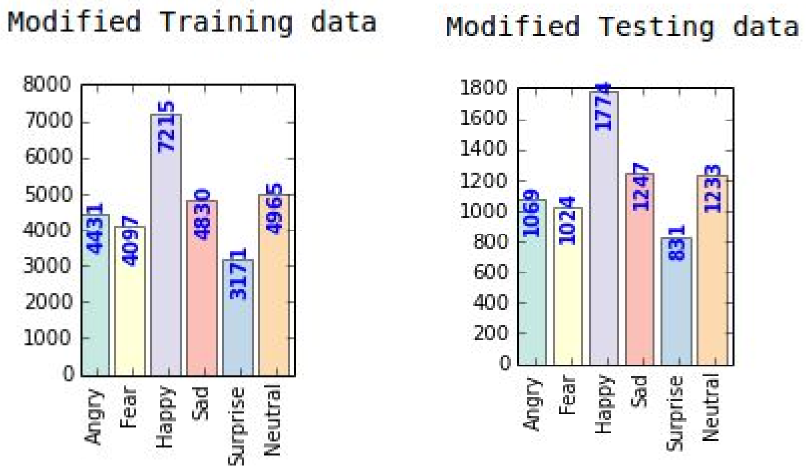


Figure 3: emotion distribution after merging Disgust and Angry class

So after merging our final data distribution is as shown in figure 3 above.

Algorithms and Techniques

For this research I experimented and studied previous work on various combinations of Convolutional,Pooling and feed forward layers for emotion recognition task. Previous research states that with limited data,network with depth of 5 / 6 performs much better than deeper networks[11] . After some research and experiment I finalised to create a network inspired from AlexNet [12].

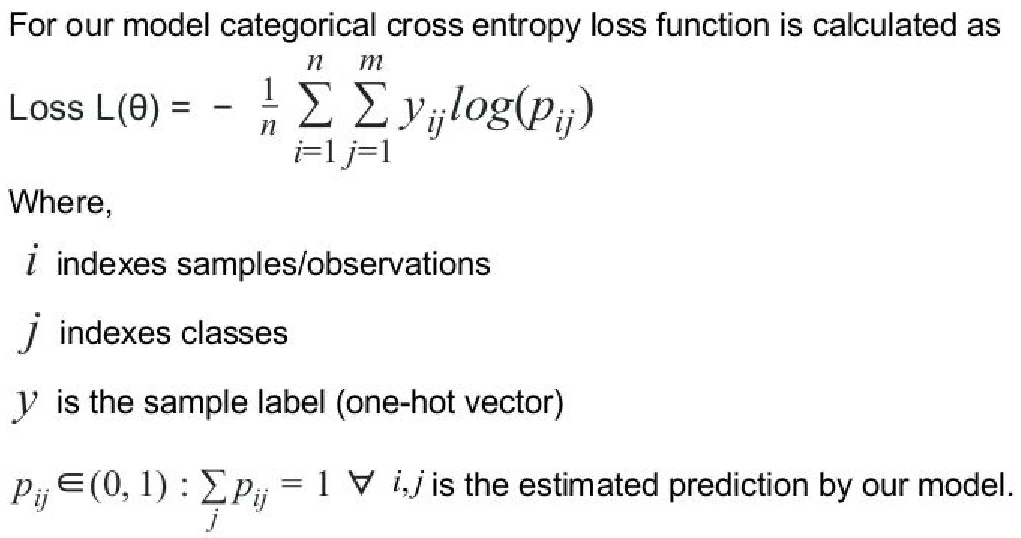
While designing the final convolutional network ,There are number of different design decisions were taken and parameters were selected and tuned properly.There are explained below.

Structure of the model: It is the most important design decision for creating a good model. While deep neural networks are capable of learning very complex features from the data,it requires a lot of training data and computation time to generalise very well. So considering all this a model with 5 / 6 layers is planned. 2D convolution layers are used directly on images to extract features from it. Max pooling layer is used after Convolution to reduce the dimensionality of the intermediate data,which in turn helped in reducing the number of model parameters.In the end multiple feed forward layers are used to learn some more complex features from the data and to have some overall interpretation from the image. A final architecture of the model is shown in the Implementation section below.

Activation Function: The choice of activation function introduces the nonlinearity into the model that is required to let the network model arbitrary functions. Rectified Linear Units (ReLUs),a recent development is shown to perform better than many of the traditional activation functions. Alexnet also used ReLU as activation functions for the feed forward layers.

Dropout: using a dropout layer is a powerful way to prevent overfitting a model by randomly setting a fixed percentage of output to zero during training . This builds robustness into the model by forcing it to build redundant representation because it cannot count on presence of any given neuron to form its prediction. In my network I used one dropout layer before the first feed forward layer.

Loss Function: To train a model we need a loss or objective function. For my model I used **Categorical cross entropy** loss method as this is one of the most popular and efficient loss function for multilabel classification tasks.



For our model categorical cross entropy loss function is calculated as

$$\text{Loss } L(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} log(p_{ij})$$

Where,

i indexes samples/observations

j indexes classes

y is the sample label (one-hot vector)

$p_{ij} \in (0, 1) : \sum_{j} p_{ij} = 1 \ \forall \ i,j$ is the estimated prediction by our model.

Optimizer: In my work I used Adagrad optimizer for optimizing the objective functio. Adagrad is an adaptive learning rate method originally proposed by Duchi et al [13].Because of it's quick learning ability I used this optimizer in my work which trained my model pretty well in limited computing environment.

Batch Size: The choice of batch size is decided considering both the efficiencies gained by training on multiple inputs at the same time with the requirement of holding all concurrent data within available processing memory,in my case GPU memory. In my work I used a batch size of 128,which is very common in current research.

Benchmark

For my benchmark model,I implemented a softmax classifier using features from a single convolutional layer. The architecture was one convolutional layer, followed by one fully connected layer, and a final softmax layer. The initial baseline did not use any regularization, dropout. A visual representation is shown below.

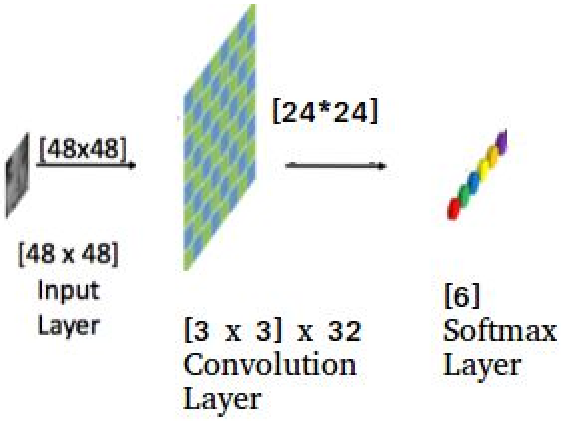


Figure 4:Benchmark model

For this benchmark model ,we got an accuracy of 40% on test data. We will use this as an reference to analyse our final model.I also considered human accuracy on the dataset as a reference parameter.

III. Methodology

Data Preprocessing

As explained earlier I merged disgust and Angry class into single class Angry. So the label for Samples containing Disgust has been changed to Angry.

My Convolutional model will directly learn from the image so I do not need any preprocessing on data.

As I used a deep network in my model,so amount of training data required to train the model is very high to overcome the problem of overfitting. Number of samples available in FERC-2013 dataset is very less in comparison to the sample we require. To overcome this issue I implemented combination of various Data augmentation technique ,which generates new data . For data augmentation I used,rotation ,Horizontal and vertical shift,Different zoom levels,horizontal flip.

Implementation

After following previous research and some experiments on various combinations of network layers I build a model with 3 Convolutional layers and 2 feed forward layers. In the end the output probability for each label is calculated using a softmax layer. A graphical diagram for my model is provided below in Figure 5.

First the weights are initialized using **glorot_uniform_initializer** [14], which is the default initialization technique in Tensorflow. Then loss for each sample is calculated and the loss is backpropagated using Adagrad optimizer. For training I started with an learning rate of 0.001. Learning rate is modified automatically by adagrad optimizer .Performance of the model is analysed throughout the training process to determine its saturation point.

Model is programmed using Keras 2,Deep learning library and Tensorflow was used as a backend to keras. I trained the model using parallel computing from NVIDIA GPU using CUDA library.

I trained the model until there is not any improvement in the accuracy on the test set for 30 consecutive epochs. Then I saved the weights into a file. Model as well as weights were used for prediction of test images and it can be used for detection for any image of face.

Refinement

Initially I trained the model with all seven emotions,but my models performed very poorly for disgusting class. So i merged it with angry class as both of the emotions are almost comparable. After doing this my model performed much better.

I also tried with much deeper CNNs,but there were not much difference on accuracy of the model ,while taking a lot of time to process. After trying various combination of layers I have finalised on the model shown below,which was seemed to be learning the model well with comparatively lesser processing time.

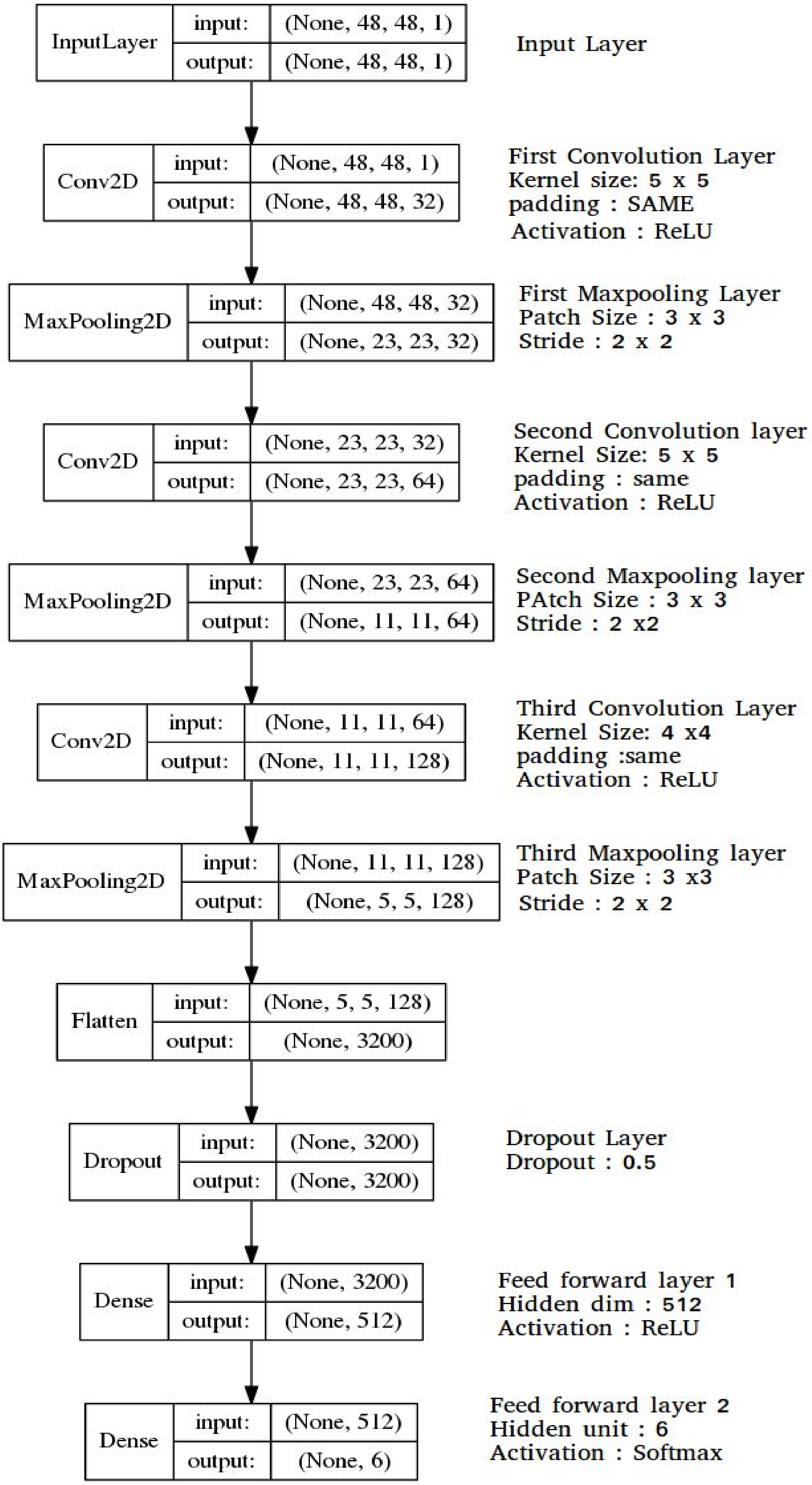


Figure 5 : final model

IV. Results

Model Evaluation and Validation

After training the model up to its saturation point ,we then finally freezed the weights. We calculated the final Test set accuracy.

	Accuracy
Human Accuracy	65.5%
Benchmark model	40.0%
CNN Model	64.1%

Table 1: Model accuracy

As we can observe from the above result that ,our model performed really well as compared to the benchmark model.Also the accuracy score is much close to that of human accuracy. Hence this model is good enough to be used for practical purpose.

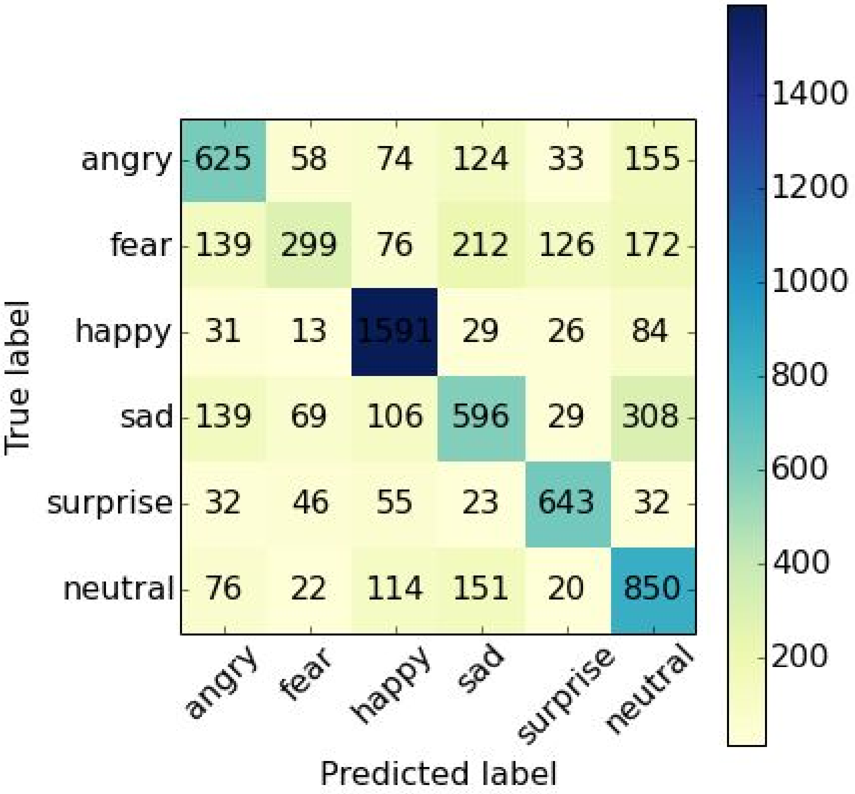


Figure 6: Confusion matrix of final model

Confusion matrix for the final model on test data is shown above in figure 6. As we can see our model was able to predict happy emotion pretty well with almost 90% accuracy. As happy emotion has the maximum number of training data,hence we can infer this high accuracy is related to that. It is also able to recognise other emotions except fear with some good accuracy level.Confusion matrix shown above also implies the ambiguity of facial emotion recognition.Also we can observe from the confusion matrix that maximum number of incorrectly predicted emotion is neutral,which is an important observation because some face structure of people influence emotion recognition a lot,so until it is clear which emotion the face represent it is safe for the model to consider it as neutral,as predicting it as some other expression might have some negative impact on real world application of FER.

For independent emotions accuracy level is as shown in the table below.

Emotion	Accuracy
Angry	58.46%
Fear	29.19%
Happy	89.68%
Sad	47.79%
Surprise	77.37%
Neutral	68.93%

Table 2: Accuracy for emotions

Justification

My model worked well in detecting some of the emotions,but it failed in some of the cases.

As we have seen from table 2 ,accuracy for fear is very less. Below I have shown some of the cases where the model was not able to predict the correct emotion Fear.

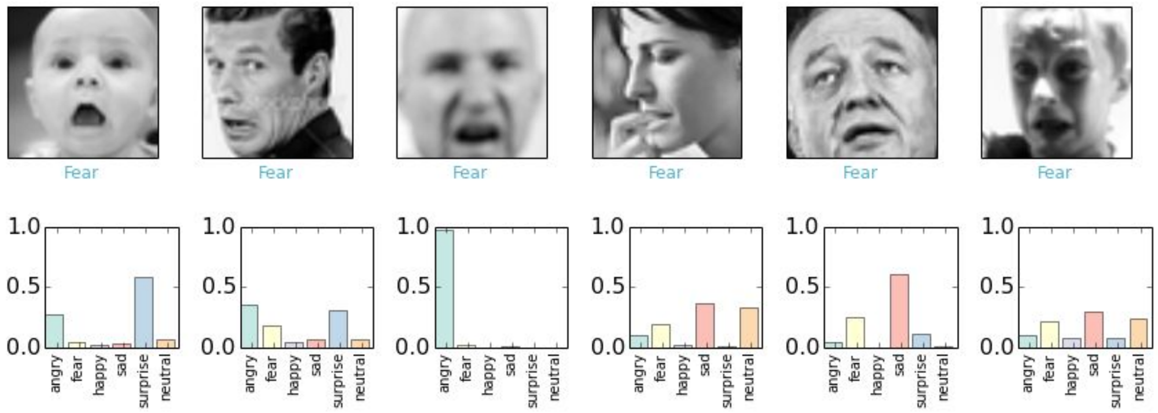


Figure 7: Wrong predictions for fear emotion

From the above image figure 7 ,as we can see,it is difficult for a human being to conclude that all images represent fear. Even the predictions somewhat look justifiable. For example the third image seems to represent angry,but it is labeled as Fear.

I have also shown some more images below where the model failed to predict correct label.Here we can observe that in most of the cases even though the model did not predict well but the second best prediction is the correct label.We can take advantage of this information while designing a real world application,where there can be some other model to confirm between possible alternatives.

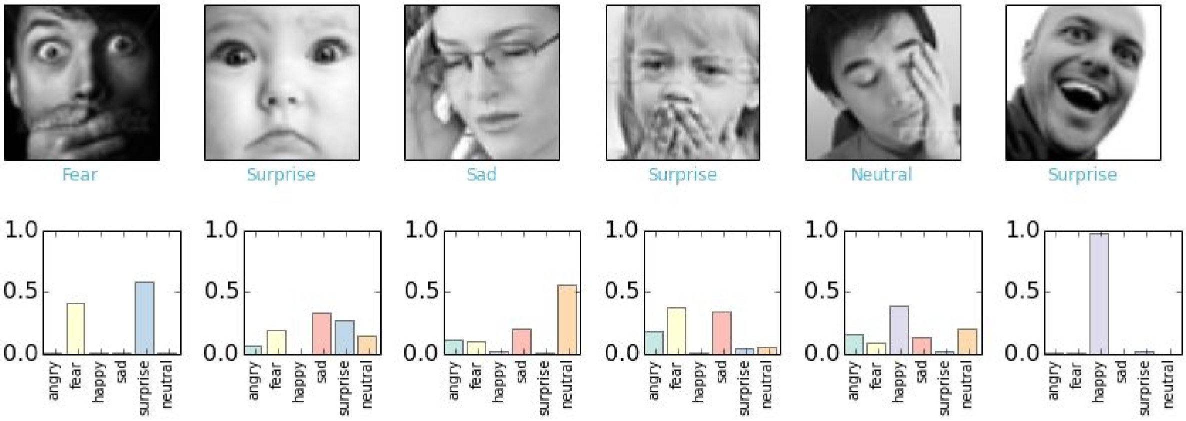


Figure 8: Wrong predictions

So from above justifications we can conclude that the model did a great job in predicting emotions similar to what a human might have done.

V. Conclusion

Free-Form Visualization

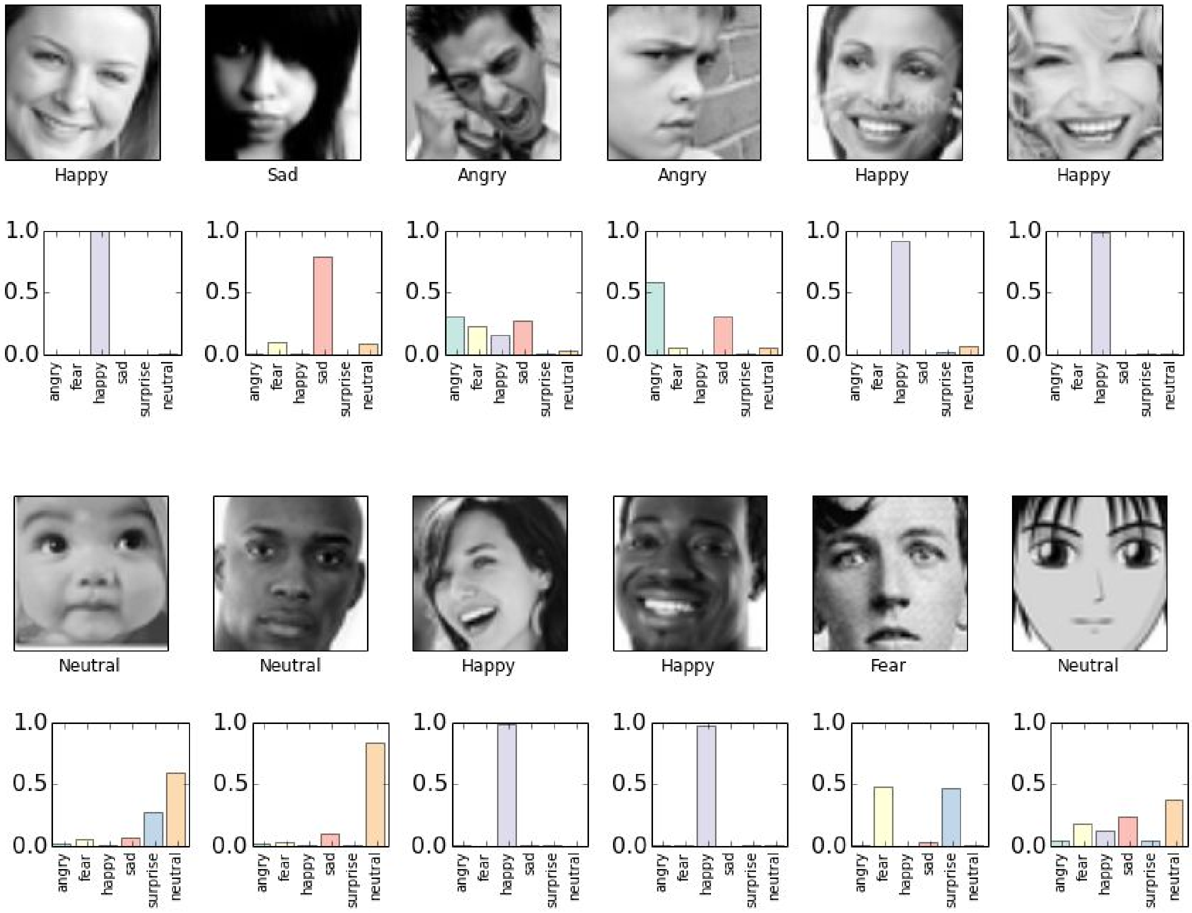


Figure 9: Correct predictions

In the figure above I have shown the cases where the model predicted the correct emotion. As we can observe the model is very confident when the face shows clear sign of happy emotion. But it is somewhat confused when the image of the face is a bit ambiguous. From this it is clear that the model has learnt to recognise emotion upto some extent.

Reflection

I have seen convolutional neural network working very well for image classification tasks.However ,at the starting of the project,I was skeptical if this will be able to recognise emotions,which seemed to be too complex for a machine to learn. But deep learning is made to learn complex things.Also some research in this domain provided me confidence to approach this with Convolutional neural networks.But the final result achieved is pretty satisfying for me.Even though the accuracy number the model achieved does not look very great,but if we consider the complexity of the task of emotion recognition and the human accuracy on FERC-2013 ,then the accuracy can be considered pretty reasonable. Even humans can not detect emotion correctly all the time.

One of the challenging task in using deep learning algorithms is the resources required for training . But the availability of tensorflow saved a lot of effort in making the model trainable using a GPU,which reduced the training time very much. Also keras deep learning framework reduced the coding effort a lot and made it easy to try out different combinations of CNN,Pooling and dense layers.

I also used the model to predict emotions captured from a webcam/mobile camera.I used opencv to detect face portion from the image. The model was able to predict my emotions. I was very satisfied with this result.I also tested with some images downloaded from internet and it worked perfectly .

Improvement

As amount of data is a crucial factor in deep learning,hence the performance of the model can be improved with the help of additional data. Even though I used data augmentation for generating more data, but a model trained on more real data is expected to perform better.

We can also let the model train for more time,as it the result might improve. Only after running the model for ample amount of time,we can get the real . Some researchers also consider using adagrad optimization algorithm to be an aggressive approach. So we can experiment on other optimization techniques,if it can improve the learning.

With more amount of data ,we can also make a more deeper model with more network parameters,so that the model can learn even more complex features from the data.

References

[1] C. Shan, S. Gong, and P. W. McOwan, “Facial expression recognition based on Local Binary Patterns: A comprehensive study,” Image Vis. Comput., vol. 27, no. 6, pp. 803–816, May 2009.
[2] J. Whitehill and C. W. Omlin, “Haar features for FACS AU recognition,” in 7th International Conference on Automatic Face and Gesture Recognition (FGR06), 2006, p. 5 pp.-pp.101.

[3] S. Berretti, A. D. Bimbo, P. Pala, B. B. Amor, and M. Daoudi, “A Set of Selected SIFT Features for 3D Facial Expression Recognition,” in 2010 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 4125–4128.

[4] C. Liu and H. Wechsler, “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition,” IEEE Trans. Image Process., vol. 11, no. 4, pp. 467–476, Apr. 2002.

[5] Z. Wang and Z. Ying, “Facial Expression Recognition Based on Local Phase Quantization and Sparse Represe

[6] B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee, “Hierarchical committee of deep convolutional neural networks for robust facial expression recognition,” J. Multimodal User Interfaces, vol. 10, no. 2, pp. 173–189, Jan. 2016

[7] Kaggle. Challenges in representation learning:Facial expression recognition challenge ,2013

[8] P.Lucey, J.F Cohn, T. Kanade: The extended chn-kanade dataset(CK+):A complete dataset for action unit and emotion specified expression. Im Computer vision and pattern recognition workshop(CVPRW),2010 IEEE Computer society conference on page 94-101,IEEE 2010.

[9] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. 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, “Challenges in representation learning: A report on three machine learning contests,” Neural Networks, vol. 64, pp. 59–63, 2015.

[10] P. Ekman and W.V. Friesen Constants across cultures in the face and emotion. Journal of personality and social psychology,17(2):124,1971

[11]C.Pramerdorfer,M.Kampel Facial Expression Recognition using Convolutional Neural Networks: State of the Art

[12] A. Krizhevsky,I.Sutskever, G E. Hinton ImageNet Classification with Deep Convolutional Neural Networks

[13] John Duchi, Elad Hazan, Yoram Singer; Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,12(Jul):2121−2159, 2011.

[14] Xavier Glorot,Yoshua Bengio : Understanding the difficulty of training deep feedforward neural networks