Machine Learning Engineer Nanodegree

Capstone Project

Peter Raafat

Date

I. Definition

Project Overview

Human beings convey emotions verbally and non-verbally. Facial expressions are one of the methods human beings convey emotions non-verbally. In the field of computer vision facial expression detection has been a topic of interest and a challenging problem. It can be used as a method for lie detection [1][2]. In clinical applications it can be used as a method to assess the patient pain regardless of their age [3]. In the field of automotive detection of human facial expressions can prove of high importance for smart cars, in case of high weather temperature and elevated state of non-comfort adjusting the air conditioner temperature might be necessary, or in case of fear, alarming the control system of a possible accident might help avoid the accident or if a presence of a human threat outside the vehicle, locking the doors and closing the windows might be appropriate. In order to achieve human facial expressions a convolutional neural network classifier will be trained on a labeled dataset containing faces of human beings and their respective way of showing emotions.

Problem Statement

Detection of human facial expressions can be a challenging task for a computer. In this project I would like to analyze the possibility of using machine learning techniques particularly CNNs in order to solve the problem of finding what emotion the human being is experiencing from his facial patterns. The basic human emotions can be categorized into neutral, happiness, anger, disgust, fear, sadness and surprise. The CNN classifier will be trained on the JAFFE and extended cohn-kanade datasets after a sequence of pre-processing is applied to the raw images.

Metrics

The model is evaluated using the categorical cross-entropy as loss function which is useful for multiclass classification. It is a measure of the how close a probability decided by the model is to the real target. It also proves more useful than accuracy for weight adjustments when the dataset is unbalanced and because it is a differentiable function so it can be minimized using gradient descent. The categorical cross-entropy function is calculated for the validation data and it can be calculated using the following formula:

$$CE = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Where y are the true labels and $\hat{y}$ is probability that the point belongs to this label.

Accuracy is also used to evaluate the final model performance on the testing dataset and to compare its performance to the benchmark model which also uses accuracy as an evaluation metric. Accuracy is a simple metric to visualize and it works with classification problems. It is a measure of the number of points classified correctly divided by the total number of points in the dataset. It can be calculated using the following formula:

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP = true positives, TN = true negatives, FP = false positives and FN = false negatives.

II. Analysis

Datasets and Inputs

For this problem the Japanese Female Facial Expression (JAFFE) database is a great candidate as the input data. The database contains 213 images in total representing 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. The images are all in grey scale and in size of 256x256 pixels. The number of classes are as follows:

Class	Quantity
Neutral	30
Happy	31
Sad	31
Surprised	30
Angry	30
Disgust	29
Fear	32

The database is available free of charge on the following link http://www.kasrl.org/jaffe.html

The extended cohn-kanade database contains 593 sequences across 123 subjects. Each of the sequences contains images from onset (neutral frame) to peak expression (last frame). The peak frame was reliably FACS coded for facial action units (AUs) [5]. Unfortunately, only 327 of the 593 come with an emotion label. The images are 640x490 pixels, mostly consisting of grayscale images but some are RGB. The number images of each class at the peak of the emotion are as follows:

Class	Quantity
Contempt	18
Happy	69
Sad	28
Surprised	83
Angry	45
Disgust	59
Fear	25

The dataset classes are highly unbalanced as we can see surprised class has 83 images and only 18 images make up the contempt class. The dataset is available free of charge at the following link: http://www.consortium.ri.cmu.edu/ckagree/

The total number of images in each class after extracting frames from the sequences at which emotion starts to build up is as follows:

Class	Quantity
Contempt	142
Happy	755
Sad	306
Surprised	765
Angry	566
Disgust	507
Fear	303

Exploratory Visualizations

In this section I would like to refer to some facial features that define how we express emotions and how drastically they can differ from a person to another.

In figures 1,2 and 3 we can see three subjects expressing happiness. A common feature in the three subjects is an increase in the mouth width compared to what they would normally be in neutral state, however figure 1 and 2 show a lift in the eyebrows, the subject in figure 2 shows squinting of eyes. Subjects in Figure 2 and 3 show nasolabial folds. Subjects in figures 1 and 2 show their teeth while subject in figure 3 does not.

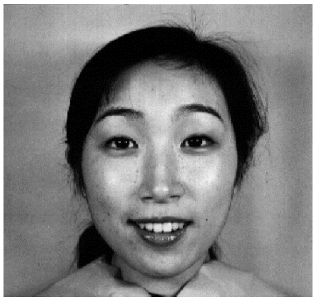


Figure 3

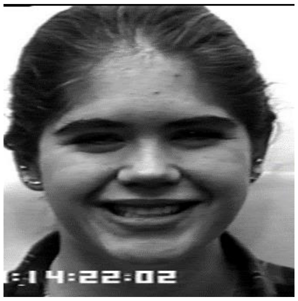


Figure 1

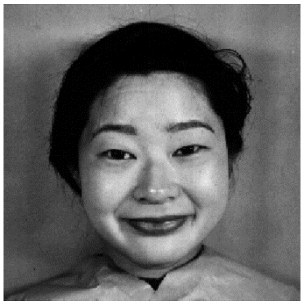


Figure 2

Figures 4, 5 and 6 shows three subjects expressing surprise. Extremely difficult for the human eye to detect any common features at the mouth, eye, eyebrows or cheeks. Subjects in figures 5 and 6 shows a wide opening of mouth while subject in figure 1 has her mouth tightly closed. Subjects in figures 4 and 5 show wide opening of eyes and lift in eyebrows while subject in figure 6 does not.

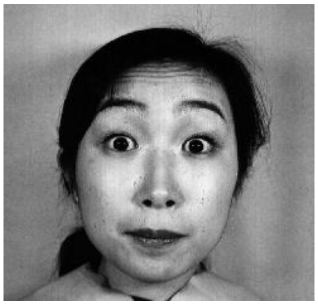


Figure 4

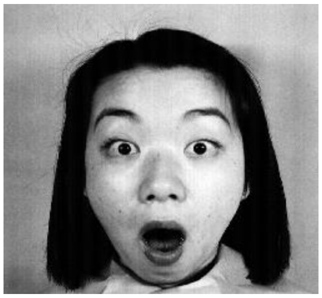


Figure 5

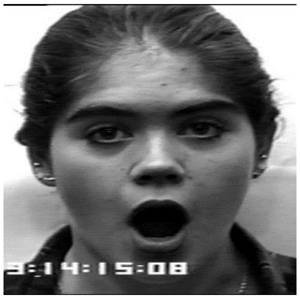


Figure 6

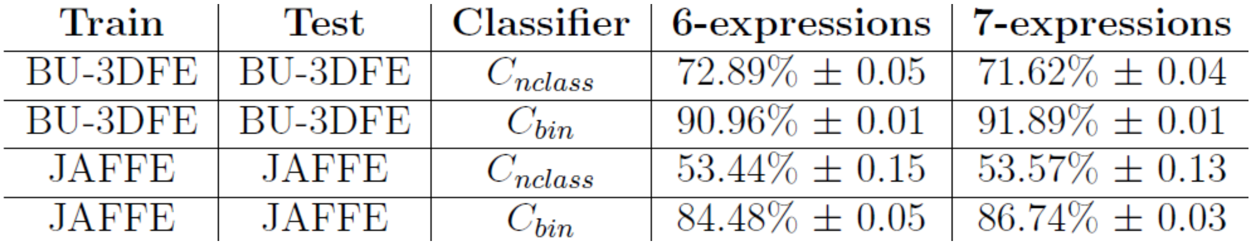
In conclusion the facial features that describe how we express emotions can be found in the eyes, eyebs, mouth and cheeks areas.

Algorithms and techniques

For the given problem a CNN classifier is used as it is one of the strongest techniques when it comes to image classification. The downside of using a CNN classifier is that a large number of data samples are needed to obtain good results, unfortunately the datasets used are small compared to the complexity of the problem. Two models were created and trained separately on the JAFFE dataset and the extended cohn-kanade dataset. The first model was trained and tested using the 213 JAFFE images with a train-test split of 0.2. The second model was trained and tested on 3344 image sequences that show emotion building up with a train-test split of 0.2. Random search was used to change the model architecture parameters. Every iteration a model was created using a random kernel size and initial number of filters, the initial number of filters was multiplied by two for each following convolutional layer. The learning rate which dictates how much the model weights change each backpropagation step and momentum which helps prevent getting stuck in a local optimum were left as the default values for the RMSProp optimizer. Each model was trained for 1000 epoch and batch size of 20.

Benchmark

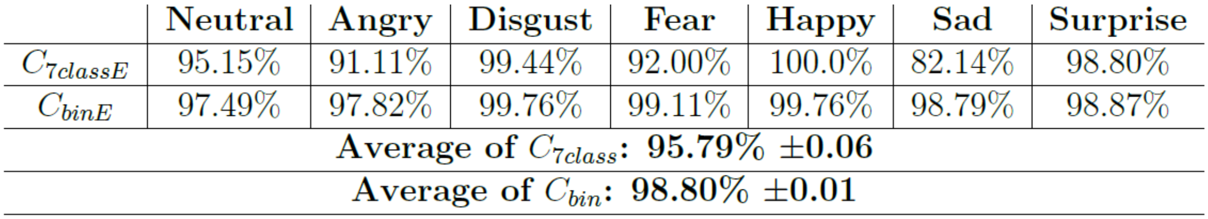
The benchmark model is that created by Lopes, A. T., de Aguiar, E., De Souza, A. F., & Oliveira-Santos, T.[4]. In their model they used a CNN classifier to predict human facial expressions using the JAFFE, extended cohn-kanade and BU-3DFE which unfortunately I could not obtain. Their approach also introduced two approaches, one that outputs a probability vector for all classes and the other consists of a model for each label where each model outputs a binary. The approach I chose was one model that outputs the probability vector for all labels. They also added rotation correction to remove any difference in camera angles or pose-specific motions, however this is a step I did not apply as I intendedly augmented the dataset with rotated and skewed images. Figure 8 shows the accuracy of their model that is trained and tested on JAFFE training dataset. The C_{nclass} is the accuracy obtained by the model that produces a single probability vector for all labels while C_{bin} is the average accuracy for all models that produce a binary output. For fair comparison the C_{nclass} will be used.



Train	Test	Classifier	6-expressions	7-expressions
BU-3DFE	BU-3DFE	C_{nclass}	72.89% ± 0.05	71.62% ± 0.04
BU-3DFE	BU-3DFE	C_{bin}	90.96% ± 0.01	91.89% ± 0.01
JAFFE	JAFFE	C_{nclass}	53.44% ± 0.15	53.57% ± 0.13
JAFFE	JAFFE	C_{bin}	84.48% ± 0.05	86.74% ± 0.03

Figure 8

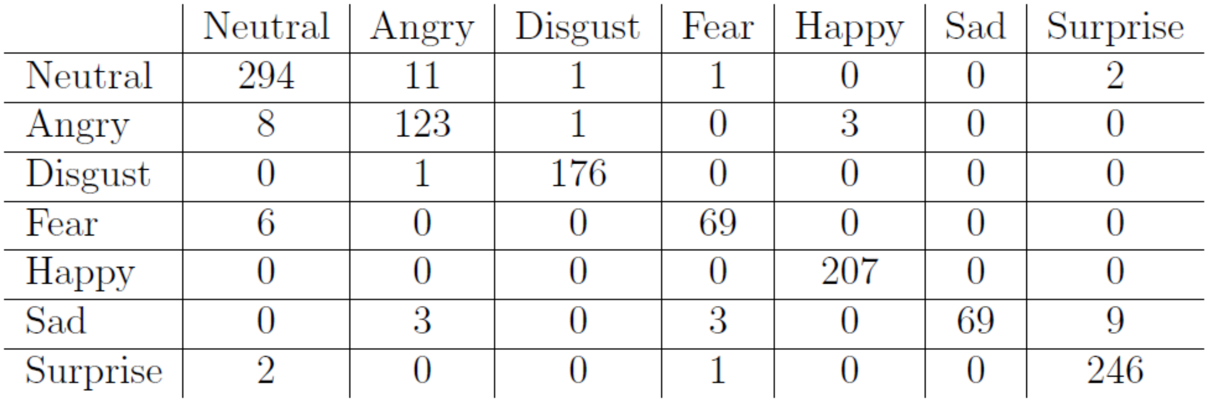
Figure 9 shows the accuracy of the model trained and tested on the cohn-kanade dataset, the C_{7class} is the accuracy of the model that was trained on the seven emotions and outputs a probability vector for all labels.



	Neutral	Angry	Disgust	Fear	Happy	Sad	Surprise
$C_{7classE}$	95.15%	91.11%	99.44%	92.00%	100.0%	82.14%	98.80%
C_{binE}	97.49%	97.82%	99.76%	99.11%	99.76%	98.79%	98.87%
Average of C_{7class}: 95.79% ±0.06							
Average of C_{bin}: 98.80% ±0.01							

Figure 9

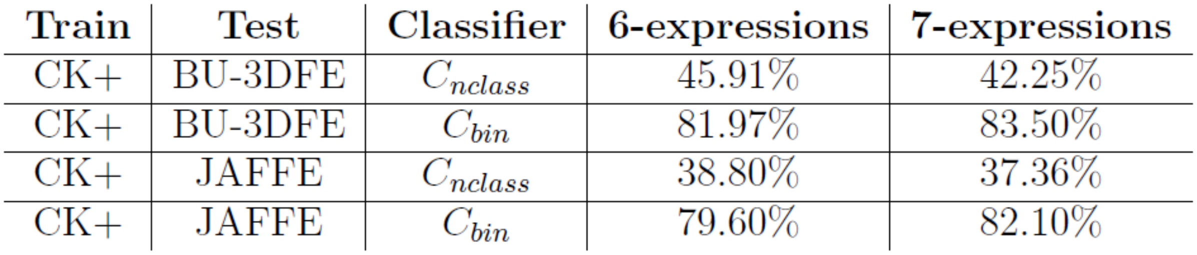
Figure 10 shows the confusion matrix for the previous model.



	Neutral	Angry	Disgust	Fear	Happy	Sad	Surprise
Neutral	294	11	1	1	0	0	2
Angry	8	123	1	0	3	0	0
Disgust	0	1	176	0	0	0	0
Fear	6	0	0	69	0	0	0
Happy	0	0	0	0	207	0	0
Sad	0	3	0	3	0	69	9
Surprise	2	0	0	1	0	0	246

Figure 10

Figure 11 shows the performance of the model trained on the cohn-kanade dataset when tested using the BU-3DFE and JAFFE dataset. Also for fair comparison the 7-expressions model is used as benchmark.



Train	Test	Classifier	6-expressions	7-expressions
CK+	BU-3DFE	C_{nclass}	45.91%	42.25%
CK+	BU-3DFE	C_{bin}	81.97%	83.50%
CK+	JAFFE	C_{nclass}	38.80%	37.36%
CK+	JAFFE	C_{bin}	79.60%	82.10%

Figure 11

III. Methodology

Data Preprocessing

The data preprocessing techniques used were to address a few problems. The first one is the very small number of unique data samples that makes up the datasets, the JAFFE dataset consists of 213 images and the partition of the dataset used of the cohn-kanade is 3344 images of which only 327 images are unique this a large amount of augmentation was needed. This also made the neural network more robust to noise or rotations in images. Also to reduce the number of unnecessary features, the image was cropped to the size of face and resized to a size of 197x197 pixels which is the minimum size that can be accepted by models like Resnet50, it proved more convenient to use the same size for all models. The following steps describe in details the data preprocessing.

- The datasets images and labels are loaded
- The datasets are split into training and testing datasets
- The labels are encoded using one-hot-encoding
- The datasets are augmented using random Gaussian noise, random rotation and random skew, where the JAFFE dataset was augmented to a total of 600 images and the cohn-kanade dataset was augmented to a total of 6000 images
- Using the openCV face detection algorithm the images were cropped to face and resized to a size of 197x197 pixels
- In case of the cohn-kanade dataset, RGB images were converted to grayscale
- The datasets were then normalized using intensity normalization and converted to 4D tensors

Implementation

The CNN architecture used is shown in figure 7.

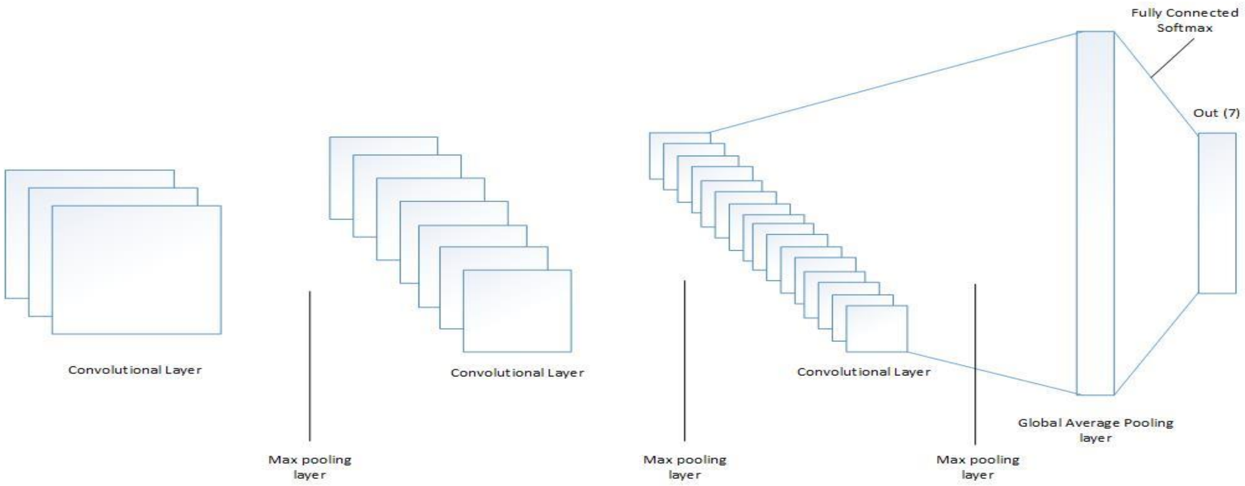


Figure 7

Random search was used to tune the model architecture. The model architecture parameters tuned were the number of filters in each convolutional layer and the kernel size which is constant in all convolutional layers. The following shows the pseudo code for model creation and training.

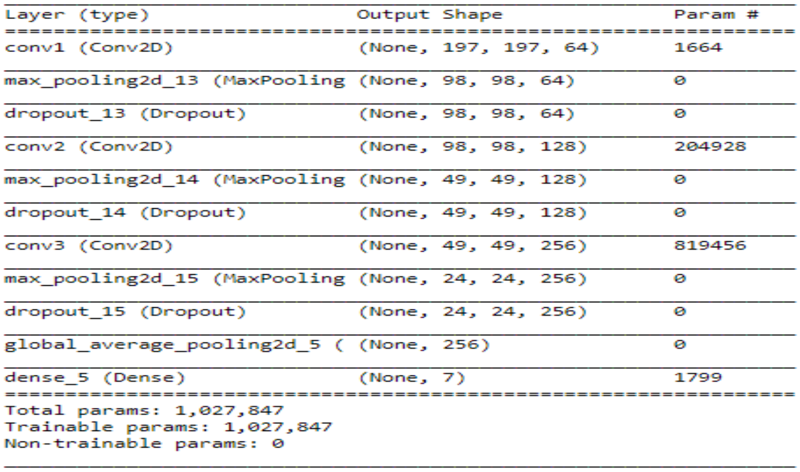
- For a number of iterations
- Pick a random kernel size and initial number of filters
- Create a model where the first convolutional layer consists of kernel size and initial number of filters, the second convolutional layer consists of the same kernel size but number of filters equal twice the number of initial filters, and the third convolutional layer consists on the same kernel size but number of filters equal four times the initial number of filters
- Train the model using the training dataset with validation split equal 0.33 batch size = 20 for 1000 epochs
- Check pointer is used to save model best weights and early stopping is used to avoid continuation of overfitting models, patience = 50 epochs
- Compare the model validation loss to the minimum validation loss so far and save the model if it scored a smaller minimum validation loss
- After the number of iterations load the best model and test it on the test dataset

  Transfer learning was also trained and tested on the JAFFE images and cohn-kanade images. For the JAFFE dataset Resnet50 was used and for the cohn-kanade dataset InceptionV3 was used both preloaded with imagenet weights.

The accuracy for each model was calculated using its respective testing dataset and the confusion matrix was also obtained. The model trained using the cohn-kanade images was also tested on the JAFFE images since it had a larger number of training input samples, the accuracy and confusion matrix were also obtained.

Refinement

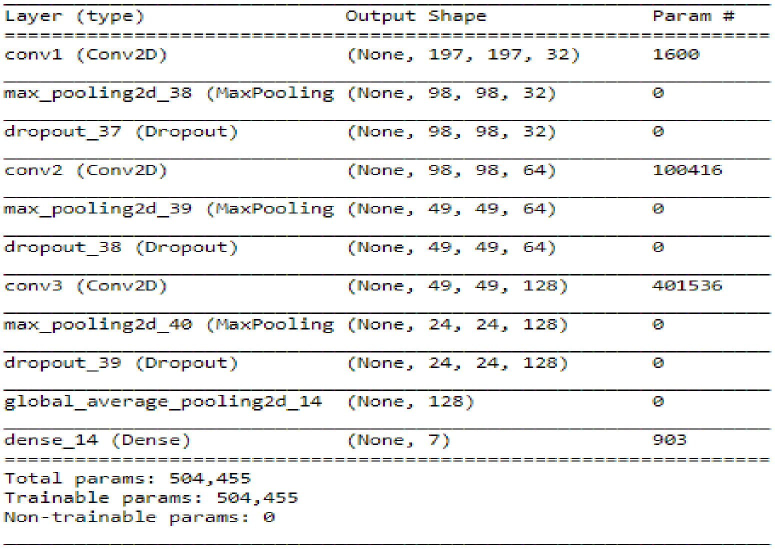
As discussed above random search technique was used to obtain a good model in the search space. The merits of using random search is that you do not have to search the entire search space only a few samples are picked, however this does not guarantee optimal model in the search space. The search space consisted of the following, kernel size between 2 and 7 and initial number of filters between 16 and 64. Ten random models were created for the JAFFE dataset since it is relatively small and 5 random models were created for the cohn-kanade dataset. For the JAFFE dataset the best obtained model had initial number of filter of 64 and kernel size 5. Figure 12 shows the model summary.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1 (Conv2D)               (None, 197, 197, 64)      1664      
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 98, 98, 64)        0         
_________________________________________________________________
dropout_13 (Dropout)         (None, 98, 98, 64)        0         
_________________________________________________________________
conv2 (Conv2D)               (None, 98, 98, 128)       204928    
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 49, 49, 128)       0         
_________________________________________________________________
dropout_14 (Dropout)         (None, 49, 49, 128)       0         
_________________________________________________________________
conv3 (Conv2D)               (None, 49, 49, 256)       819456    
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 24, 24, 256)       0         
_________________________________________________________________
dropout_15 (Dropout)         (None, 24, 24, 256)       0         
_________________________________________________________________
global_average_pooling2d_5 ( (None, 256)               0         
_________________________________________________________________
dense_5 (Dense)              (None, 7)                 1799      
=================================================================
Total params: 1,027,847
Trainable params: 1,027,847
Non-trainable params: 0
_________________________________________________________________
```

Figure 12

For the cohn-kanade dataset the best obtained model consists of 32 initial number of filters and kernel size of 7. Figure 13 shows the model summary.



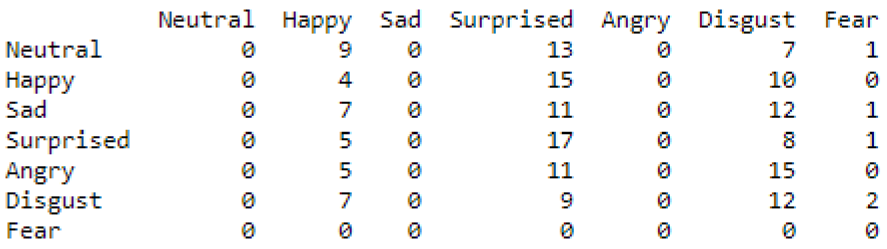
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1 (Conv2D)               (None, 197, 197, 32)      1600      
_________________________________________________________________
max_pooling2d_38 (MaxPooling (None, 98, 98, 32)        0         
_________________________________________________________________
dropout_37 (Dropout)         (None, 98, 98, 32)        0         
_________________________________________________________________
conv2 (Conv2D)               (None, 98, 98, 64)        100416    
_________________________________________________________________
max_pooling2d_39 (MaxPooling (None, 49, 49, 64)        0         
_________________________________________________________________
dropout_38 (Dropout)         (None, 49, 49, 64)        0         
_________________________________________________________________
conv3 (Conv2D)               (None, 49, 49, 128)       401536    
_________________________________________________________________
max_pooling2d_40 (MaxPooling (None, 24, 24, 128)       0         
_________________________________________________________________
dropout_39 (Dropout)         (None, 24, 24, 128)       0         
_________________________________________________________________
global_average_pooling2d_14  (None, 128)               0         
_________________________________________________________________
dense_14 (Dense)             (None, 7)                 903       
=================================================================
Total params: 504,455
Trainable params: 504,455
Non-trainable params: 0
_________________________________________________________________
```

Figure 13

IV. Results

Model Evaluation and Validation

The JAFFE model was able to score an 84% accuracy on the JAFFE dataset. Figure 14 shows the confusion matrix for the model. We can see that model is not able to predict sadness and fear with high accuracy.

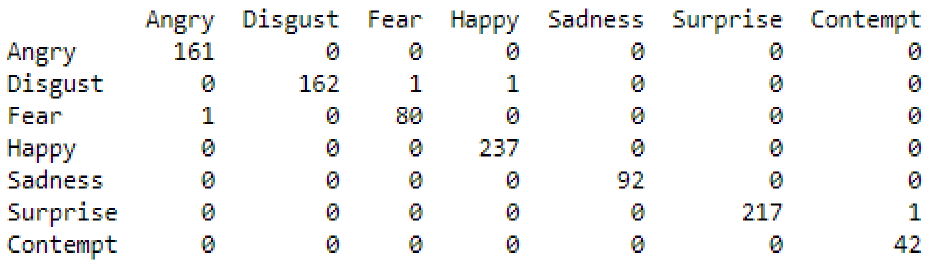


```
           Neutral  Happy  Sad  Surprised  Angry  Disgust  Fear
Neutral          0      9    0         13      0        7     1
Happy            0      4    0         15      0       10     0
Sad              0      7    0         11      0       12     1
Surprised        0      5    0         17      0        8     1
Angry            0      5    0         11      0       15     0
Disgust          0      7    0          9      0       12     2
Fear             0      0    0          0      0        0     0
```

Figure 14

The Resnet50 model that was trained on the JAFFE dataset scored only 15% accuracy.

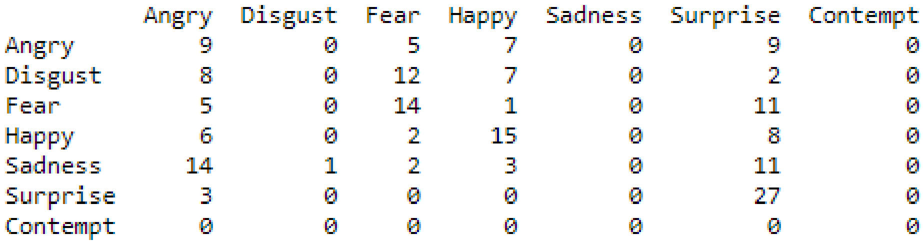
For the cohn-kanade model, it was able to score a 99.598% accuracy when tested on the cohn-kanade test dataset. Figure 15 shows the confusion matrix.



```
          Angry  Disgust  Fear  Happy  Sadness  Surprise  Contempt
Angry       161        0     0      0        0         0         0
Disgust       0      162     1      1        0         0         0
Fear          1        0    80      0        0         0         0
Happy         0        0     0    237        0         0         0
Sadness       0        0     0      0       92         0         0
Surprise      0        0     0      0        0       217         1
Contempt      0        0     0      0        0         0        42
```

Figure 15

The cohn-model was also tested on the entire JAFFE dataset to check for robustness and it scored an accuracy of 35.714%. Figure 16 shows the confusion matrix obtained from prediction of the JAFFE dataset. We can observe that the model fails completely at prediction of sad and disgust labels and can predict surprise and to a certain extent happiness.



```
           Angry  Disgust  Fear  Happy  Sadness  Surprise  Contempt
Angry          9        0     5      7        0         9         0
Disgust        8        0    12      7        0         2         0
Fear           5        0    14      1        0        11         0
Happy          6        0     2     15        0         8         0
Sadness       14        1     2      3        0        11         0
Surprise       3        0     0      0        0        27         0
Contempt       0        0     0      0        0         0         0
```

Figure 16

An InceptionV3 model was also trained on the cohn-kanade dataset but it scored only 32% when tested on the cohn-kanade dataset and 18% when tested on the JAFFE dataset.

Justification

The benchmark model produced only 53.57% accuracy when trained and tested on the JAFFE dataset. The model I made achieved an accuracy of 84%, however the JAFFE dataset is extremely small and no model that depends solely on such dataset can be robust enough to unseen data. For the cohn-kanade dataset, the benchmark model achieved an accuracy 95.79% when tested on the cohn-kanade dataset. The model I created achieved 99.598% when tested on the cohn-kanade dataset however they both achieved close results when tested on the JAFFE model where the benchmark achieved 37.36% and mine achieved 35.714%. In conclusion this is not robust enough to predict human facial expressions from unseen data. The biggest reason is that the dataset consists of very small amount of sample, hence larger datasets are needed.

V. Conclusion

Free Form Visualization

Figure 17 shows some images from the cohn-kanade testing dataset labeled with the true label and predicted label.

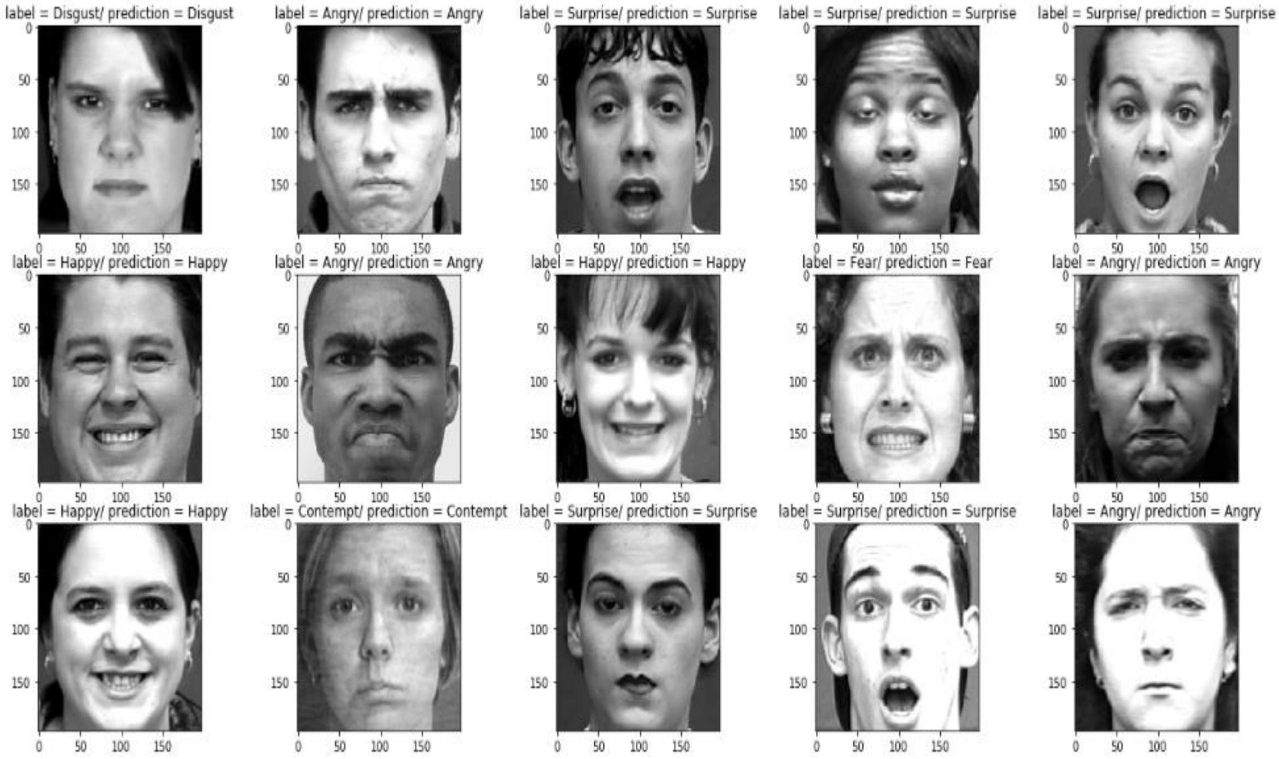

label = Disgust/ prediction = Disgust
label = Angry/ prediction = Angry
label = Surprise/ prediction = Surprise
label = Surprise/ prediction = Surprise
label = Surprise/ prediction = Surprise
label = Happy/ prediction = Happy
label = Angry/ prediction = Angry
label = Happy/ prediction = Happy
label = Fear/ prediction = Fear
label = Angry/ prediction = Angry
label = Happy/ prediction = Happy
label = Contempt/ prediction = Contempt
label = Surprise/ prediction = Surprise
label = Surprise/ prediction = Surprise
label = Angry/ prediction = Angry
0
50
100
150

Figure 17

Figure 18 shows some of the JAFFE images tested with the cohn-kanade model with their labels and predictions.

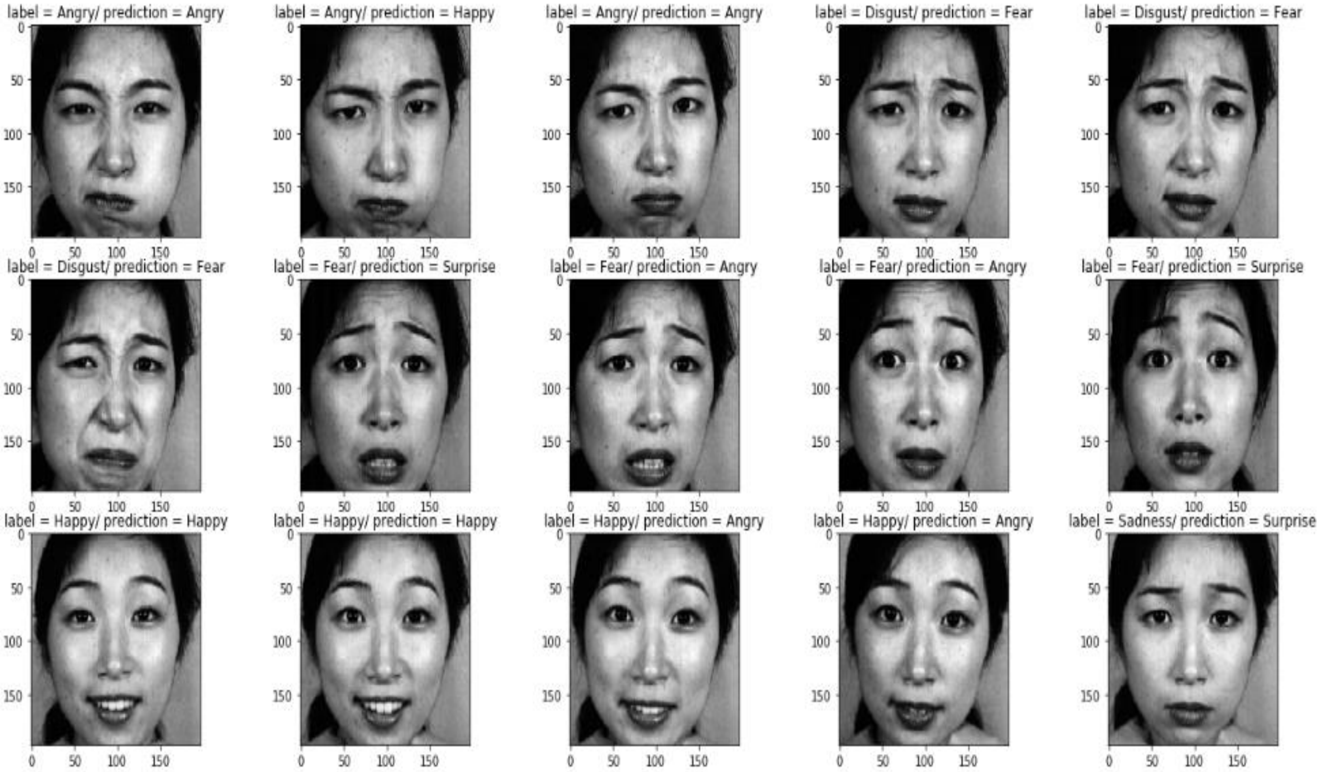

label = Angry/ prediction = Angry
label = Angry/ prediction = Happy
label = Angry/ prediction = Angry
label = Disgust/ prediction = Fear
label = Disgust/ prediction = Fear
label = Disgust/ prediction = Fear
label = Fear/ prediction = Surprise
label = Fear/ prediction = Angry
label = Fear/ prediction = Angry
label = Fear/ prediction = Surprise
label = Happy/ prediction = Happy
label = Happy/ prediction = Happy
label = Happy/ prediction = Angry
label = Happy/ prediction = Angry
label = Sadness/ prediction = Surprise
0
50
100
150

Figure 18

This shows that the model is not robust enough to deal the unseen data like those in the JAFFE dataset.

Reflection

The project can be summarized as the following:

- Available datasets like cohn-kanade and JAFFE were downloaded and imported
- Preprocessing such as augmentation and cropping was used
- Two models for each dataset were created and trained on their respective dataset
- Random search was used to find a local optimal model
- The final two models achieved greater accuracy than the benchmark model on their respective dataset
- Transfer learning was also tested but did not prove helpful for this problem

In general, the preprocessing part was the most difficult, it took a lot of research and trials. The most interesting part is how the cohn-kanade model achieved nearly 100% on the testing dataset which consists of a large variety of ways to show a particular emotion yet did not achieve such good accuracy with the JAFFE dataset, also transfer learning achieving very low accuracies was interesting.

Improvement

For future work I would like to try the binary method introduced by the benchmark model by making a model for each label and making them produce only true or false for their respective label. Also I would like to try to obtain a larger dataset since I believe that a very large impact on why the model is not robust enough is due to lack of samples. Lastly, in case large accuracies were obtained, I would like to introduce PCA to the problem to decrease the image dimensionality hence training and prediction time.

References

[1] Chen, J., Chen, Z., Chi, Z., & Fu, H. (2014, August). Facial expression recognition based on facial components detection and hog features. In *International workshops on electrical and computer engineering subfields* (pp. 884-888).

[2] Gosavi, A. P., & Khot, S. R. (2013). Facial expression recognition using principal component analysis. *International Journal of Soft Computing and Engineering (IJSCE)*, *3*(4), 2231-2307.

[3] Jiang, M., Gia, T. N., Anzanpour, A., Rahmani, A. M., Westerlund, T., Salanterä, S., ... & Tenhunen, H. (2016, April). IoT-based remote facial expression monitoring system with sEMG signal. In *2016 IEEE Sensors Applications Symposium (SAS)* (pp. 1-6). IEEE.

[4] Lopes, A. T., de Aguiar, E., De Souza, A. F., & Oliveira-Santos, T. (2017). Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recognition*, *61*, 610-628.

[5] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010, June). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops* (pp. 94-101). IEEE.