Practical Machine Learning

Maria-Simona Cioclov

Project 1 - Supervised Methods

Contents

1 Dataset **1**

2 Features **1**
2.1 HOG features 1
2.2 Sobel features 3

3 Adaptive Boosting **3**
3.1 Algorithm 3
3.2 Hyperparameter tuning 3
3.3 Results 3

4 Extra Trees (Extremely Randomized Trees) **6**
4.1 Algorithm 6
4.2 Hyperparameter tuning 6
4.3 Results 6

1 Dataset

The dataset consists of 15000 colored images of 300x200 pixels each. The images are organised in three classes of 5000 samples each, representing hand gestures from the Rock-Paper-Scissors game. The task is to predict which of the three gestures - Rock, Paper, or Scissors - the hand in an image illustrates.

I preprocessed the dataset by first converting each sample to a grayscale image while keeping the original size of the images, as illustrated in Figure 1. Additional preprocessing steps were applied separately for each feature extraction method, as I will present next. Finally, I split the dataset into training and test parts using a ratio of 80% and 20% of the whole dataset respectively. In both subsets, the classes are distributed equally.

2 Features

2.1 HOG features

One type of feature I used for classifying the hand gestures was extracted using the Histogram of Oriented Gradients method, which is a popular technique for detecting objects in computer vision. It works by splitting the grayscale image into smaller cells and computing the gradient of each pixel inside them. For each cell, a histogram of the orientations of the corresponding pixel gradients is created. These histograms are then normalized in groups of cells called blocks, to reduce the influence of lighting or contrast variations. In the end, the feature vector for a sample is formed by concatenating all the histograms of oriented gradients.

I experimented with different parameters for creating the features while plotting the points in an interactive 3D plot, such that I can visualize how well the features separate the different clusters. I used the t-SNE method to reduce the dimensionality of the dataset before plotting the clusters as it can be seen in Figure 2. The best cluster separation was achieved using this configuration: cells of 32x32 pixels, blocks of 4x4 cells, 10 orientation bins per histogram and Euclidean distance for normalizing the blocks. With this setup, each image is now described by 2880 features. To reduce dimensionality, I applied PCA while keeping 95% of the variance, which resulted in a final feature vector of 144 features per sample.

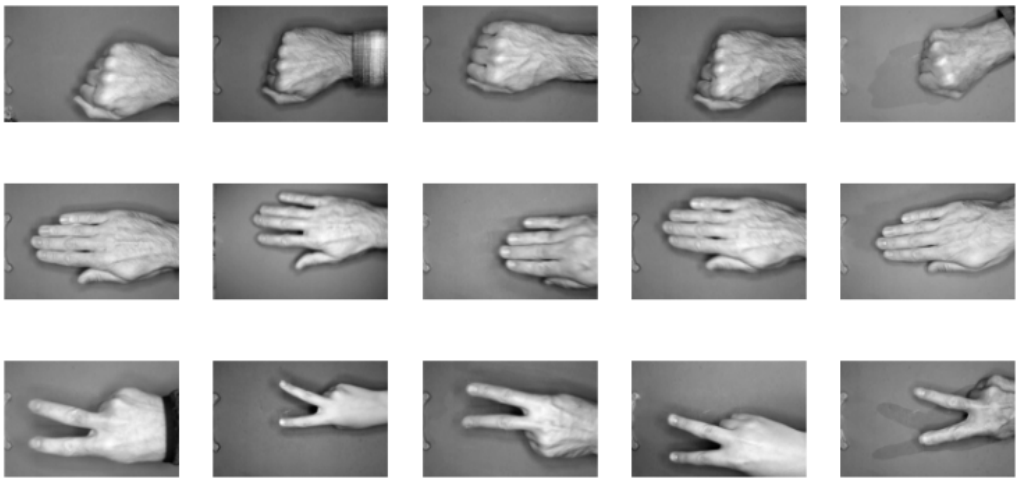


Figure 1: Example grayscale images from the dataset. Each row represents one of the three classes: Rock, Paper, and Scissors.

Figure 2: Labels represent: 0 - rock; 1 - paper; 2 - scissors

2.2 Sobel features

Another type of feature used for classifying the hand gestures was extracted using the Sobel operator, which detects edges in an image, as illustrated in Figure 3. This method computes the gradients of pixels in vertical and horizontal directions, detecting regions with high changes in intensity. For each direction, a specific convolution filter is applied to generate gradient maps. In the end, the two gradient maps are combined to compute the orientation and magnitude of each pixel. The feature vector is formed by concatenating the pixel magnitudes.The best cluster separation was achieved using 7x7 pixel kernels, with Gaussian blur applied to the image beforehand to reduce noise. With this setup, each image is now described by 60000 Features. To reduce dimensionality, I applied PCA while keeping 90% of the variance, which resulted in a final feature vector of 638 features per sample.

A representation of data represented by Sobel features can be seen in Figure 4 where I used t-SNE method to plot the data in a 3D space.

3 Adaptive Boosting

3.1 Algorithm

AdaBoost is a powerful classifier that works by combining contributions of many weak classifiers. Here, a weak classifier is a "stump" - a decision tree that makes only one decision - created using the Gini impurity formula to find the best feature and split point. At first, the algorithm associates to each sample in the dataset the same importance value.The algorithm then trains classifiers iteratively, with each new stump correcting errors from the previous one. This correction step is achieved by increasing the importance of misclassified samples and decreasing the importance of the correctly classified ones. This way, each new classifier focuses more on harder-to-classify examples. The error of a classifier is given by the total importance of misclassified samples. Next, an importance value is also assigned to the classifier, that it's inversely proportional to its error. After a fixed number of iterations, multiple classifiers are combined to make the final decision, with each contributing proportionally to its importance.

3.2 Hyperparameter tuning

For tuning the hyperparameters I used the grid search technique that evaluates all possible combinations of specified hyperparameter values, as shown in Table 1. For validating the results I used cross-validation using 5 folds. This method ensures that the model's performance is evaluated on multiple subsets of data, revealing its generalization capability and reducing the risk of overfitting.

After training, the best parameters found are the same for both HOG and Sobel features: `learning_rate=1.0`, `n_estimators=300`.

In Figures 5 and 6 I analyzed how each hyperparameter values affect best model's performance for both HOG and Sobel features.

To further analyze the relationship between hyperparameter values and model performance, I generated heatmaps that illustrate how the accuracy score changes for different hyperparameter combinations. Figures 7a and 7b show the results for HOG and Sobel features, respectively.

3.3 Results

I tested the best model found on the test dataset. In Figure 8 there are the confusion matrices for both versions of the problem, the one using HOG and the one using Sobel features. Additionally, Table 3 and Table 2 summarize the precision, recall, and F1-score for each class, separately for models trained using HOG and Sobel features.

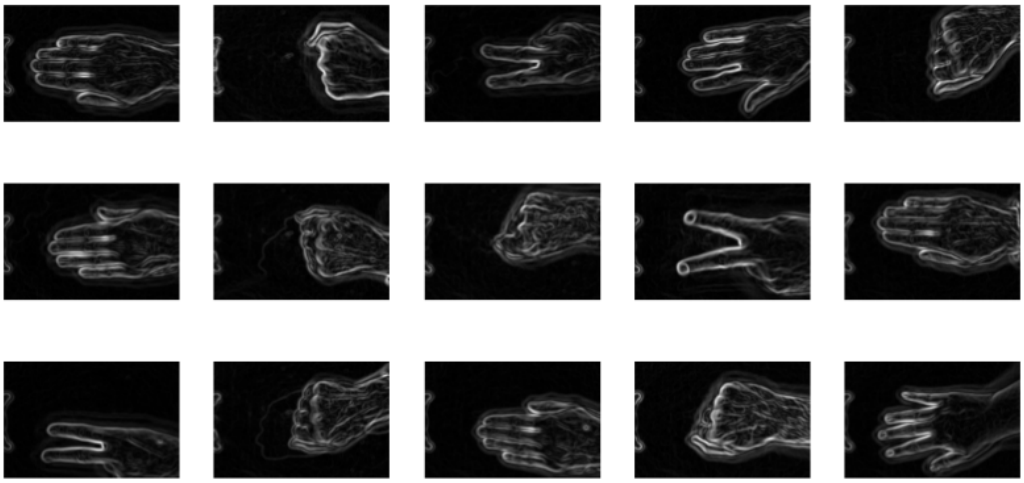


Figure 3: Example images after applying the Sobel filter.

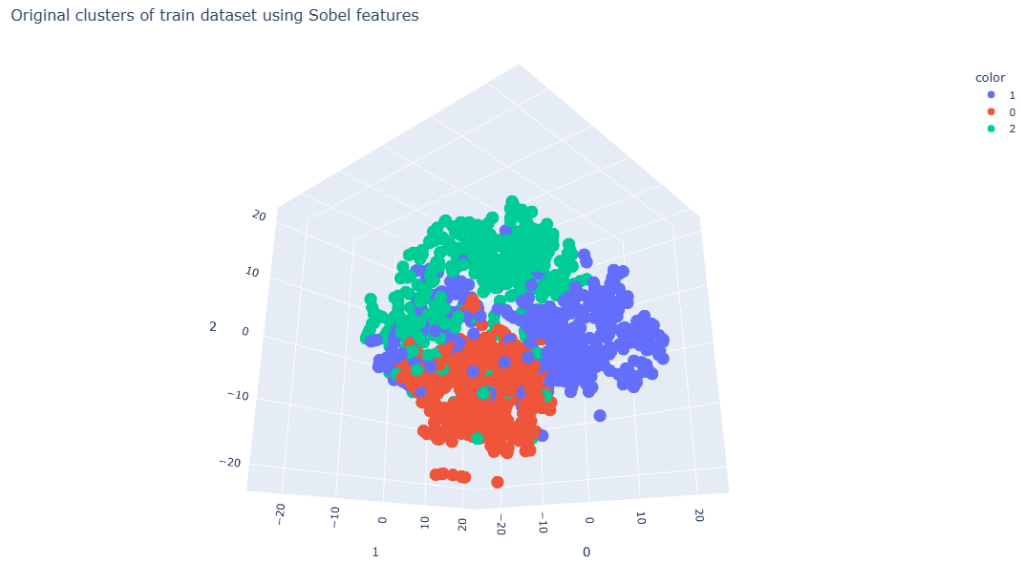


Figure 4: Labels represent: 0 - rock; 1 - paper; 2 - scissors

Parameter	Values Tried	Description
`n_estimators`	50, 100, 200, 300	number of weak classifiers used
`learning_rate`	0.01, 0.1, 1.0, 10.0	weight assigned to each weak classifier's contribution

Table 1: Parameters tuned and values tried for Grid Search

Table 2: Classification Report for Sobel Features

Class	Precision	Recall	F1-score
0	0.87	0.91	0.89
1	0.70	0.87	0.78
2	0.81	0.58	0.67
Accuracy		0.79	

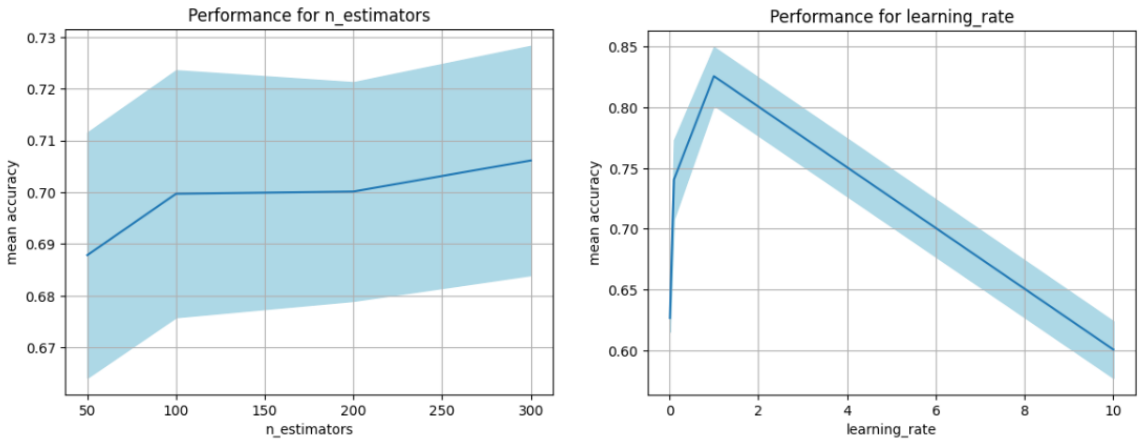


Figure 5: Performance variation with hyperparameters (HOG features)

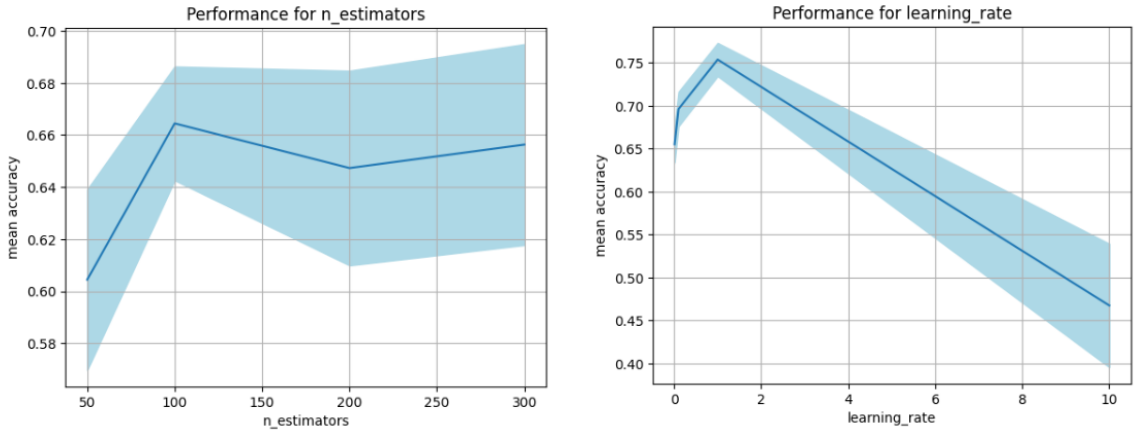
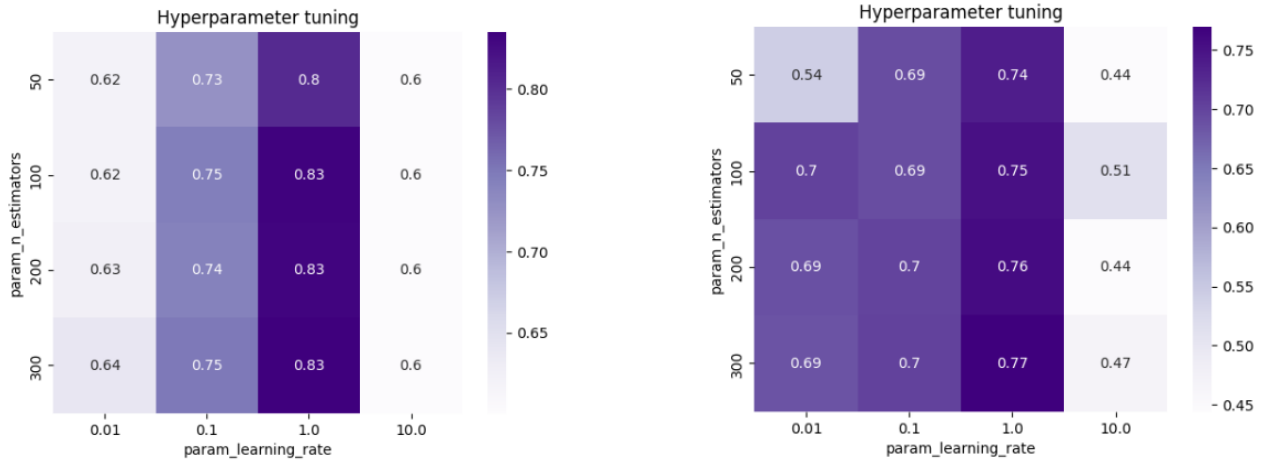


Figure 6: Performance variation with hyperparameters (Sobel features)



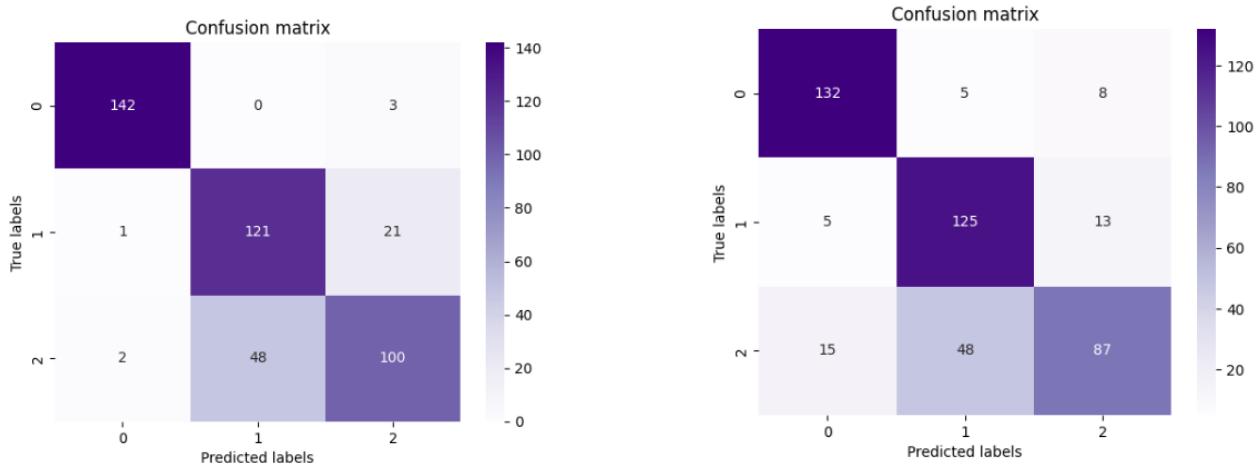
(a) HOG Features

(b) Sobel Features

Figure 7: Heatmaps illustrating accuracy score variations for different hyperparameter combinations using HOG and Sobel features.

Table 3: Classification Report for HOG Features

Class	Precision	Recall	F1-score
0	0.98	0.98	0.98
1	0.72	0.85	0.78
2	0.81	0.67	0.73
Accuracy		0.83	



(a) HOG Features

(b) Sobel Features

Figure 8: Confusion matrices for the best AdaBoost model using HOG and Sobel features.

4 Extra Trees (Extremely Randomized Trees)

4.1 Algorithm

Extremely Randomized Trees (ExtraTrees) is also a classification algorithm that works by combining predictions of multiple decision trees. What differentiates it from the Random Forest algorithm is its greater level of randomness as it selects both the feature and the split threshold randomly, rather than choosing the best threshold based on a criterion like Gini impurity. Also, it uses the entire dataset for building each tree,instead of a random subsample. The increased randomness makes ExtraTrees not only faster, as it skips computing the optimal splits, but also decreases the risk of overfitting the data. One disadvantage, however, is that predictions become less precise, as variation (given by randomness) gets higher. The final decision of a tree is given by a vote of majority in the leaf.

4.2 Hyperparameter tuning

For tuning the hyperparameters I also used the grid search technique as it can be seen in Table 4. I also used cross-validation using 5 folds for validating the results.

After training using HOG features, the best parameters found are: `max_depth=20`, `min_samples_leaf=2`, `min_samples_split=2`, `n_estimators=200`. After training using Sobel features, the best parameters found are: `max_depth=30`, `min_samples_leaf=2`, `min_samples_split=10`, `n_estimators=200`.

In Figures 9 and 10 I analyzed how each hyperparameter values affect best model's performance for both HOG and Sobel features.

4.3 Results

I tested the best model found on the test dataset. In Figure 11 there are the confusion

Parameter	Values Tried	Description
`n_estimators`	50, 100, 200	number of decision trees used
`max_depth`	10, 20, 30	maximum depth of each tree
`min_samples_split`	2, 5, 10	minimum number of samples required to split a node
`min_samples_leaf`	1, 2, 4	minimum number of samples required to be a leaf node

Table 4: Parameters tuned and values tried for Grid Search

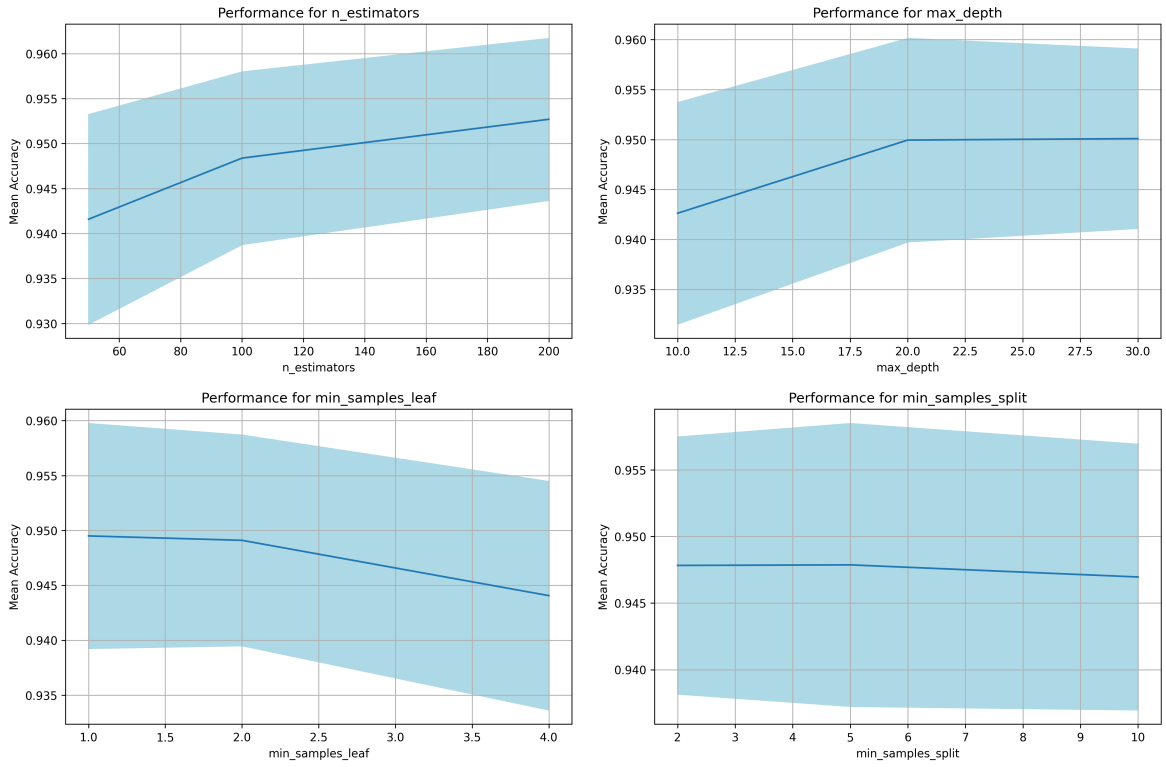


Figure 9: Performance variation with hyperparameters (HOG features)

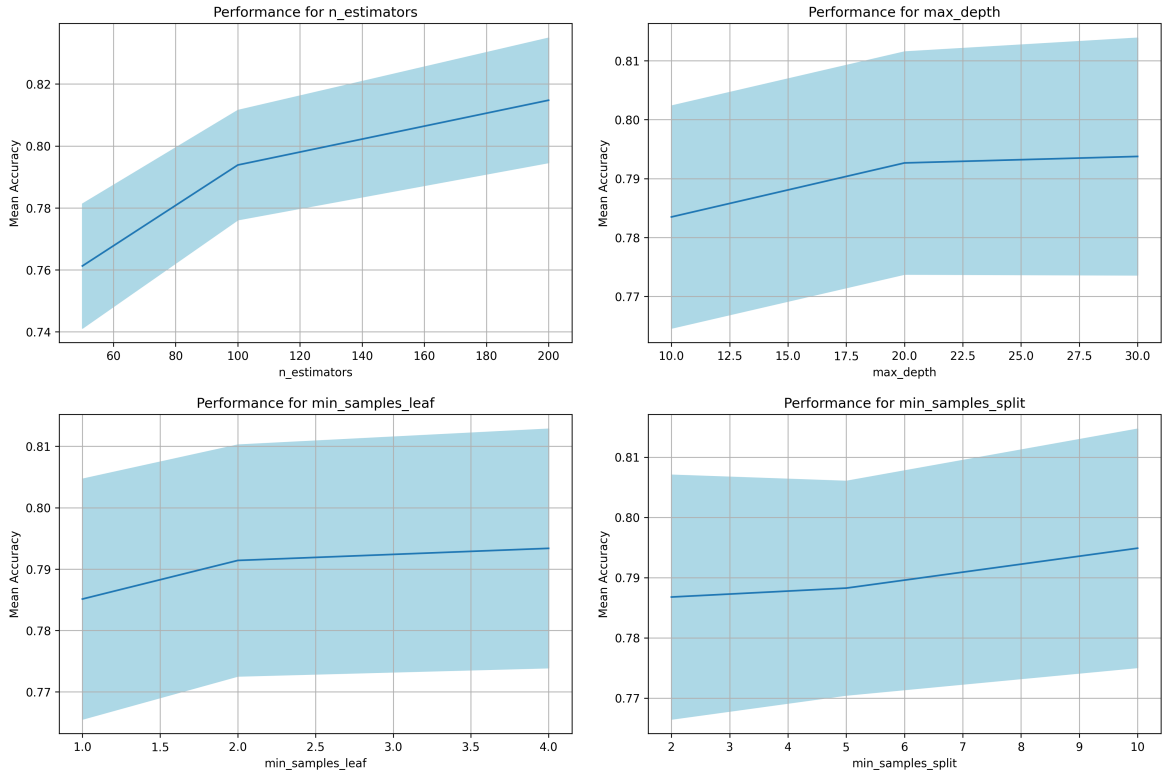


Figure 10: Performance variation with hyperparameters (Sobel features)

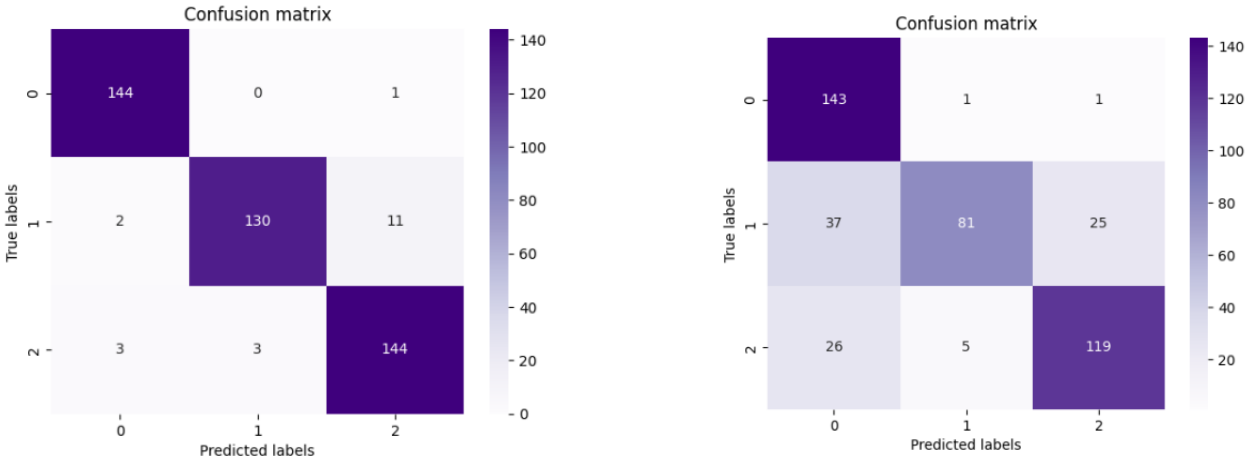
matrices for both versions of the problem, the one using HOG and the one using Sobel features. Additionally, Table 6 and Table 5 summarize the precision, recall, and F1-score for each class, separately for models trained using HOG and Sobel features.

Table 5: Classification Report for Sobel Features

Class	Precision	Recall	F1-score
0	0.69	0.99	0.81
1	0.93	0.57	0.70
2	0.82	0.79	0.81
Accuracy		0.78	

Table 6: Classification Report for HOG Features

Class	Precision	Recall	F1-score
0	0.97	0.99	0.98
1	0.98	0.91	0.94
2	0.92	0.96	0.94
Accuracy		0.95	



(a) HOG Features

(b) Sobel Features

Figure 11: Confusion matrices for the best Extra Trees model using HOG and Sobel features.