Final Project: Exploration of Unsupervised Learning and Binary and Multi-Class Classification through the Fashion-MNIST Dataset

Emmy Castator, HyoJung Kang, and Youngseo Ko

School of Statistics

University of Illinois

Urbana, Illinois 61801 USA

ecasta9@illinois.edu, arielkang1@gmail.com, and ko37@illinois.edu

Project Description and Summary

The objective of this project is to utilize the Fashion-MNIST dataset, comprised of 60,000 training observations and 10,000 testing observations associated with different types of clothing images, to test model performance through binary and multi-class classification. Each image for all ten classes is composed of 28 by 28 pixels, producing 784 pixels in aggregate. Images are classified as a t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, or ankle boot which respectively correspond to a digit from 0 to 9.

Initial investigation of the data will be accomplished with unsupervised learning tactics using Principal Component Analysis (PCA) and the Self-Organizing Map (SOM) clustering algorithm. These approaches will be useful in acquiring information before models are built from the dataset.

Primary predictive measures include binary classification using solely coat and shirt labels, associated with labels 4 and 6. Models with high predictive power will be obtained through model performance tuning based on sensitivity and specificity criterion using cross-validation. This will be further validated with visualizing the Receiver Operating Characteristic (ROC) curve and subsequently evaluating the Area Under the Curve (AUC) criterion. Multiple classification models will be chosen to fit the training data, namely Penalized Logistic Classification with Elastic-Net methods as a combination of Lasso and Ridge penalties. Additionally, Support Vector Machine (SVM) models will be implemented as an alternative approach. Multi-class classification will be performed with all clothing types to assess the predictive power of different models. These models include K-Nearest Neighbors, Linear Discriminant Analysis, and Random Forests. The best-performing model for the binary and multi-class models separately will be chosen based on AUC criterion and testing accuracy.

Literature Review

Recognition of images through machine learning is an integral facet across disciplines. Many machine learning algorithms and deep learning networks are instrumental in improving image classification accuracy. The greatest difficulty in classifying clothing types arises from the "richness of the clothes properties and the high depth of clothes categorization (Kayed et al. 2020). Currently, the best accuracy modeled on this dataset is 96.91 with a 3.09 percent error (Tanveer et al. 2020). Such high accuracy was achieved with Fine-Tuning DARTS for Image Classification.

Greeshma and Sreekumar researched classification using the dataset based on Histogram Oriented Gradient (HOG) feature descriptor using SVM. These researchers honed in on using the dataset to train models in hopes of mitigating the issues in driverless car technology geared towards classifying nearby objects as pedestrians or cars. HOG is a simple and effective method for object detection in image processing and computer vision. SVM was implemented after HOG as a multi-class classification tool on the reduced feature space defined by HOG. Evaluation on the testing data produced a classification accuracy of 0.8653 (Greeshma and Sreekuman 2019).

From a more consumerism-directed approach, many fashion businesses have used Convolution Neural Network (CNN) to solve problems regarding clothing recognition for clothes searching and recognition (Kayed et al. 2020). CNN is an artificial neural network mainly used for "image processing, classification, segmentation, and others" (Kayed et al. 2020). These researchers built a model for CNN-LeNet-5 which achieved a high accuracy of 0.988 tuned on the parameters for learning rate, the number of training set samples, and the number of times the algorithm processes a complete dataset. Notably, the high accuracy came at the expense of high computational cost for a test-set run-time averaging 80 minutes.

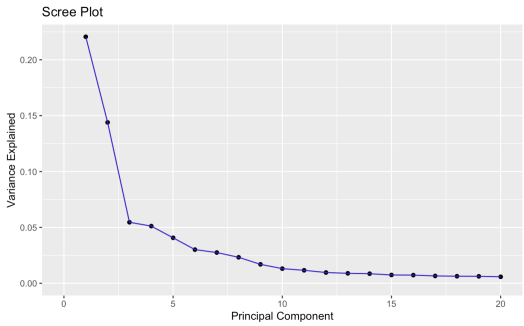
Summary Statistics: Data Processing and Unsupervised Learning

A frequency table of the outcome table for both the training and testing set was found for introductory exploration of the data. The data is balanced between each digit and therefore will not lead to issues for prediction in future analysis.

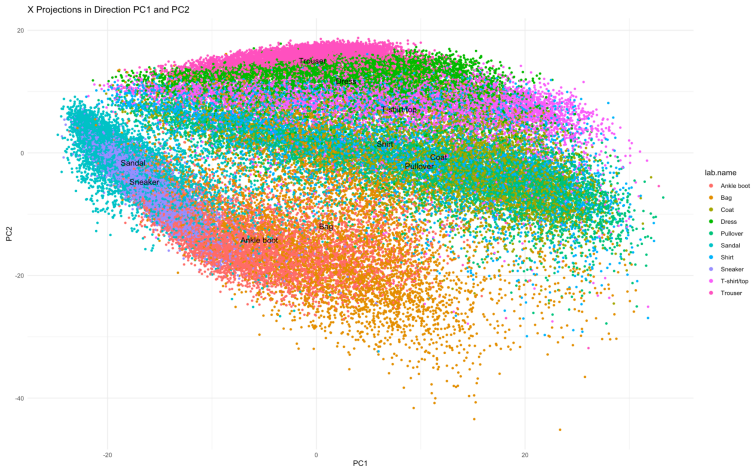
```
   0    1    2    3    4    5    6    7    8    9
6000 6000 6000 6000 6000 6000 6000 6000 6000 6000

   0    1    2    3    4    5    6    7    8    9
1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
```

PCA was initially performed on the centered and scaled observations for dimension reduction and visualization purposes. The eigenvalue scores of all 784 principal components were found and the first 20 were plotted in a scree plot to visualize the percentage of data explained by each component. The first principal component explains 22% of the variation within the dataset, and the second component explains 14%. There is a drop-off at around 12 components and upon further calculation, approximately the first 200 components are needed to explain 93.6% of the variation. This is evidence that there exists redundancy within the data which can be accounted for through marginal screening.

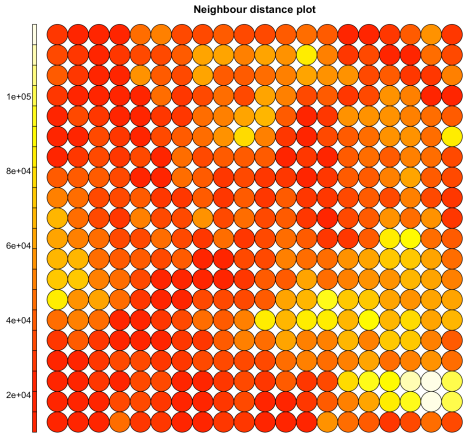


To visualize how well the first 2 principal components separated the data, the data with their respective labels was plotted in the first 2 dimensions. The text labels are assigned to each clothing type and their placements are based on the mean of the first 2 principal components. Category clustering is

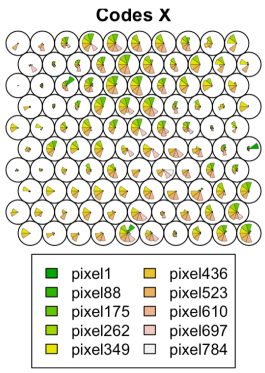


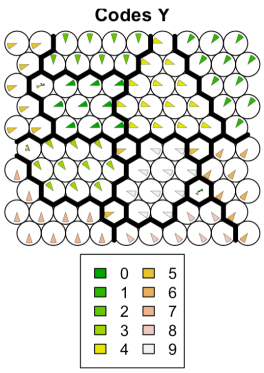
evident as the shoe types are clumped together on the left side of the plot and the bag is grouped more towards the center. Further, long-sleeve items, shirt, coat, and pullover, aggregate together while trouser sits atop the dress and top group in its own clump. This creates about 6 clusters.

The clustering algorithm for the Self-Organizing Map was performed on the training data, using 20 randomly chosen pixels to define each image. The neighbor distance plot illustrates the distance between neighbors with darker nodes indicating a smaller distance. This introduces a natural boundary between node clusters and from this, the number of clusters visually appears to be approximately 6 while the true number of clusters is known to be 10. K-means clustering was implemented to determine the optimal number of clusters where it was found that 10 clusters produced the smallest total variation. Thus, 10 clusters were used for future evaluation.



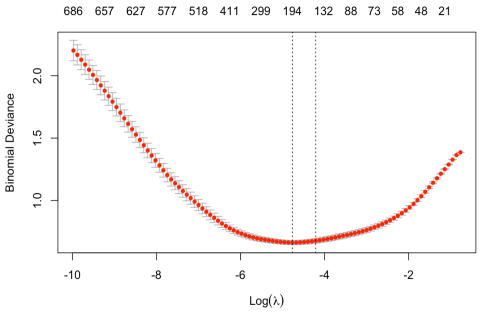
Next, the node weight vectors were plotted using 10 randomly chosen pixels, where each vector for the node's weight is representative of the samples mapped onto that node. The clustering boundary is created through supervised SOM with the true number of clusters being 10. The Codes Y plot displays the frequency for each cluster and each cluster has a pure outcome for each label 0-9. Therefore the dominating y-label for each cluster corresponds to its respective 0-9 label in all 10 clusters. Even without the borders, there is a clear pattern in the distribution of y-labels. Thus, the clusters unequivocally help to separate the labels.

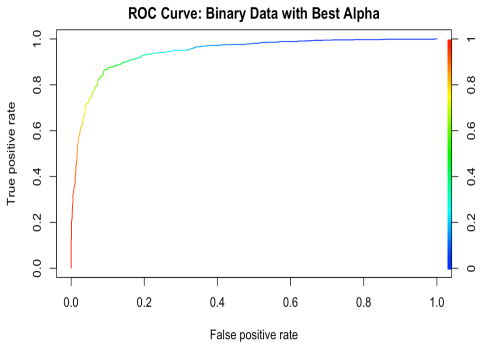




Binary Classification: Coat vs. Shirt

As indicated through PCA, coat and shirt labels clump together which may lead to issues when distinguishing between them. This can be verified through binary classification of the coat and shirt labels. Additionally, models were fit on 2000 randomly chosen observations from the training data to lower computational cost. Binary Logistic Classification with Elastic Net penalty was chosen to train the model using variable selection through Lasso and accounting for multicollinearity issues with Ridge. The initial model was built on all pixels in the training set and was tuned to find the best α and λ values through 10-fold cross validation with α sequencing from 0 to 1 on 0.2 increments. Here it was found that an α value of 0.4 produced the lowest cross-validation error of 0.663. Since Lasso is specified by an α value of 1 and Ridge is specified by an α value of 0, this model drifts more towards a Ridge penalty. The minimum λ value was found to be 0.0085 based on the binomial deviance as seen by the cross-validation plot. A logistic model was then built on these tuning parameters and the ROC curve, AUC, and testing accuracy were calculated. The AUC was 0.9432 which indicates that this model has high specificity and sensitivity. Further, the testing accuracy was 0.883, calculated from the confusion matrix, which indicates that this model has high predictive power.

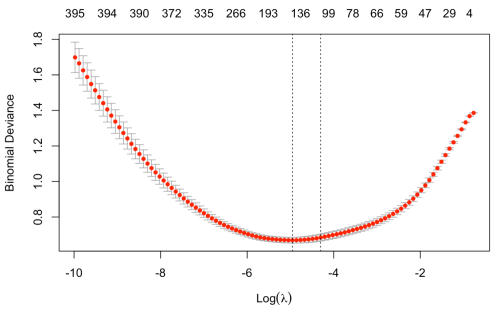




Confusion Matrix: ENet All Pixels

	4	6
FALSE	889	123
TRUE	111	877

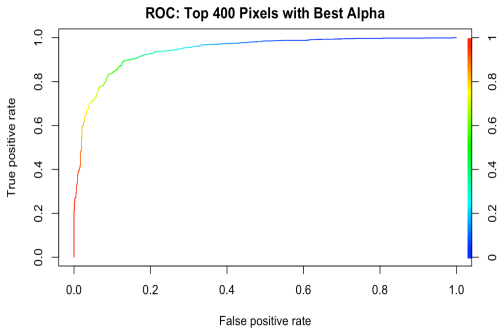
Marginal screening was subsequently incorporated to remove pixels with low variation as those pixels do not provide useful information. As demonstrated through PCA, low information



pixels pose redundancy to the dataset. The variance of all pixels was calculated, ranked, and the 400 pixels with the highest variance were selected. Next, a model was built using the top 400 variance pixels and tuned using the elastic net sequence of α values with 10-fold cross-validation. This resulted in the best α of 0.4 and a minimum $\boldsymbol{\lambda}$ of 0.0071, and the AUC decreased slightly to 0.9419 as did the testing accuracy to 0.875. Finally, a logistic model with a Lasso penalty of α equal to 1 was fit on the top 400 pixels to determine if it performed better based on testing accuracy criterion. The model produced a testing accuracy of 0.88 which performed slightly better than the marginally screened elastic net model but slightly worse than the elastic net fit on all pixels.

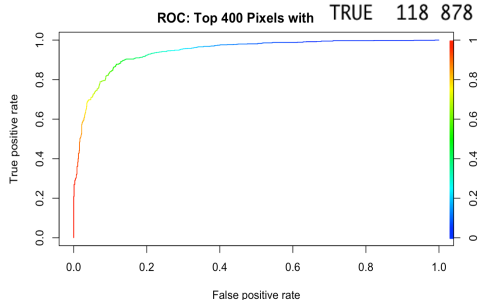
Confusion Matrix: ENet Top 400 Pixels

	4	6
FALSE	879	130
TRUE	121	870



Confusion Matrix: Lasso Top 400 Pixels

	4	6
FALSE	882	122
TRUE	118	878



Support vector machine (SVM) was chosen for model building as it is better suited for 2-class problems while random forests is intrinsically suited for multi-class problems. Since SVM optimizes the distance between observations, it will function well with large distances among points. SVM models were then built on the marginally screened data using the 2000 randomly chosen training observations to determine if they performed better than the penalized logistic models based on testing accuracy criterion. The first model with SVM pre-scaled and centered the data and used a linear kernel tuned on 5 different cost values ranging from 0.01 to 3.5 using 10-fold cross-validation. Cost is needed to introduce balance to the model which is

significant in non-separable problems. A cost value of 0.01 produced the highest training accuracy of 0.8605. This is a small cost value which indicates that the cost for having a wrong classification is low. A model was then fit using a linear combination for the kernel and the cost equal to 0.01. Testing accuracy for this model was found to be 0.878 as calculated from the confusion matrix.

Confusion Matrix: Linear SVM Top 400 Pixels

pred	4	6
4	879	123
6	121	877

A more flexible form of SVM was then implemented using a radial kernel. This is advantageous as this method introduces more possibilities for model fitting using non-linear processes. This model was tuned using 5-fold cross-validation and values 0.1 and 5 for the cost with values 0.01 and 0.05 for sigma. The highest training accuracy of 0.8775 was produced with a cost value of 5 and a sigma value of 0.01. A cost of 5 is relatively high and therefore miss-classification is penalized more heavily. Sigma was evaluated to be 0.01 which is low, resulting in a decision boundary that is more linear. A model was then built on this cost value and sigma value to evaluate the testing accuracy. From the confusion matrix, the testing accuracy was established to be 0.909.

Confusion Matrix: Radial SVM Top 400 Pixels

pred	4	6
4	899	82
6	101	918

Conclusively, the model built on SVM using a radial kernel with a cost of 5 and sigma of 0.01 produced the best model based on testing data accuracy for binary classification. This model produced a testing accuracy of 0.909 and classification error of 0.092. Overall this model has high predictive power and performs well on the data.

Multi-Class Classification

For classification of all labels, the entire dataset was used, subset again on the top 400 pixels. Subsetting measures were further incorporated in randomly choosing 10,000 observations from the training data, in order to reduce run-time. Firstly, a k-Nearest Neighbors (kNN) model was fit on the training set using 5-fold cross-validation with 3 repeats to determine the optimal number of neighbors based on the value that was associated with the smallest root-mean-squared error. The tuning parameter, defined by the number of neighbors, ranged from 2 to 12, incremented by 2, landing on 8 as the best value for the model. An 8-NN model was then fit on the data which resulted in a training time of 3.861 minutes. The classification accuracy is 0.842, as calculated from the confusion matrix. Moreover, label 4 and label 6 had the lowest classification accuracy. k-NN for classification utilizes majority voting rather than averaging as in k-NN for regression. Therefore, this model used majority voting to classify each label based on the 8 closest neighbors. For classification purposes, this model performs well and is relatively computationally efficient.

```
k-Nearest Neighbors

10000 samples
  400 predictor

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 7999, 8000, 8001, 8000, 8000, 7998, ...
Resampling results across tuning parameters:

  k   RMSE      Rsquared   MAE
   2  1.433844  0.7652153  0.6622299
   4  1.332470  0.7917898  0.6789272
   6  1.304838  0.7988384  0.6940865
   8  1.295678  0.8010069  0.7088263
  10  1.297046  0.8002683  0.7222693
  12  1.300474  0.7990086  0.7330326

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 8.
```

```
Confusion Matrix and Statistics


knn.fit   0   1   2   3   4   5   6   7   8   9
      0 883   8  15  50   8   2 235   0   5   0
      1   0 968   2  19   3   0   3   0   1   0
      2  24   4 824  17 172   0 152   0  14   0
      3  11  14   7 870  32   1  18   0   3   0
      4   3   1  80  32 721   0  79   0   4   0
      5   0   0   0   0   0 796   0   8   1   3
      6  67   5  59  10  58   4 499   0   5   0
      7   1   0   0   0   0 108   0 944   5  43
      8  11   0  13   2   6  13  14   2 961   1
      9   0   0   0   0   0  76   0  46   1 953

Overall Statistics

               Accuracy : 0.8419
                 95% CI : (0.8346, 0.849)
    No Information Rate : 0.1
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8243

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8 Class: 9
Sensitivity            0.8830   0.9680   0.8240   0.8700   0.7210   0.7960   0.4990   0.9440   0.9610   0.9530
Specificity            0.9641   0.9969   0.9574   0.9904   0.9779   0.9987   0.9769   0.9826   0.9931   0.9863
Pos Pred Value         0.7322   0.9719   0.6827   0.9100   0.7837   0.9851   0.7058   0.8574   0.9394   0.8857
Neg Pred Value         0.9867   0.9964   0.9800   0.9856   0.9693   0.9778   0.9461   0.9937   0.9957   0.9947
Prevalence             0.1000   0.1000   0.1000   0.1000   0.1000   0.1000   0.1000   0.1000   0.1000   0.1000
Detection Rate         0.0883   0.0968   0.0824   0.0870   0.0721   0.0796   0.0499   0.0944   0.0961   0.0953
Detection Prevalence   0.1206   0.0996   0.1207   0.0956   0.0920   0.0808   0.0707   0.1101   0.1023   0.1076
Balanced Accuracy      0.9236   0.9824   0.8907   0.9302   0.8494   0.8973   0.7379   0.9633   0.9771   0.9697
```

Next, Discriminant Analysis was applied. Linear Discriminant Analysis (LDA) is beneficial in this high dimensional setting due to its rapid run-time. The LDA model was fit over all pixels with 30,000 randomly selected observations from the training set. No tuning was

applied to this model, it was more so used as a baseline for run time. The resulting testing accuracy was 0.8232 and a run-time of 1.61 minutes. This is a quick run-time but increased testing accuracy outweighs increased computational cost. This accuracy is lower than 8-NN, so a better model with improved classification power is needed.

```
Confusion Matrix and Statistics

          0   1   2   3   4   5   6   7   8   9
  0 768   0  17  71   2   3 124   0  15   0
  1   2 940  10  36   0   1  11   0   0   0
  2  21   0 712   9 150   2  95   0  11   0
  3  23   4  22 878  24   4  45   0   0   0
  4   0   1  75  29 785   1 107   0   2   0
  5   1   0   0   0   0 884   3  71  13  28
  6 150   0 101  45  89   3 592   0  20   0
  7   0   0   0   0   0  70   0 855   0  75
  8   2   0   6  11   4  17  45   4 910   1
  9   0   0   0   0   0  40   0  52   0 908

Overall Statistics

               Accuracy : 0.8232
                 95% CI : (0.8156, 0.8306)
    No Information Rate : 0.1079
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8036
```

Random Forests was the final model that was investigated for the multi-class classification problem. The data used to tune the model was subset again on the top 400 pixels and further reduced to 5000 randomly chosen training observations. The ranger package was used to tune the model as it is a computationally fast process. As tuning measures, 5-fold cross-validation alongside 2 values for minimum node size, 5 and 10, coupled with 2 values for the number of variables randomly sampled as candidates at each split, 4 and 6, were integrated to tune the model. The number of trees was fixed to be 200.

```
Random Forest

5000 samples
 400 predictor
  10 classes: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 3999, 4000, 3999, 3999, 4003
Resampling results across tuning parameters:

  mtry  min.node.size  Accuracy   Kappa
  4      5            0.8156046  0.7950430
  4     10            0.8146054  0.7939367
  6      5            0.8203994  0.8003747
  6     10            0.8224030  0.8026029

Tuning parameter 'splitrule' was held constant at a value of gini
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were mtry = 6, splitrule = gini and min.node.size = 10.
```

Chosen based on the highest accuracy criterion, an mtry value of 6 combined with a minimum node size of 10 produced an accuracy of 0.8224. This is consistent with the information provided through the clustering algorithms as it was verified that using 10 clusters produced the smallest error. Next, a model was built with the best tuning parameters, on the top

```
Call:
 randomForest(x = rf.X, y = as.factor(rf.y), xtest = x.test, ytest
               Type of random forest: classification
                     Number of trees: 200
No. of variables tried at each split: 6

        OOB estimate of  error rate: 15.25%
Confusion matrix:
     0    1    2    3    4    5    6    7    8    9 class.error
0 1709    4   46   71   16    0  124    0   20    0  0.14120603
1    3 1903   13   52   13    0   13    0    2    0  0.04802401
2   18    0 1577   12  222    0  148    0   19    0  0.20991984
3   63    5   19 1782   61    0   58    0    9    0  0.10766149
4    5    5  262  117 1494    0  146    0   11    0  0.26764706
5    0    0    0    1    0 1839    1   96    9   36  0.07214934
6  389    6  291   48  171    1 1082    0   44    0  0.46751969
7    0    0    0    0    0   73    0 1748    1  120  0.09989701
8    1    1   22    6   11    7   23    3 1897    2  0.03852002
9    0    0    0    0    0   62    0   68    1 1918  0.06393363
                Test set error rate: 15.08%
Confusion matrix:
    0   1   2   3   4   5   6   7   8   9 class.error
0 841   2  22  36   4   2  77   0  16   0       0.159
1   2 953  12  24   2   1   5   0   1   0       0.047
2   6   2 796   5 109   0  71   0  11   0       0.204
3  30   9  10 900  31   0  19   0   1   0       0.100
4   1   4 122  42 773   0  53   0   5   0       0.227
5   0   0   0   0   0 919   0  53   7  21       0.081
6 224   3 126  23  76   0 527   0  21   0       0.473
7   0   0   0   0   0  36   0 892   0  72       0.108
8   2   1  14   4   2   3  11   5 957   1       0.043
9   0   0   0   0   0  21   1  41   3 934       0.066
```

400 pixels for 20,000 randomly selected training observations and 10,000 randomly selected testing observations. Training time ran for 2.872 minutes which is relatively low considering the large number of observations. Again, the highest classification error is produced by labels 4 and 6. From the confusion matrix of the random forest model, the classification accuracy of the testing data is 0.849. This is the highest accuracy obtained across all multiclass classification models and therefore is the best model.

Residual findings reveal that labels 4 and 6, corresponding to coat and shirt respectively, consistently produced the highest classification error. Possible reasons for this include that these two clothing items habitually cluster together and therefore are classified almost identically. Labels 1, 5, 8, and 9, corresponding to trouser, sandal, bag, and ankle boot, have persistently low classification error. As evidenced by PCA and clustering algorithms, these clothing items never group together and are therefore more easily distinguishable.

Conclusively, the best model produced over multi-class classification was found to be Random Forests built on the top 400 pixels with a mtry value of 6, the minimum number of nodes equal to 10, the number of trees fixed at 200, with 20,000 training and 10,000 testing observations. Further, the classification accuracy of the testing data was evaluated to be 0.849.

References

Greeshma, K. V., & Sreekumar, K. (2019). Fashion-MNIST classification based on HOG feature descriptor using SVM. *International Journal of Innovative Technology and Exploring Engineering*, *8*(5), 960-962.

Kayed, M., Anter, A., & Mohamed, H. (2020, February). Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture. In *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)* (pp. 238-243). IEEE.

Tanveer, M. S., Khan, M. U. K., and Kyung, C. M. (2020). Fine-Tuning DARTS for Image Classification. *arXiv preprint arXiv:2006.09042.*