ML - Image feature extraction

Alexandru MIHAI, 341C3

November 25, 2024

1 Feature Extraction

1.1 FashionMNIST

The FashionMNIST set consists of 10 classes: T-Shirt/Top(0), Trouser(1), Pullover(2), Dress(3), Coat(4), Sandal(5), Shirt(6), Sneaker(7), Bag(8), Ankle boot(9). The training set consists of 6000×10 28×28 images and the testing set consists of 1000×10 images.

I split the training set in 80% training and 20% validation in order to select the hyperparameters without interfering with the testing set.

The framework for this dataset follows 2 main steps: (1) extract PCA features and visualize their effects of the initial images, as well as the increase of the cummulative variance with the number of PCA and (2) extract the HOG features as 128×64 images, which are then rescaled back to 28×28 (the original HOG features exceeded the RAM capacity).

I decided not to choose ORB features because there were very few key points that were extracted for this dataset and did not bring value to most of the images. The contour features were relative redundant as the same information is prevailed with the HOG extraction and proved not to add more value to the training data.

1.2 Fruits-360

The Fruits360 set consists of 141 classes. The training set consists of 71773 100×100 and the testing set consists of 24051 images across 70 classes.

I split the training set in 80% training and 20% validation in order to select the hyperparameters without interfering with the testing set.

The framework for this dataset follows 2 main steps: (1) extract the ORB features and flatten them inside an 1×32 array averaging all ORB features (I decided to go with this approach as the number of ORB features might differ from image to image, sometimes being actually 0), visualize the effect of the ORB features and (2) extract the HOG features as 128×64 images, which are then rescaled to 32×32 images (as larger images would exceed the RAM capacity).

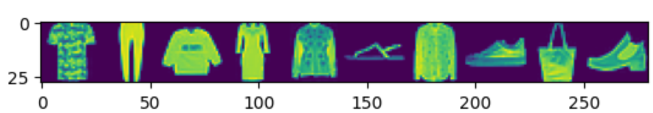

Figure 1: Example of images from each class - FashionMNIST.

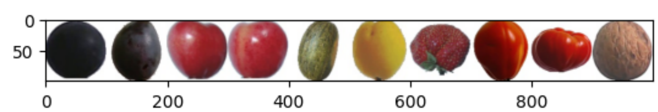

Figure 2: Example of images from the 10 most frequent classes - Fruits-360

2 Feature visualization

2.1 Class balancing analysis

For the FashionMNIST dataset the classes are equally distributed in both the training and testing set.

For the Fruits-360 dataset the classes are roughly equally distributed as present in the histogram below. This inbalance did not affect the training of classic ML algorithms.

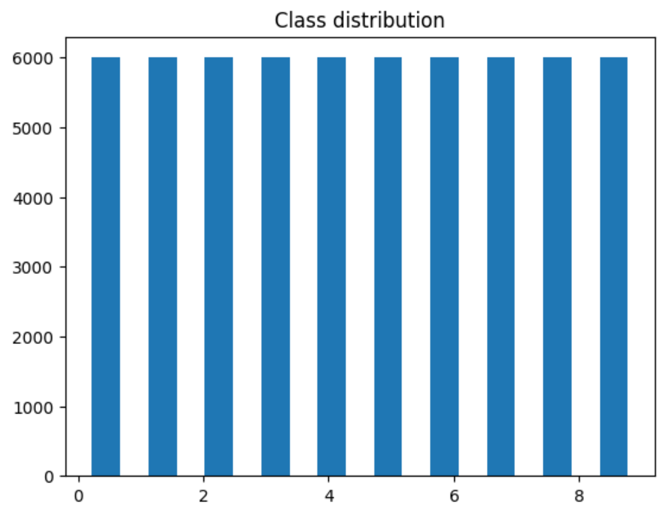

Figure 3: Train class distribution - FashionMNIST.

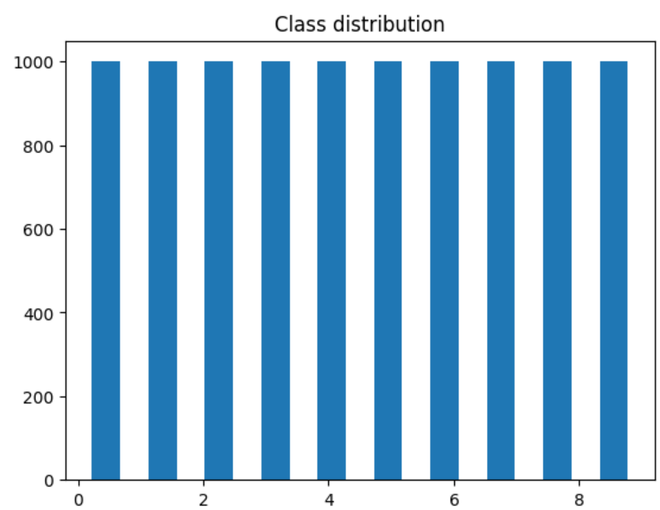

Figure 4: Test class distribution - FashionMNIST.

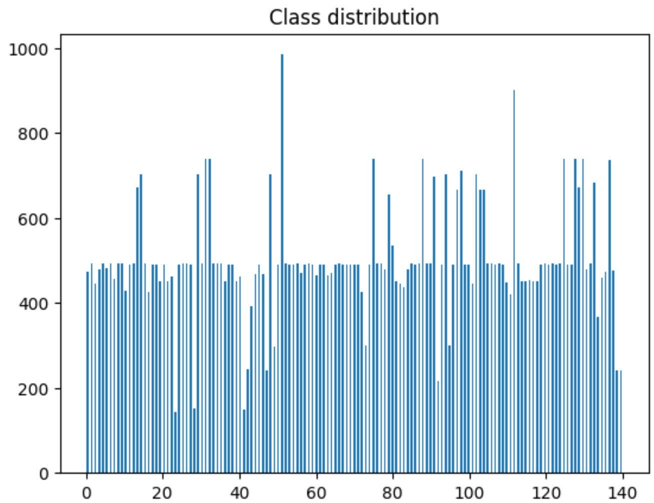

Figure 5: Train class distribution - Fruits-360

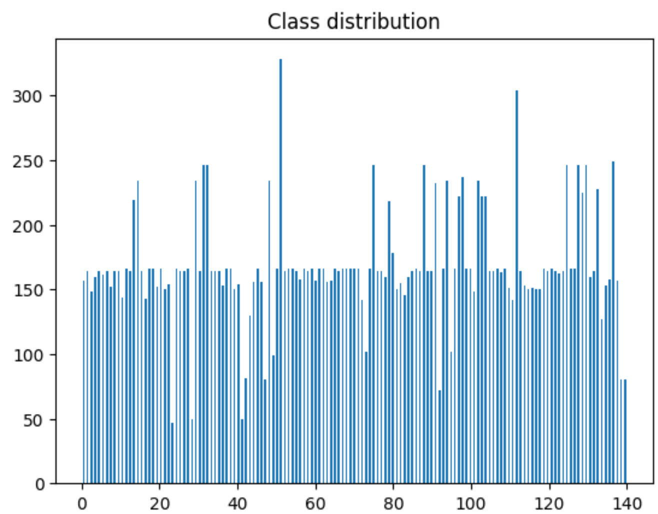

Figure 6: Test class distribution - Fruits-360

2.2 FashionMNIST - PCA visualization

Principal Component Analysis is used to reduce the initial 28×28 images to 20 components that summarize the key features of the image. As presented in Figure 7, the total cumulative variance increases with the number of principals components chosen from the initial image, reaching 1 when all pixels are taken into consideration. I have chosen a total of 20 PC as testing with more did not bring considerable more value to the total cumulative variance.

The Figures 8-17 describe the effect of using PCA on each image from the dataset. The pixels that contribute the most to differentiate between images have a higher weight and are more prominent when taking into consideration the transformed image.

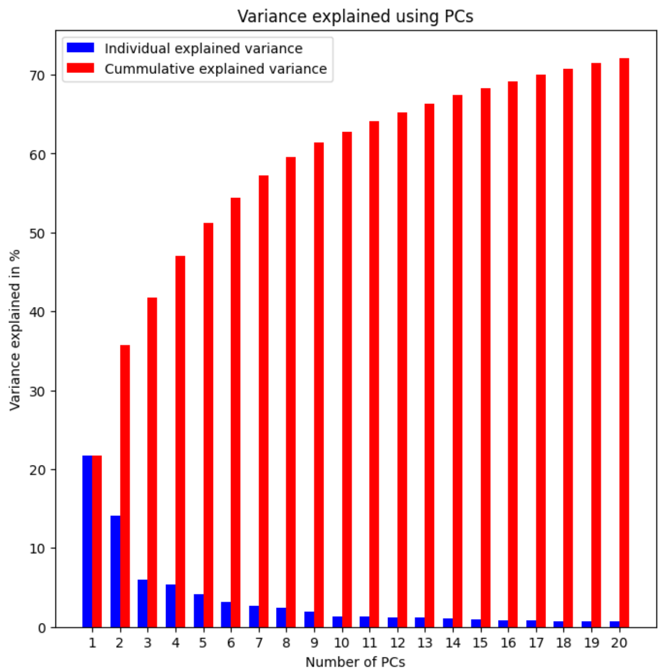

Figure 7: Individual and cumulative variance relative to the number of principal components

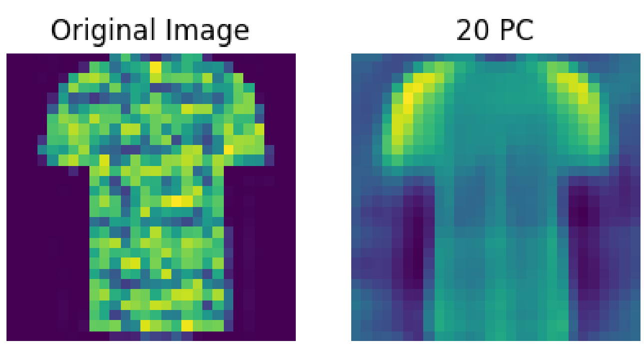

Figure 8: Class 0 - PCA analysis

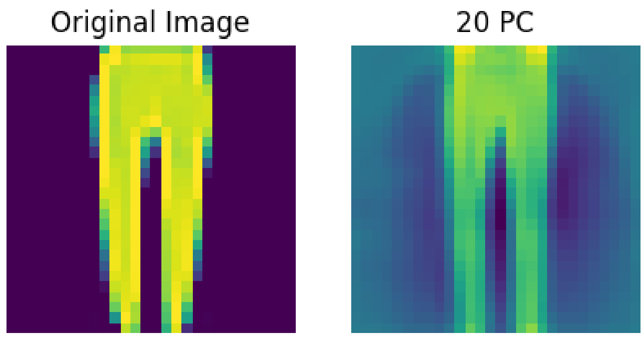

Figure 9: Class 1 - PCA analysis

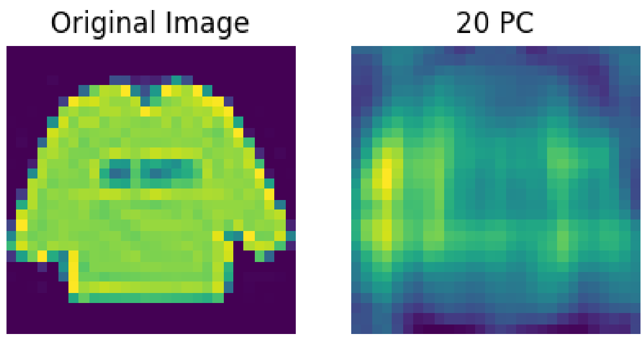

Figure 10: Class 2 - PCA analysis

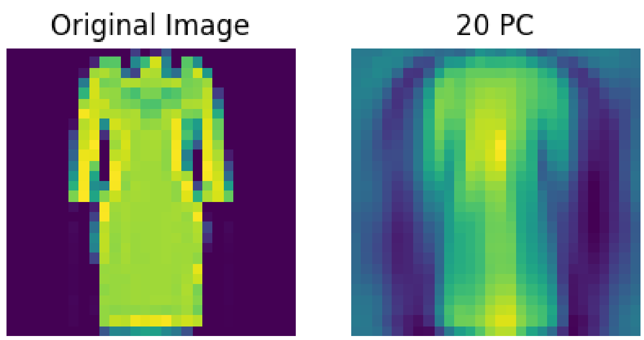

Figure 11: Class 3 - PCA analysis

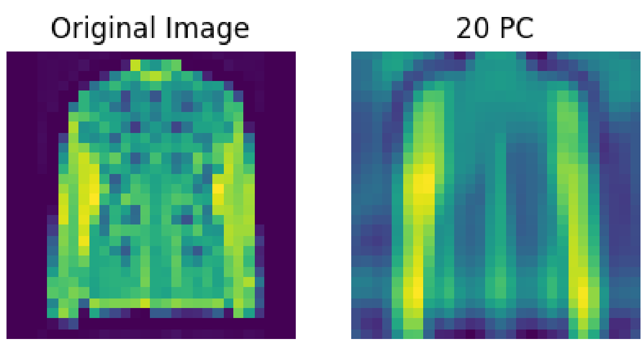

Figure 12: Class 4 - PCA analysis

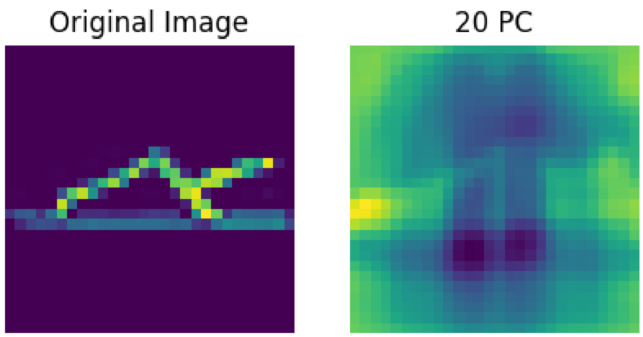

Figure 13: Class 5 - PCA analysis

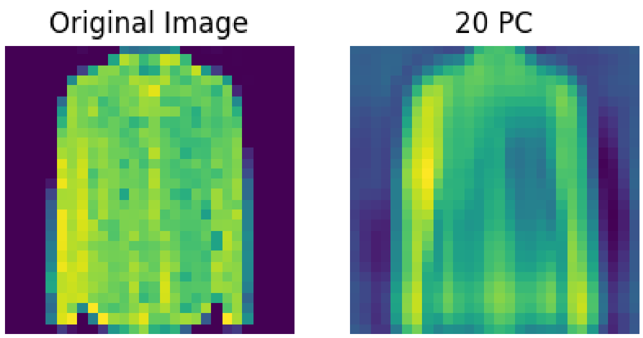

Figure 14: Class 6 - PCA analysis

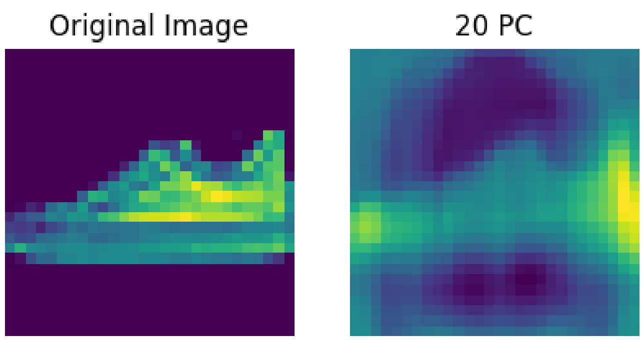

Figure 15: Class 7 - PCA analysis

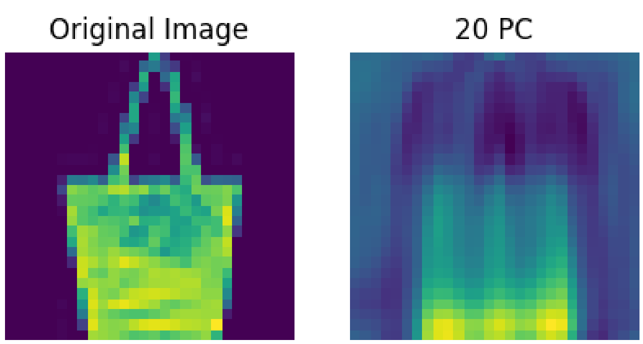

Figure 16: Class 8 - PCA analysis

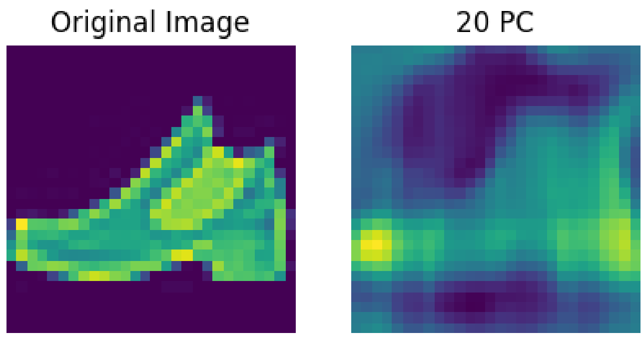

Figure 17: Class 9 - PCA analysis

2.3 FashionMNIST - HOG visualization

Histogram of Oriented Gradients was used to describe the shape of the clothing article, as most of them have noticeable distinct shapes. The initial HOG images were 128×64 which were rescaled back to 28×28 in order to fit the total RAM available on the system. Generally, the transformed images describe straight lines that follow the curvature of the initial images. Noticeably, class 4 and 6 have striking similarities and proved to be difficult to distinct during the training process.

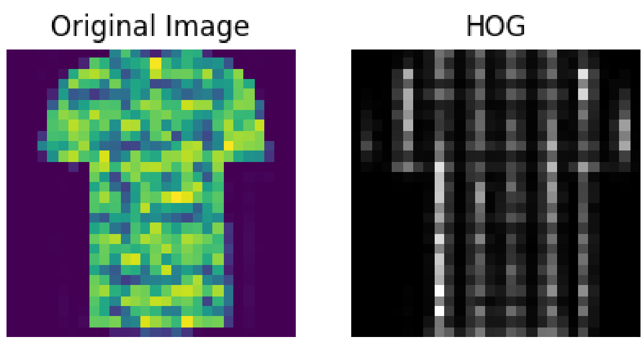

Figure 18: Class 0 - HOG analysis

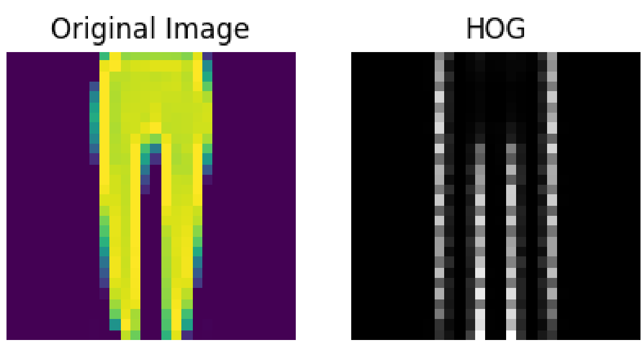

Figure 19: Class 1 - HOG analysis

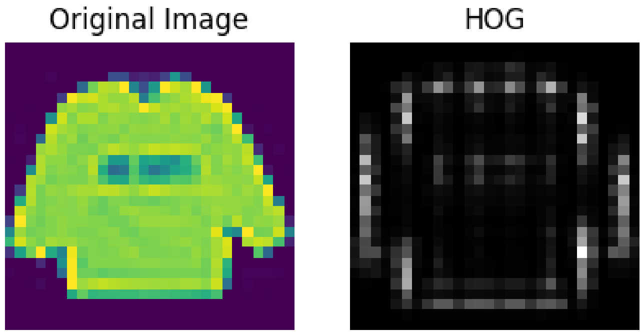

Figure 20: Class 2 - HOG analysis

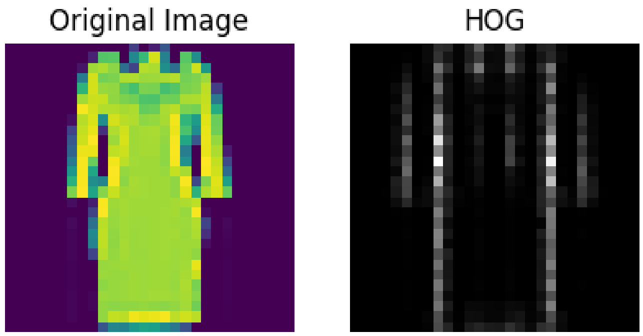

Figure 21: Class 3 - HOG analysis

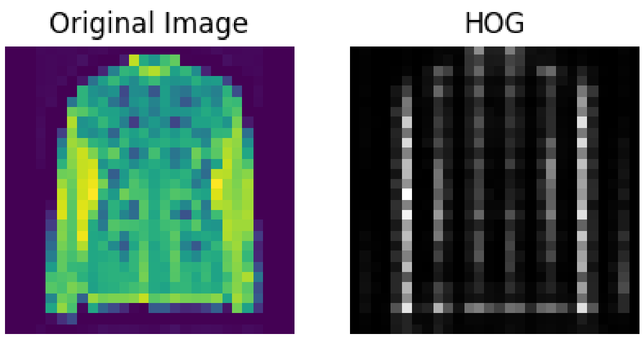

Figure 22: Class 4 - HOG analysis

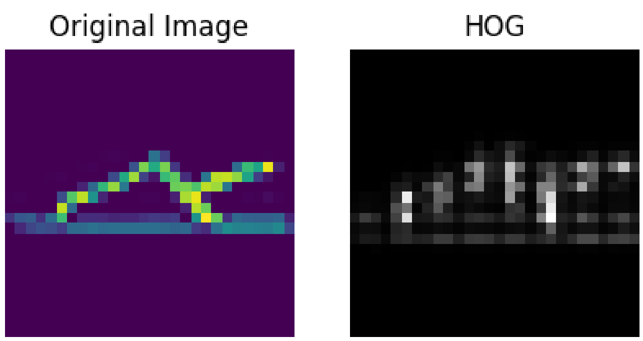

Figure 23: Class 5 - HOG analysis

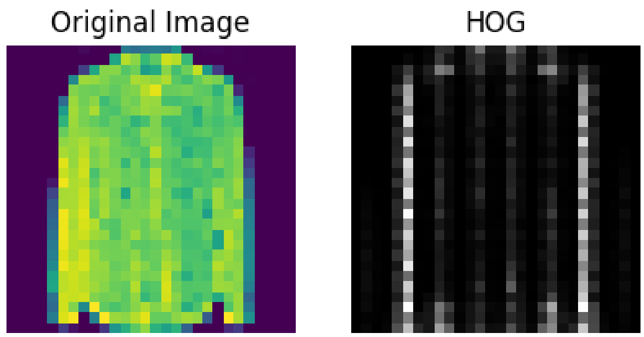

Figure 24: Class 6 - HOG analysis

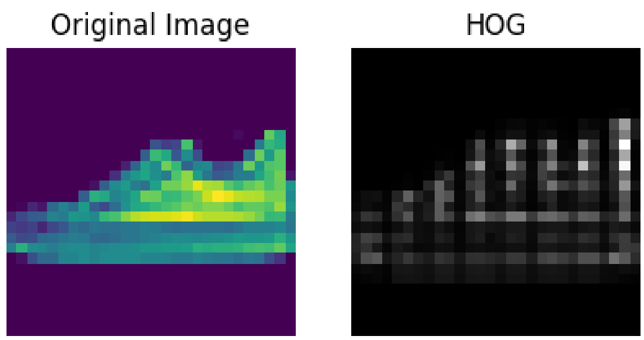

Figure 25: Class 7 - HOG analysis

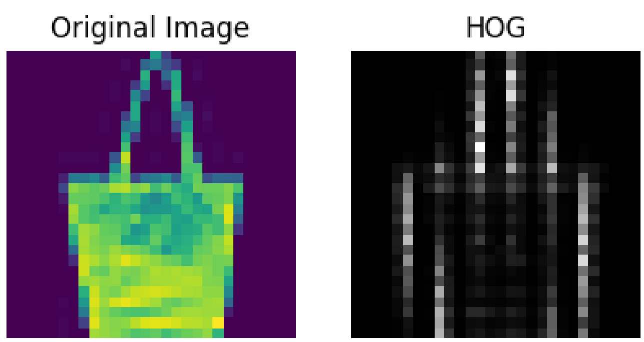

Figure 26: Class 8 - HOG analysis

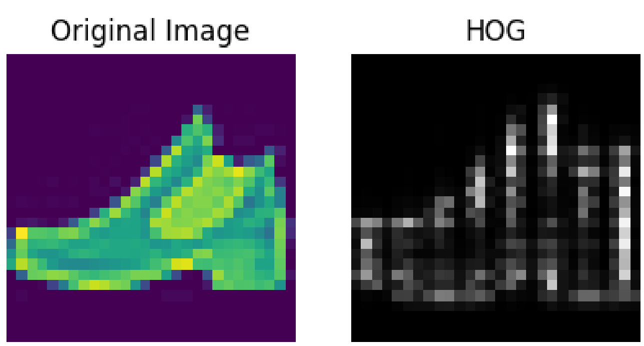

Figure 27: Class 9 - HOG analysis

2.4 Fruits-360 - ORB visualization

Oriented FAST and Rotated BRIEF is an algorithm that tries to identify the key points for an image. The descriptors have a position and an associate size as reflected in the Figures 28-37 for the top 10 most frequent fruits. In order to ensure a constant feature dimensionality these key points descriptors are flatten in a 1×32 vector that represents the mean of all key points descriptors provided. The main idea behind this trick is that an image would provide roughly the same mean descriptor regardless of the rotation and lighting variances.

Figure 28: Most frequent class 0 - ORB analysis

Figure 29: Most frequent class 1 - ORB analysis

Figure 30: Most frequent class 2 - ORB analysis

Figure 31: Most frequent class 3 - ORB analysis

Figure 32: Most frequent class 4 - ORB analysis

Figure 33: Most frequent class 5 - ORB analysis

Figure 34: Most frequent class 6 - ORB analysis

Figure 35: Most frequent class 7 - ORB analysis

Figure 36: Most frequent class 8 - ORB analysis

Figure 37: Most frequent class 9 - ORB analysis

2.5 Fruits-360 - HOG visualization

For this dataset, HOG extracted round shapes for most of images, however it is quite interesting when looking at a few pixel inside these shapes. Most of these pixels include specific lighting features. The images were scaled back from 100×100 for a more fair comparison with the original image, however they are stored as 32×32 images in order to not exceed the available RAM.

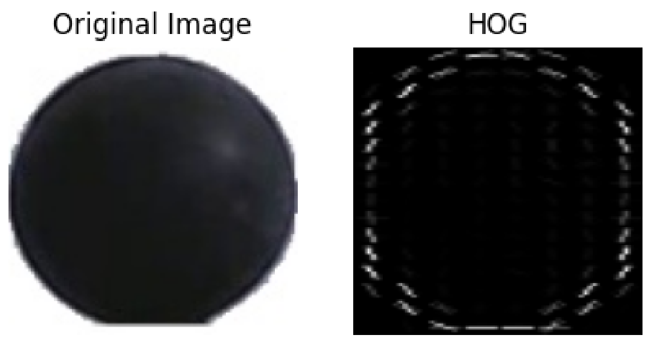

Figure 38: Most frequent class 0 - HOG analysis

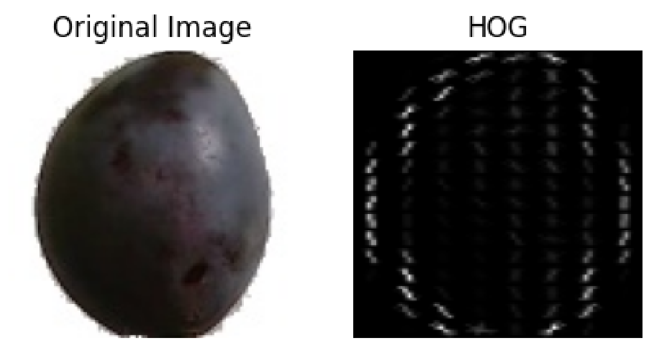

Figure 39: Most frequent class 1 - HOG analysis

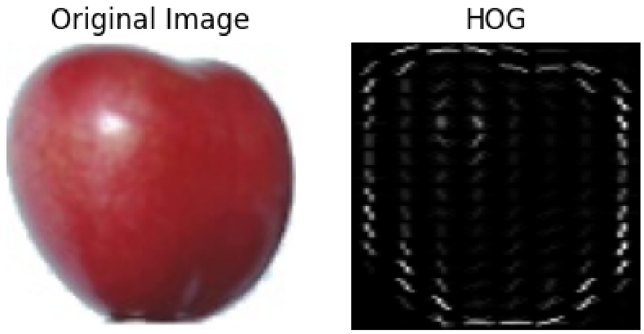

Figure 40: Most frequent class 2 - HOG analysis

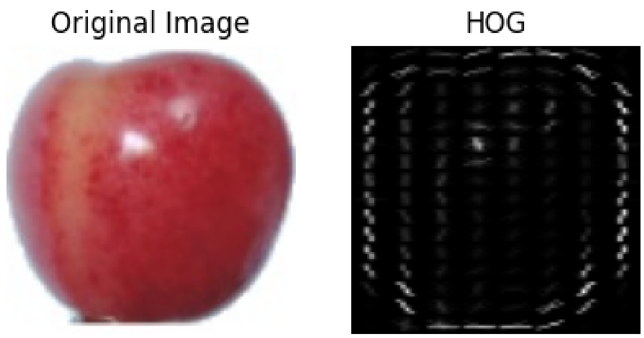

Figure 41: Most frequent class 3 - HOG analysis

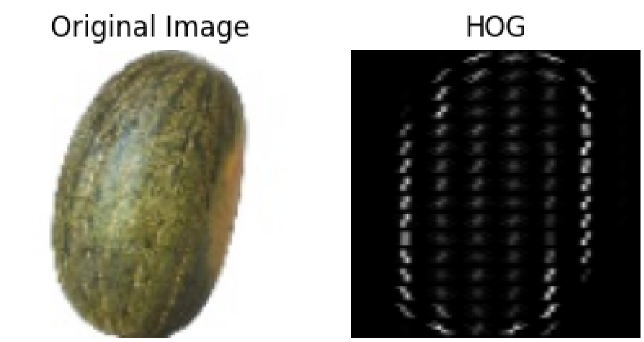

Figure 42: Most frequent class 4 - HOG analysis

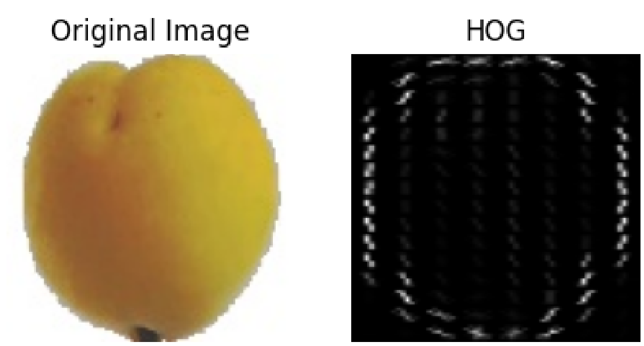

Figure 43: Most frequent class 5 - HOG analysis

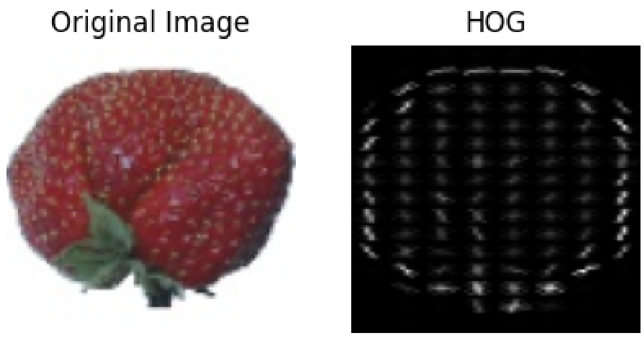

Figure 44: Most frequent class 6 - HOG analysis

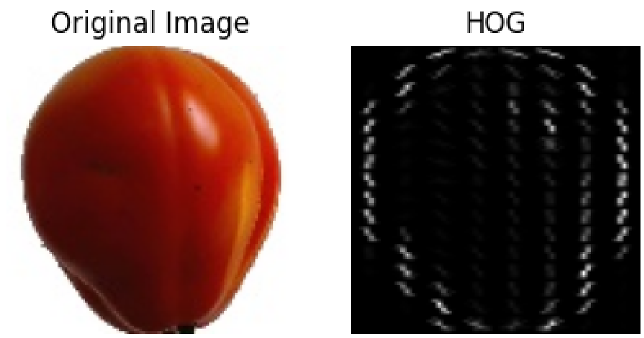

Figure 45: Most frequent class 7 - HOG analysis

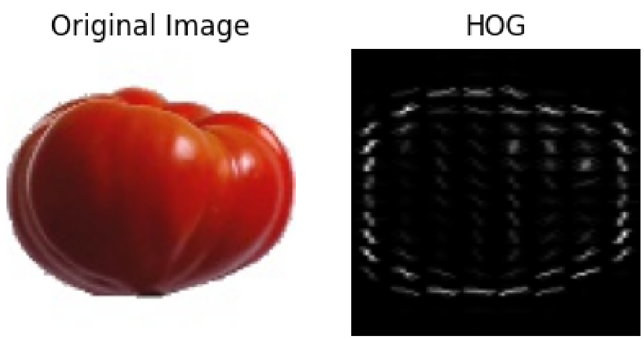

Figure 46: Most frequent class 8 - HOG analysis

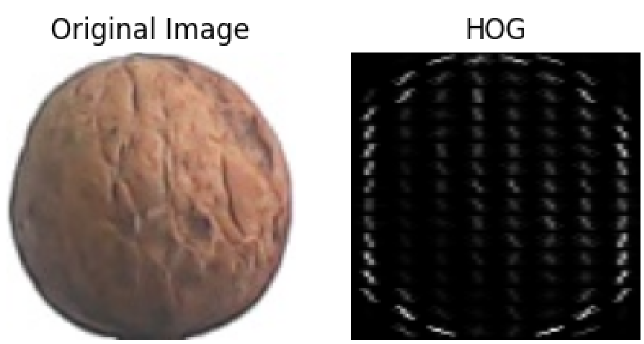

Figure 47: Most frequent class 9 - HOG analysis

3 Feature Standardization & Selection

For the feature standardization I have chosen a StandardScaler to compute upon selective the most relevant features extract for each dataset.

The selection for both sets was done using a VarienceThreshold selection that is choosing the features that have significant variations based on a threshold. For the threshold, as smaller values would prune all the HOG features extracted, at the same time decreasing the efficience of the ML models used, I decided to go with a p value of 0.95 for FashionMNIST and 0.97 for Fruits-360, thus ensuring that the extracted feature must differ at least in one place to be consider relevant. A more ideal approach would have been to get a value close to the proportion of each class $(1 - 0.1 = 0.9)$ meaning that a class actually has very different HOG features.

For the FashionMNIST dataset: before feature selection - 3800 features, after feature selection - 119 (the rest probably have the same value and would slow the model training down without bringing any other value - i.e. insignificant pixel features).

For the Fruits-360 dataset: before feature selection - 3812 features, after feature selection - 286.

4 Classic ML algorithms

4.1 Logistic Regression

For the FashionMNIST dataset hyperparameters: regularization - $l2$, regularization strength - $C = 0.8$, multi_class - *ovr* (binary problem for each label), max_iter - 500.

For the Fruits-360 dataset hyperparameters: regularization - $l2$, regularization strength - $C = 0.2$, multi_class - *multinomial* (the loss minimised is the multinomial loss fit across the entire probability distribution), max_ier - 1000.

4.2 SVM

For the FashionMNIST dataset hyperparameters: regularization strength - $C = 0.6$, kernel - rbf (radial basis function), gamma - *auto* (coefficient for rbf kernel).

For the Fruits-360 dataset hyperparameters: regularization strength - $C = 0.2$, kernel - *linear*, gamma - *auto*.

4.3 Random Forest

For the FashionMNIST dataset hyperparameters: n_estimators - 100, max_depth - 20, max_samples - 10000 (in a batch).

For the Fruits-360 dataset hyperparameters: n_estimators - 100, max_depth - 20, max_samples - 10000 (in a batch).

4.4 Gradient Boosted Trees

For the FashionMNIST dataset hyperparameters: n_estimators - 100, max_depth - 20, learning rate - 0.003.

For the Fruits-360 dataset hyperparameters: n_estimators - 100, max_depth - 20, learning rate - 0.005.

4.5 Results

Overall, the best performing algorithm on the extracted features is SVM. It succeeded to reach the highest for all metrics across both datasets. For the logistic regression, apart from the maximum iterations which led to non converged solutions, the other hyperparameters did not influence the training, obtaining similar results with other configurations of hyperparameters. The SVM proved a similar concept, outperforming the other algorithms, however, it did not change the accuracy that much relative to the modifications of the hyperparameters. Random Forest and Gradient Boosted Trees faced related issues: the depth of the individual trees tremendously influenced the output. I opted for larger trees (20 depth compared to my initial 10 depth) which significantly improved the accuracy with at least 5%. For Random Forest another important hyperparameter was the max batch samples, as for low values the model did not learn much, and I decided to go with aprox. 15% of the training set, as batch samples.

The results are presented below alongside their confusion matrix for the first 10 most frequent classes (for the Fruits-360 dataset plotting all 141 classes did not bring much value and understanding of the results).

For the FashionMNIST all algorithms performed poorly on the class 4 (Coat) and class 6 (Shirt). Probably those are the most generic items of clothing and most other classes (0, 2, 3) have similar features extracted. Relative to the HOG vizualization results we can see that those classes with high False Positives results actually have similar shapes, SVM performed best on classes 1, 3, 7, 8 (Trouser, Dress, Sneaker, Bag) as those had the most distinctive shapes and reduced some bias for the more confusing classes compared to the other algorithms.

The Fruits-360 the confusion matrix does not provide as much value as the number of classes is not reflected accurate in the plots. However, it can clearly be inferred that SVM outperformed all other algorithms, as it has the highest values on the principal diagonal of the confusion matrix. About 7 from the most frequent 10 classes proved to have a high accuracy when compared to the others, especially class 51 (Grape blue) that probably had very distinctive ORB features. We can see that there are a couple of false positives for the classes 31 and 32 (Cherry 2 & Cherry Rainier 1) as they have similar shapes and color. This result might be because as we have extracted HOG and ORB features, the color of the Grape blue is the most distinctive compared to all other colorful fruits.

Algorithm	Accuracy	Precision	Recall	F1
Logistic Regression	0.80	0.80	0.80	0.80
SVM	**0.84**	**0.84**	**0.84**	**0.84**
Random Forest	0.83	0.83	0.83	0.83
Gradient Boosted Trees	0.82	0.81	0.82	0.81

Table 1: ML algorithms analysis for FashionMNIST.

Algorithm	Accuracy	Precision	Recall	F1
Logistic Regression	0.72	0.73	0.72	0.72
SVM	**0.80**	**0.81**	**0.80**	**0.80**
Random Forest	0.69	0.70	0.69	0.68
Gradient Boosted Trees	0.67	0.67	0.67	0.66

Table 2: ML algorithms analysis for Fruits-360.

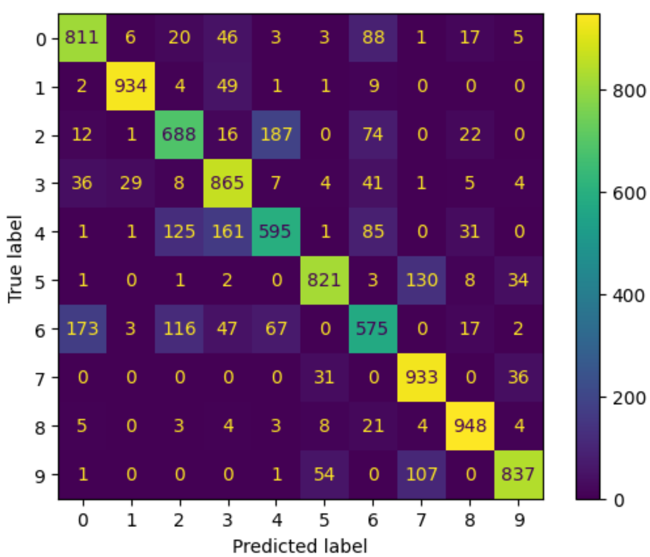

Figure 48: Confusion Matrix for Logistic Regression - FashionMNIST

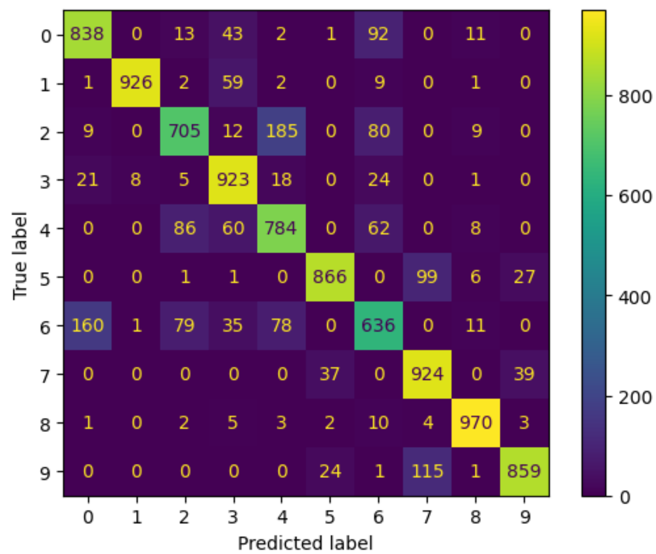

Figure 49: Confusion Matrix for SVM - FashionMNIST

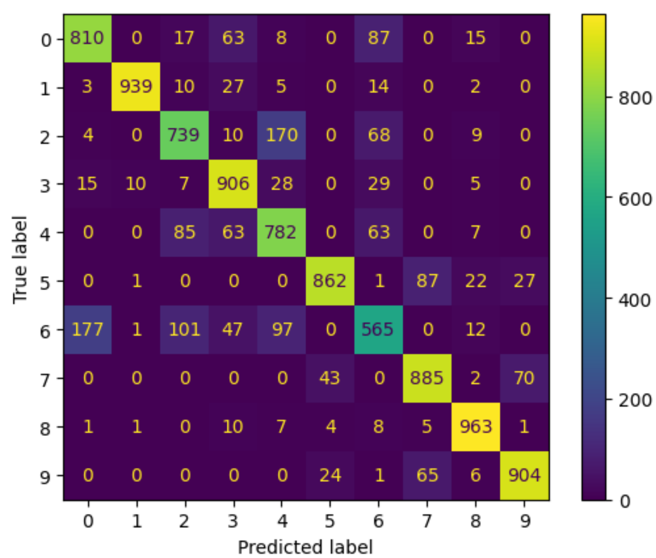

Figure 50: Confusion Matrix for Random Forest - FashionMNIST

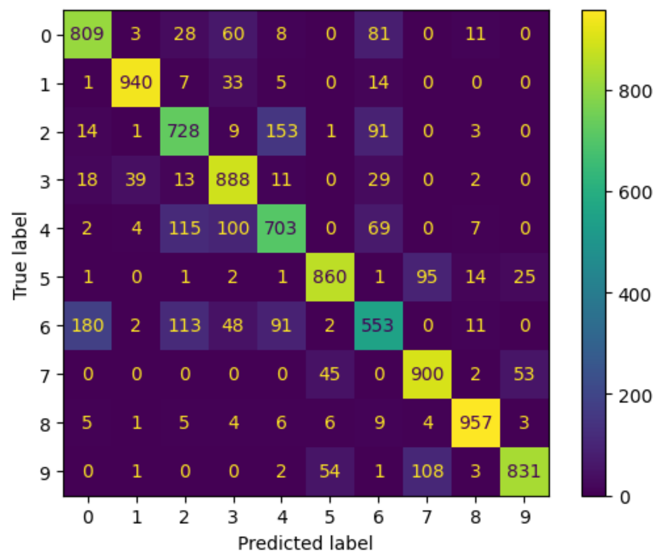

Figure 51: Confusion Matrix for Gradient Boosted Trees - FashionMNIST

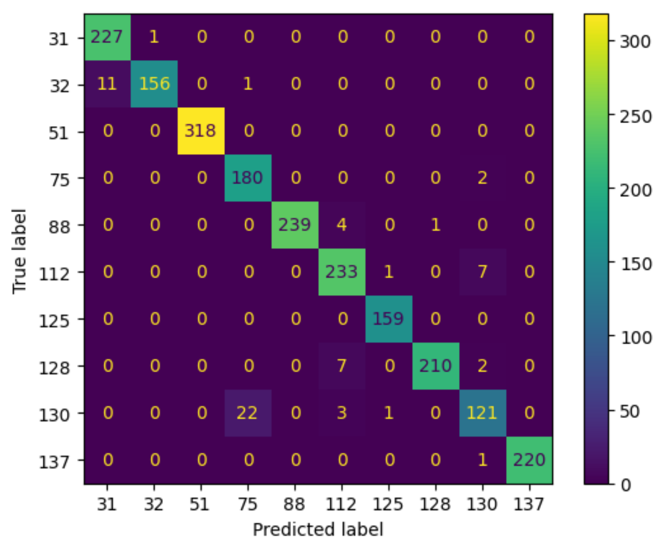

Figure 52: Confusion Matrix for Logistic Regression - Fruits-360

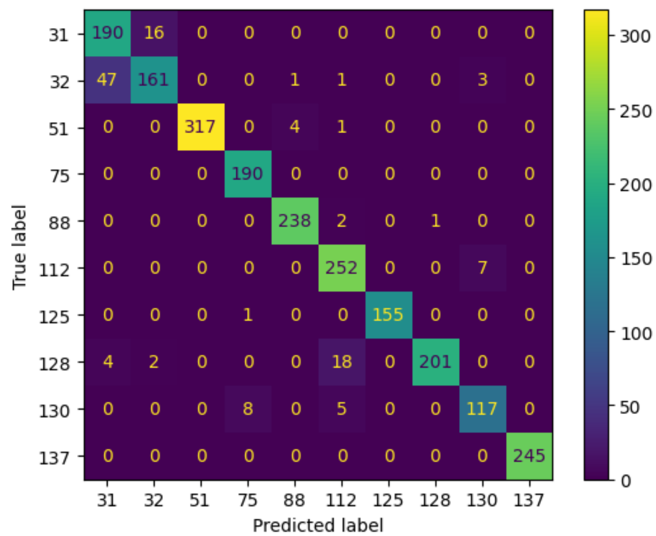

Figure 53: Confusion Matrix for SVM - Fruits-360

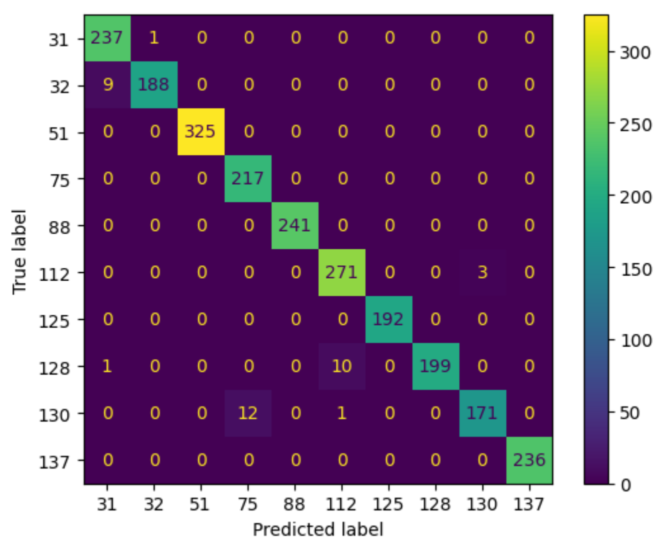

Figure 54: Confusion Matrix for Random Forest - Fruits-360

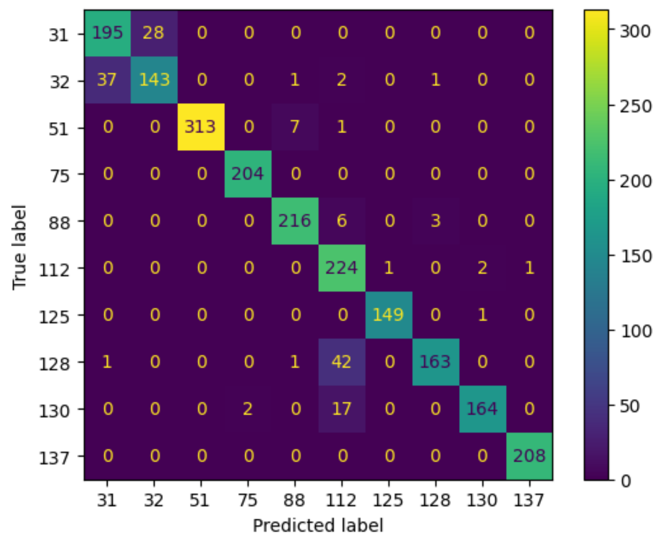

Figure 55: Confusion Matrix for Gradient Boosted Trees - Fruits-360