For this project I was given a dataset (FLIR Thermal Data) that contains 5k images from 5 classes [‘person’, ‘sign’, ‘bike’, ‘bus’, ‘car’]. The files that I was given will contain the 256 features that are extracted from all the provided 5k images (using CNN) and the image folder which contains 5 files for the different classes to look up images to those specific data sample.

For the beginning of this project, I will be using unsupervised training to see if I can reduce my features, use PCA, and find trends that might increase my ARI and Silhouette Scores which will give me more ideas on how this preprocessing will help me when I select 4 different types of classifiers later. I will be using python in Jupyter notebook.

To first start off on my project I went ahead and uploaded the data and image files to my notebook. I looked at no scaling at first with SSE vs Number of Clusters to get an idea on how many clusters I might need.

**Figure 1**

A graph with blue dots

Description automatically generated

Just based on the figure 1 display I use the ‘elbow’ method and decided that cluster 8 will be the cluster I am going with. I then printed out the silhouette scores to get an idea of my clusters.

**Figure 2**

A graph of different colors

Description automatically generated

Cluster 4 and Cluster 2 appear to have big separation, but the other cluster don’t have good silhouette scores which makes me think that using unscaled data with no feature selection or even PCA is going to give me optimal ARI scores.

When checking ARI I got .7523 with clusters at 8 with unscaled which is pretty good I believe considering I kept all features and have not normalized anything to see if it can change. However, I am trying to optimize my ARI and Silhouette scores so I will be mixing up my parameters. When printed out my clusters with its 5 core samples and 2 boundary samples I noticed that Cluster 2 is person and Cluster 4 is sign which shows that it might be easier to distinguish those clusters. Some of the clusters like in Cluster 8 have a bike with a bike on it which might be confusing to distinguish which ones which so have a sub section is also practical. I also observe that car is easier to spot on the images compared to busses. Even for boundary samples the images appear to be blurry.

I was missing 940 images from Class 5(signs) so I only displayed those images on what I had to display. I do wonder if I start to change my parameters if the images, I get are off better quality since I am getting core and boundary which could mean that the blurriness on some of the images can be improved since it might select blurry photos for core now.

I will include on my PDF files I created to look at with no normalization and then my final pdf file with my features, PCA etc. at end of the unsupervised training section of this report.

Below are just two core samples examples of different clusters to get an idea on what I am discussing. Figure 3 and Figure 4 as reference.

**Figure 3**

A close-up of a sample

Description automatically generated

**Figure 4**

A screenshot of a computer

Description automatically generated

Next, I wanted to normalize my data using min\_max and standard\_scandard scaler to see If my clusters might change in numbers or if my ARI and Silhouette Scores might increase or decrease.

**Figure 5**

A graph with a line

Description automatically generatedA graph with blue dots

Description automatically generated

Clusters appear to be the same with 8 being the drop off. Even my Silhouette scores appear to be like unscaled data. Figure 6 as reference.

**Figure 6**

A chart of different colors

Description automatically generatedA chart of different colors

Description automatically generated

However, min\_max does have a slightly better ARI with a score of .76433 compared to standard\_scaler with .75998 and unscaled .76020.

I then tried DBSCAN clustering and went with min\_max and standard\_scaler due to wanting to keep to normalizing my data. I varied my EPS values from .05, .1, .2, .4, .5, .7, .8, .9, 1.0, 1.1 and my min\_samples\_values from 2,3,5,10,15, and 20.

I printed out my top ten DBSCAN scores based on ARI and my silhouette scores, and ARI scores are low compared to K-means clustering. Based on the results below in Figure 7 I will be using K-means clustering to see if I can improve that score while PCA and feature selection.

**Figure 7**

A screenshot of a computer

Description automatically generated

When using standard scale on my data and using PCA on my features I was able to print out an image that shows me the variance of data that is covered within the increment increase of PCA. When I used min\_max it was a similar PCA variance to standard scaler.

**Figure 8**

**A graph with a blue line

Description automatically generatedA table with numbers and letters

Description automatically generated**

I will be using PC3 and PC4 to test on my ARI and silhouette scores with just PCA without feature selection yet. PC4 seems to be the beginning of the peak where my cumulative does not increase too much after that.

When selecting min\_max as my parameter to run for PC3 and PC3. I get PC3 with an ARI score of .714159 and silhouette score of .414627. For PC4 I get an ARI of .764595 with a silhouette score of .475295. Based on these results keeping all the features tends to outperform or be equal to keeping all features without PCA.

I will now try using feature selection and then PCA with that feature selection to see if I can increase my ARI and silhouette scores. I checked to see if any variance was 0 and there was not in the data.

Next, I used unscaling, min\_max scaling and standard\_scaling to see if feature selection will make a difference. I varied my cluster options to 5,6,7 and 8 and feature counts to 10,20,40,50 and 100.

When looking at my output I get .70, .80 or .90 ARI scores now with silhouette scores of .40 to .53 with the majority of all the parameters above. Just based on this I will be selecting Feature counts of 50 and 100 with PCA components of 2,3, and 4 as well as cluster of 5 and 6. I will print out figures and explain what gave me the best results.

**Figure 9**

**A graph with colored lines and numbers

Description automatically generated**

**Figure 9**

**A graph with a line graph

Description automatically generated with medium confidence**

Just based on Figure 8 and Figure 9 my ARI scores have approved big time compared to what I have had so far with Feature Reduction with PCA components. I get an ARI score of .94 with Features 100, PCA 4, cluster 5 with min\_max scaling. This will be what I will be going off of as my parameters when I start to select my classifier types.

**Figure 10 Figure 11**

**A graph with colored lines and numbers

Description automatically generatedA graph with lines and numbers

Description automatically generated**

Even based on my silhouette scores I will still be picking features 100, PCA 4, Cluster 5 with Min\_Max as my best parameter off the top when I start to pick my classifiers. Below in Figure 12 is my silhouette scores from my best parameters for my clusters. I get good cluster scores that range from .5 at the worse to .7 at the top.

**Figure 12**

A graph of different colors

Description automatically generated

I printed the visualization of PCA 3 and PC4 at 2d and 3d to get a feel of what kind of classifier types I might use.

**Figure 13**

A graph of clustering colors

Description automatically generated with medium confidenceA diagram of clustering dots

Description automatically generated with medium confidenceA diagram of a cluster of dots

Description automatically generated with medium confidence

Based on just looking at the visualization I will be Kernal SVM, Random Forest, Logistic Regression and KNN. However, I was given harder extra\_hard\_samples.csv file so now I will be training and testing that data with the parameters above.

Before training my SVM model, I selected 70 samples of each class from Extra\_hard\_samples.csv and combined them to my data.csv data. I made sure all the columns matched.

I will be doing an 80/20 split when training and testing my data. I will use PC4 with 100 top features based on MI scores. I got the best results in my preprocessing steps with the above features. When testing and training my data I will use both standard\_scaler and min\_max scaler.

**SVM**

I will be using k-fold = 4 with parameters set at C\_values = [0.01, 0.1, 1, 10], gamma\_values = [0.01, 0.1, 1, 10], kernels = ['linear', 'rbf'] and scalers = { "MinMaxScaler": MinMaxScaler(),

"StandardScaler": StandardScaler( }. I set top ranked features at 100 based on MI scores and PCA components set at 4.

When running my results I will be outputting Average Train Accuracy, Average Cross Validation Accuracy, F1-Score, AUC-ROC and Time Taken.

**Figure 14**

A graph with green and blue lines

Description automatically generated

When looking at figure 14, It appears that minmax appears to be more stable in average training accuracy and validation accuracy compared to Standard Scaler. Standard Scaler has higher training accuracy scores in comparison to min\_max. However, validation scores appear to be similar. Linear kernel has better results on validation scores. Good generalization occurs unless its kernel is RBF with low C of .01. This is not always the case, but it does occur more often.

**Figure 15**

**A graph with orange and purple lines

Description automatically generated**

F1-Score is the measure between precision and recall which is very important which trying to balance these metrics importance. For AUC-ROC this is a good measure on how well your model can predict correctly. Based on observation in Figure 15 AUC-ROC will always be higher than F1-score. I also observe that Figure 15 when compared to Figure 14 shows me that usually when Training and Validation accuracy are high or low it correlates with AUC-ROC and F1- Score.

Based on my observation I will select C = 1.0, Gamma = Auto with Linear Kernaal and Min\_Max Scaler. High overall on all my parameters. However, I will try bagging to see if I can increase my scores.

For bagging SVM I will use the same parameters I had prior but now I will be adding bagging classifier with additional parameters. n\_estimators = [5, 10, 25] and max\_samples = [0.2, 0.5, 0.7].

Figure 16 will show output of all my parameters I ran when coming training and validation accuracy. When looking at this chart in Figure 16 it is really bunched together so I will separate them later. I also notice I get a lot higher training accuracy with bagging; I also get slightly higher validation accuracy. On the other side I get overfitting a lot more often with bagging which can happen with bagging.

**Figure 16**

A graph showing a graph of data

Description automatically generated with medium confidence

When looking at figure 16, I did not notice anything unusual until I looked at min\_max and standard\_scaler differences. I observed that the accuracy and validation still correlate with AOC-ROC and F1-Scores, I also noticed that standard scaler did not do as well in generalization since it appeared too always overfit. Figure 17,18,19 show this.

**Figure 17** **Figure 18**

A graph of training and validation accuracy

Description automatically generated A graph with numbers and lines

Description automatically generated

**Figure 19**

**A graph with numbers and lines

Description automatically generated**

Just based on my results, I will not use bagging since it was like not using bagging. My validation accuracy was around 92 to 93 percent which is what I had without bagging. I could get higher training accuracy, but I like my training and validation accuracy to be closer to each other. I will be selecting no bagging with the parameters shown here. C = 1.0, Gamma = Auto with Linear Kernal and Min\_Max Scaler.

**KNN**

For KNN I used the parameters as stated below.

k\_values = [1, 3, 5, 7, 10, 13, 16, 20, 25, 30, 50]

weights = ['uniform', 'distance']

metrics = ['euclidean', 'manhattan']

scalers = {

"MinMaxScaler": MinMaxScaler(),

"StandardScaler": StandardScaler()

}

I kept the feature count and PCA the same as SVM as well as the k-fold=4. Figures 20, 21,22,23 will visualize my min\_max and standard scaler training/validation accuracy, AUC-ROC, F1 Scores.

**Figure 20 Figure 21**

A graph with blue and green lines

Description automatically generatedA graph with blue and purple lines

Description automatically generated

**Figure 22 Figure 23**

A graph with blue and green lines

Description automatically generatedA graph with blue and purple lines

Description automatically generated

For figures 20-23, I have K parameter set up with its unique parameters as well. Its seperated by min\_max and standard scaler. While looking at all these features, both standard scaler and min\_max performed similiar. I have overfitting when all normalization have a weight that is distance. Uniform performs better on generalization. I will select K = 7 with min\_max scaler, with metric of manhattan and weight as uniform. Even on AUC-ROC and F1-Scores I appear to stabilize at K=7.

**Random Forest**

I will continue using the same parameters as before with k-fold, PCA and feature selection. However, with Random Forest Classifier I will use N\_estimators set at 2,3,5,7,10,25,50,100 and Max\_depths set at 2,3,4,5,10,20. Normalization will be the same as before with min\_max and standard\_scaler.

**Figure 24 Figure 25**

A graph with blue and green lines

Description automatically generated**A graph with different colored lines

Description automatically generated**

**Figure 26 Figure 27**

**A graph with blue and green lines

Description automatically generatedA graph with different colored lines

Description automatically generated**

Just based on observation from the figures 24-27, Min\_max and standard\_scaler once again have similar results. I tend to overfit a lot more often than my previous two classifiers. More often than naught, when my max depth is 10 and above, I get overfitting. This would make sense since I am able to fit more into my training model which improves my training accuracy but is not able to perform as well to cross validation compared to its training. I will be using min\_max since it performs as well as standard\_scaler. N\_Estimator = 3 is where I start to stabilize on all my generalization as well as my AUC-ROC and F1-Scores. Since this is where I don’t really have any huge improvements after this. I will stick with N\_Estimator = 3. With N= 5 and max\_depth = 5, I get my best results on all metrics. Figure 28 visualizes this. I will select these parameters moving forward for Random Forest.

**Figure 28**

A screenshot of a graph

Description automatically generated.