ASL Sign Language Translation

Beard, Bennion, Carlson, Napolitano

**Data Science Capstone Project   
Exploratory Data Analytics Report**

Date:

[5/27/2022]

Team Members:

Name: Tyler Beard

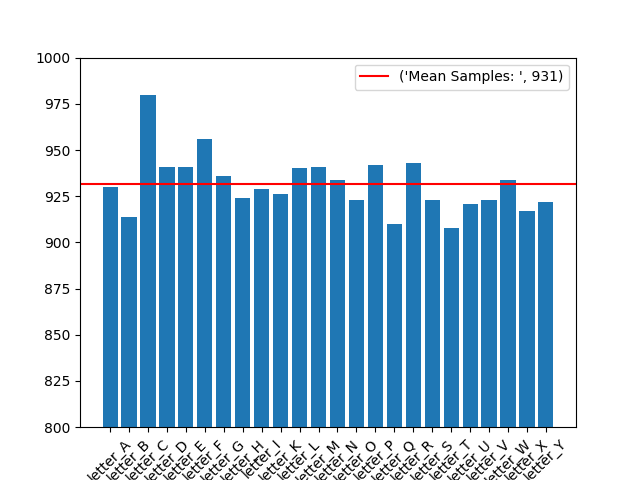
Name: Adam Bennion

Name: Zach Carlson

Name: Andrew Napolitano

**Analysis of class balance**

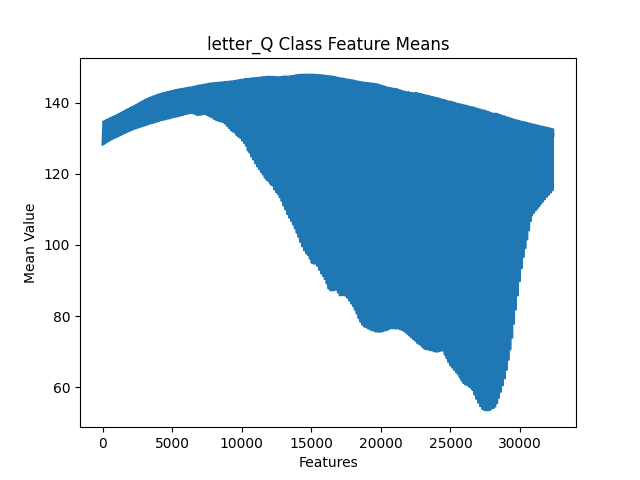
Since the dataset was created by our team specifically for this project, we were able to control several key aspects of the dataset, including the class balance. The plot below demonstrates that the dataset is fairly balanced across all 24 classes, with only one class (Letter C) having noticeably more samples than the other classes. The Letter C sample contains 980 samples which is 49 images or only about 5% deviation from the Sample Mean of 931. Since the deviation is only 5%, our team concluded that this will not have a material effect on our model and decided not to remove samples at this time. In the training phase it will be trivial to undersample if necessary, or in other words reduce all class sample counts to the same quantity.

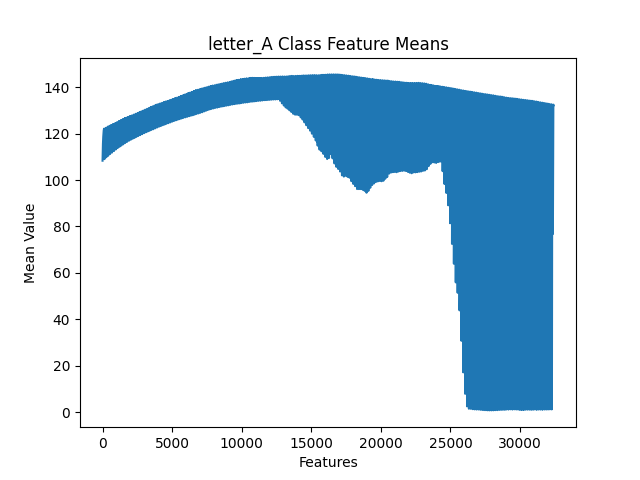
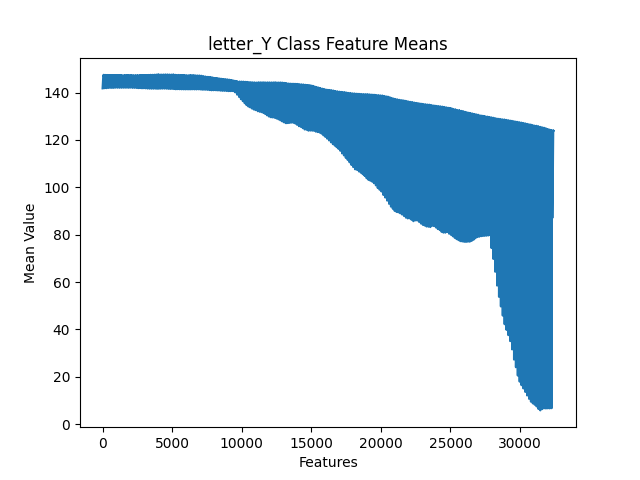


**Analysis of the basic metrics of variables**

The dataset consists of images and therefore the feature space is composed entirely of pixels. As of this report, each feature is a value between 0 and 255 and thus continuous/numerical. This value range represents a greyscale from black to white. This value range can be seen in the feature-mean space plots later in the report.

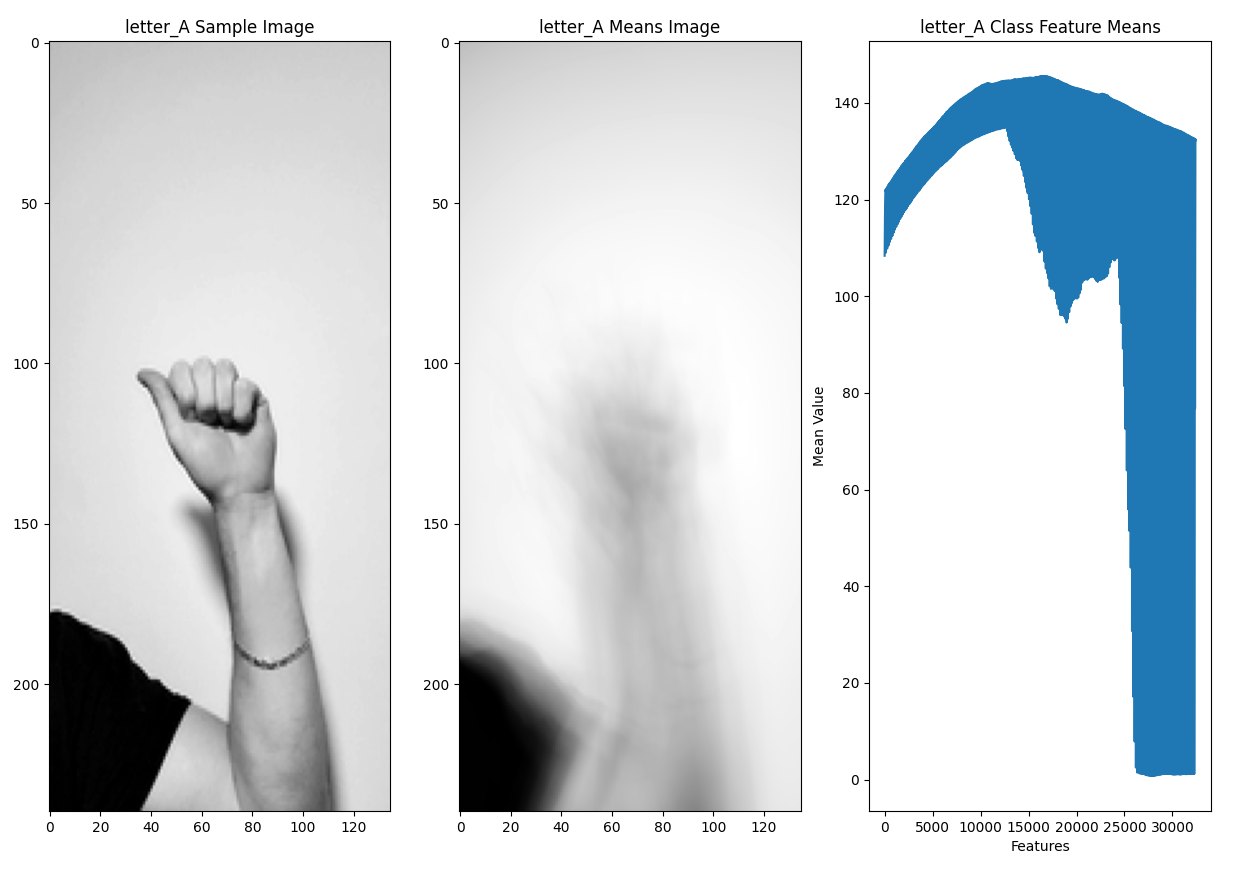
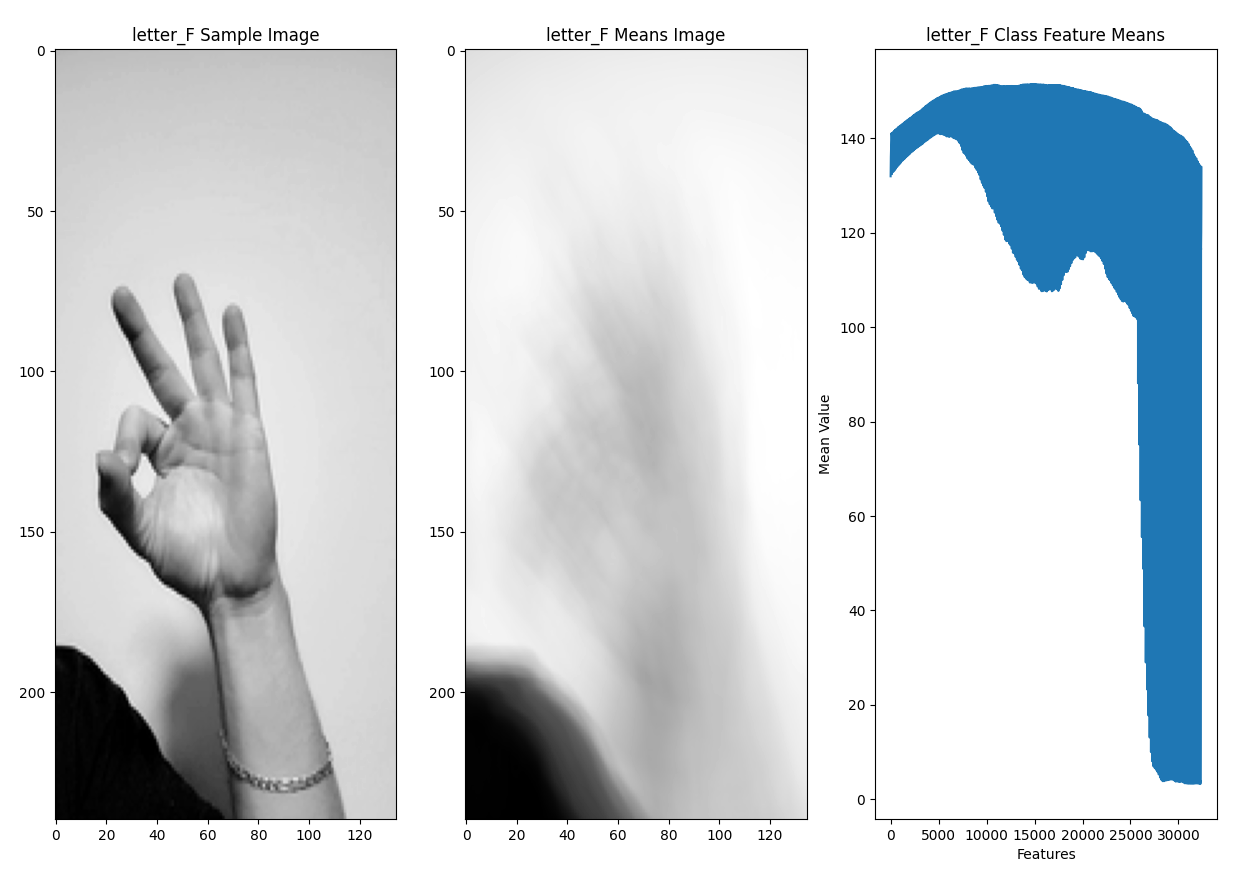
A plot was generated for each class, showing mean value per feature. Since there are 32400 features, each plot is quite dense. However, you can see clear distinctions between classes so it is logical that classes have potential to be fairly separable (plots on the next page).





**Non-graphical and graphical univariate analysis**

To gain better visualization in exploratory data analysis, for each class the mean values of each feature were turned back into an image and compared to a random sample from the class. Combining these two images with the aforementioned feature mean plots allowed for an apt visual comparison of classes to get a general sense of where classes overlap in similarity, and a general sense of deviation.



**Missing value analysis and outlier analysis**

Due to the nature of an image dataset no values are missing.

**Feature engineering and analysis**

Due to the nature of 24 classes and 32400 features, the team deemed standard deviation amongst classes for each feature as the key metric to be explored. Standard deviation was computed across classes for each feature to highlight least-informative features. The goal of identifying the least-important features is to minimize the featurespace down to the most useful features which will assist with accuracy and avoid overfitting as it prevents an over-complicated solution. Overfitting is especially likely when dealing with a feature space larger than the total number of samples.

First, the team consolidated the mean pixel values for each class into one DataFrame, where each column was a given pixel (pixel1, pixel2, etc.) and each row was a class (A, B, C, etc.). Standard deviation was calculated across each column, converted to an array, and then reshaped to the original image format of 240x135 pixels. The reshaping was required to see where the least important pixels were located in the image.

Before visualization, the grayscale image was inverted, so that the darker regions represented areas with higher standard deviation. The lighter regions are areas that are relatively consistent across all classes.

As expected, the top ⅓ to ½ of the image is consistent across all classes and isn’t needed for training. The team opted to subset the DataFrame to exclude these pixel values. However, before training using cropped images, the team will visualize a random subset of images for each class to ensure the entire hand sign is still captured.

**Training research**

Due to the nature of our dataset, EDA is somewhat limited, therefore we have already begun to consider our training and pre-training process.

**Transfer Learning Investigation:**

The biggest problem with transfer learning is the datasets used to train the pre-trained weights. As discussed in our acquisition report, most readily available datasets include incorrect hand signs for some letters. This was the case with the following transfer learning source:

https://www.freecodecamp.org/news/asl-recognition-using-transfer-learning-918ba054c004/

Another issue with transfer learning is the different dataset image sizes. The aforementioned transfer learning source generated weights for a 200x200 image dataset, meaning our team would need to resize and crop all ~21000 of our images to match the pre-trained weights.

We will likely utilize VGG16 or Resnet directly instead of using an existing ASL weight set if we employ transfer learning. Again, this will require a re-scaling and cropping of our images to match the 224x224 image size default for the pretrained models. This can most readily be achieved by running our acquisition and preprocessing pipelines but with a rescaling that squishes the images into a square aspect ratio. We would also remove the greyscale step from preprocessing due to the expectations of colored images in VGG16.

**Documentation for Transfer Learning:**

https://keras.io/api/applications/vgg/

https://www.kaggle.com/code/praanj/transfer-learning-vgg-19-resnet-50-with-kfold/notebook

https://koushik1102.medium.com/transfer-learning-with-vgg16-and-vgg19-the-simpler-way-ad4eec1e2997

https://github.com/fchollet/deep-learning-models/releases/

https://machinelearningmastery.com/use-pre-trained-vgg-model-classify-objects-photographs/

**Traditional Convolutional Neural Network (no transfer learning)**

CNN would likely follow the rather traditional route of the following code:

Conv2D**,** MaxPool2D**,** Flatten**,** Dense**,** Dropout**,** BatchNormalization

model **=** Sequential**()**

model**.**add**(**Conv2D**(**filters**=**32**,** kernel\_size**=(**3**,**3**),** activation**=**"relu"**,** input\_shape**=(**28**,**28**,**1**)))**

model**.**add**(**MaxPool2D**((**2**,**2**),**padding**=**'SAME'**))**

model**.**add**(**Dropout**(**rate**=**0.2**))**

model**.**add**(**Conv2D**(**filters**=**64**,** kernel\_size**=(**3**,**3**),** activation**=**"relu"**,** input\_shape**=(**28**,**28**,**1**)))**

model**.**add**(**MaxPool2D**((**2**,**2**),**padding**=**'SAME'**))**

model**.**add**(**Dropout**(**rate**=**0.2**))**

model**.**add**(**Conv2D**(**filters**=**521**,** kernel\_size**=(**3**,**3**),** activation**=**"relu"**,** input\_shape**=(**28**,**28**,**1**)))**

model**.**add**(**MaxPool2D**((**2**,**2**),**padding**=**'SAME'**))**

model**.**add**(**Dropout**(**rate**=**0.2**))**

model**.**add**(**Flatten**())**

model**.**add**(**Dense**(**units**=**521**,** activation**=**"relu"**))**

model**.**add**(**Dense**(**units**=**256**,** activation**=**"relu"**))**

model**.**add**(**Dropout**(**0.5**))**

model**.**add**(**Dense**(**units**=**24**,** activation**=**"softmax"**))**

reference: https://www.kaggle.com/code/uditsharma72/sign-language-mnist-acc-98

This code segment serves as an outline for our own implementation of a baseline model to compare our potential transfer learning model against.

**EDA and early training study conclusion**

**Issues identified in this phase**

1. Deviation and mean feature statistics highlighted the value of cropping the images. There is ample deviation in samples due to the presence of a forearm and shirt in the images. Ideally shirt and forearm positing should have less effect on the training model for future application and proof of concept.
2. Readily available libraries for transfer learning are most compatible with square images.
3. A slight imbalance of classes in the dataset

**Solutions proposed**

1. Revisiting the preprocessing phase and modifying previous scripts to successfully crop images down to square images. A proposed code solution for accurate cropping would involve traversing images row by row from top to bottom. Upon reaching a high contrast between non-edge pixels (due to the start of the hand), the script would crop into a square based on the row position.
2. When revisiting the preprocessing phase, images can be scaled to more appropriate dimensions (224x224) compatible with VGG16 for transfer learning.
3. A slight imbalance of classes in the dataset may require trivial class balancing via undersampling.

**Revisiting the Preprocessing for a Stronger Dataset**

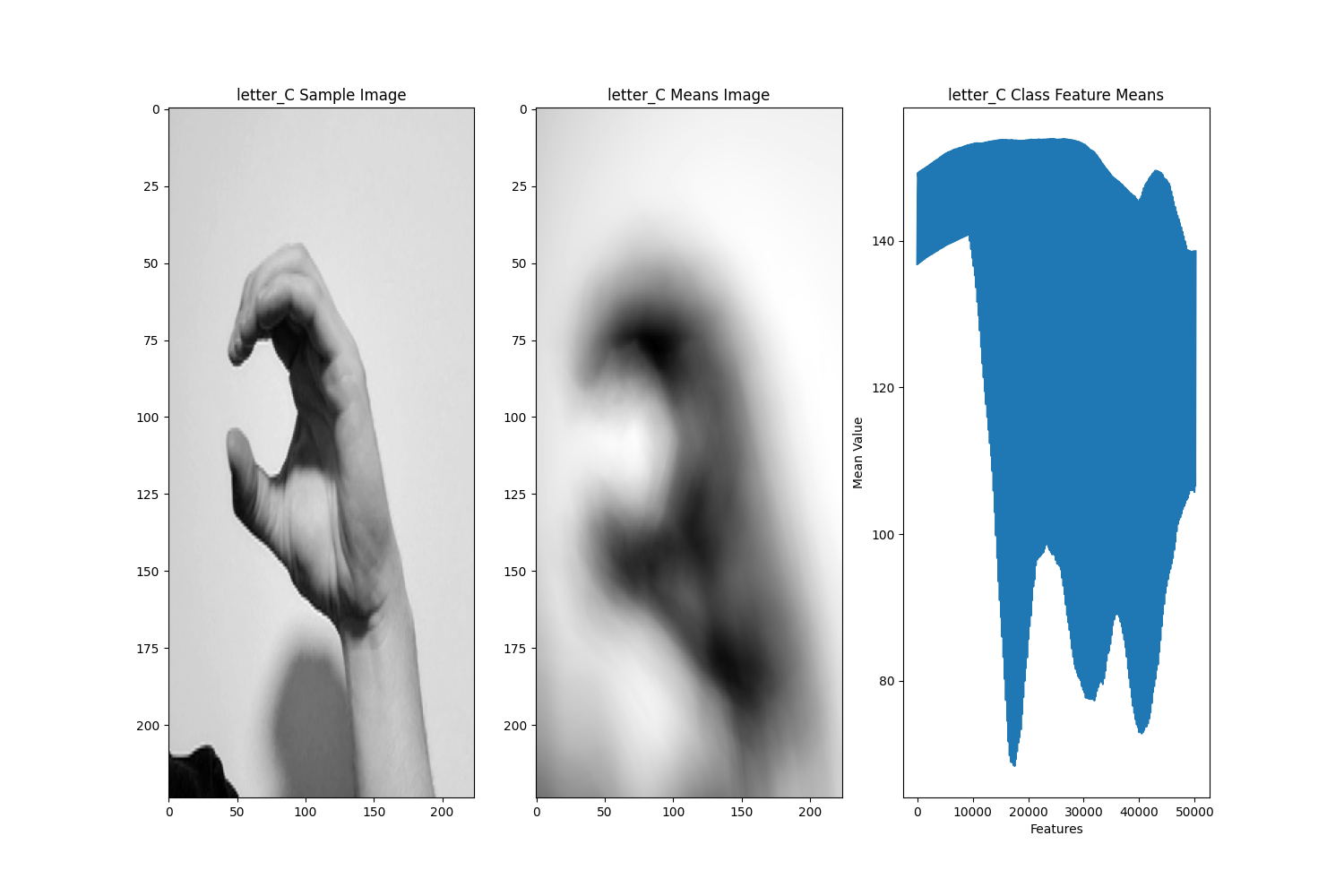
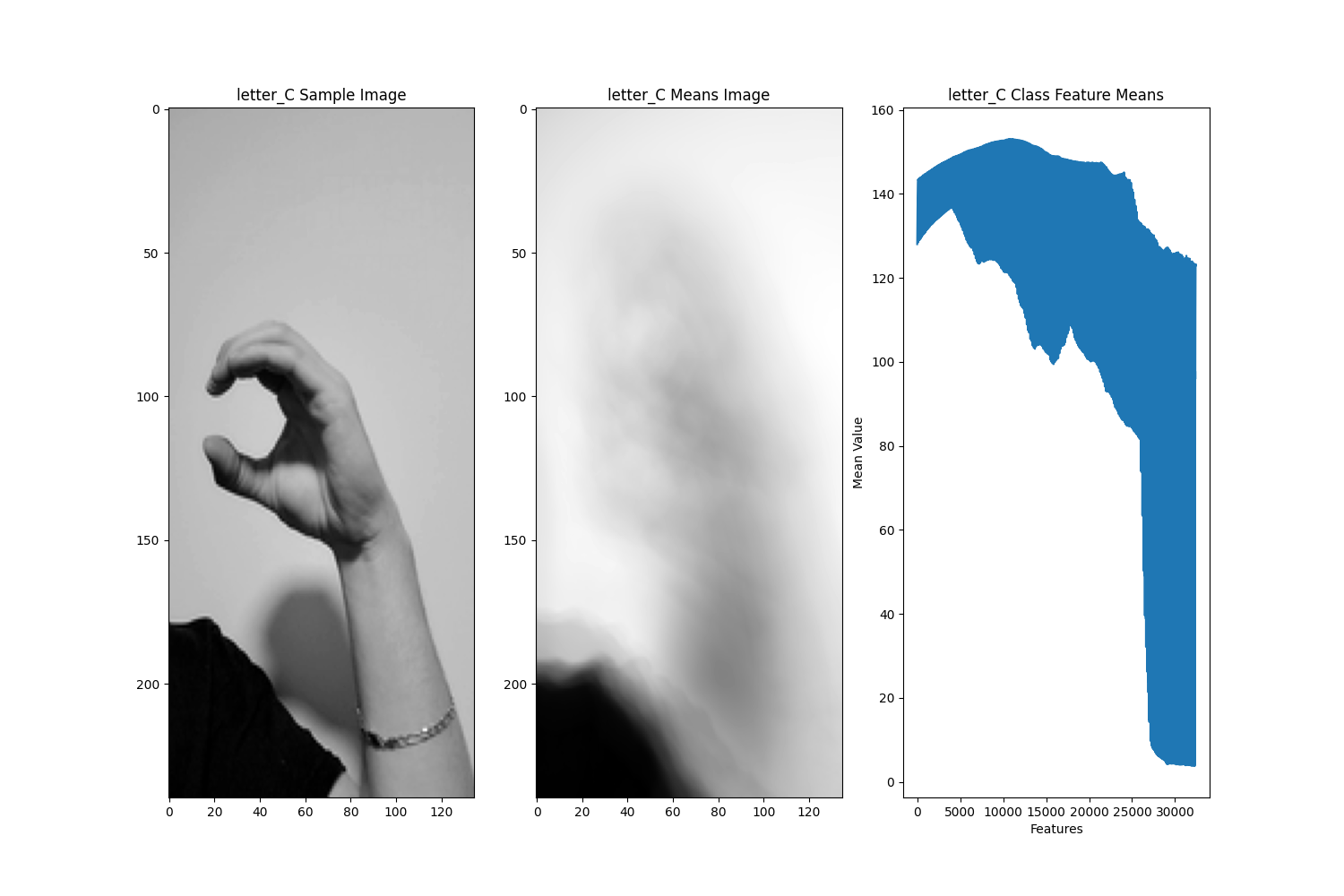
After talking with the professor and amongst the team we decided to take another crack at our image data in order to get the best possible experience for the model training. We were mainly worried about 1 thing: the amount of unnecessary pixels. The images contained too much blank space and not the hand making the letter itself. We wanted to revisit our pre processing to see if we could clean them up in order to help out our model training in the future.

**Steps Taken**

1. Run steps 2 through 4 with adjustments until subject centering and cropping was satisfactory
2. Shrink original 1080p images to 224x398
3. Add y axis centering so that the hand is the main focal point when cropping
4. Crop unwanted space so that the images become 224x224
5. Rerun grayscaling

**Results**

* The top image is the first round of pre processing and you notice that the means image(center image) is very bright where the hand is. Additionally, you can see that the feature means graph is sparse looking at the top left. This is due to the amount of space at the top of the cropped hand image. Now looking at the bottom image(2nd round of pre processing after EDA) the sample image is much more focused in on the hand itself. This makes the means image much more contrasted about the hand and is further supported by the feature means graph.



* The figure on the left (first preprocessing round), there was a lot of undeviated space at the top of the image with the hand being barely noticiable in the middle. The image on the right (2nd round of pre processing after EDA) has a much better hand distinction and is prominently deviated.

****

**Appendix**

Note: The provided code snippets are not 100% of the code involved in the EDA phase. For the complete code please contact the team.

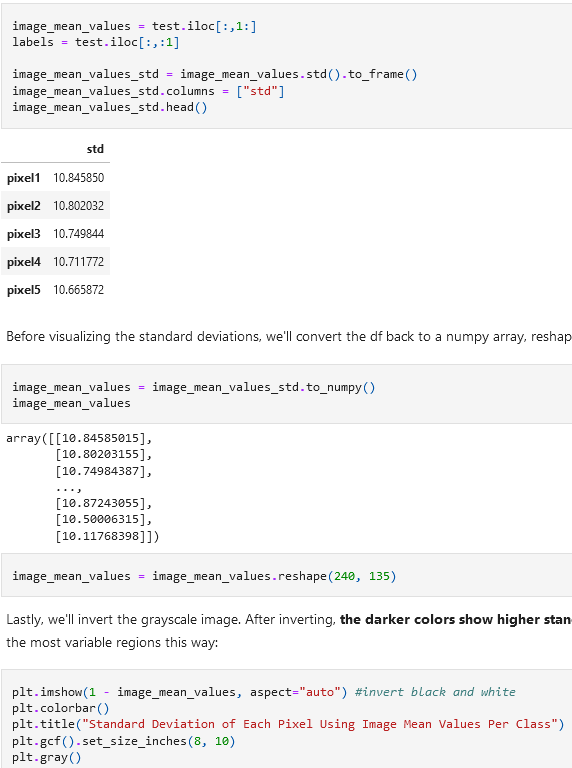
Data Wrangling and mean image code:

****

Feature Means Code:



Standard Deviation code snippet:



Class Balance code:



Table of Contributions

The table below identifies contributors to various sections of this document.

|  | **Section** | **Writing** | **Editing** |
| --- | --- | --- | --- |
| **1** | **Analysis of class balance** | **AN** | **TB** |
| **2** | **Analysis the basic metrics of variables** | **AB** | **ZC, AN, TB** |
| **3** | **Non-graphical and graphical univariate analysis** | **AB, ZC** | **ZC, AN, TB** |
| **4** | **Missing value and outlier analysis** | **AB** | **AN, TB** |
| **5** | **Feature engineering and analysis** | **AB, ZC** | **ZC, AN, TB** |
| **6** | **Appendix** | **TB, AB** | **AB** |

**Grading**

The grade is given on the basis of quality, clarity, presentation, completeness, and writing of each section in the report. This is the grade of the group. Individual grades will be assigned at the end of the term when peer reviews are collected.