Predicting Facial Expression/Emotion from Images

...

Luka Kelly
9/19/23

Problem Statement

The goal of my project is to build 1-3 Convolutional Neural Network (CNN) models that can take in images of individuals with different facial expressions and classify them into the correct emotion classes with above 70% accuracy in both the training and testing set. The emotion classes that were identified in the source data were: Angry, Fear, Happy, Sad, Surprised, and Neutral.

The purpose of investigating this topic is to see how well a model can interpret facial expressions as simply as possible, with as little resources as possible. If a model can predict facial expressions with an accuracy of 70% without needing to purchase extra resources from services like Google Colab, it may elude to the possibility of smaller companies and individuals being able to implement these models for their own uses and products, without needing the resources of large tech companies.

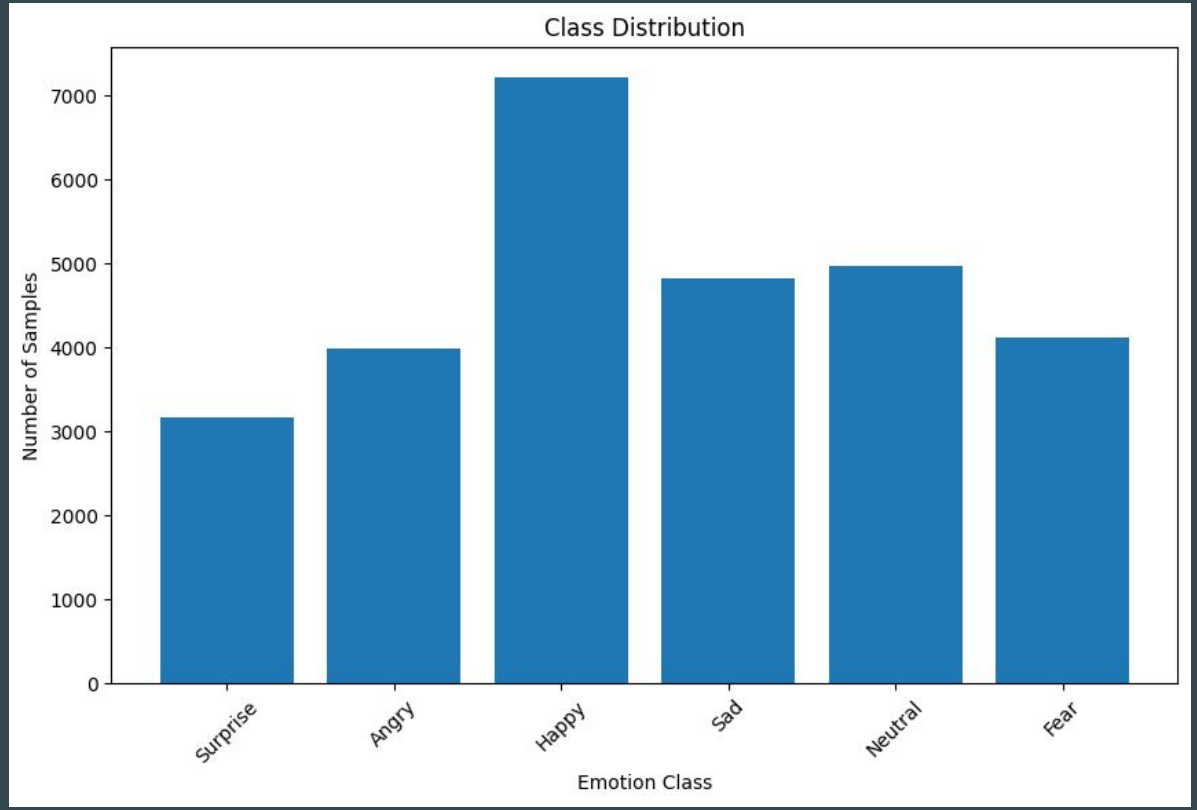
Additionally, if these models can easily and relatively accurately predict facial expression from images, it may improve certain technologies such as VR, AR, facial recognition technologies, etc., as well as aid in studies in psychology and related fields.

EDA and Preprocessing

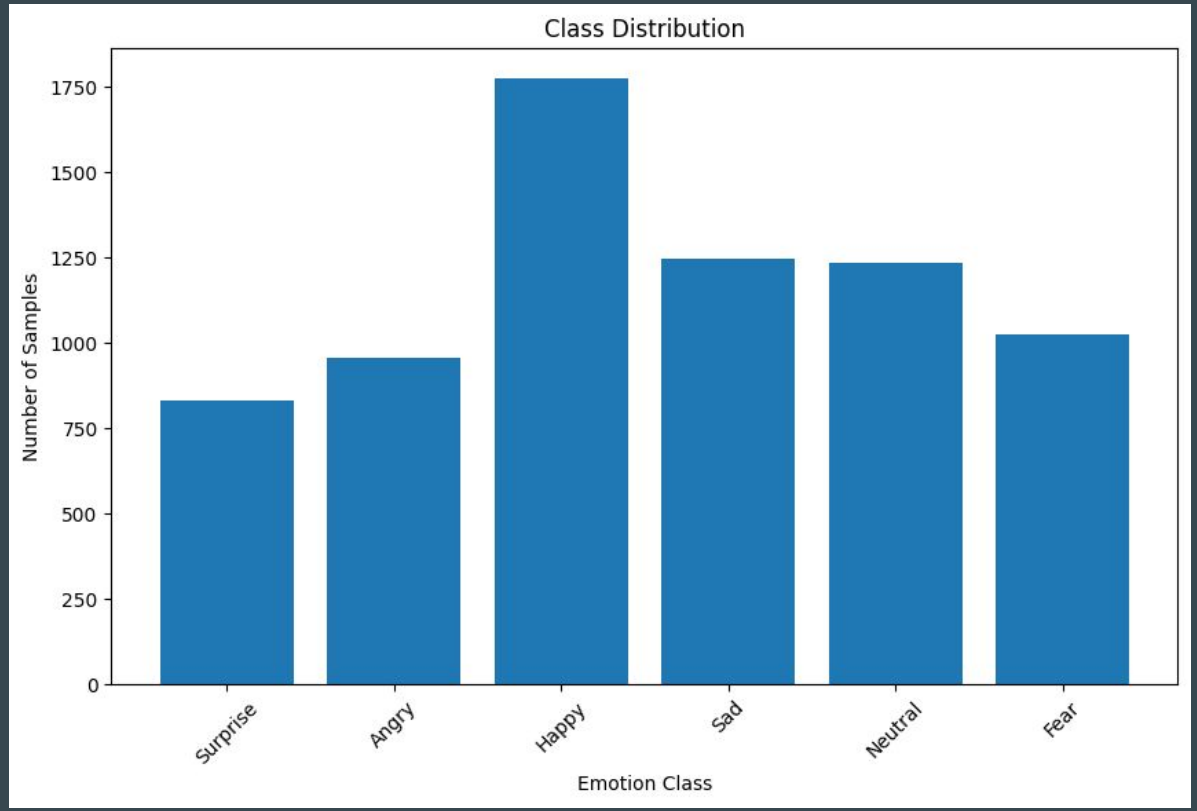
Sample Images from Each Emotion Category - Training Set:

Sample Images from Each Emotion Category - Testing Set:

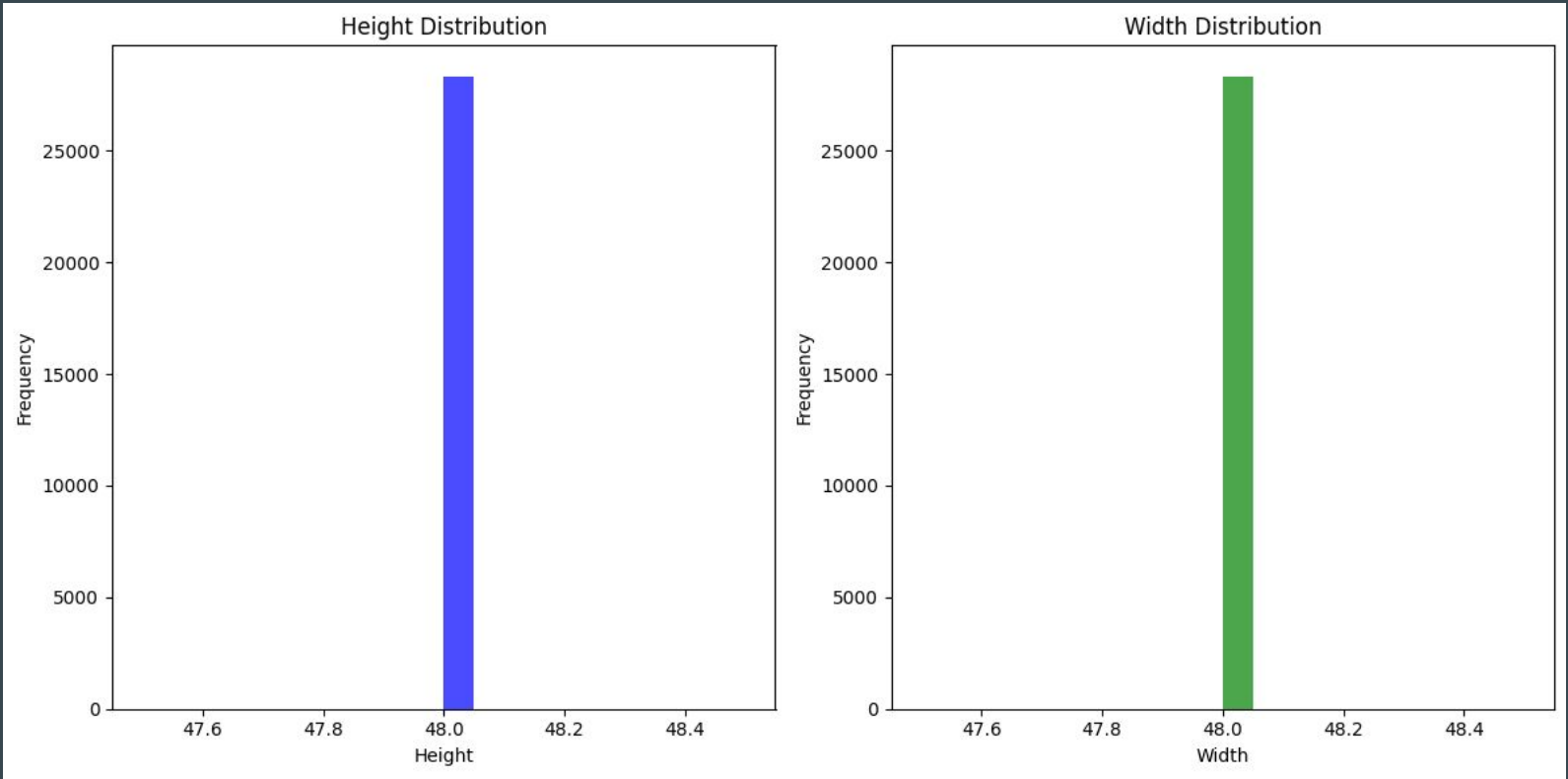
Emotion Class Distribution - Training Set



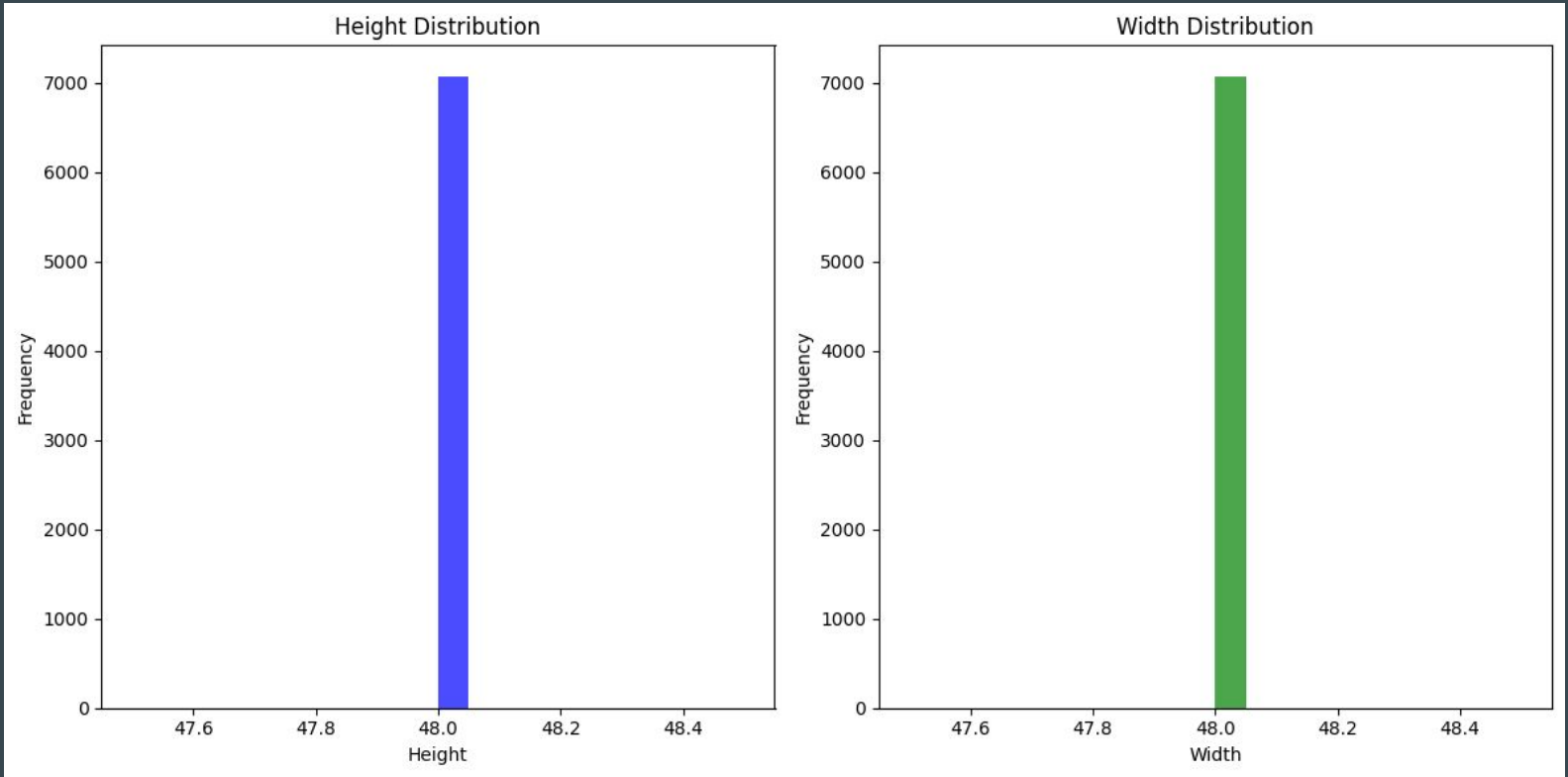
Emotion Class Distribution - Testing Set



Histogram of Image Dimensions - Training Set:



Histogram of Image Dimensions - Testing Set:



Computing and Visualizing Image Statistics - Training Set:

```
Emotion Class: Surprise
Mean Pixel Value: 146.26
Standard Deviation of Pixel Values: 64.93
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Angry
Mean Pixel Value: 126.23
Standard Deviation of Pixel Values: 65.19
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Happy
Mean Pixel Value: 129.43
Standard Deviation of Pixel Values: 63.38
Mean Image Dimensions (Height, Width, Channels): [48. 48.]
```

```
Emotion Class: Sad
Mean Pixel Value: 121.28
Standard Deviation of Pixel Values: 64.87
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Neutral
Mean Pixel Value: 123.91
Standard Deviation of Pixel Values: 65.30
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Fear
Mean Pixel Value: 135.56
Standard Deviation of Pixel Values: 64.96
Mean Image Dimensions (Height, Width, Channels): [48. 48.]
```

Computing and Visualizing Image Statistics - Testing Set:

```
Emotion Class: Surprise
Mean Pixel Value: 146.34
Standard Deviation of Pixel Values: 65.34
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Angry
Mean Pixel Value: 127.80
Standard Deviation of Pixel Values: 66.07
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Happy
Mean Pixel Value: 127.64
Standard Deviation of Pixel Values: 63.22
Mean Image Dimensions (Height, Width, Channels): [48. 48.]
```

```
Emotion Class: Sad
Mean Pixel Value: 120.10
Standard Deviation of Pixel Values: 64.31
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Neutral
Mean Pixel Value: 123.84
Standard Deviation of Pixel Values: 65.59
Mean Image Dimensions (Height, Width, Channels): [48. 48.]

Emotion Class: Fear
Mean Pixel Value: 135.06
Standard Deviation of Pixel Values: 65.63
Mean Image Dimensions (Height, Width, Channels): [48. 48.]
```

Modeling

Basic CNN Model - 5 Epochs, no Early Stopping or Data Augmentation

```
Epoch 1/5
885/885 [==============================] - 3424s 4s/step - loss: 1.6362 - accuracy: 0.3284 - val_loss: 1.4702 - val_accuracy: 0.4180
Epoch 2/5
885/885 [==============================] - 3388s 4s/step - loss: 1.4168 - accuracy: 0.4368 - val_loss: 1.3470 - val_accuracy: 0.4664
Epoch 3/5
885/885 [==============================] - 3364s 4s/step - loss: 1.2737 - accuracy: 0.5030 - val_loss: 1.2991 - val_accuracy: 0.4921
Epoch 4/5
885/885 [==============================] - 3362s 4s/step - loss: 1.1362 - accuracy: 0.5594 - val_loss: 1.2954 - val_accuracy: 0.4978
Epoch 5/5
885/885 [==============================] - 3388s 4s/step - loss: 1.0065 - accuracy: 0.6117 - val_loss: 1.3247 - val_accuracy: 0.5111
```

Evaluation:

- Concluded with a training accuracy of 0.61 and testing accuracy of 0.51
- both the training and testing accuracy scores improved from epoch 1(training: 0.33; testing: 0.42) to epoch 5(training: 0.61; testing: 0.51)
- model became increasingly overfit as the epochs increased
- Did not meet the success criteria of 70% accuracy on both the training and testing sets

CNN Model - 10 Epochs with Early Stopping, no Data Augmentation

```
Epoch 1/10
885/885 [==============================] - 9198s 10s/step - loss: 1.6694 - accuracy: 0.3119 - val_loss: 1.4797 - val_accuracy: 0.4111
Epoch 2/10
885/885 [==============================] - 2968s 3s/step - loss: 1.4527 - accuracy: 0.4255 - val_loss: 1.3403 - val_accuracy: 0.4663
Epoch 3/10
885/885 [==============================] - 3022s 3s/step - loss: 1.3091 - accuracy: 0.4859 - val_loss: 1.3446 - val_accuracy: 0.4796
Epoch 4/10
885/885 [==============================] - 3004s 3s/step - loss: 1.1840 - accuracy: 0.5447 - val_loss: 1.3064 - val_accuracy: 0.4996
Epoch 5/10
885/885 [==============================] - 3014s 3s/step - loss: 1.0387 - accuracy: 0.6019 - val_loss: 1.3156 - val_accuracy: 0.5013
Epoch 6/10
885/885 [==============================] - 3029s 3s/step - loss: 0.9114 - accuracy: 0.6555 - val_loss: 1.4048 - val_accuracy: 0.5057
Epoch 7/10
885/885 [==============================] - 3010s 3s/step - loss: 0.7899 - accuracy: 0.6964 - val_loss: 1.4375 - val_accuracy: 0.5056
```

Evaluation:

- Adding more epochs to the model improved the training accuracy(~0.70 after 7 epochs, vs ~0.61 after 5 epochs in my previous model)
- Although testing accuracy began to stagnate around 0.5
- Model became increasingly overfit as it moved higher in the number of epochs
- Model stopped at epoch 7 due to the early stopping clause
- Did not meet the success criteria of 70% accuracy on both the training and testing sets

Conclusion and Recommendations:

- Do not think that one can conclude with certainty one way or the other
- Attempted to build a CNN model that ran for 10 epochs with early stopping and included significant data augmentation to combat overfitting/high variance
    - continued to reproduce a "FileNotFoundError"
- If model mention above model is able to be produced, model could produce 70%+ accuracy scores on both the training and testing set
- Recommendation:
    - When attempting to build that can take in images of individuals with different facial expressions and classify them into the correct emotion classes with a 70% accuracy on both the training and testing set, one should include significant data augmentation to combat overfitting/high variance

Future Steps

- Plan to install TensorFlow/Keras onto my local machine so I can try to run the CNN model with 10 epochs, early stopping, and significant data augmentation using a different resource, such as VS Code or Jupyter Lab
    - May help to resolve “FileNotFoundError” issue, making running my models a much smoother process