FACIAL EXPRESSION IDENTIFICATION

CHIEN-CHIEH, HU (UIN: 670782410)
DURGASHANKAR M P (UIN: 671192091)
KARTHIK RANGAPPA (UIN: 677923220)
LAAVANYA GANESH (UIN: 654324917)
SAVITHA TAMRAPARNI-VITTAL (UIN: 662405004)
VINEET KUMAR SINGH (UIN: 654489295)

IDS 594-
Machine Learning Applications with Python
Final Project Report

INTRODUCTION

The major emotions which have been categorized universally are anger, disgust, fear, happiness, sadness, neutral and surprise. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind.
Human facial recognition has become a widely-researched topic in recent times. In 2013, International Conference on Machine Learning, Facial Emotion Recognition challenge was first introduced.
It has become an important aspect to understand how well computers can do a job in recognizing human emotions.

We can measure the effects that content and services have on the audience/users through an easy and low-cost procedure. For example, retailers may use these metrics to evaluate customer interest. Healthcare providers can provide better service by using additional information about patents' emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content.

The objective of this project is to classify human emotions into these discrete categories. Presently, there is 75% accuracy for unseen data achieved from the deep learning models. The highest accuracy has been achieved through ensemble of 8 deep nets.

The project idea is to gain understanding from such existing models and infuse more approaches with the CNNs to get better accuracy on unseen data.
Another scope of project lies in tackling images with side shots, partial shots of faces and different rotations and scaling required as when compared to frontal shots of images.

DATASET DESCRIPTION

The Dataset used for this project has been taken from Kaggle.
The data consists of 48x48 pixel gray scale images of faces. The total dataset has 35,899 examples.

The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise).
The Kaggle dataset has another category 6=Neutral in case the face does not match any of the other emotions.

The training set consists of 28,709 examples. The training dataset contains two columns emotion and pixels. The emotion column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image.
The pixel column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. The remaining 7,179 examples were labelled as test. These remaining examples were then equally divided into validation and test data to increase the accuracy of the final chosen model.

Link for dataset: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

CHALLENGES FACED:

- **Unbalanced data:** Through an exploratory analysis of the data, it can be seen that Happy has the maximum number of labels (7,215) in the dataset whereas, disgust has a very low number (113). Hence, the dataset looks like a very unbalanced one.

Figure 1: Overview of Distribution of Categories in the dataset

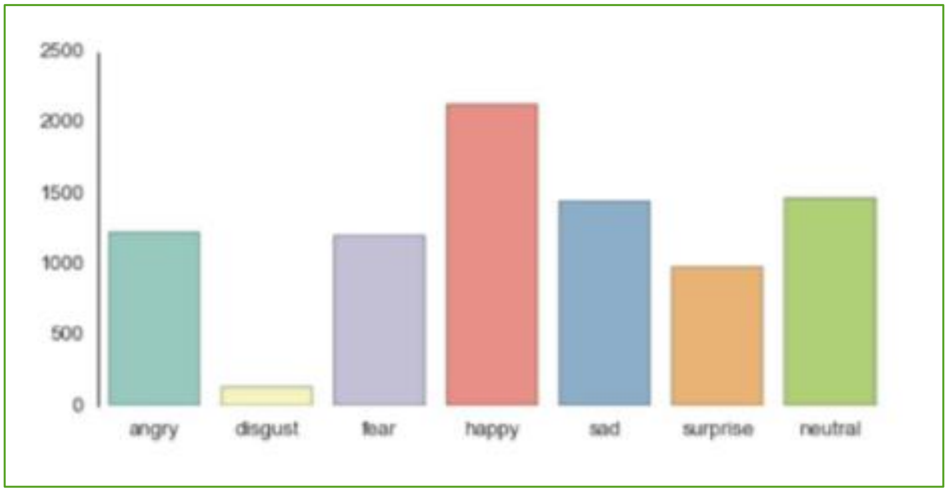


Table 1: Overview of Distribution of Categories in the dataset

Emotions	Training Set Count
Angry	4318
Disgust	113
Fear	4097
Happy	7215
Sad	4830
Surprise	3171
Neutral	4965

- **Computational Power:** Both the models used were computationally expensive given the fact that the data dealt with was image data. It took almost 48 hours to run each model.

- **Class Variation:** Certain facial emotions are difficult for humans to detect. For example, the subtle difference between sad and neutral is difficult even by human discretion.

DATA PRE-PROCESSING

The data pre- processing yielded a significant increase in validation and test accuracy in our model. The two methods used for data pre-processing were:

- **Normalization to ensure centering of data (BOTH FOR SVM & CNN)**

  Each image pixel was subtracted from its' individual mean.

- **Data Augmentation (ONLY FOR CNN)**

  a. Random Flipping of Images was achieved using Image Data Generator
  b. 20% of zooming was done on images to focus on more on specific muscle movements of face during each expression. Ex- frown in forehead when angry, curving of lips when happy etc.

MODEL ALGORITHMS

The following two models were used to classify the emotions into 7 discrete categories:

- **SVM:**

  In machine learning, support vector machines (SVMs, also support vector network) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

  A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

  SVMs can be used to solve various real world problems:

  a. SVMs are helpful in text and hypertext categorization
  b. Classification of images
  c. Hand-written characters can be recognized
  d. Biological and other sciences to classify proteins

- **Convolutional Neural Nets (CNN)**

  A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers are either convolutional, pooling or fully connected. CNNs share weights in convolutional layers, which means that the same filter is used for each receptive field in the layer; this reduces memory footprint and improves performance.

  a. Convolutional

  Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli. Each convolutional neuron processes data only for its receptive field. Tiling allows CNNs to tolerate translation of the input image (e.g. translation, rotation, perspective distortion). Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. In other words, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using backpropagation.

  b. Pooling

  Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters at one layer into a single neuron in the next layer. For example, max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Another example is average pooling, which uses the average value from each of a cluster of neurons at the prior layer.

  c. Fully-connected

  Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP).

RESULTS FROM SEVERAL TRIALS

Best parameters for SVM model used:

- **SVM**

  Principal Component Analysis (PCA) was used for dimensionality reduction in order to transform a 48*48 vector into 150*1 eigen faces. Grid Search was used to tune model parameters. Best estimator found by grid search:

  SVC (C=1000.0, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape =None, degree=3, gamma=0.01, kernel='rbf', max_iter=-1, probability=False, random_state =None, shrinking=True, tol=0.001, verbose=False)

- **CNN**

  a. Number of CNN layers – 2, 3 & 4
  b. Number of FC layers – 1 & 2
  c. Dropout- With 0.2 Dropout or Without Dropout
  d. Learning Rate – Constant & Adaptive Scheduler

Table 2: Accuracies from those models based on changing different parameters

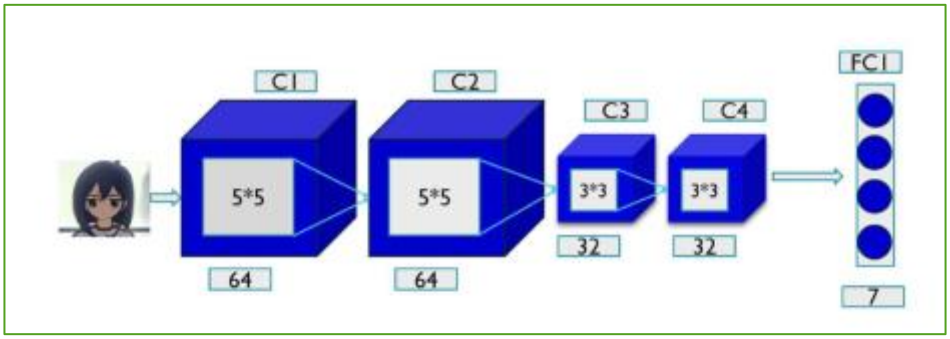
Architecture	Train Acc	Validation Acc	Test Acc	Findings
SVM	99.8%	N/A	48.9%	Very low in testing data (overfitting)
2 CNNs with 1 FC layer	99.6%	59%	59.6%	High training accuracy and low test accuracy indicates strong over-fitting in the model
3 CNNs with 1 FC layer	88.7%	63%	64%	The model with 3 layers definitely reduced overfitting & showed remarkable improvement in test accuracy (5%)
3 CNNs with 2 FC layers	77%	64%	65.4%	The model with one more FC layers was better than the previous one by 1%
4 CNNs with 1 FC layer	70%	65.5%	65.6%	Increasing one more CNN layer increased the model accuracy by 0.2%

The final model was chosen based on test accuracy which was 4 CNNs with 1 FC layer, with test accuracy 65.6%.

BEST MODEL ARCHITECTURE

The input images are of dimension 1*48*48. The best model has been built by taking 4 CNN layers with 1 Fully Connected Layer. In all the CNN layers, activation function which has been used is ReLu. The initial two CNN layers have 64 filters of 5*5 dimension. Batch normalization has been used along with 2*2 strides in Max pooling with Zero Padding. The next two layers have 32 filters of 3*3 along with Max Pooling, Zero Padding and Batch Normalization. Additionally, Dropout=0.25 has been used in these two final layers of CNN. In the fully connected layer, dropout of 0.2, activation function of soft max and cross entropy loss.

Figure 2: CNN Best Model Architecture - 4 CNN, 1 FC

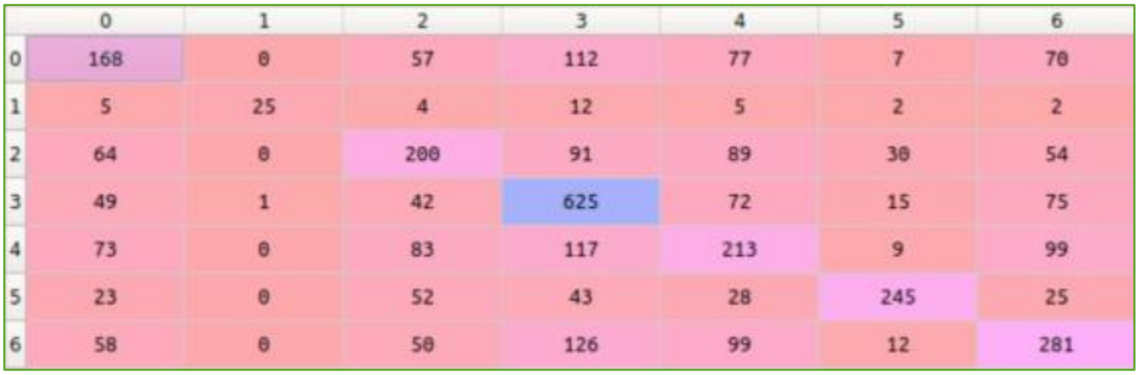


RESULTS BASED ON CONFUSION MATRICES

Figure 3: SVM Confusion Matrix

PREDICTED

ACTUAL

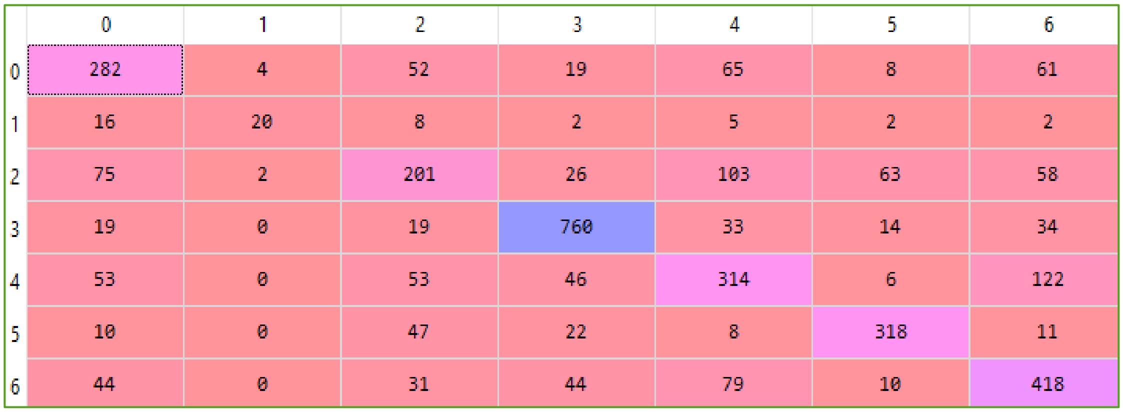


	0	1	2	3	4	5	6
0	168	0	57	112	77	7	70
1	5	25	4	12	5	2	2
2	64	0	200	91	89	30	54
3	49	1	42	625	72	15	75
4	73	0	83	117	213	9	99
5	23	0	52	43	28	245	25
6	58	0	50	126	99	12	281

Figure 4: CNN Confusion Matrix

PREDICTED

ACTUAL



	0	1	2	3	4	5	6
0	282	4	52	19	65	8	61
1	16	20	8	2	5	2	2
2	75	2	201	26	103	63	58
3	19	0	19	760	33	14	34
4	53	0	53	46	314	6	122
5	10	0	47	22	8	318	11
6	44	0	31	44	79	10	418

The matrix gives the counts of emotion predictions and some insights to the performance of the multi-class classification model.

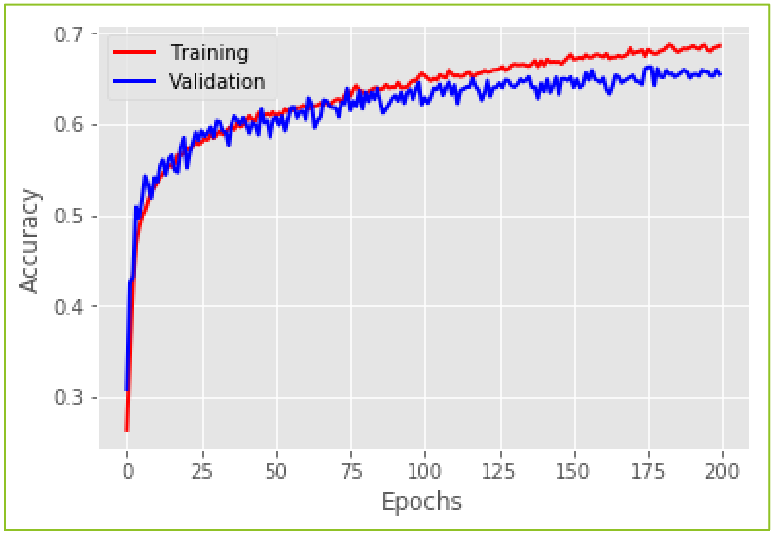
CNN: The model performs well on classifying positive emotions resulting in relatively high precision scores for happy and surprised. 760 happy emotions were correctly classified. Happy has a precision of 76.7% which could be explained by having the most examples (~7000) in the
training set. Interestingly, surprise also has high rate of accuracy with 318 faces being classified correctly. Sad gets misclassified with the next closest negative emotion fear. 103 classifications of sad class were in fear and 122 classifications of neutral class were in sad. This is a classic example of the challenge of class variation that was discussed earlier.
The model frequently misclassified angry, fear and neutral as sad. In addition, it is most confused when predicting sad and neutral faces because these two emotions are probably the least expressive (excluding crying faces).

SVM: SVM too performed well in classifying positive emotions. 625 happy emotions were correctly classied. But the number of false positives were higher as compared to CNN.

This is one of the reasons why CNN was chosen over SVM apart from other factors like better accuracies and less computationally expensive.

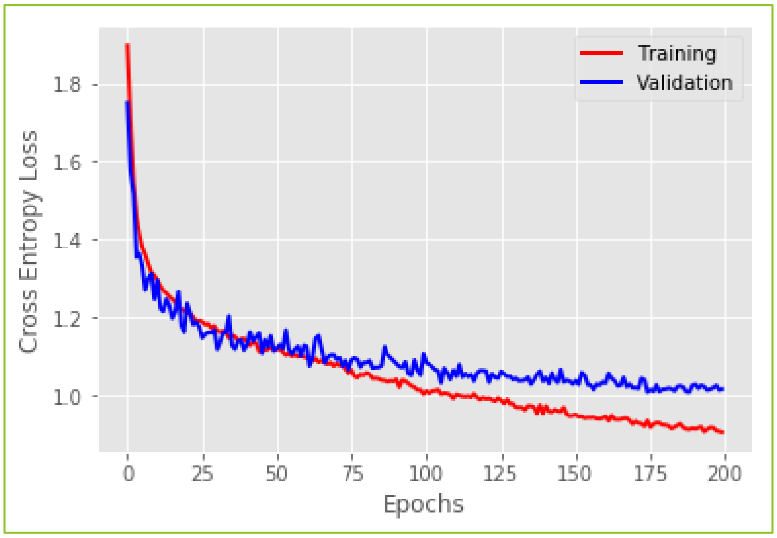
RESULTS: CNN TRAINING V/S VALIDATION

Figure 5: CNN TRAINING V/S VALIDATION: Accuracy



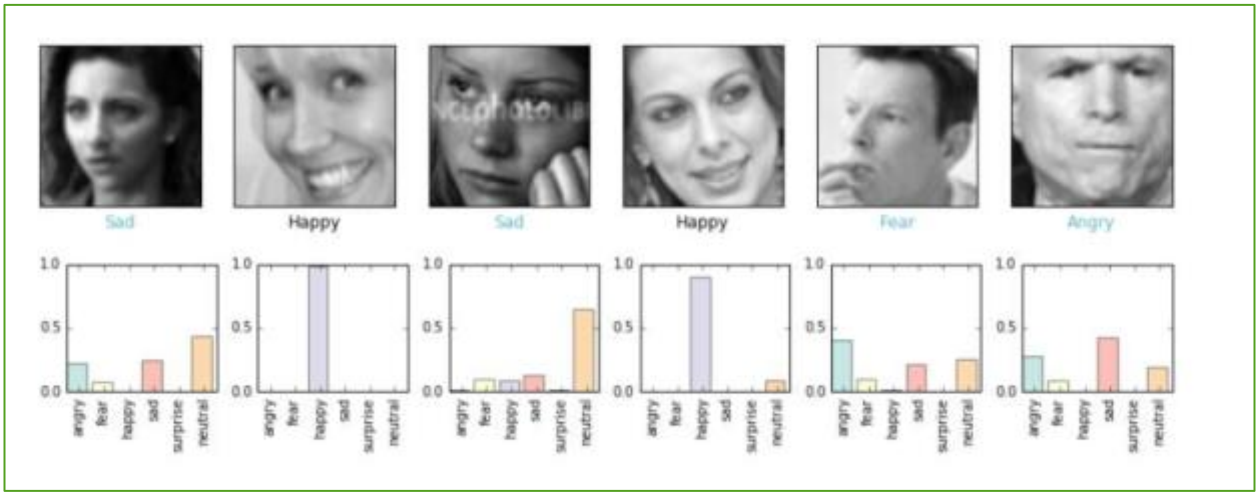
Training Accuracy and Validation improves with each epoch iteration.

Figure 6: CNN TRAINING V/S VALIDATION: Cross Entropy Loss



Cross Entropy Loss decreases with each epoch iteration.

Figure 5: Example of some of the faces which got correctly predicted



REFERENCES

- Ali Mollahosseini, David Chan, and Mohammad H. Mahoor
  “Going Deeper in Facial Expression Recognition using Deep Neural Networks”

- Peter Burkert, Felix Trier, Muhammad ZeshanAfzal, Andreas Dengel and Marcus Liwicki
  “DeXpression: Deep Convolutional Neural Network for Expression Recognition”

- YichuanTang, “Deep Learning using Linear Support Vector Machines”

- Christopher Pramerdorfer, Martin Kampel, “Facial Expression Recognition using Convolutional Neural Networks: State of the Art”

CONCLUSION

The final model accuracy could not reach the state of art accuracy achieved last year which was 75%.
The ensemble deep nets would have been the next future steps to improve the accuracy further.
But the model accuracy had improved remarkably after data augmentation and normalization as compared to first report results where no data pre-processing was done.
Although the accuracy is lower than 75% on closer inspection of misclassified classes it was observed, emotion is getting labeled to next best emotion category.

GROUP MEMBER CONTRIBUTION

GROUP MEMBER NAME	UIN	CONTRIBUTION
CHIEN-CHIEH, HU	670782410	SVM and report
DURGASHANKAR M P	671192091	CNN and report
KARTHIK RANGAPPA	677923220	SVM and presentation
LAAVANYA GANESH	654324917	CNN and report
SAVITHA TAMRAPARNI-VITTAL	662405004	SVM and presentation
VINEET KUMAR SINGH	654489295	CNN and report