

Human Classification/Detection using SVM and Random Forest

[Classification Using Linear HOG and SVM \(Results\)](#)

Performance Scores

Precision Score: 0.95

Recall Score: 0.98

F1 Score: 0.96

Accuracy Score: 0.94

True Positive Rate: 0.97

False Positive Rate: 0.13

Confusion matrix:

Predicted	0	1	Total
Actual			
0	392	61	453
1	27	1105	1132
Total	419	1166	1585

Qualitative Classification Results for SVM:

Predicted Class: [1] Predicted Class: [1] Predicted Class: [1]



Predicted Class: [0] Predicted Class: [0] Predicted Class: [0]



Classification Using HOG and Random Forest Classifier (Results)

Precision Score: 0.98

Recall Score: 1.00

F1 Score: 0.99

Accuracy Score: 0.99

True Positive Rate: 1

False Positive Rate: 0.04

Confusion matrix:

Predicted	0	1	Total
Actual			
0	434	19	453
1	0	1132	1132
Total	434	1151	1585

Qualitative Classification Results for Random Forest:

Predicted Class: [1] Predicted Class: [1] Predicted Class: [1]



Predicted Class: [0] Predicted Class: [0] Predicted Class: [0]



Python Code Files Summary:

- 1- **SVM Classifier.py**: Computes the model for SVM Classifier and stores it in "SVM Model.sav"
- 2- **Random Forest Classifier.py**: Computes the model for Random Forest Classifier and stores it in "RandomForest.sav"
- 3- **"Positive resize" and "Neg Image Resize"**: Resize the positive and negative images, respectively, to the same dimensions (96,160). The positive images (RGBA, .png) and negative images (RGB, .jpg) required different steps to resize to the same diensions.
- 4- **Model scores**: Loads and then tests a model on the given training set. The model can be specified in the code. It returns various scoring parameters.