Project Part 1
3/22/19
Team TRM
Trung, Nikita, Rosemond & Michael

1.) **How good is your performance when you train (and/or validate) using all but one teammember's data and test on the remaining teammember's data? Repeat this for each teammember being the test case multiple times (at least 3x each). Back up your response with graphs and/or tables of results.**

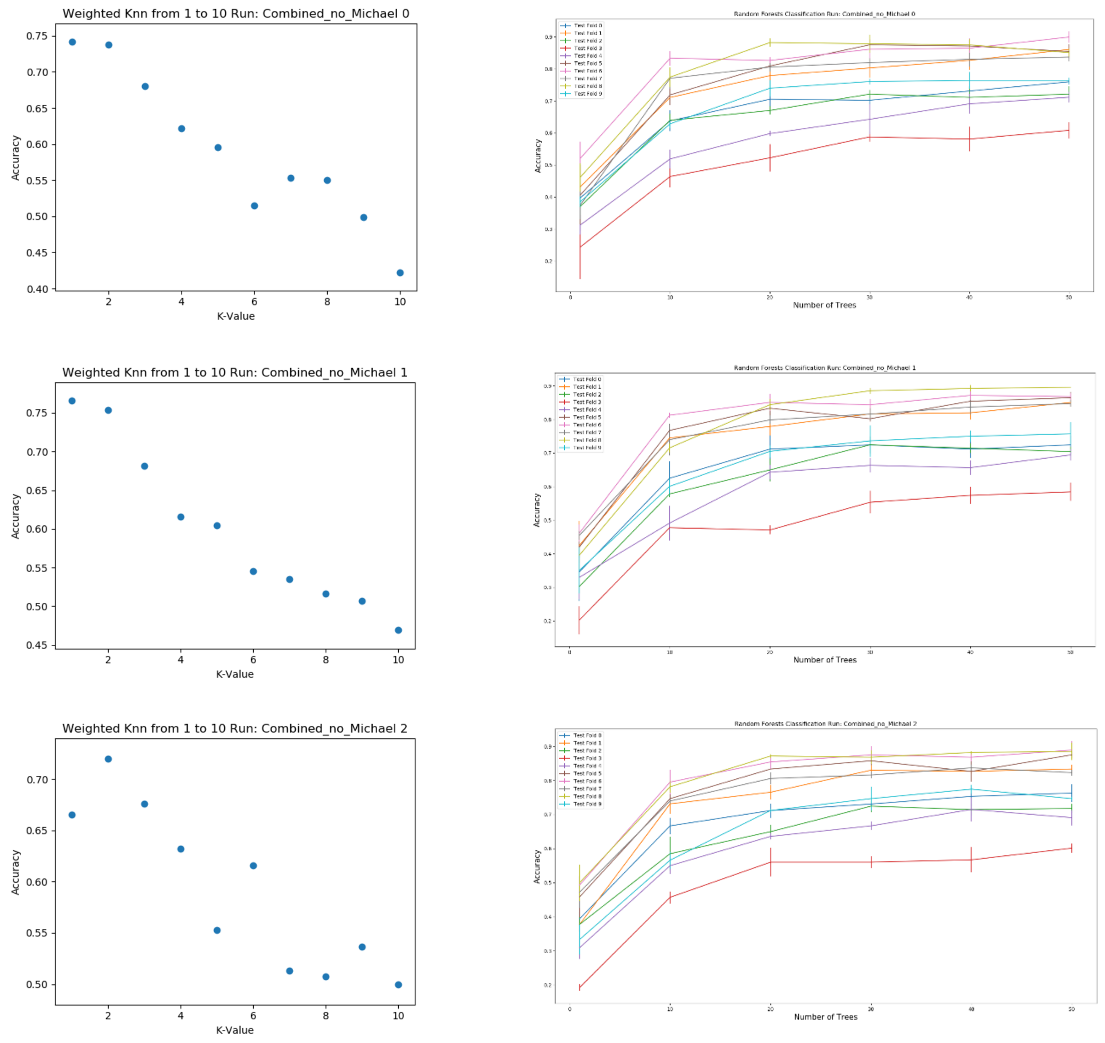

Figure 1: The graphs above depict the results when Michael’s data was taken away. Ran 3 times. K-Value: 1 – 10. Number of trees: 0 – 50.

Project Part 1
3/22/19
Team TRM
Trung, Nikita, Rosemond & Michael

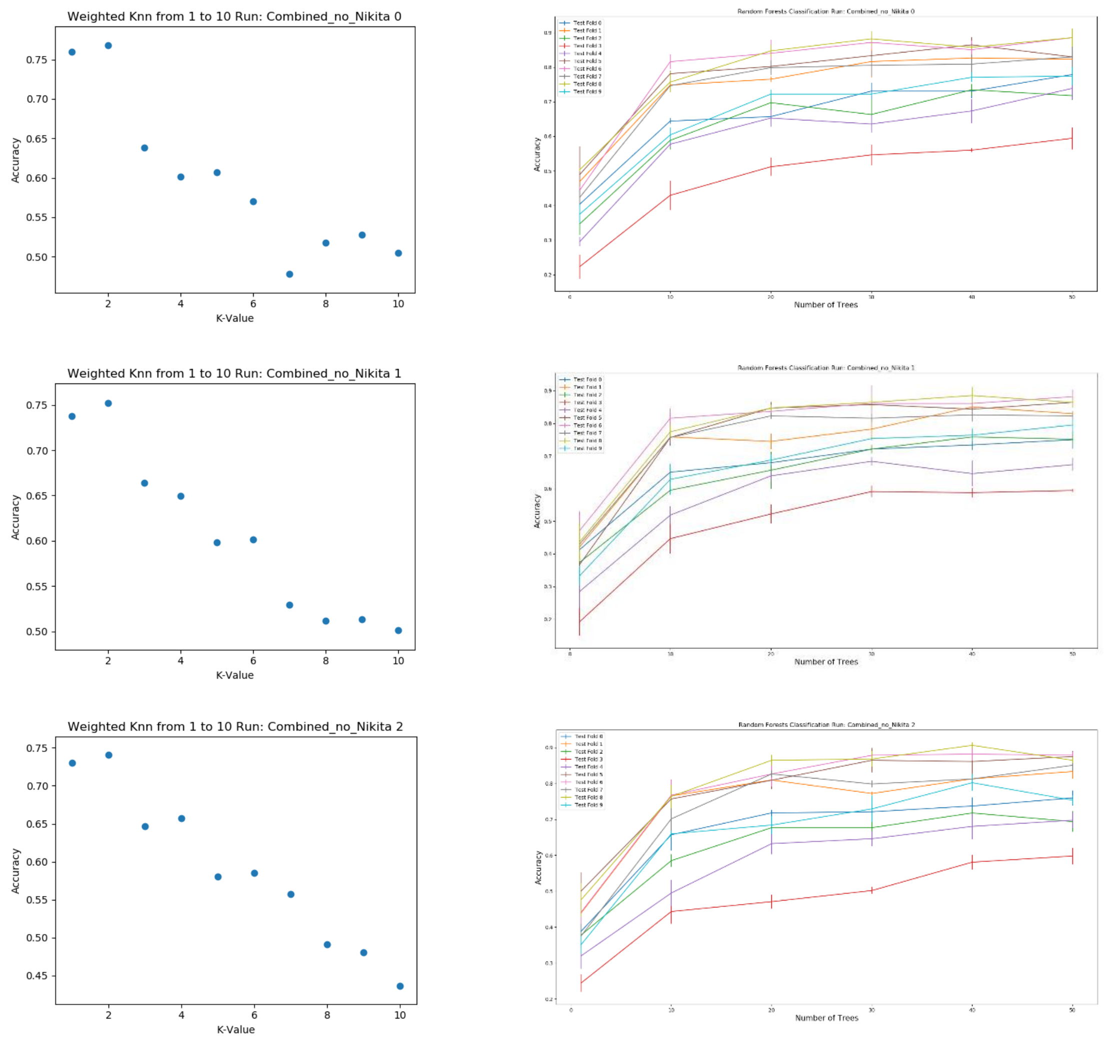

Figure 2: The graphs above depict the results when Nikita’s data was taken away. Ran 3 times. K-Value: 1 – 10. Number of trees: 0 – 50.

Project Part 1
3/22/19
Team TRM
Trung, Nikita, Rosemond & Michael

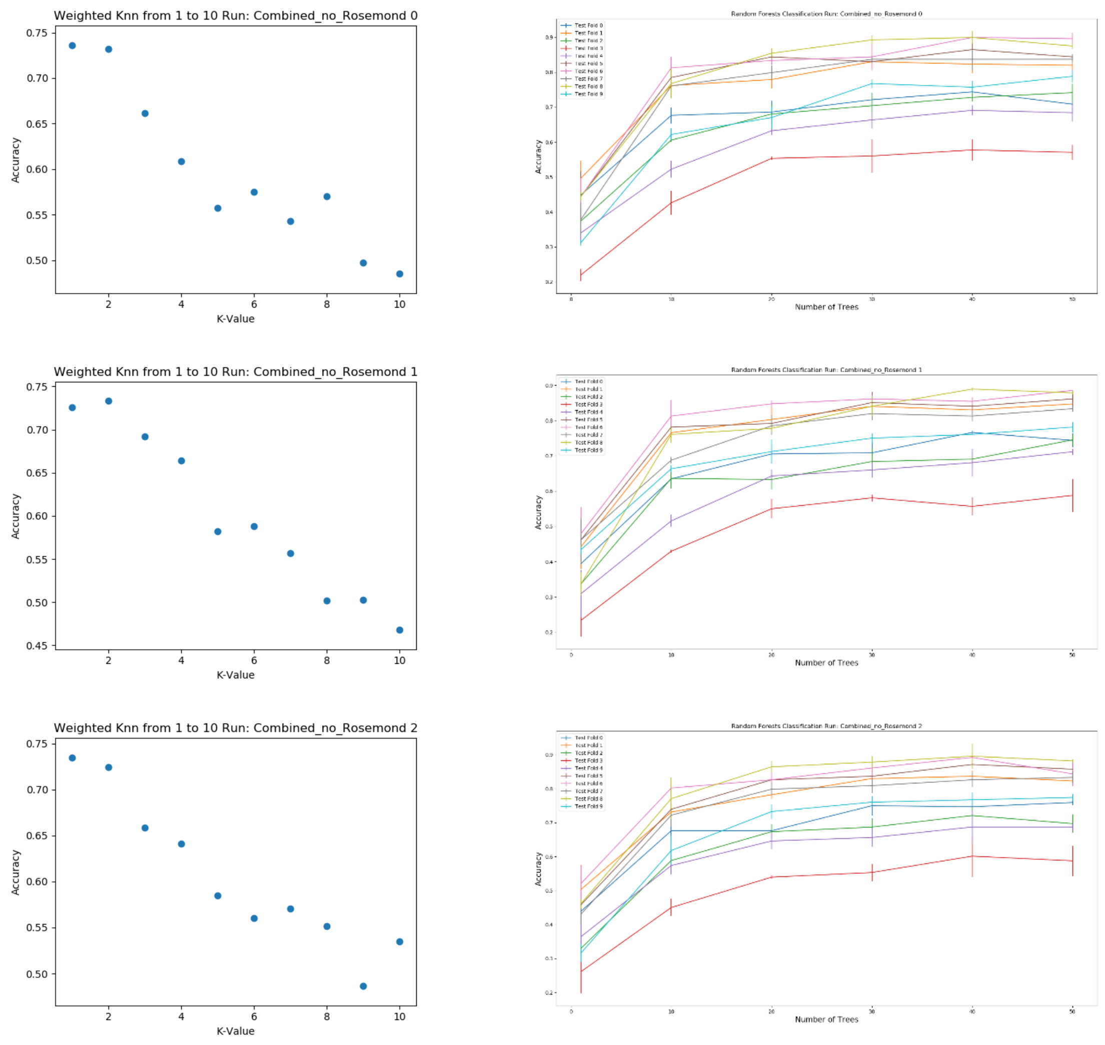

Figure 3: The graphs above depict the results when Rosemond's data was taken away. Ran 3 times. K-Value: 1 – 10. Number of trees: 0 – 50.

Project Part 1
3/22/19
Team TRM
Trung, Nikita, Rosemond & Michael

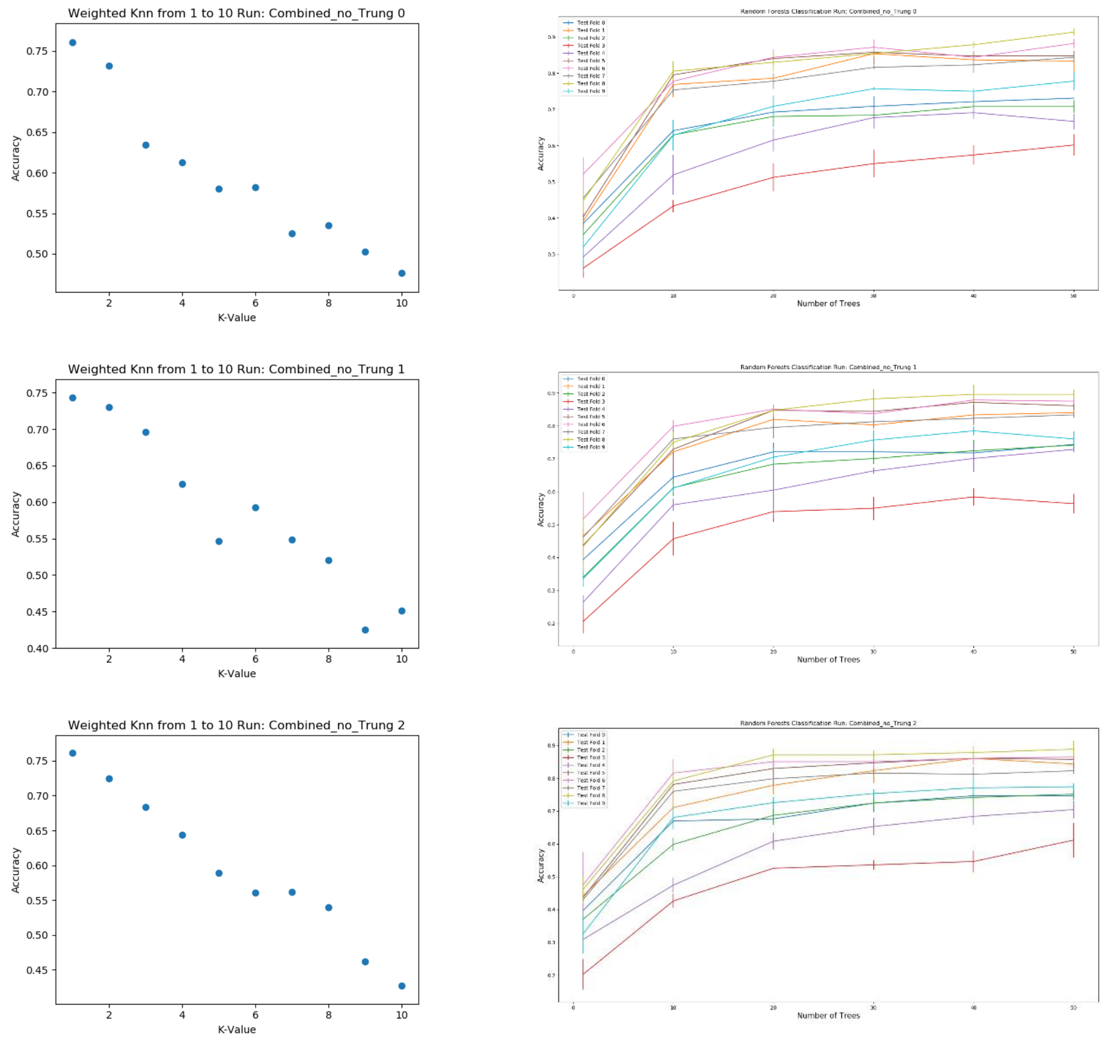

Figure 4: The graphs above depict the results when Trung's data was taken away. Ran 3 times. K-Value: 1 – 10. Number of trees: 0 – 50.

Project Part 1
3/22/19
Team TRM
Trung, Nikita, Rosemond & Michael

As you can see from Figures 1 through 4. All our resulting graphs give relatively the same accuracy. Looking at all our data and the images that we took individually, it would make sense these scores are the same across the board. The only surprise is that the scores are the same even though Rosemond, whose hand skin color is a bit darker, give the same results for the most part. Even with the preprocessing that we did, we still thought that his data would throw off some of the accuracy.

Of course, we did our best to make sure that skin color and tone would be accounted for. That was why we opted to use HSV instead of RGB. We discovered that looking at the hue and saturation of the image gave us a better difference between the hand and the background. That way, we can zoom into the hand and ignore any other objects of non-importance in the image. Looking at the hues of the image, there was not any big difference in values from Rosemond's hand compared to the rest of the team. The results make more sense when you consider that fact.

All of us also made sure to follow the instructions for taking photos and had our hands centered for the most part and a white, clear background with as little noise as possible.

Looking at all the figures, you can see that as we increase the K-Value for the KNN classifier, our accuracy decreases. I think we may need to extract better features to increase our accuracy when a K-Value of 1 is chosen. Right now, it's approximately around 75% accurate. That's not good considering the lower the K-Value, the more influence noise will have on the results. Our noise is minimal for the most part thanks to our preprocessing cropping out the hand, but we need to extract more features other than edge histogram and ORB key points.

For the random forest, all the resulting graphs reach between 80 – 90 percent accuracy when the number of trees is 50. That makes sense considering most of the features ranked were all the pixels from the dataset. As we increase the number of trees, the accuracy increases and will plateau at some point. This will have to be tuned in the future once we rank the features and throw out the ones that does not increase the accuracy or hurts our results.

Overall, the results came out decent. These graphs will look a lot differently once we train/test on the class' dataset. We will probably have to rethink our preprocessing and also some of our feature extractions based on some of the horrific images that we have seen in the class' dataset.

Project Part 1
3/22/19
Team TRM
Trung, Nikita, Rosemond & Michael

2.) How good is your performance when you shuffle all teammembers' data and use 70% for training (and/or validation) and the remaining 30% for test? Repeat this multiple times (at least 3x). Back up your response with graphs and/or tables of results.

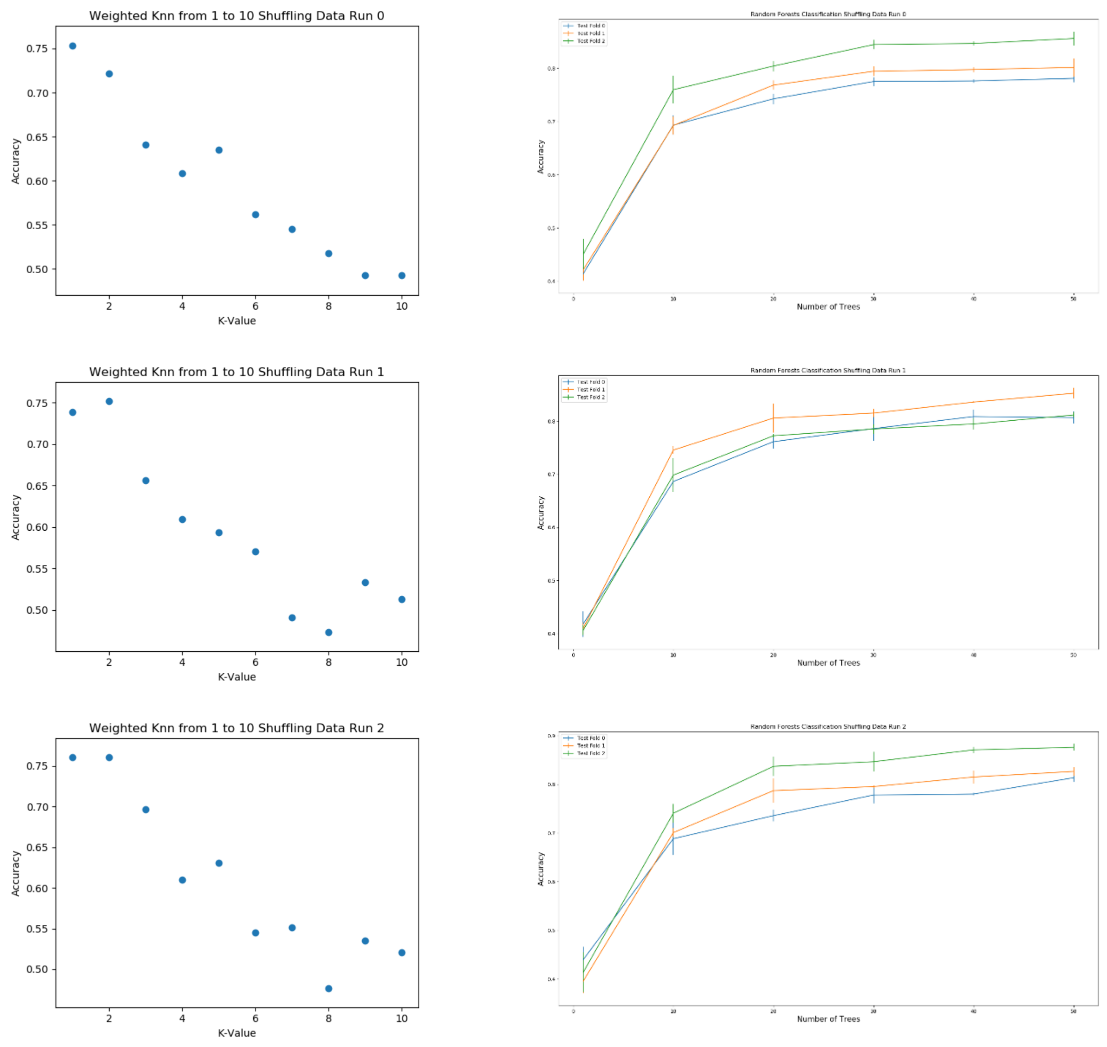

Figure 5: The graphs above depict the results when the data was shuffled and ran with a 70/30 percent split. Ran 3 times. The number of trees used ranged from 0 to 50.

From looking at Figure 5, we can see that the same trends from Figures 1 through 4 are shown here as well. The KNN graph gives us an accuracy of 70 – 75 percent. For the random forest however, the accuracy now is at approximately 80 percent with the number of trees at 50. This is a huge decrease from our results in Figures 1 – 4. We did use less test folds so that may have something to do with the decrease. However, we are using more data, and the split between the training and testing sets are random. There might be occasions where someone's hand makes up more of the training than testing and vice versa which might decrease our accuracy.

3.) Are there any issues with your preprocessing or feature extraction? What are they? Why are they issues? What is your plan for future development in project?

We've had several issues preprocessing our images. The first thing we noticed in our dataset vs other datasets is that our images are clear and have little noise. In other images people left in their shoulders, arms, and various other background noise. We decided to create some preprocessing algorithms which try to find the hand and crop around it, this worked effectively for hand symbols which were condensed, but for letters like 'L' or 'G' part of the finger strays away from the condensed mass so we lost a part of our fingers in preprocessing. It was difficult to find the correct libraries which had things we could use for this project. We haven't previously done any work with feature extraction or preprocessing, and most of the articles and research papers we found were already advanced enough that they didn't really need to explain what they were doing.

Another difficulty with preprocessing was finding a data structure that we could use that was compatible with the libraries that we found. We settled with a 3d matrix, but encountered some issues flattening it down for other libraries.

We looked at several feature extraction algorithms, mainly "SURF", "SIFT", "ORB", edge extraction, and others. Surf and Sift became patented so they were removed from the opencv library but were replaced by a new and improved "ORB". We found several orb implementations, namely on opencv and scikit-learn image. We ran into an issue with the opencv version because there was a lack of documentation regarding how the images should be loaded into Orb. With the opencv version, we found that images had to be grayscale. The opencv version also required images to vary in contrast between pixels. We will continue looking at the opencv version of ORB as well as other feature extraction techniques that we haven't seen yet.

For the future we plan on exploring more preprocessing and feature extraction algorithms and how we could modify them to work for our specific project. There has already been a lot of research done on hand symbol recognition, but that uses video feed and background subtraction strategies to get the hand. We also will look into using Convolutional Neural Nets once we begin using PyTorch and learn more about them. It seems that CNN's are the modern way of dealing with problems like this, and since PyTorch has much more support than the libraries we were looking at, there might be some new possibilities for tackling this problem.