Image Classification Analysis

Mîrza Ana-Maria, 341C1
Faculty of Automation and Computer Science

Keywords: Machine Learning, Logistical Regression, Neural Network Model

1 General Description

This article describes the process of data analysis, processing, attribute selection and model training of images from two datasets in order to obtain a valid classification. The two datasets used are Fruits-360, a dataset containing 70 different fruits and vegetables and Fashion-MNIST, a dataset containing 10 different clothing objects.

2 Fruits-360

2.1 Data Analysis

The visual representation of this dataset is presented in the images below.

Figure 1: Top 10 fruits

This figure shows samples of images from the dataset with the 10 most frequent fruits.

Next, we want to see the frequency of each class in the dataset.

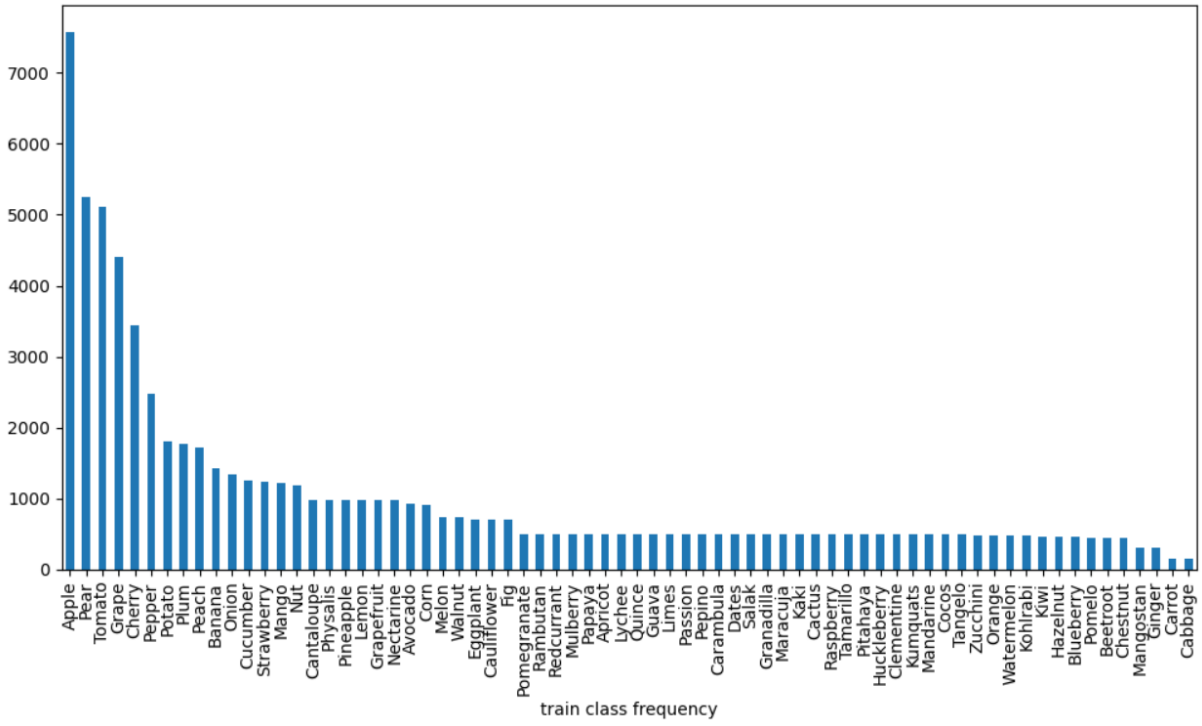

Figure 2: Class

We can see that the dataset is not balanced, which may cause the models to learn harder or to overfit for classes with more examples.

2.2 Attribute Extraction: PCA

The next step is attribute extraction. We will apply PCA (Principal Component Analysis) because it's one of the fastest methods of attribute extraction and we can directly control the number of features extracted based on the capacity of the machine we are training the model on.

2.2.1 Variance Analysis

First, we look at the variance graph in order to determine how many features to extract and not lose too much information.

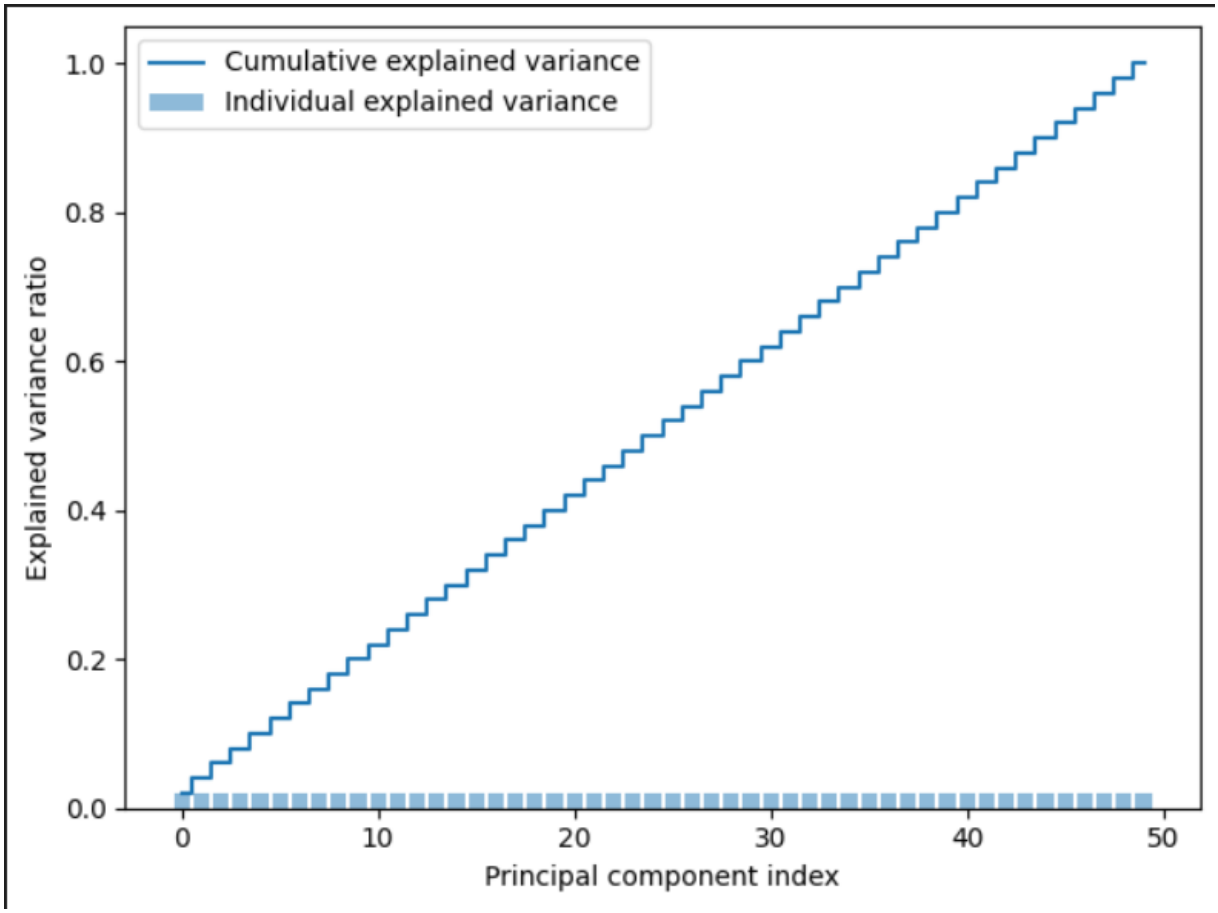

Figure 3: Class

Since we want to have over 90% of the information retained, we will extract at least 50 principal components from the images.

2.2.2 Normalization

Before applying PCA, the dataset needs to be normalized so that the features extracted will give better results. Below we have an image of the dataset scaled.

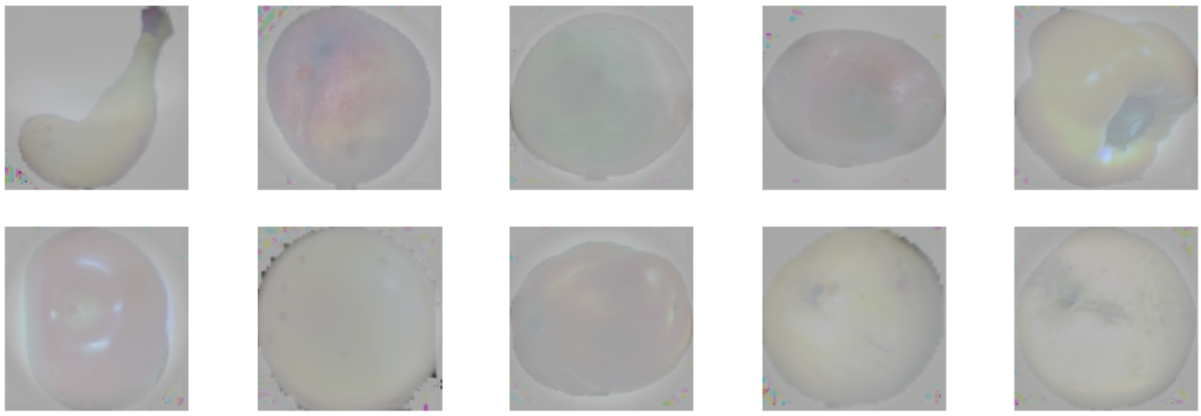

Figure 4: Class

2.2.3 Scaling and PCA

Because the dataset is big and there is not enough memory to load the whole dataset in the memory, we started by scaling the images from 100x100 to 20x20, and we applied PCA with 70 component extraction to get the following output:

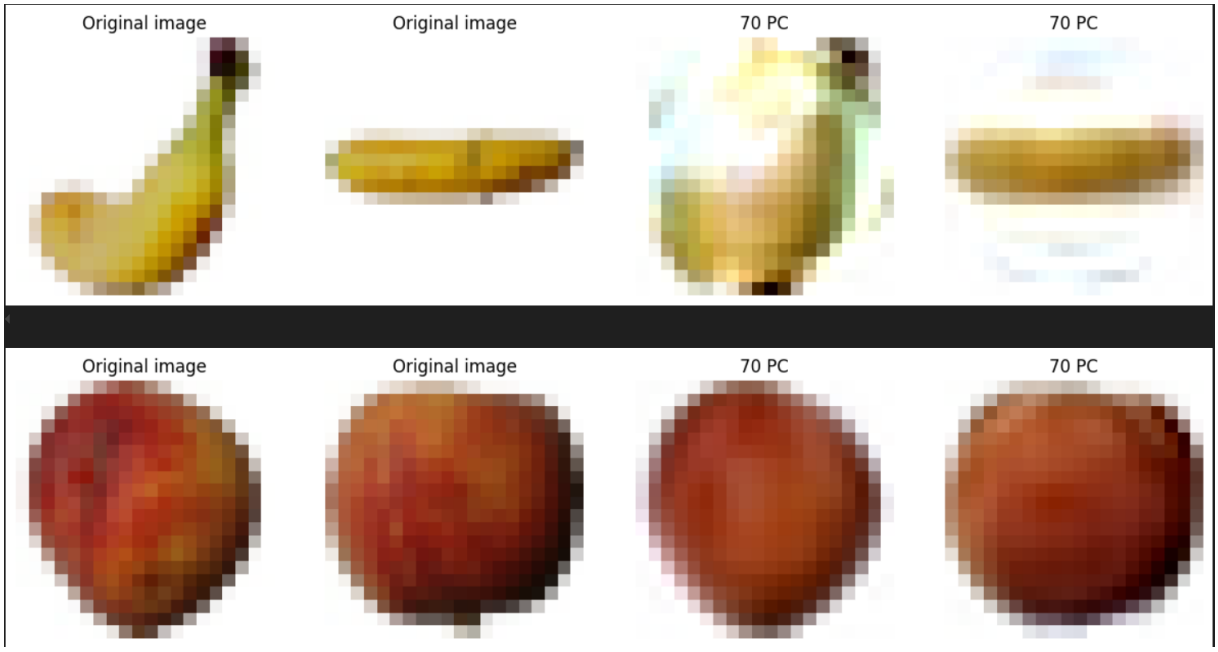

Figure 5: Class

The images obtained after extracting 70 principal components are very similar to the original ones, which will help the models to differentiate between the fruits easier.

2.3 Model Results - PCA

After testing the accuracy results of the models with several number of principal components extracted with PCA, we could observe that the scores were getting higher when the number of features extracted was larger. We will include the results of the 70 components extracted for several models trained.

The training dataset was split in 80% training set and 20% validation set for the hyperparameter search.

2.3.1 Logistic Regression

Below, we have the scores of logistic regression after applying Grid Search for hyperparameter tuning.

param_C	param_solver	param_multi_class	mean_test_score
1.000000	newton-cg	multinomial	0.863741
0.100000	newton-cg	multinomial	0.854058
1.000000	newton-cg	ovr	0.812957
0.010000	newton-cg	multinomial	0.811285
0.100000	newton-cg	ovr	0.804458
0.010000	newton-cg	ovr	0.768931

Figure 6: Hyperparameter Tuning

The hyperparameters we searched for were C, the solver, and multi_class. The highest score was obtained for C = 1.0, newton-cg as solver, and multinomial as multi-class, with an accuracy of 86% on the validation set.

	precision	recall	f1-score	support
Mangostan	0.50	0.19	0.27	102
Cherry	0.79	0.66	0.72	1148
Grape	0.68	0.78	0.73	1476
Nectarine	0.27	0.30	0.28	324
Kohlrabi	0.94	0.76	0.84	157
Physalis	0.88	0.94	0.91	328
Carrot	0.78	1.00	0.88	50
Melon	0.71	0.85	0.77	246
Tomato	0.66	0.65	0.65	1707
Potato	0.44	0.32	0.37	601
Apple	0.61	0.63	0.62	2525
Beetroot	0.83	0.47	0.60	150
Chestnut	0.62	0.69	0.65	153
Avocado	0.88	0.93	0.90	309
Pear	0.59	0.62	0.60	1761
Grapefruit	0.50	0.58	0.54	330
Kiwi	0.87	0.69	0.77	156
Nut	0.46	0.54	0.50	396
Cauliflower	0.53	0.73	0.61	234
Guava	0.97	0.89	0.93	166
Mulberry	0.99	0.99	0.99	164
Walnut	0.88	1.00	0.94	249
Pineapple	0.81	0.97	0.88	329
...				
accuracy			0.72	24051
macro avg	0.80	0.78	0.78	24051
weighted avg	0.72	0.72	0.72	24051

Figure 7: Class Score

Figure 7 presents the score obtained on the different fruit classes. The overall accuracy of the model is 71.78%.

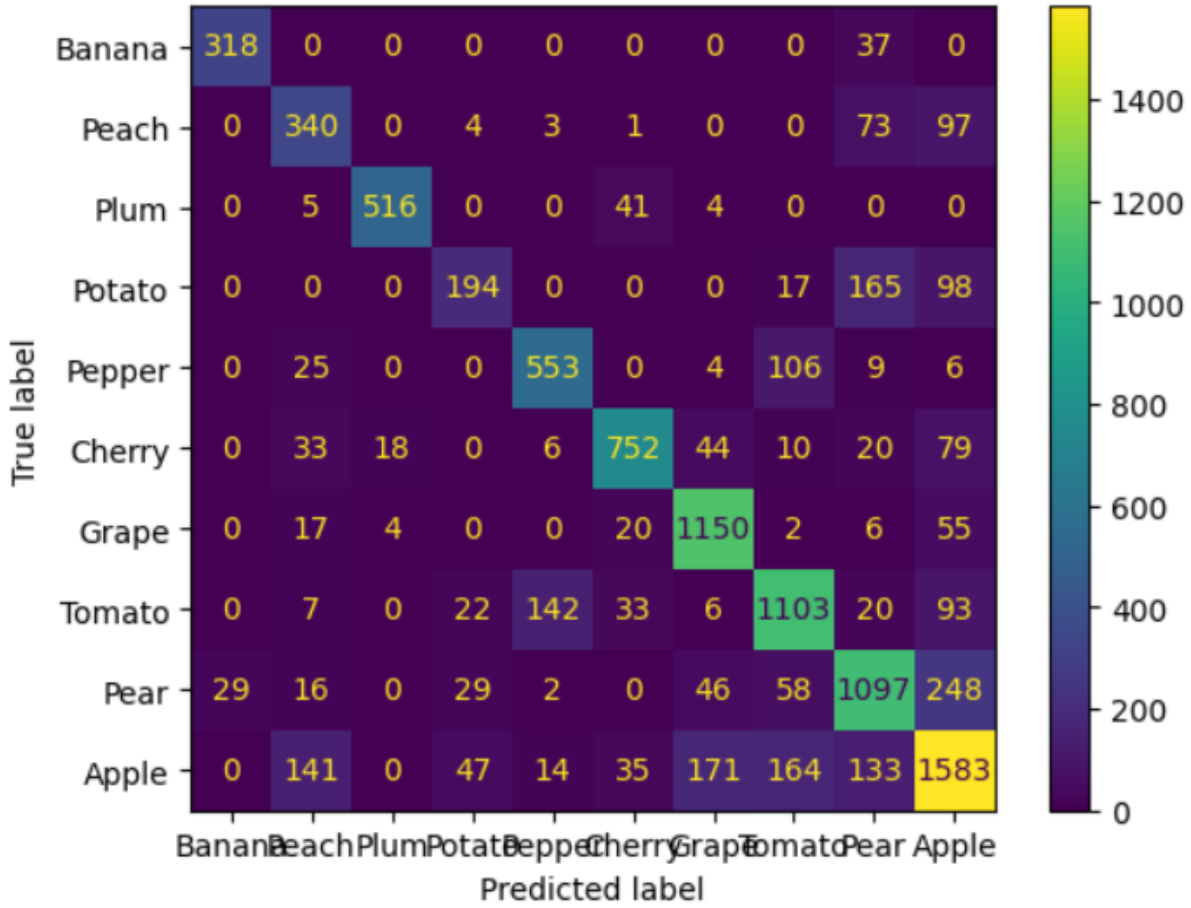

Figure 8: PCA Confusion Matrix

The confusion matrix is displayed for the 10 most frequent classes of fruits in the dataset since there are 70 classes in the dataset.

2.3.2 SVM

The scores obtained for the model on the different hyperparameteter are the following:

param_C	param_gamma	param_kernel	param_random_state	mean_test_score
0.100000	auto	poly	0	0.975061
0.100000	auto	poly	16	0.975061
0.100000	auto	poly	19	0.975061
0.100000	auto	poly	None	0.975061
10.000000	scale	rbf	None	0.974922
10.000000	scale	rbf	19	0.974922
10.000000	scale	rbf	16	0.974922
10.000000	scale	rbf	0	0.974922
1.000000	auto	poly	0	0.974364
1.000000	auto	poly	16	0.974364
1.000000	auto	poly	19	0.974364
1.000000	auto	poly	None	0.974364
10.000000	auto	poly	None	0.974295
10.000000	auto	poly	19	0.974295
10.000000	auto	poly	16	0.974295
10.000000	auto	poly	0	0.974295
0.010000	auto	poly	None	0.973668
0.010000	auto	poly	19	0.973668
0.010000	auto	poly	16	0.973668
0.010000	auto	poly	0	0.973668
10.000000	scale	poly	0	0.964751

Figure 9: Hyperparameter Score

	precision	recall	f1-score	support
Mangostan	0.45	0.38	0.41	102
Cherry	0.94	0.93	0.94	1148
Grape	0.87	0.86	0.86	1476
Nectarine	0.97	0.79	0.87	324
Kohlrabi	1.00	1.00	1.00	157
Physalis	0.99	1.00	0.99	328
Carrot	0.62	1.00	0.76	50
Melon	0.95	1.00	0.98	246
Tomato	0.99	0.98	0.98	1707
Potato	0.89	0.92	0.90	601
Apple	0.93	0.94	0.94	2525
Beetroot	0.71	0.72	0.72	150
Chestnut	0.81	0.93	0.87	153
Avocado	0.95	0.99	0.97	309
Pear	0.94	0.86	0.90	1761
Grapefruit	0.50	0.62	0.55	330
Kiwi	0.88	1.00	0.93	156
Nut	0.94	0.98	0.96	396
Cauliflower	0.98	1.00	0.99	234
Guava	1.00	1.00	1.00	166
Mulberry	1.00	1.00	1.00	164
Walnut	0.84	1.00	0.91	249
Pineapple	0.73	1.00	0.85	329
...				
accuracy			0.92	24051
macro avg	0.93	0.93	0.93	24051
weighted avg	0.93	0.92	0.92	24051

Figure 10: Class Score

In figure 10, we can see the scores obtained on several classes. The overall accuracy of the model was 92.35%. This score demonstrates that SVM is better suited for multi-class datasets, rather than Logical Regression.

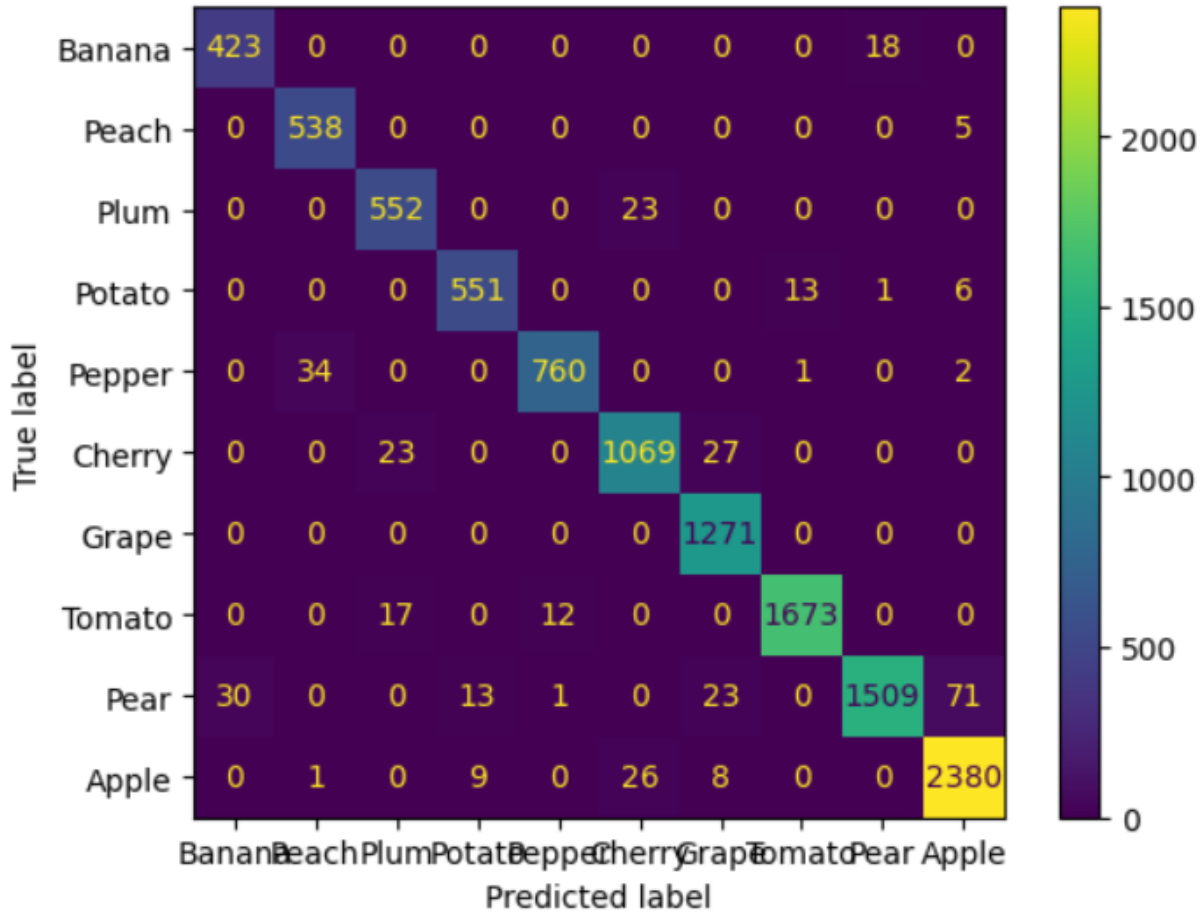

Figure 11: SVM Confusion Matrix

2.3.3 Random Forest

Random forest had an average accuracy of 87.31%. Below are images with the hyperparameter scores and class precision, along with the confusion matrix.

param_n_estimators	param_max_depth	param_max_samples	mean_test_score
100	None	None	0.952072
50	None	None	0.949147
100	None	0.700000	0.947544
50	None	0.700000	0.943365
100	None	0.300000	0.926576
50	None	0.300000	0.918844
100	8	None	0.684848
50	8	0.700000	0.680669
50	8	None	0.679624
100	8	0.700000	0.678997
100	8	0.300000	0.670010
50	8	0.300000	0.661512
100	5	None	0.428979
100	5	0.700000	0.421456
100	5	0.300000	0.419854
50	5	None	0.417346
50	5	0.300000	0.414699
50	5	0.700000	0.412052

Figure 12: Hyperparameter Score

	precision	recall	f1-score	support
Mangostan	0.49	0.44	0.47	102
Cherry	0.88	0.95	0.91	1148
Grape	0.82	0.87	0.85	1476
Nectarine	0.94	0.59	0.72	324
Kohlrabi	0.97	0.68	0.80	157
Physalis	0.98	1.00	0.99	328
Carrot	0.72	1.00	0.84	50
Melon	0.93	0.99	0.96	246
Tomato	0.87	0.96	0.91	1707
Potato	0.74	0.78	0.76	601
Apple	0.74	0.99	0.84	2525
Beetroot	0.68	0.39	0.49	150
Chestnut	0.85	0.76	0.80	153
Avocado	0.95	0.93	0.94	309
Pear	0.82	0.79	0.80	1761
Grapefruit	0.50	0.51	0.51	330
Kiwi	0.99	0.94	0.96	156
Nut	0.93	0.76	0.84	396
Cauliflower	0.96	0.82	0.89	234
Guava	1.00	1.00	1.00	166
Mulberry	0.94	0.99	0.96	164
Walnut	0.94	1.00	0.97	249
Pineapple	0.96	1.00	0.98	329
...				
accuracy			0.87	24051
macro avg	0.93	0.86	0.89	24051
weighted avg	0.88	0.87	0.87	24051

Figure 13: Class Score

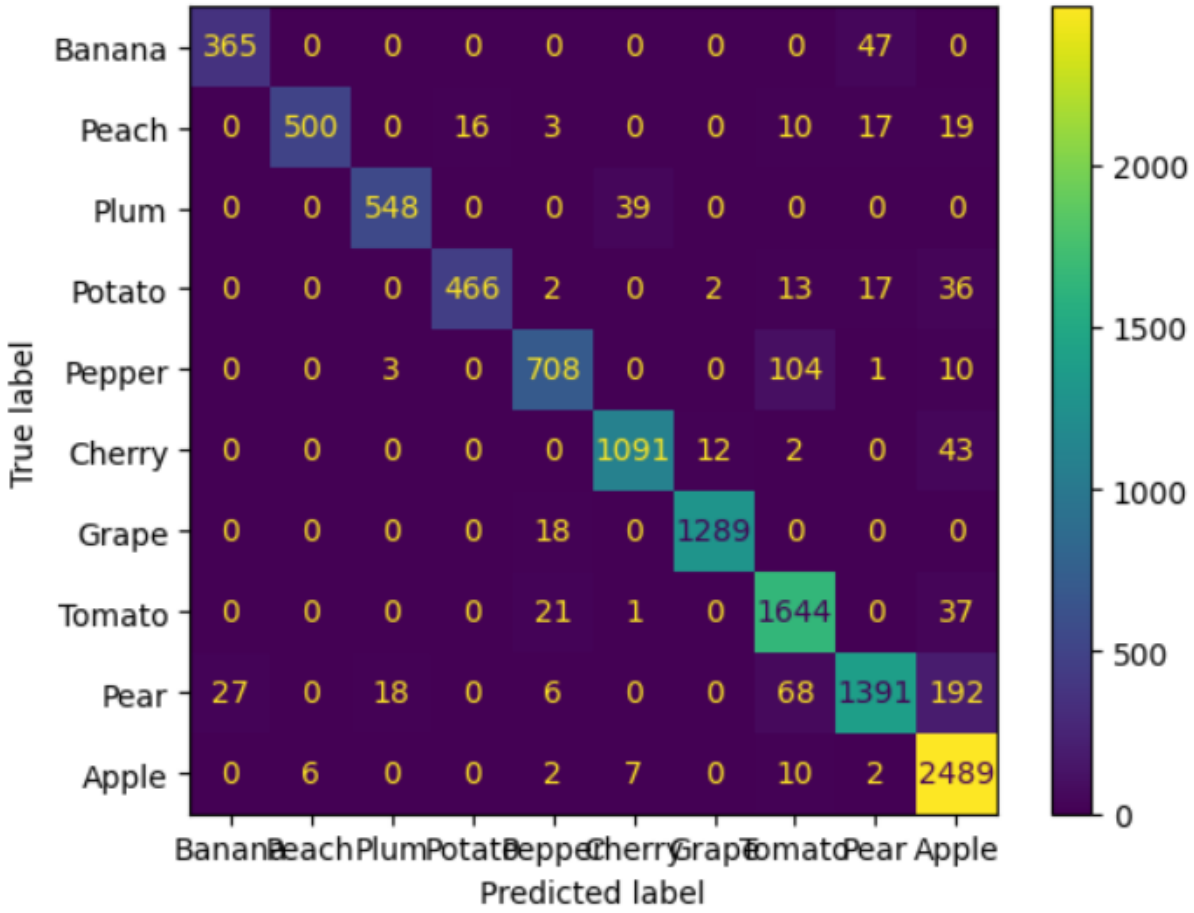

Figure 14: Random Forest Confusion Matrix

2.3.4 Gradient Boosted Trees

The gradient boosted trees model ended with an accuracy of 81.16% after hyperparameter tuning. The hyperparameters tested for were depth of the tree, number of parallel trees and the learning rate.

param_max_depth	param_num_parallel_tree	param_eta	mean_test_score
4	6	0.300000	0.935980
4	4	0.300000	0.935772
6	4	0.300000	0.931452
6	6	0.300000	0.931452
4	4	0.450000	0.921491
4	6	0.450000	0.921282
6	6	0.450000	0.909370
6	4	0.450000	0.909300

Figure 15: Hyperparameter Score

	precision	recall	f1-score	support
Pineapple	0.91	0.98	0.95	329
Cocos	0.96	0.82	0.89	166
Carrot	1.00	0.96	0.98	50
Guava	1.00	1.00	1.00	166
accuracy			0.95	711
macro avg	0.97	0.94	0.95	711
weighted avg	0.95	0.95	0.95	711

Figure 16: Class Score

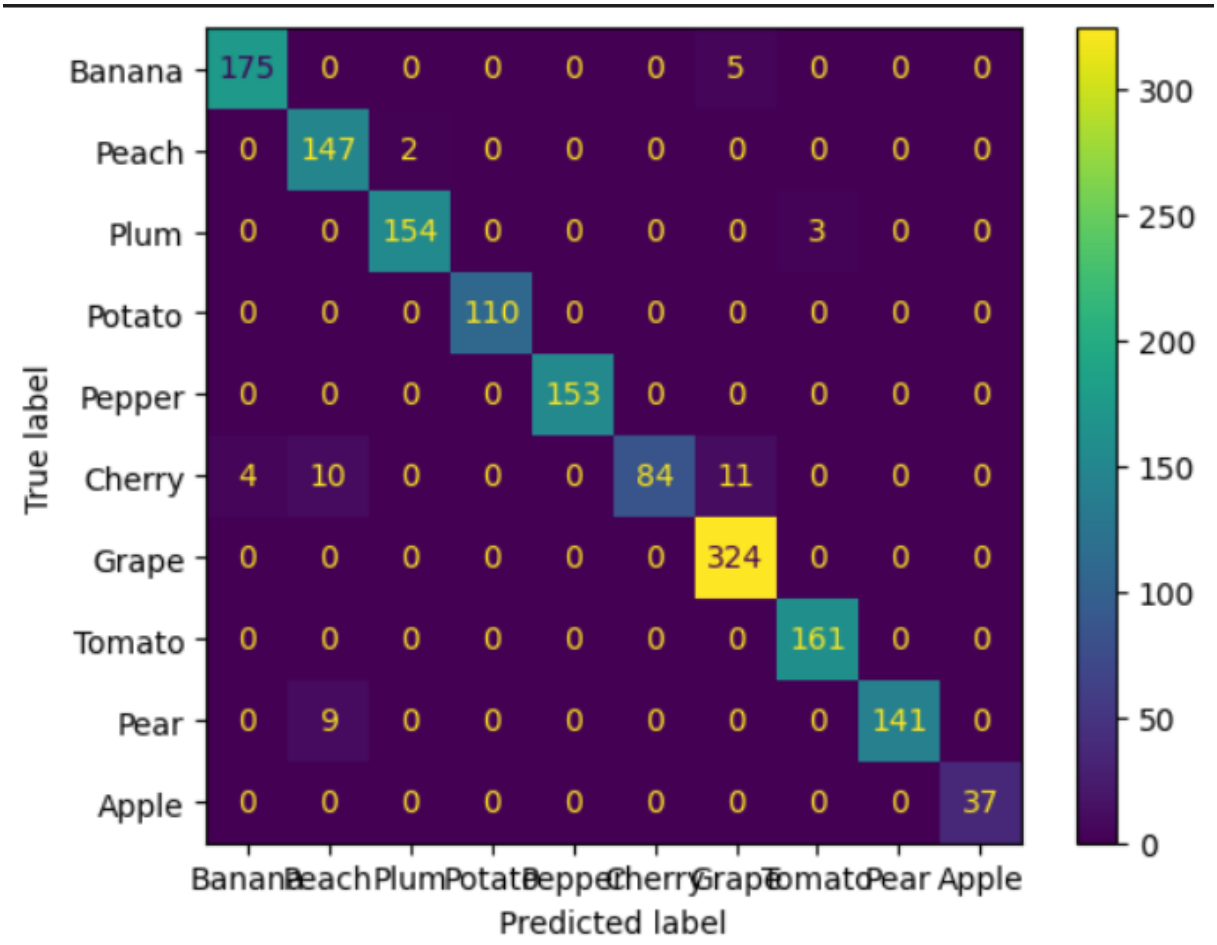

Figure 17: Gradient Boosted Trees Confusion Matrix

2.4 Attribute Extraction: Combined HIST - HOG

For contrast, we used another method of extracting attributes. The procedure followed was finding the HIST keypoints, taking the region surrounding them and extracting HOG attributes of the areas. The next step applied was applying k-means to cluster the attributes into a vocabulary of 300 attributes that was used to build histograms for each image in the dataset based on the frequency of apparition of the features in the vocabulary in the image. To get more details on the implementation, check the following link.

The figure below displays the keypoints extracted with the SIFT algorithm.

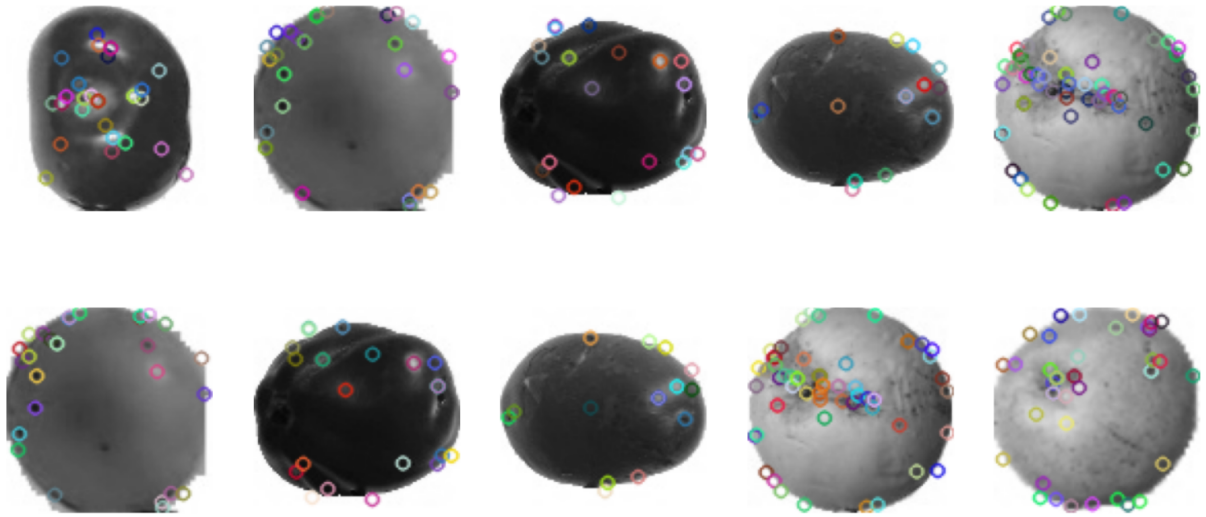

Figure 18: SIFT

The keypoints extracted might not be the best for the models to learn the dataset since most of the fruits and vegetables and round.

2.5 Model Results - HIST

A summary of the models accuracy is as listed below.
Logistic Regression: 61.78%
SVM: 70.89%
Random Forest: 59.46%
Gradient Boosted Trees: 59.38%

2.5.1 Logistic Regresion

param_C	param_solver	param_multi_class	mean_test_score
20	lbfgs	multinomial	0.615019
20	lbfgs	ovr	0.611208
50	sag	multinomial	0.609780
50	saga	multinomial	0.609463
20	lbfgs	ovr	0.486422
20	lbfgs	multinomial	0.485470
50	saga	multinomial	0.484514
50	sag	multinomial	0.483562
50	lbfgs	multinomial	0.483562
50	newton-cg	multinomial	0.483562
50	lbfgs	ovr	0.483083
10	sag	ovr	0.478800
10	lbfgs	ovr	0.478325
10	saga	ovr	0.478324
10	newton-cg	ovr	0.478324
20	lbfgs	ovr	0.373392
10	newton-cg	ovr	0.372004
10	sag	ovr	0.372004
50	lbfgs	multinomial	0.371963
50	saga	multinomial	0.371963
10	saga	ovr	0.370576

Figure 19: Hyperparameter Score

	precision	recall	f1-score	support
Apple	0.45	0.31	0.37	102
Apricot	0.71	0.65	0.68	1148
Avocado	0.57	0.68	0.62	1476
Banana	0.26	0.22	0.24	324
Beetroot	0.66	0.75	0.70	157
Blueberry	0.75	0.75	0.75	328
Cabbage	0.92	0.92	0.92	50
Cactus	0.78	0.74	0.76	246
Cantaloupe	0.64	0.68	0.66	1707
Carambula	0.40	0.33	0.36	601
Carrot	0.46	0.56	0.51	2525
Cauliflower	0.33	0.21	0.25	150
Cherry	0.38	0.31	0.35	153
Chestnut	0.66	0.73	0.70	309
Clementine	0.50	0.48	0.49	1761
Cocos	0.45	0.27	0.34	330
Corn	0.61	0.56	0.58	156
Cucumber	0.50	0.48	0.49	396
Dates	0.96	1.00	0.98	234
Eggplant	0.75	0.51	0.60	166
Fig	0.96	0.98	0.97	164
Ginger	0.93	0.94	0.93	249
Granadilla	0.99	0.98	0.98	329
...				
accuracy			0.62	24050
macro avg	0.68	0.64	0.65	24050
weighted avg	0.62	0.62	0.61	24050

Figure 20: Class Score

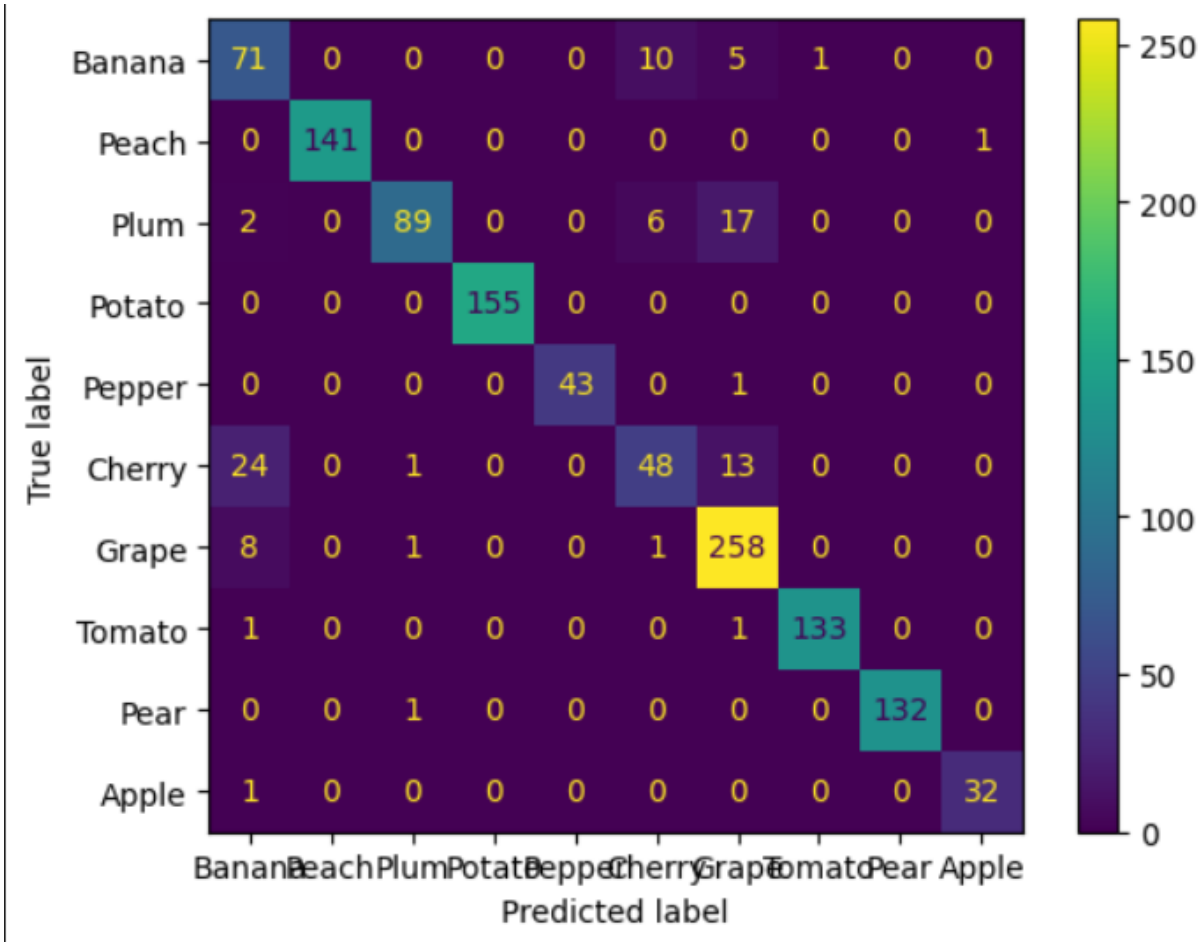

Figure 21: Confusion Matrix

2.5.2 SVM

param_C	param_gamma	param_kernel	mean_test_score
10	scale	rbf	0.762366
5	scale	rbf	0.761878
1	scale	rbf	0.702731
10	scale	poly	0.686778
5	scale	poly	0.686778
1	scale	poly	0.672844
10	auto	rbf	0.310715
5	auto	rbf	0.215550
1	auto	rbf	0.108959
1	auto	poly	0.108959
5	auto	poly	0.108959
10	auto	poly	0.108959

Figure 22: Hyperparameter Score

	precision	recall	f1-score	support
Apple	0.46	0.12	0.19	102
Apricot	0.79	0.82	0.81	1148
Avocado	0.61	0.80	0.69	1476
Banana	0.44	0.34	0.38	324
Beetroot	0.81	0.81	0.81	157
Blueberry	0.90	0.87	0.88	328
Cabbage	0.94	0.92	0.93	50
Cactus	0.84	0.76	0.80	246
Cantaloupe	0.71	0.80	0.75	1707
Carambula	0.48	0.50	0.49	601
Carrot	0.53	0.79	0.64	2525
Cauliflower	0.68	0.27	0.39	150
Cherry	0.65	0.35	0.45	153
Chestnut	0.75	0.83	0.79	309
Clementine	0.64	0.66	0.65	1761
Cocos	0.46	0.24	0.32	330
Corn	0.69	0.54	0.60	156
Cucumber	0.62	0.52	0.57	396
Dates	0.99	1.00	0.99	234
Eggplant	0.78	0.49	0.61	166
Fig	0.97	1.00	0.98	164
Ginger	0.95	0.93	0.94	249
Granadilla	1.00	0.99	1.00	329
...				
accuracy			0.71	24050
macro avg	0.79	0.68	0.71	24050
weighted avg	0.72	0.71	0.70	24050

Figure 23: Class Score

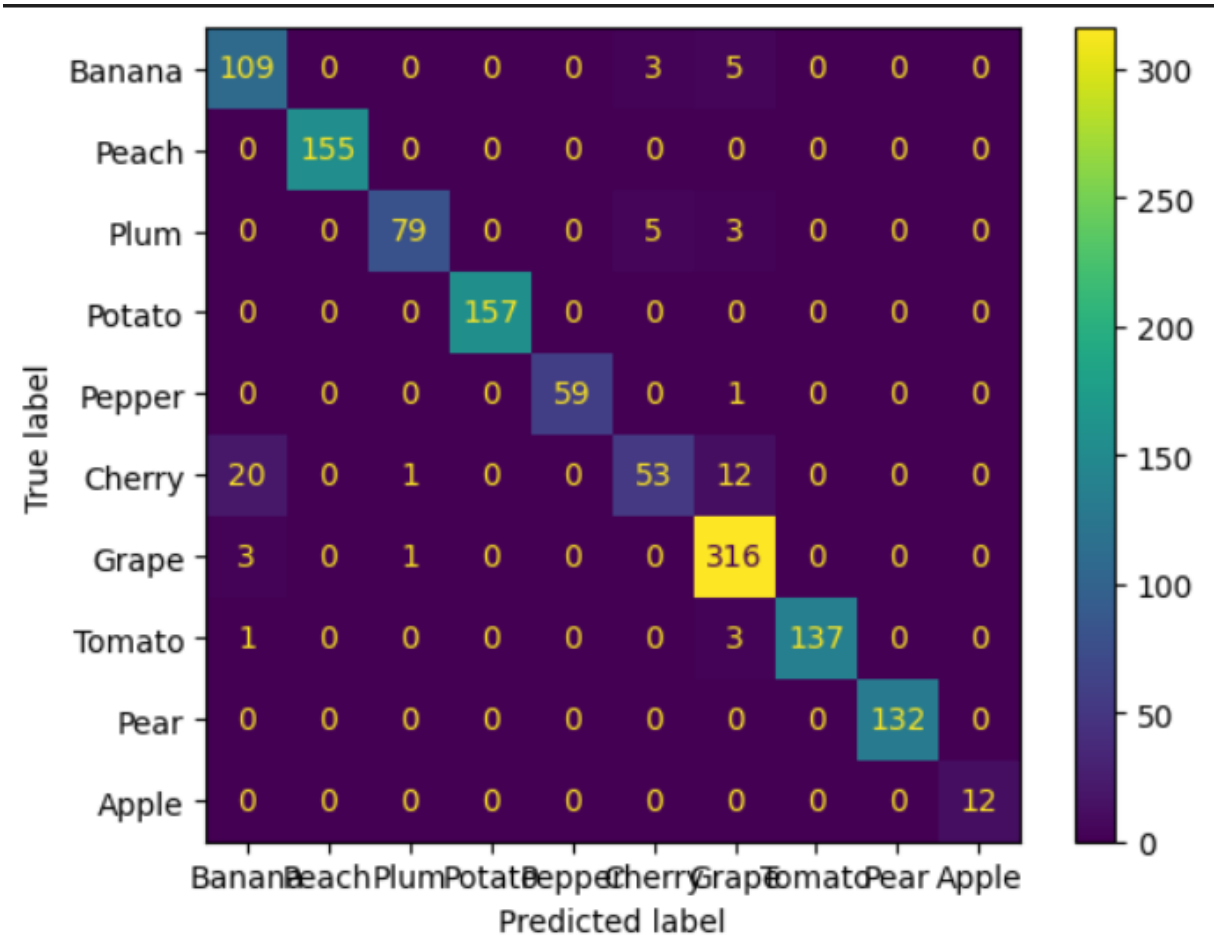

Figure 24: Confusion Matrix

2.5.3 Random Forest

param_n_estimators	param_max_depth	param_max_samples	mean_test_score
500	300	1.000000	0.610562
500	200	1.000000	0.609518
500	100	1.000000	0.609239
250	300	1.000000	0.606104
250	200	1.000000	0.605685
250	100	1.000000	0.602272
200	100	1.000000	0.601993
200	300	1.000000	0.601924
200	200	1.000000	0.598580
500	300	0.700000	0.593843
500	200	0.700000	0.592101
500	100	0.700000	0.591613
250	300	0.700000	0.587990
250	100	0.700000	0.587990
200	300	0.700000	0.587502
250	200	0.700000	0.587154
200	200	0.700000	0.585413
200	100	0.700000	0.584576
500	200	0.500000	0.575102
500	300	0.500000	0.575032
500	100	0.500000	0.574614

Figure 25: Hyperparameter Score

	precision	recall	f1-score	support
Apple	0.00	0.00	0.00	102
Apricot	0.72	0.80	0.76	1148
Avocado	0.55	0.76	0.64	1476
Banana	0.48	0.14	0.22	324
Beetroot	0.78	0.73	0.75	157
Blueberry	0.94	0.66	0.78	328
Cabbage	0.82	0.84	0.83	50
Cactus	0.75	0.39	0.51	246
Cantaloupe	0.61	0.81	0.70	1707
Carambula	0.33	0.22	0.26	601
Carrot	0.36	0.80	0.49	2525
Cauliflower	0.64	0.14	0.23	150
Cherry	0.55	0.20	0.30	153
Chestnut	0.65	0.76	0.70	309
Clementine	0.44	0.65	0.52	1761
Cocos	0.47	0.08	0.14	330
Corn	0.78	0.13	0.23	156
Cucumber	0.54	0.29	0.38	396
Dates	0.94	1.00	0.97	234
Eggplant	0.95	0.12	0.21	166
Fig	0.95	0.99	0.97	164
Ginger	0.80	0.85	0.83	249
Granadilla	0.98	0.99	0.99	329
...				
accuracy			0.59	24050
macro avg	0.74	0.50	0.55	24050
weighted avg	0.65	0.59	0.57	24050

Figure 26: Class Score

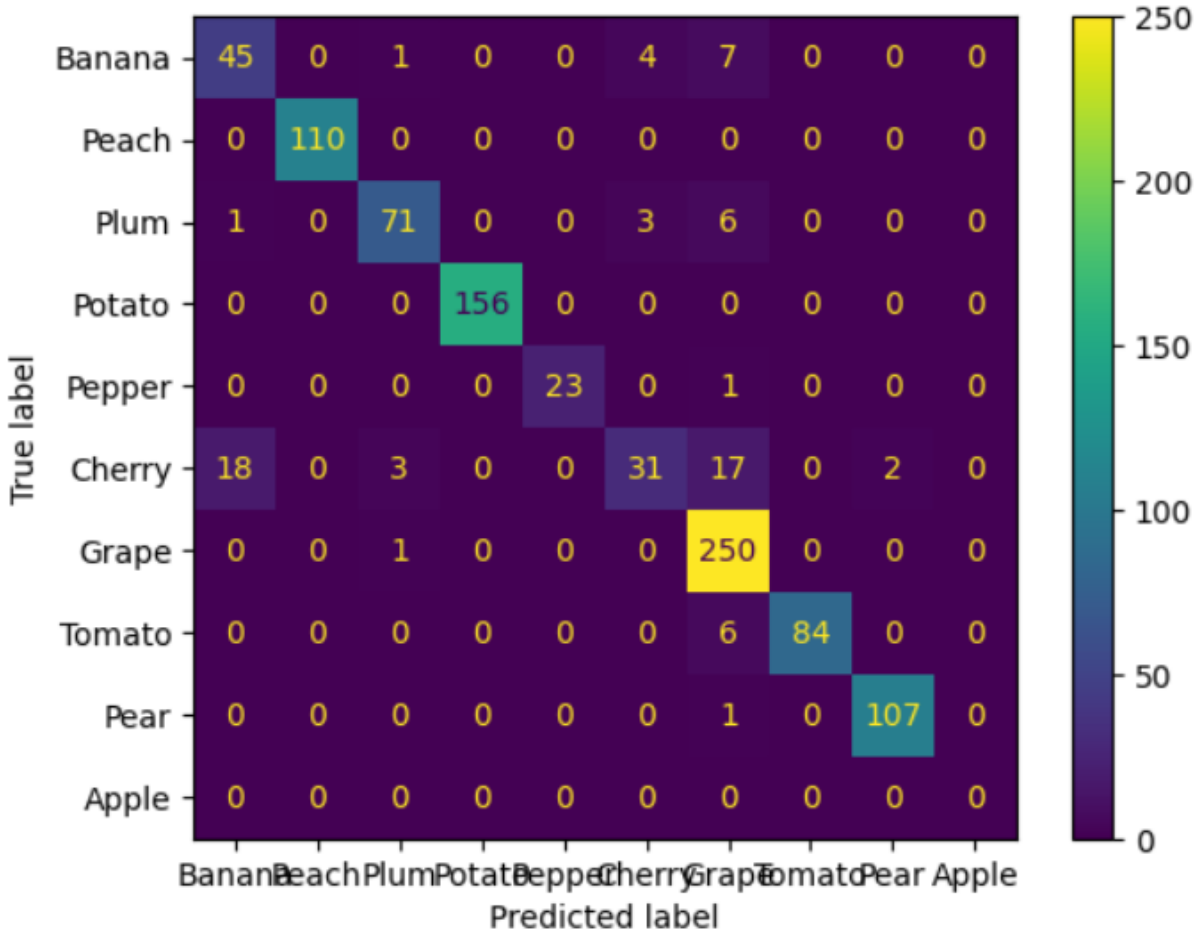

Figure 27: Confusion Matrix

2.5.4 Gradient Boosted Trees

param_max_depth	param_num_parallel_tree	param_eta	mean_test_score
6	4	0.300000	0.619765
6	6	0.300000	0.619277
6	4	0.450000	0.610983
6	6	0.300000	0.502092
6	4	0.300000	0.501883
6	4	0.450000	0.497490
4	4	0.300000	0.494979
4	6	0.300000	0.494770
6	6	0.450000	0.492469
4	6	0.450000	0.490795
4	4	0.450000	0.489749

Figure 28: Hyperparameter Score

	precision	recall	f1-score	support
Apple	0.47	0.23	0.30	102
Apricot	0.71	0.73	0.72	1148
Avocado	0.54	0.75	0.63	1476
Banana	0.37	0.24	0.29	324
Beetroot	0.77	0.66	0.71	157
Blueberry	0.80	0.76	0.78	328
Cabbage	0.87	0.82	0.85	50
Cactus	0.75	0.56	0.64	246
Cantaloupe	0.60	0.70	0.65	1707
Carambula	0.30	0.27	0.29	601
Carrot	0.42	0.68	0.52	2525
Cauliflower	0.40	0.17	0.23	150
Cherry	0.50	0.31	0.39	153
Chestnut	0.75	0.65	0.70	309
Clementine	0.45	0.57	0.50	1761
Cocos	0.41	0.23	0.30	330
Corn	0.59	0.31	0.40	156
Cucumber	0.50	0.42	0.46	396
Dates	0.97	0.93	0.95	234
Eggplant	0.74	0.34	0.47	166
Fig	0.92	0.88	0.90	164
Ginger	0.89	0.74	0.81	249
Granadilla	0.97	0.92	0.94	329
...				
accuracy			0.59	24050
macro avg	0.69	0.55	0.60	24050
weighted avg	0.62	0.59	0.59	24050

Figure 29: Class Score

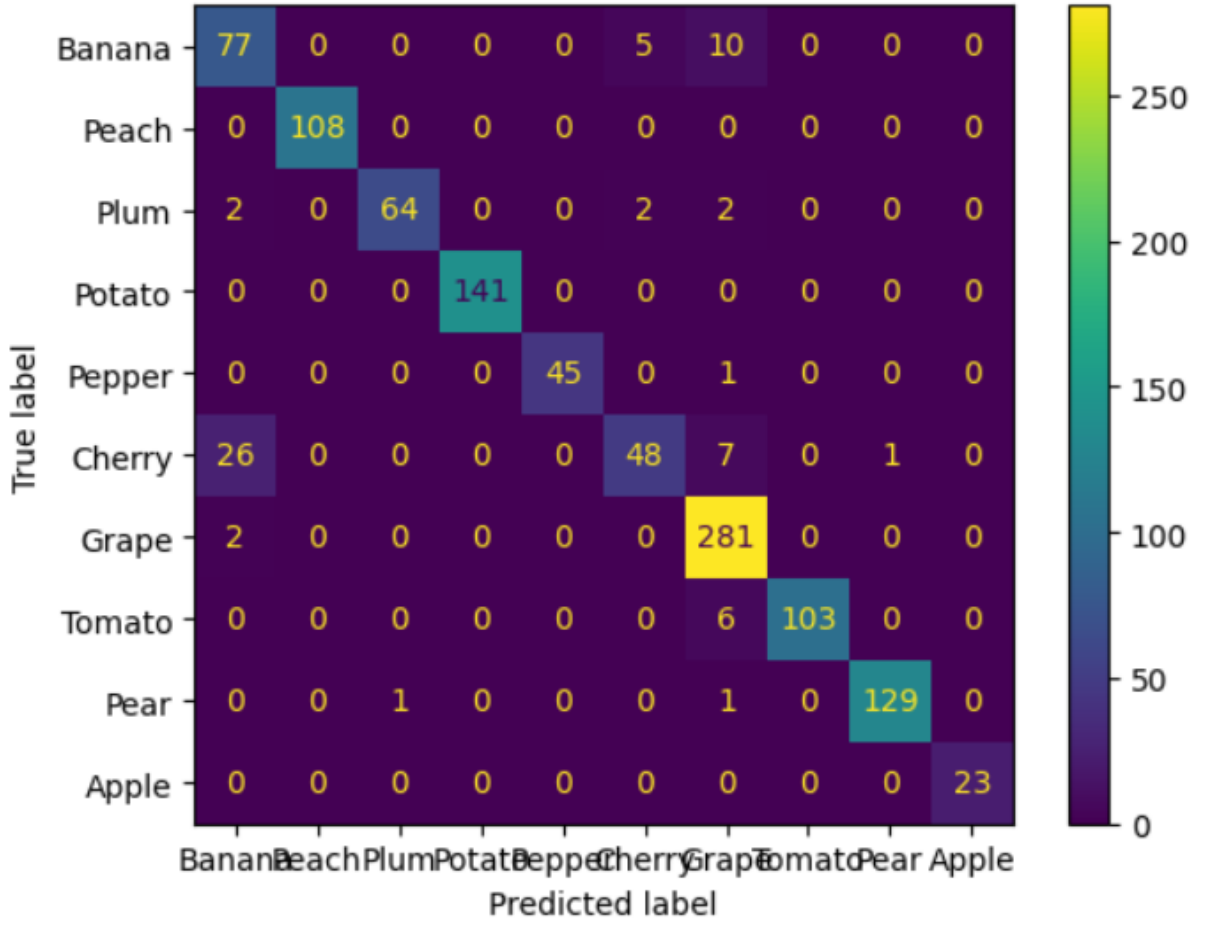

Figure 30: Confusion Matrix

The lower scores of the sift method might be due to the similarities between the classes' outline and the vocabulary size of only 300 due to memory limitations. For better results, some methods observed to increase the accuracy of the models include increasing the size of the vocabulary of keypoints, increasing the depth and the parallel trees hyperparameters in the random forest and gradient boosting trees models. The scores obtained were limited by the configurations of the machine on which the model was trained.

3 Fashion-MNIST

3.1 Data Processing

A visual representation of the images of this dataset is displayed in figure 31.

Figure 31: Fashion-MNIST

3.2 Dataset Analysis

The frequency of classes in the fashion dataset is distributed equally across the dataset, which is ideal for training models.

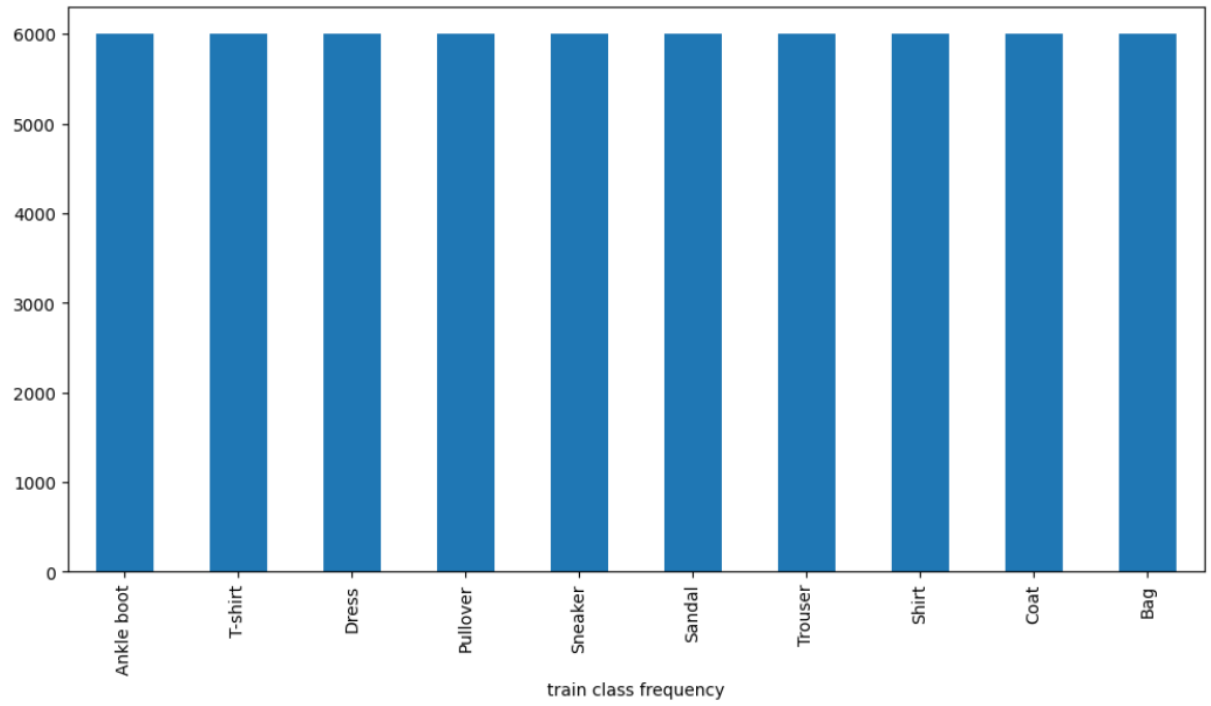

Figure 32: Fashion-MNIST

3.3 Data Normalization

We start the attribute extraction from the fashion dataset images by normalizing all the images for optimal results in extraction algorithms and training.

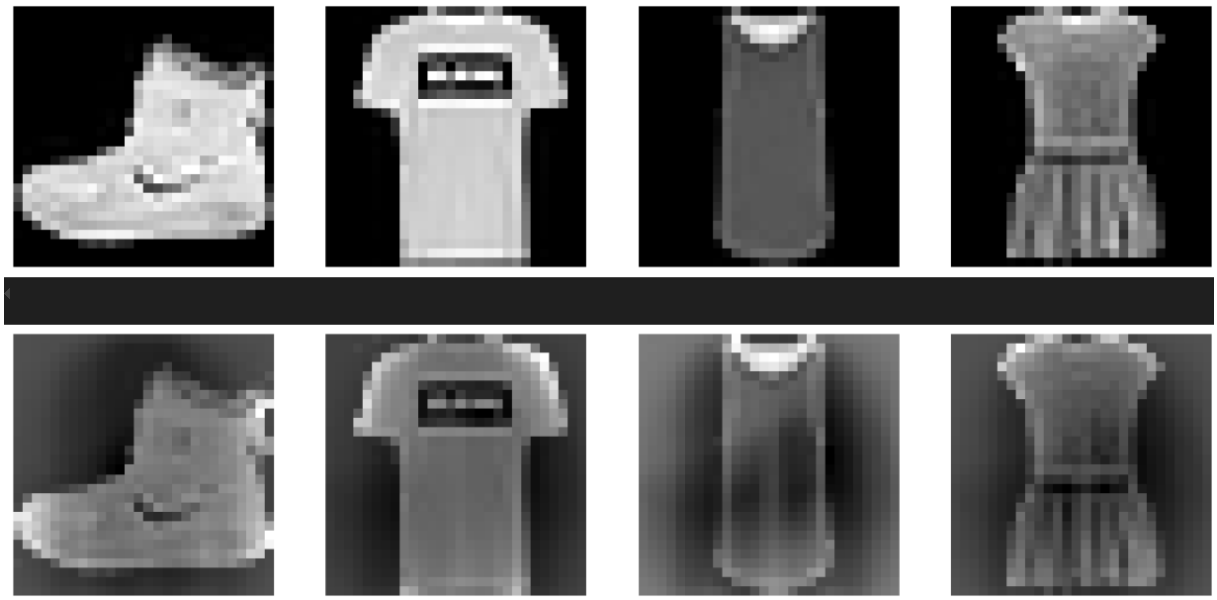

Figure 33: Fashion-MNIST Normalized

3.4 Attribute Extraction: HOG

The first extraction algorithm applied to this dataset is the HOG algorithm, which computes the gradients for each image. A visual representation is included below.

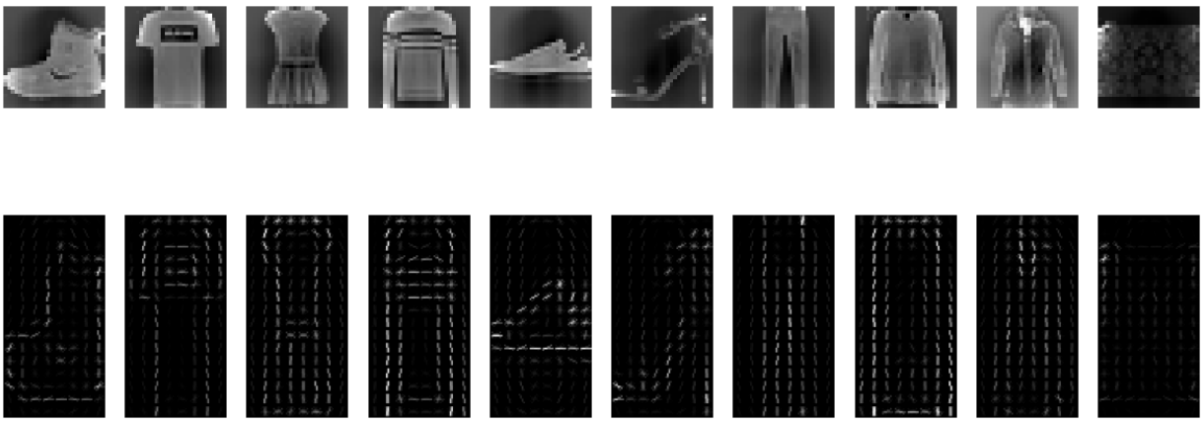

Figure 34: Fashion-MNIST

Since the clothing items are very different in their shape, this algorithm captures well the unique characteristics of each class and provides a small number of features extracted compared to the image's original size.

3.5 Attribute Extraction: HOG - PCA

Although the HOG algorithm reduces significantlty the size of the dataset by extracting the gradients, we want to further reduce the size by applying a second feature extraction method. Due to its efficiency and high capability of capturing information, we chose to apply PCA with only 20 principal components.

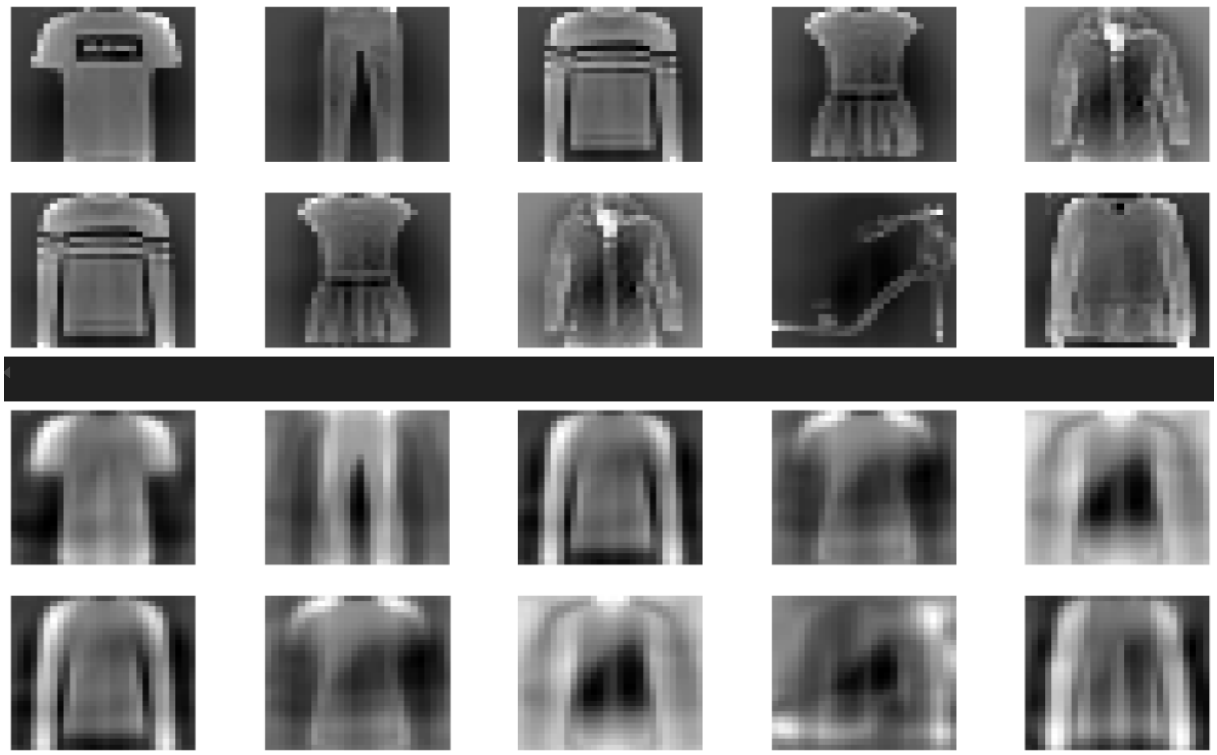

Figure 35: Fashion-MNIST

This figure presents the information captured from the original images with 20 principal components, which capture about 75% of the original information according to the variance graph.

3.6 Attribute Selection

Attribute selection is another method of reducing the dataset size when memory is an issue. For this dataset, we applied SelectPercentile() with 80% of the data retained. Eliminating too much from the features extracted will inhibit the models in learning efficiently and might lead to bad scores.

3.7 Model Scores

3.7.1 Logistic Regression

Overall accuracy: 80.98%

param_penalty	param_C	param_solver	param_max_iter	param_multi_class	mean_test_score
l2	1.000000	lbfgs	1000	multinomial	0.807533
l2	1.000000	lbfgs	500	multinomial	0.807533
l2	1.000000	newton-cg	1000	multinomial	0.807333
l2	1.000000	newton-cg	500	multinomial	0.807333
elasticnet	1.000000	saga	500	multinomial	0.807333
elasticnet	1.000000	saga	1000	multinomial	0.807333
elasticnet	1.000000	saga	1000	multinomial	0.807333
elasticnet	1.000000	saga	500	multinomial	0.807333
elasticnet	1.000000	saga	1000	multinomial	0.807200
elasticnet	1.000000	saga	500	multinomial	0.807200
l2	1.000000	sag	1000	multinomial	0.807133
l2	1.000000	sag	500	multinomial	0.807133
l2	0.100000	lbfgs	500	multinomial	0.804867
l2	0.100000	lbfgs	1000	multinomial	0.804867
elasticnet	0.100000	saga	500	multinomial	0.804600
elasticnet	0.100000	saga	1000	multinomial	0.804600
l2	0.100000	sag	1000	multinomial	0.804400
l2	0.100000	newton-cg	500	multinomial	0.804400
l2	0.100000	sag	500	multinomial	0.804400
l2	0.100000	newton-cg	1000	multinomial	0.804400
elasticnet	0.100000	saga	1000	multinomial	0.804400
elasticnet	0.100000	saga	500	multinomial	0.804400
elasticnet	0.100000	saga	500	multinomial	0.804133
elasticnet	0.100000	saga	1000	multinomial	0.804133
l2	1.000000	sag	500	ovr	0.799533

Figure 36: Hyperparameter Score

	precision	recall	f1-score	support
T-shirt	0.81	0.78	0.79	1000
Trouser	0.96	0.96	0.96	1000
Pullover	0.74	0.68	0.71	1000
Dress	0.81	0.85	0.83	1000
Coat	0.68	0.69	0.69	1000
Sandal	0.89	0.83	0.86	1000
Shirt	0.56	0.57	0.56	1000
Sneaker	0.82	0.87	0.84	1000
Bag	0.92	0.95	0.93	1000
Ankle boot	0.91	0.92	0.91	1000
accuracy			0.81	10000
macro avg	0.81	0.81	0.81	10000
weighted avg	0.81	0.81	0.81	10000

Figure 37: Class Score

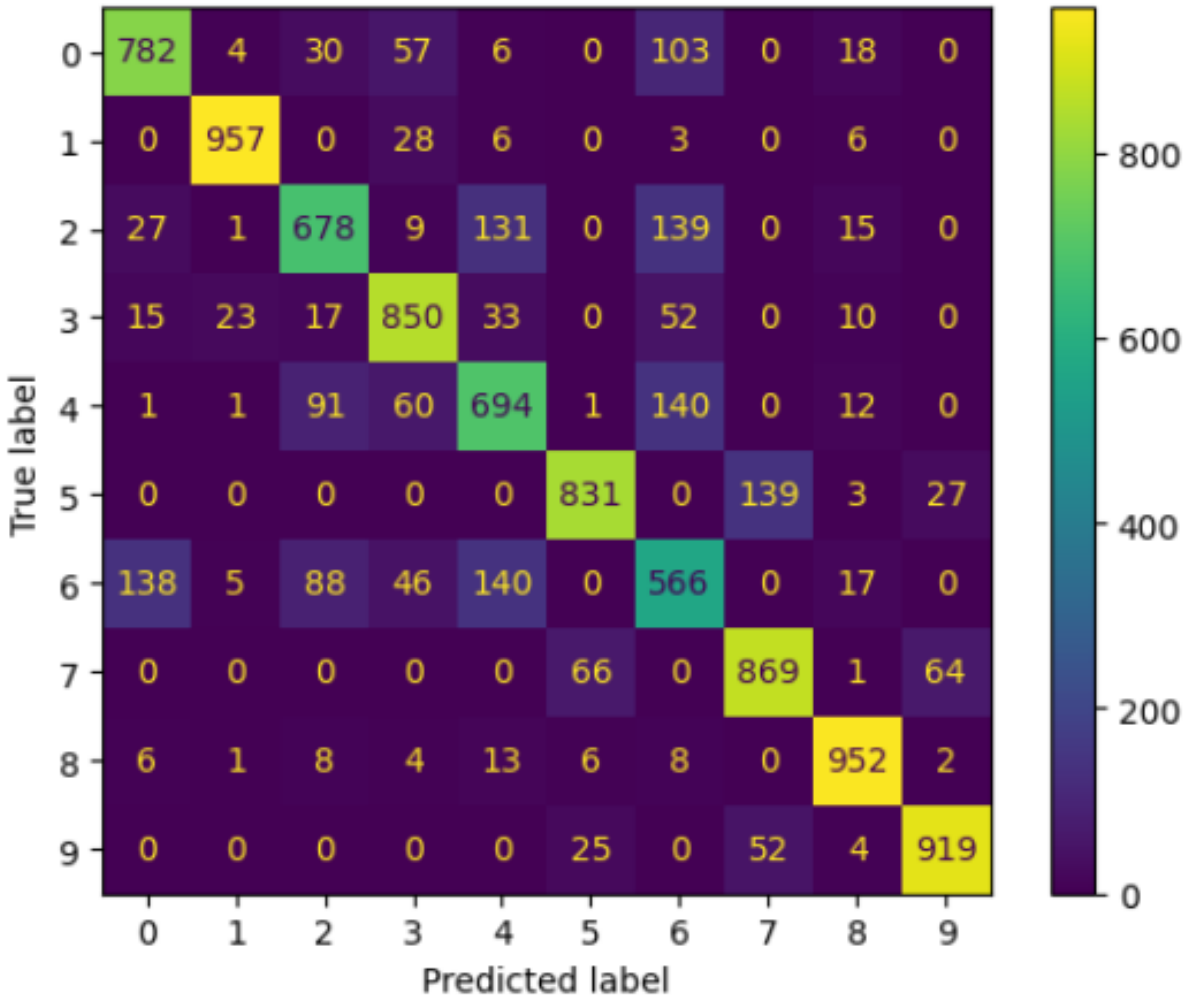

Figure 38: Confusion Matrix

3.7.2 SVM

Overall accuracy: 87.54%

param_C	param_kernel	mean_test_score
10	rbf	0.865267
30	rbf	0.864800
10	poly	0.862400
30	poly	0.861400
1	rbf	0.854333
1	poly	0.850200

Figure 39: Hyperparameter Score

	precision	recall	f1-score	support
T-shirt	0.84	0.85	0.84	1000
Trouser	0.97	0.96	0.97	1000
Pullover	0.82	0.80	0.81	1000
Dress	0.86	0.89	0.88	1000
Coat	0.78	0.81	0.79	1000
Sandal	0.96	0.91	0.94	1000
Shirt	0.71	0.66	0.68	1000
Sneaker	0.90	0.95	0.92	1000
Bag	0.97	0.98	0.97	1000
Ankle boot	0.94	0.94	0.94	1000
accuracy			0.88	10000
macro avg	0.87	0.88	0.87	10000
weighted avg	0.87	0.88	0.87	10000

Figure 40: Class Score

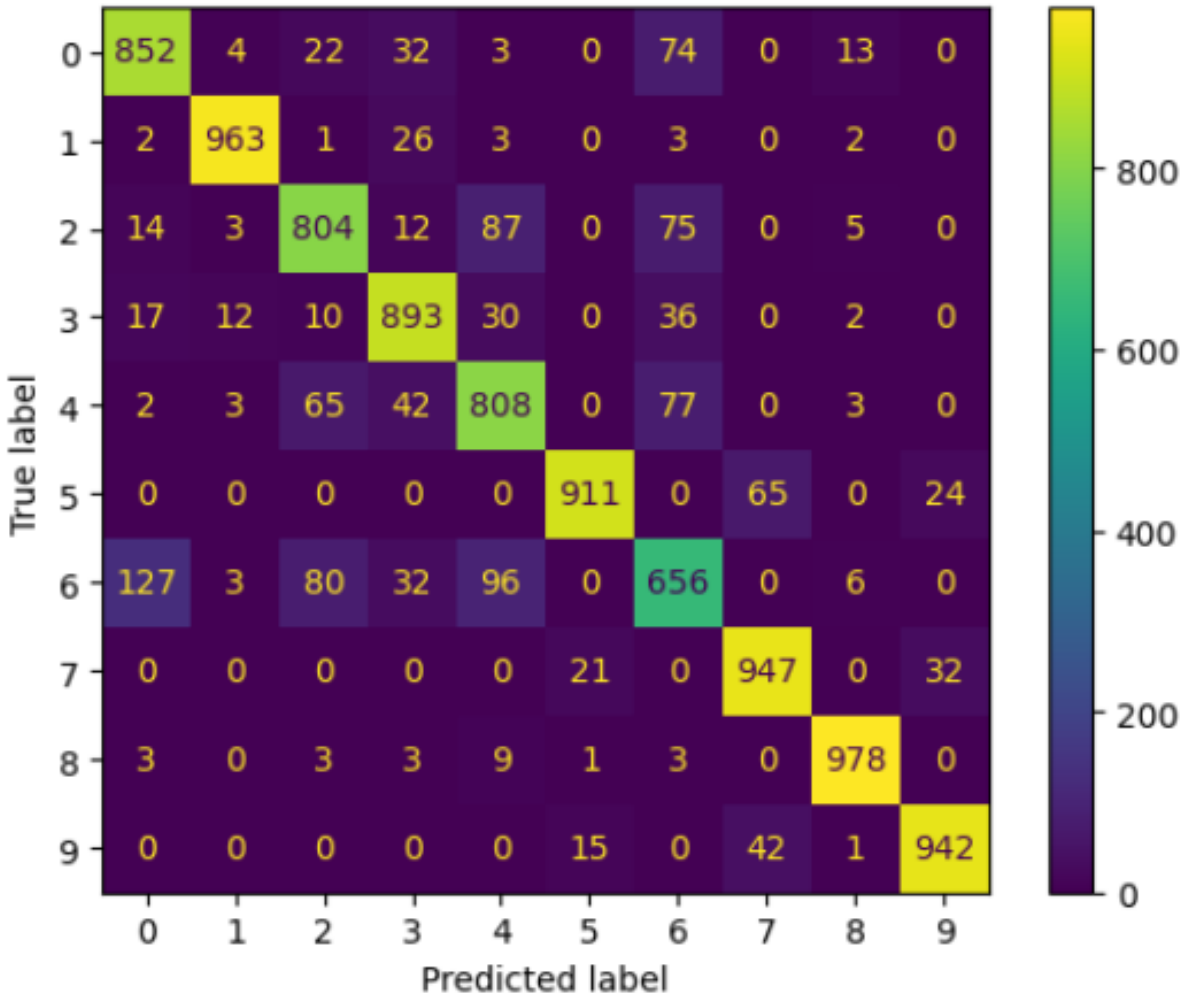

Figure 41: Confusion Matrix

3.7.3 Random Forest

Overall accuracy: 85.29%

param_n_estimators	param_max_depth	param_max_samples	mean_test_score
150	None	1.000000	0.843400
100	None	1.000000	0.841600
150	None	0.700000	0.841000
150	None	None	0.840600
100	None	0.700000	0.840267
100	None	None	0.839667
50	None	None	0.837933
50	None	1.000000	0.837800
50	None	0.700000	0.837533
150	None	0.300000	0.833600
100	None	0.300000	0.833067
150	12	None	0.832867
100	12	None	0.832667
100	12	1.000000	0.831933
150	12	1.000000	0.831667
50	None	0.300000	0.830933
100	12	0.700000	0.830667
150	12	0.700000	0.830467
50	12	None	0.829867
50	12	0.700000	0.829000
150	12	0.300000	0.828800
50	12	1.000000	0.828667

Figure 42: Hyperparameter Score

	precision	recall	f1-score	support
T-shirt	0.82	0.81	0.82	1000
Trouser	0.98	0.95	0.97	1000
Pullover	0.79	0.79	0.79	1000
Dress	0.83	0.87	0.85	1000
Coat	0.76	0.78	0.77	1000
Sandal	0.93	0.91	0.92	1000
Shirt	0.66	0.62	0.64	1000
Sneaker	0.89	0.91	0.90	1000
Bag	0.93	0.96	0.95	1000
Ankle boot	0.93	0.94	0.93	1000
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

Figure 43: Class Score

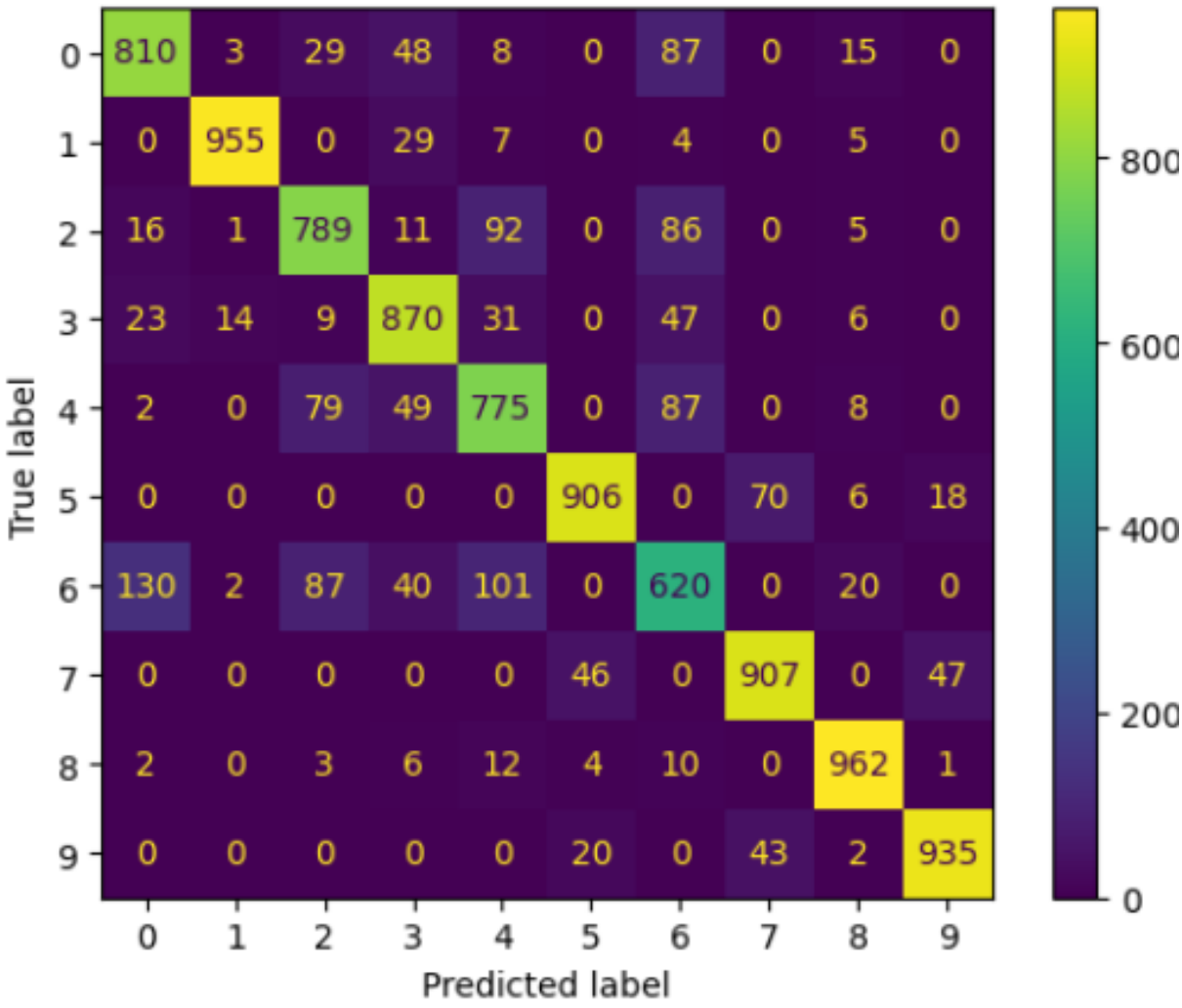

Figure 44: Confusion Matrix

3.7.4 Gradient Boosted Trees

Overall accuracy: 85.83%

param_max_depth	param_num_parallel_tree	param_eta	mean_test_score
6	5	0.300000	0.849267
6	1	0.300000	0.849200
6	3	0.300000	0.849200
6	3	0.450000	0.848200
6	5	0.450000	0.848200
6	1	0.450000	0.848200
6	1	0.150000	0.847667
6	3	0.150000	0.847667
6	5	0.150000	0.847600
4	1	0.300000	0.845000
4	3	0.300000	0.845000
4	5	0.300000	0.845000
4	5	0.450000	0.845000
4	1	0.450000	0.845000
4	3	0.450000	0.845000
4	3	0.150000	0.839467
4	5	0.150000	0.839200
4	1	0.150000	0.838933
2	3	0.450000	0.834200
2	5	0.450000	0.834200
2	1	0.450000	0.834200
2	1	0.300000	0.828400

Figure 45: Hyperparameter Score

	precision	recall	f1-score	support
T-shirt	0.83	0.82	0.82	1000
Trouser	0.98	0.96	0.97	1000
Pullover	0.79	0.77	0.78	1000
Dress	0.86	0.87	0.86	1000
Coat	0.75	0.77	0.76	1000
Sandal	0.94	0.91	0.92	1000
Shirt	0.65	0.65	0.65	1000
Sneaker	0.90	0.92	0.91	1000
Bag	0.96	0.97	0.96	1000
Ankle boot	0.94	0.94	0.94	1000
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

Figure 46: Class Score

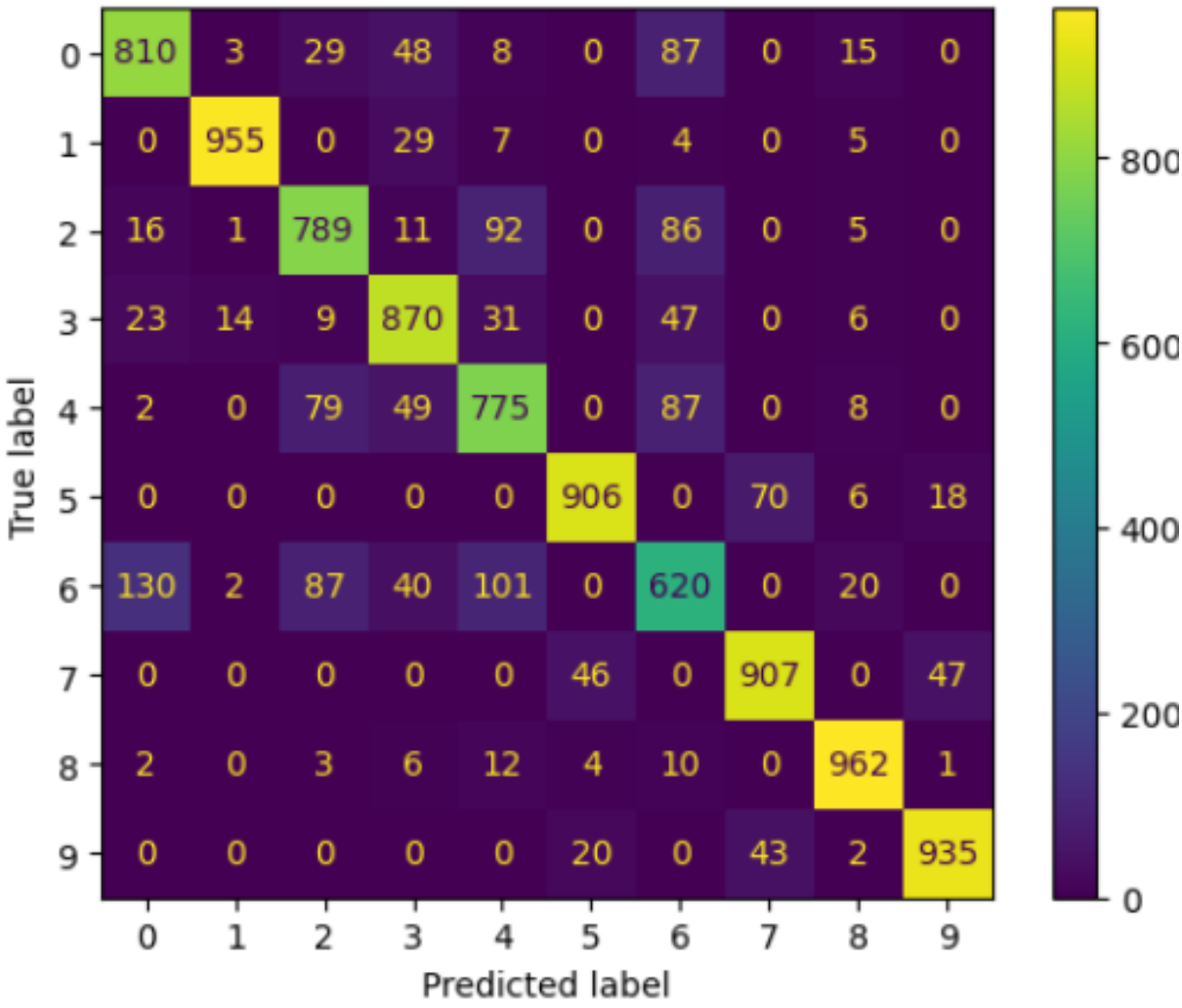

Figure 47: Confusion Matrix

4 Conclusion

In conclusion, based on the results obtained for the two datasets, the following were observed: the more balanced the dataset, the better the results obtained on the models, and SVM is the best suited for multiclass predictions, since it received on average the best results in terms of accuracy.

The benefits of data normalization before PCA extraction were also demonstrated, as the accuracy of the fashion data set increased by 40% after it was applied. After testing the models with default hyperparameters, the results were worse by 1%-5%, demonstrating the impact of the tuning. The most impactful hyperparameters were the following:

1. **Logistic regression** - number of iterations, multiclass
2. **SVM** - gamma
3. **Random forest** - number of estimators and maximum depth
4. **Gradient boosted trees** - maximum depth and number of parallel trees

Lastly, this report has demonstrated the impact of attribute extraction. Based on the method used, one might get better or worse scores on the same model with the same parameters. For the fruit data set, the combined sift-hog method had fairly bad results, compared to the PCA method, which had an acceptable range of scores. In addition, the fruit data set was extremely unbalanced, which might have been another factor that contributed to the results that were worse than those obtained for the fashion data set.