Computer Vision Research Project

Jalil Inayat-Hussain, Felix Mavrodoglu, Reilly Evans

April 2025

Group Members

- Jalil Inayat-Hussain (22751096)
- Felix Mavrodoglu (23720305)
- Reilly Evans (23615971)

1 Dataset Collection

1.1 Selecting a Dataset

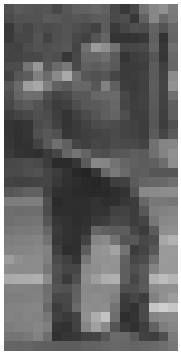
To test our human detection model, we initially used the Daimler Pedestrian Classification Benchmark Dataset [1]. This dataset was selected for three main reasons: It included non-human images, the images were already greyscale, and the images were cropped to a 2:1 ratio of height to width, removing some of the preprocessing needed to obtain the Histogram of Oriented Gradient (HOG) features. However, as seen in Figure 1a, the images in the Daimler dataset have a very low resolution, which we feared would inhibit the performance of our model.

We eventually switched to using the INRIA dataset [2] to improve our model's performance. This dataset uses much larger and more detailed images compared to Daimler, and positive instances have also been cropped to focus on human subjects, as seen in Figure 1b. However, the INRIA images are in colour and thus must be converted to greyscale before we can calculate their HOG features. Additionally, in order to centre each human subject, some of the instances have visible padding on the edges, creating parallel lines that could affect our model's predictions. Finally, the negative samples are not in the same portrait ratio as the positive samples, necessitating that we extract appropriately-sized segments from them rather than using them directly.

1.2 Dataset Preprocessing

We obtained the INRIA dataset by downloading it from the FTP server of the authors of the dataset, using the terminal command `curl ftp://ftp.inrialpes.fr/pub/lear/douze/data/INRIAPerson.tar -o INRIAPerson.tar`. This command was necessary because the original website for the INRIA dataset, `http://lear.inrialpes.fr/data`, does not load at the time of writing.

To generate smaller images to use as negative samples, we designed a Python function `segmentImages()` to randomly generate patches from the negative samples in the dataset's `INRIAPerson/Train/neg` folder. We decided against using the full INRIA dataset, as the GitHub repository we used for this project has a file size limit of 100MB while the INRIA dataset is around 400MB. After generating 5 appropriately-sized patches for each of the 1218 negative instances in the folder, we compressed them along with the cropped positive instances at `INRIAPerson/96X160H96/Train/pos` into a single tarball called `FormattedImages.tar`, containing one folder for positive examples and one for negative examples. We then designed another Python function `createDataset()` to randomly select 4520 samples from this tarball, constructed training and testing sets with them, and save these sets as their own compressed `.tar.gz` files. The functions used here can be found in `projectFunctions.py` as part of our project submission.



(a) Daimler Sample

(b) INRIA Sample

Figure 1: 2 sample images taken from the Daimler and INRIA datasets. The Daimler image is 18x36 pixels, greyscale and uses the Portable Grey Map (.pgm) file format (though it has been converted to Portable Network Graphics (.png) here in order to render it properly). The INRIA image is 96x160 pixels, in colour and uses the PNG format. Some vertical lines indicating the usage of edge padding can be seen at the bottom of the INRIA image.

Our final dataset has 3600 images in the training set and 900 images in the testing set. Both sets have 50% positive and 50% negative instances. These datasets give our models a sufficient amount of data to be trained on, while staying under GitHub's 100MB file size limit. The `createDataset()` function also selects 20 images (10 positive, 10 negative) not present in either set to use for testing our GUI.

2 Feature Extraction & Model Training

2.1 HOG Feature Extraction Properties

The final set of properties that we used for the extraction of HOG characteristics is as follows:

- Preprocessing: Raw grayscale images without gamma correction
- Orientation Binning: 12 bins covering angles 0°–180°
- Cell Size: 8x8 pixels
- Block Size: 2x2 cells (16x16 pixels)
- Block Normalisation: L2-Hys normalisation
- Block Stride: 8-pixel spacing (4-fold coverage of each cell)
- Detection Window Size: 64x128 pixels
- Classifier: Linear SVM for binary classification

After performing ablation studies on several of these properties, we found that for all parameters except the number of orientation bins, the sample parameters given in the Dalal and Triggs paper [2] produced the highest performance for the SVM model. This commonality likely results from the fact that the aforementioned paper also used the INRIA dataset.

2.2 Model Training

The model we used for this project was a linear Support Vector Machine implemented via scikit-learn's `LinearSVC` class. This class was chosen because it scales better to larger datasets compared to other SVM implementations, thus reducing the time it would take to train the dataset. We also tested the performance

of nonlinear SVMs using the `SVC` class in scikit-learn, which resulted in better predictions on the testing dataset. However, the nonlinear SVM also took exponentially longer to fit upon the training data; to ensure that waiting for the model to train did not impede our ablation studies, we decided to continue using the linear implementation instead.

The hyperparameters we used for the SVM are as follows:

- Penalty: L2 / ridge regularisation
- Loss Function: squared hinge loss
- C (Regularisation Hyperparameter): 0.01

These parameters were selected through a grid search technique, examining various combinations of parameters with 5-fold cross-validation on our training set. For our final model, the SVM was fitted onto our training set and used to make predictions on the testing set; these predictions would be used to calculate the performance metrics.

2.3 Model Evaluation

2.3.1 Performance Metrics

For this project, we have evaluated the performance of our model using the following methods:

- Accuracy
- Precision Score
- Recall Score
- F1 Score
- Analysis of Confusion Matrix
- Analysis of Receiver Operating Characteristic (ROC) Curve
- Area Under ROC Curve (AUC)
- Analysis of Detection Error Trade-off (DET) Curve

These metrics were calculated using the corresponding scikit-learn functions, the true labels for each image in the testing dataset, and the predictions from the model on the testing dataset. The Dalal and Triggs paper [2] uses DET curves for its performance analyses, but with false positives per window (FPPW) plotted on the x-axis instead of the typical false positive rate (FPR). However, our HOG feature extraction process only uses a single window for each image, which means that FPPW is equivalent to FPR on our dataset; as such, our DET curves still use FPR.

2.3.2 Results

Accuracy (%)	**Precision (%)**	**Recall (%)**	**F1-Score**	**Area Under Curve (AUC)**
97.5556	97.9821	97.1111	0.975446	0.997491

Table 1: Performance metrics of our SVM model on HOG features extracted from the INRIA dataset.

3 Ablation Study

Ablation studies will be conducted to assess the impact of normalisation techniques, the number of bins and block size.

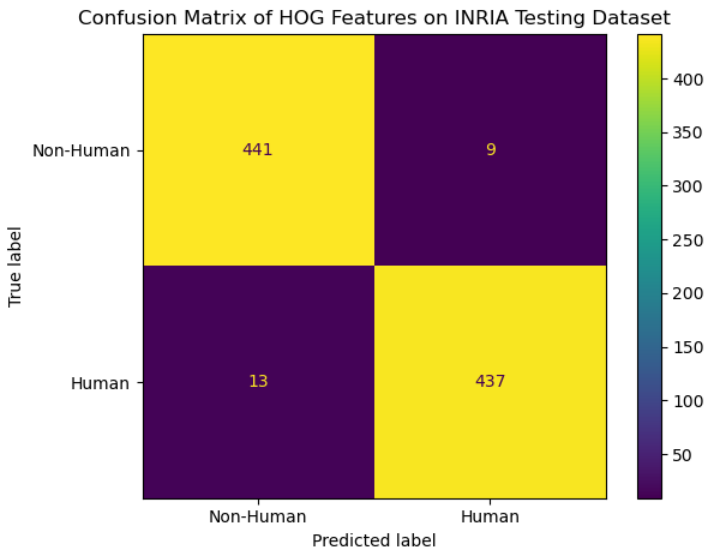


Figure 2: Confusion matrix of our SVM model on HOG features extracted from the INRIA testing dataset.

3.1 Baseline Configuration

The baseline configuration for these studies uses these parameters (excluding those identical to our final model):

- Orientation Binning: 9 bins covering angles 0°–180°
- SVM C (Regularisation Hyperparameter): 1

3.2 Normalisation Techniques

We examined the performance of 4 different normalisation techniques:

- L1 normalisation (Manhattan norm)
- L1-sqrt normalisation (square root of Manhattan norm)
- L2 normalisation (Euclidean norm)
- L2-Hys normalisation (Lowe-style clipping on Euclidean norm; default normalisation technique)

3.2.1 Results

Normalisation Technique	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score**	**Area Under Curve (AUC)**
L1	96.2222	95.2174	97.3333	0.962637	0.99401
L1-sqrt	96.4444	95.6332	97.3333	0.964758	0.995151
L2	95.5556	95.1542	96.0000	0.955752	0.993126
L2-Hys (default)	96.5556	96.4523	96.6667	0.965594	0.99557

Table 2: Performance metrics of models with different normalisation techniques.

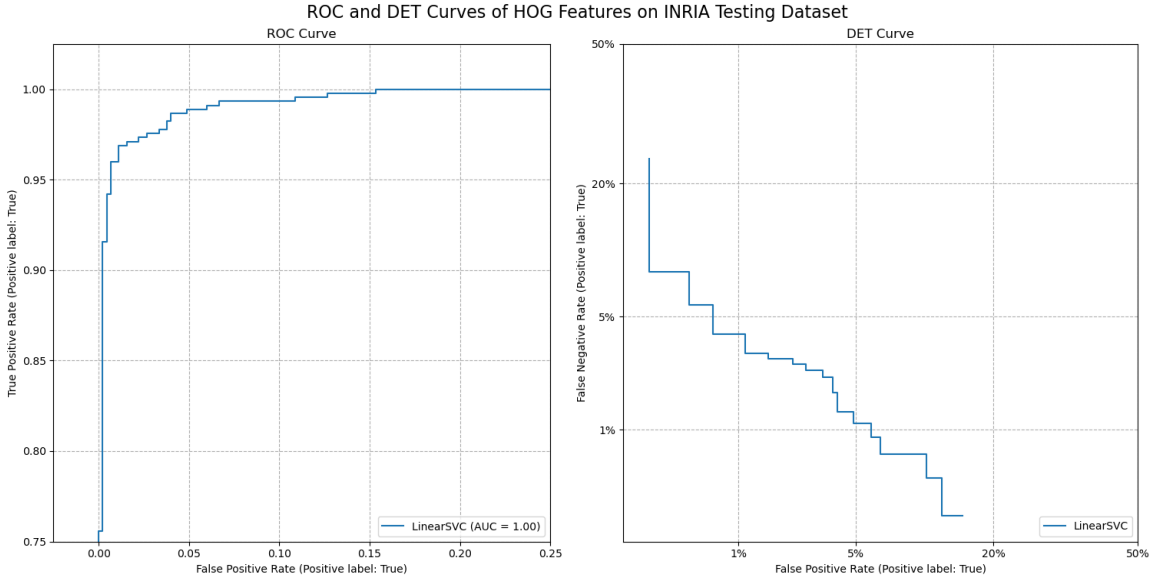


Figure 3: ROC and DET curves for our SVM model on HOG features extracted from the INRIA testing dataset. Note that the x and y limits on both curves have been adjusted to show more detail at the corner of the curve.

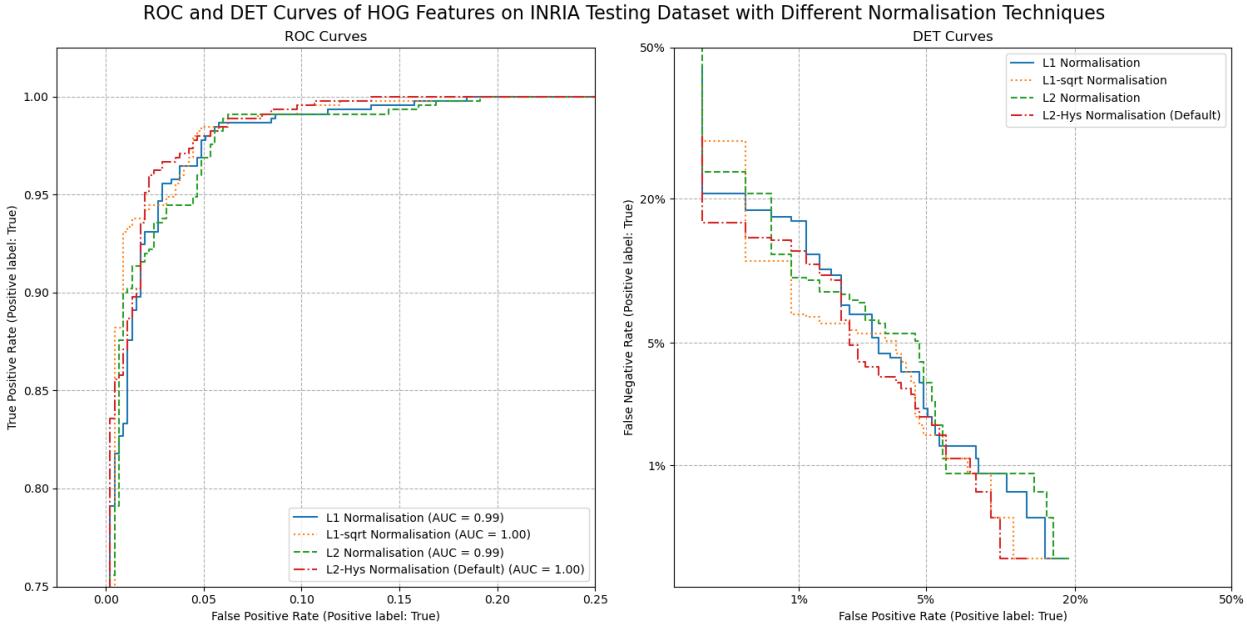


Figure 5: ROC and DET curves for models with different normalisation techniques.

3.2.2 Performance Analysis

- L2 performs the worst in all metrics.
- L2-Hys has the highest accuracy, precision, F1 and AUC.
- L1 and L1-sqrt have identical recall; L1-sqrt performs better than L1 in other metrics.
- L2-Hys had the most even split between positive/negative predictions.

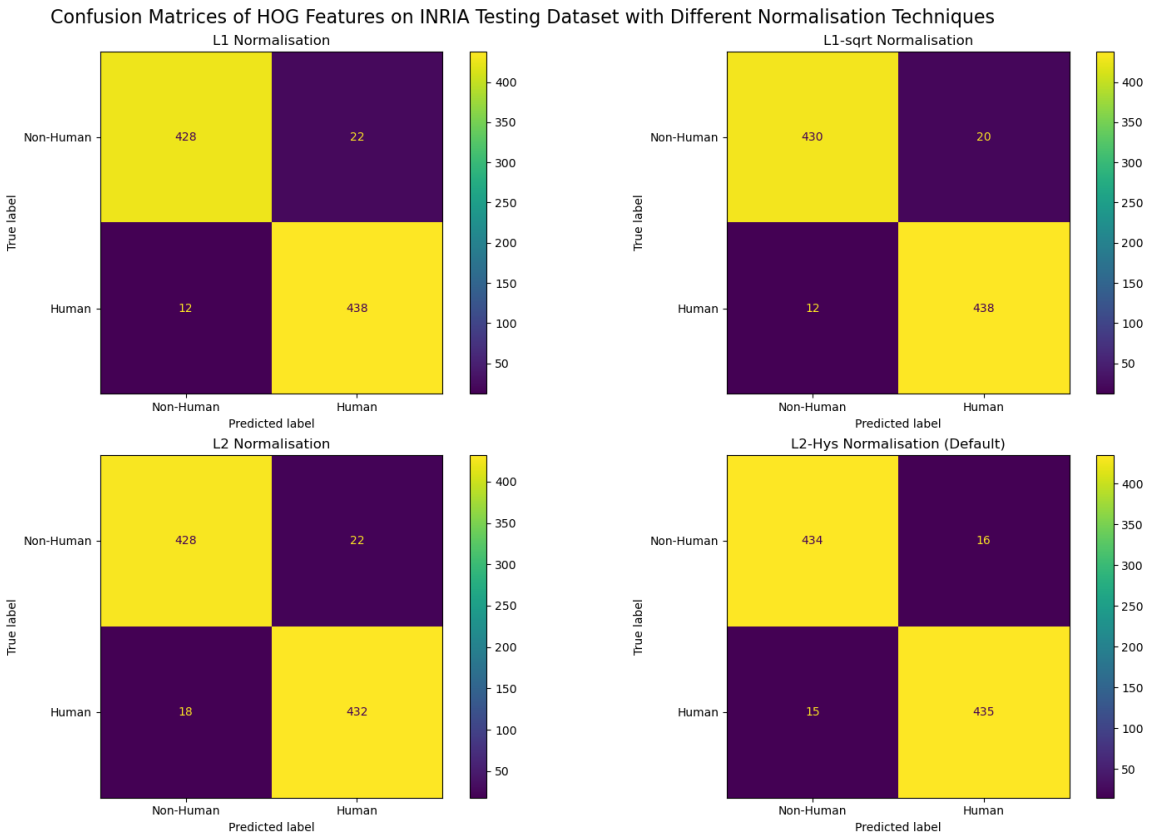


Figure 4: Confusion matrices for models with different normalisation techniques.

- Feature calculation time was similar across all techniques, but L2-Hys consistently took the longest.

3.2.3 Optimal Configuration

The results indicate that L2-Hys is optimal; its accuracy, F1 and AUC are the highest, and the balanced divide between positive/negative predictions indicates minimal skewing. The lower recall and higher computation time are minor enough to be accepted; for real-world deployment with larger datasets we might consider switching to L1-sqrt if the time difference became noticeable.

3.3 Block Size

In this study we vary the size of the HOG block, the number of adjacent 8×8 pixel cells grouped and normalised together, while keeping all other HOG parameters at their default values.

Table 3: Performance metrics of HOG+SVM on the INRIA test set for varying block sizes

Block Size	Accuracy (%)	Precision (%)	Recall (%)	F1-Score	AUC
1×1	93.00	93.10	92.89	92.99	0.98135
2×2 (default)	96.67	96.46	96.89	96.67	0.99547
3×3	96.33	96.03	96.67	96.35	0.99458
4×4	96.11	95.21	97.11	96.15	0.99374
5×5	95.33	94.74	96.00	95.36	0.99240
6×6	95.44	94.95	96.00	95.47	0.99125

3.3.1 Results

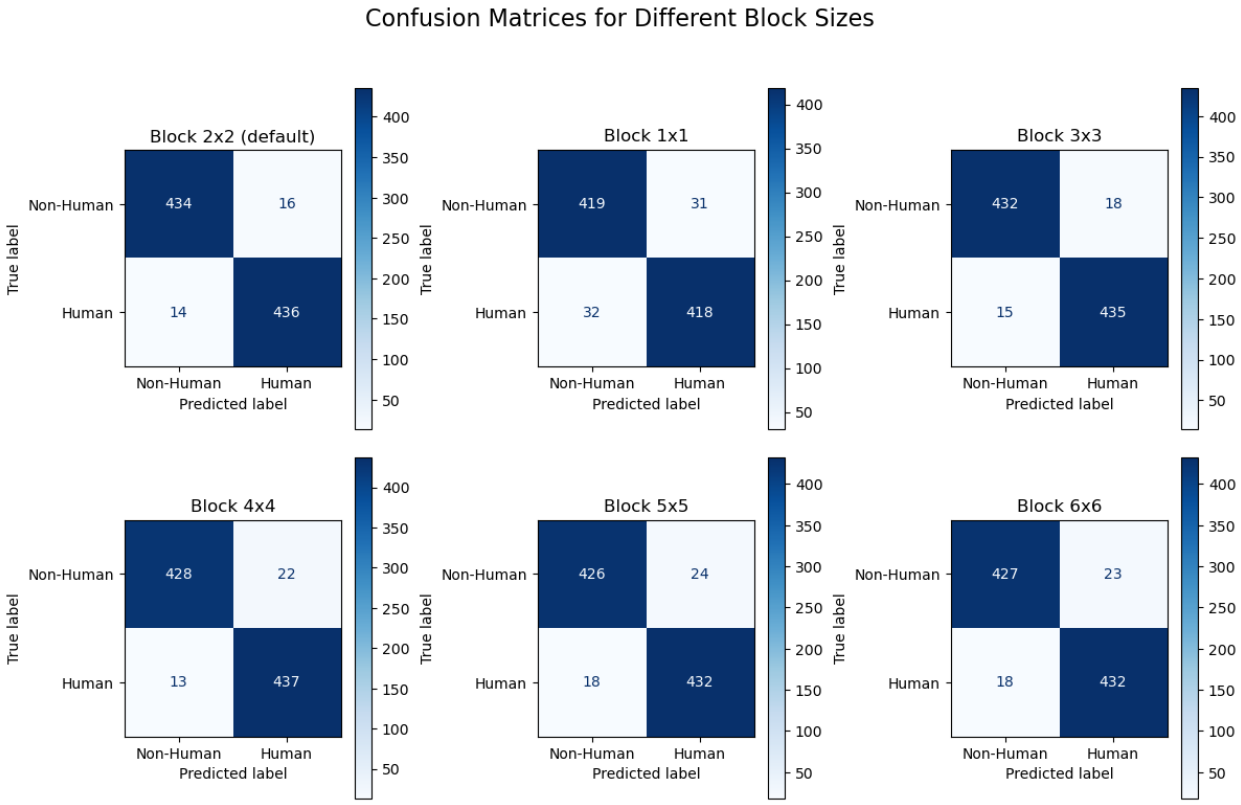


Figure 6: Confusion matrices for different HOG block sizes on the INRIA testing dataset.

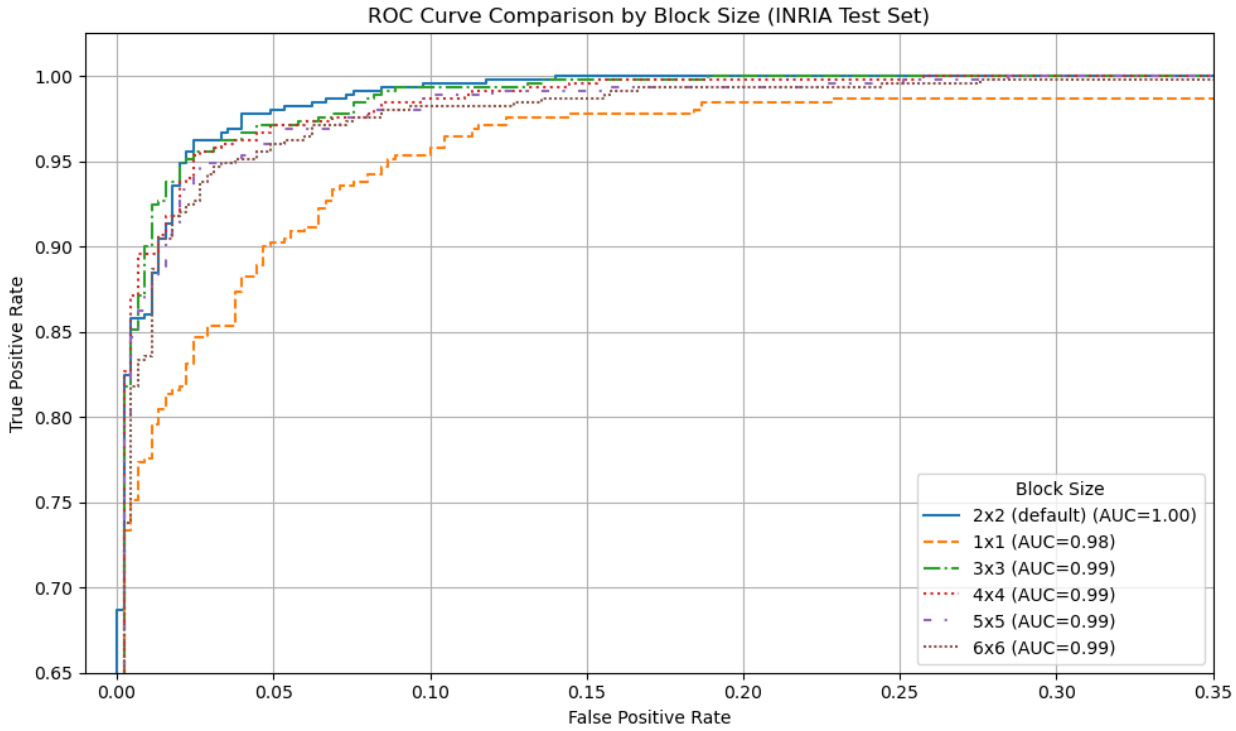


Figure 7: ROC curves for different HOG block sizes on the INRIA testing dataset.

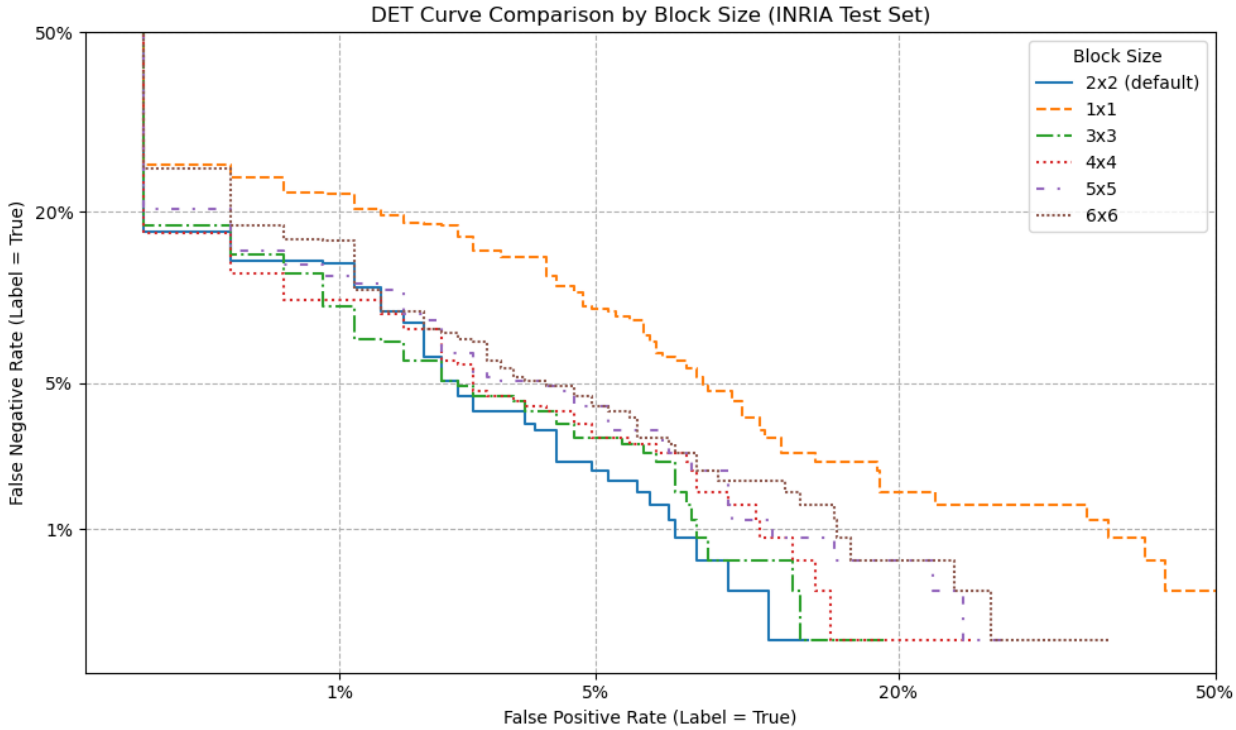


Figure 8: DET curves for different HOG block sizes on the INRIA testing dataset.

3.3.2 Performance Analysis

The ablation results in Table 3 and Figures 6–7–8 show:

- 1×1 blocks lose spatial context and perform poorly.
- 2×2 blocks hit the best detection–false-alarm balance.
- 3×3–4×4 blocks boost recall but hurt precision and accuracy.
- Blocks larger than 4×4 become too coarse, with diminishing returns.

3.3.3 Optimal Configuration

The 2×2 cell block remains the best choice, offering the strongest balance of true positive and negative rates. While a 4×4 block slightly boosts recall, it lowers precision and overall accuracy, and larger blocks perform progressively worse. Since the extraction time is essentially unchanged, we will stick with the 2×2 for the future work.

3.4 Number of Bins

3.4.1 Results and Analysis

Orientation Bins	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score**	**Feature Size**
6	96.4444	95.6332	97.3333	0.964758	2520
9 (default)	96.6667	96.4602	96.8889	0.966741	3780
12	97.0000	97.3154	96.6667	0.969900	5040
18	96.5556	96.6592	96.4444	0.965517	7560

Table 4: Performance comparison of HOG feature extraction with varying orientation bin counts

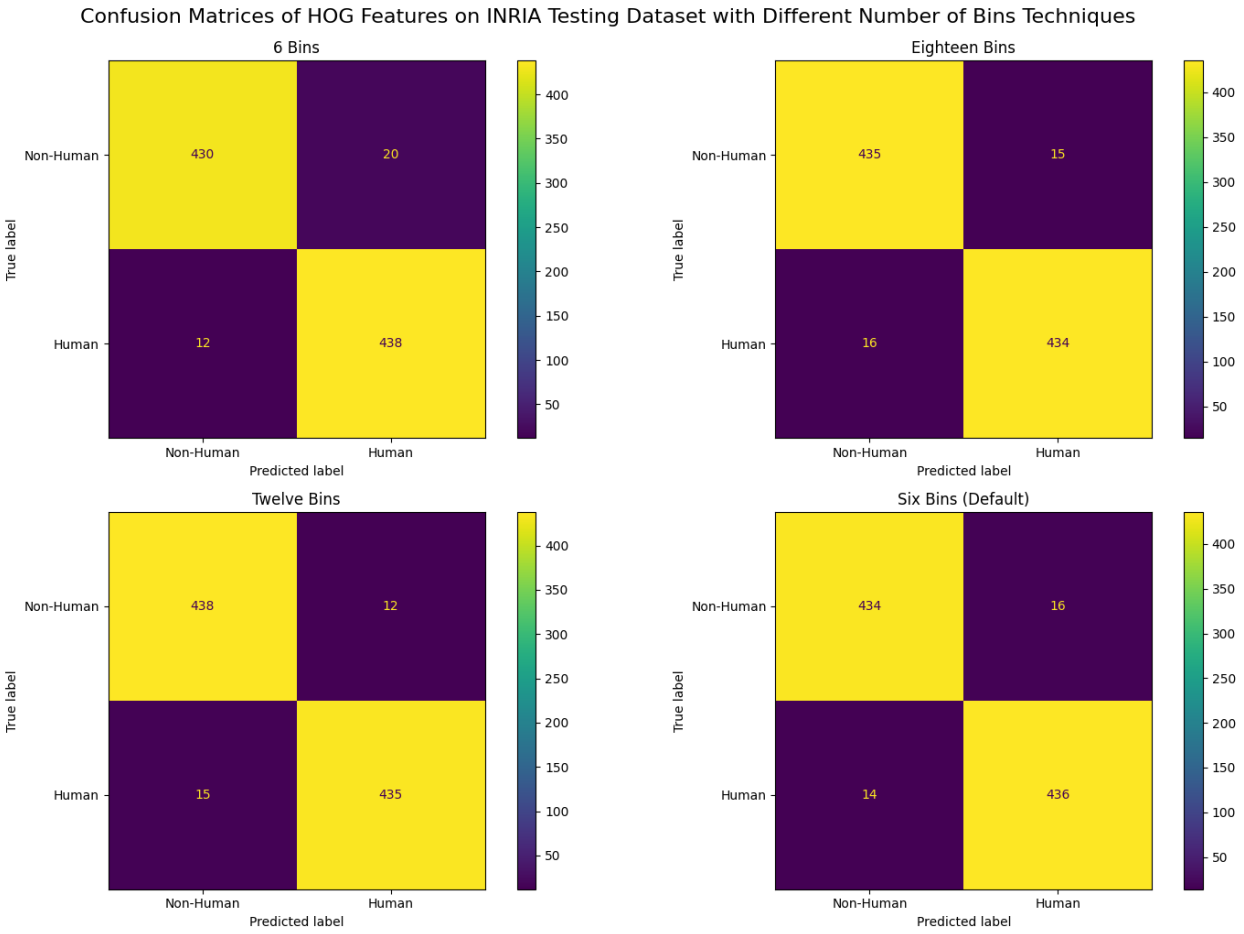


Figure 9: Confusion Matrices for Different Number of Bins

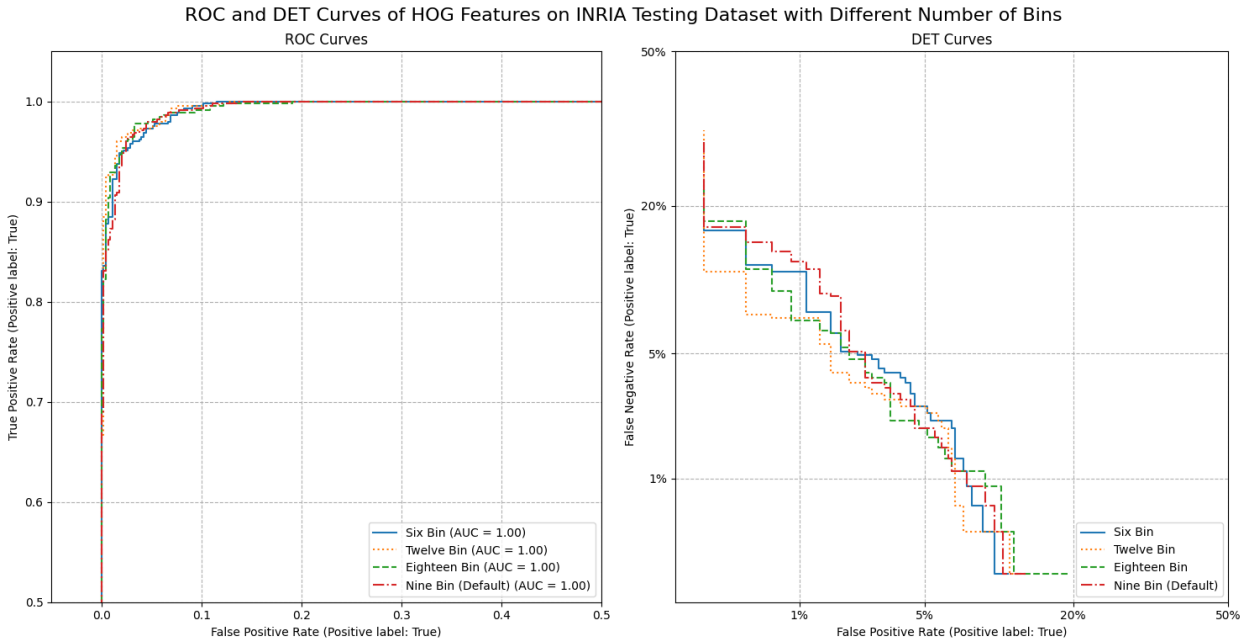


Figure 10: DET and ROC Curves for Different Number of Bins

3.4.2 Performance Analysis

The ROC curves display exceptional AUC values of 1.00 across all configurations (Six Bin, Twelve Bin, Eighteen Bin, and Nine Bin Default) while the confusion matrices reveal:

- Six bin configuration performs slightly worse with 96.4% accuracy and highest false positive rate (20)
- Twelve bin configuration achieves the best performance with 97.0% accuracy
- Eighteen and nine bin (default) configurations show nearly identical performance at 96.6% and 96.7% accuracy respectively
- Performance peaks at twelve bins with minimal computational overhead compared to eighteen bins

3.4.3 Optimal Configuration

Based on the results, the twelve bin configuration represents the optimal choice:

1. It achieves the highest classification accuracy (97.0%) among all tested configurations
2. Demonstrates the lowest false positive rate (2.7%), critical for human detection applications
3. Maintains balanced error distribution with only 12 false positives and 15 false negatives
4. Provides optimal trade-off between feature granularity and computational efficiency

For real-world deployment, especially in security systems or automated surveillance where accurate human detection is critical, the twelve bin provides the best compromise between detection performance and computational requirements.

References

[1] S. Munder and D.M. Gavrila. "An Experimental Study on Pedestrian Classification". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.11 (2006), pp. 1863–1868. DOI: 10.1109/TPAMI.2006.217.

[2] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 886–893.