North South University

Final Report

Stop Sign Detection

CSE445
Machine Learning
Section: 5

Submitted to
Dr. Mohammad Shifat-E-Rabbi

Assistant Professor, Dept. of ECE

Submitted by
Group - 08
14th December, 2025

Name	ID
Md. Rakibul Islam Rakib	1911977042
Md. Mobenul Islam Chowdhury	2012022042
Unaiza Nawar Tanisha	2131103642
Yakin Nazif	2131862642

1. Abstract

The stop sign detection project uses traditional machine learning models on 50 street images. The SVM model outperformed the others, achieving a test accuracy of 40.25% and a training accuracy of 89.74%.

2. Introduction

2.1 Background and Motivation:

Stop Sign Detection plays an important role in Intelligent Transportation Systems and Self-Driving. As the systems must recognize traffic signs and respond accordingly, accurate detection thus means safer roads. Deep learning models run a risk of overfitting when dealing with a small dataset. In contrast, traditional machine learning models, such as support vector machine (SVM), K nearest neighbour (KNN), and random forest, can work very well with a smaller dataset.

2.2 Problem Statement:

The objective is to automatically detect stop signs in 50 street images using machine learning models. The challenge lies in achieving high accuracy with a limited dataset and varying image conditions.

2.3 Objectives:

- To collect and preprocess 50 street images containing stop signs.
- To implement and compare SVM, KNN, and Random Forest models and select one.

3. Literature Review

This research compares three machine learning models, SVM, Random Forest, and KNN, to determine the most efficient for stop sign detection, considering dataset quality and preprocessing, and aims to improve traffic sign detection performance.

4. Methodology

4.1 Dataset Collection:

- **Source:** 50 images were collected from various online sources, ensuring each image contained at least one visible stop sign.
- **Image Characteristics:**
  - Different backgrounds (urban, rural).
  - Varying lighting conditions.
  - Different angles and distances from the stop sign.

4.2 Preprocessing:

- **Grayscale Conversion:** Simplifies processing by reducing color complexity.
- **Resizing:** All images resized to 128x128 pixels to standardize input dimensions.
- **Normalization:** Pixel values normalized to improve model performance.

4.3 Feature Extraction

- **Histogram of Oriented Gradients (HOG):**
  - Extracts edge and shape information, crucial for identifying the distinct octagonal shape of stop signs.
  - HOG features are robust to variations in lighting and pose.

4.4 Model Selection:

- **Support Vector Machine (SVM):**
  - Effective for binary classification tasks.
  - Uses hyperplanes to separate data points.

```
model = svm.SVC(kernel='linear', C=1.0)
model.fit(X_train, y_train)
✓ 0.0s

SVC
SVC(kernel='linear')
```

- **K-Nearest Neighbors (KNN):**
  - A distance-based classifier that assigns class labels based on the majority class among the k-nearest neighbors.

```
# Train the KNN model
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train, y_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```

  - Simple to implement and interpret.
- **Random Forest:**
  - Ensemble method that builds multiple decision trees.

```
# Train a Random Forest classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train_pca, y_train)
```

4.5 Training and Testing:

Train-Test Split: 80% of the data was used for training, and 20% for testing.

- **Metrics Used:**
  - Accurac
  - Precision
  - Recall
  - F1 Score

4.6 Tools:

- Python, Jupiter Notebook

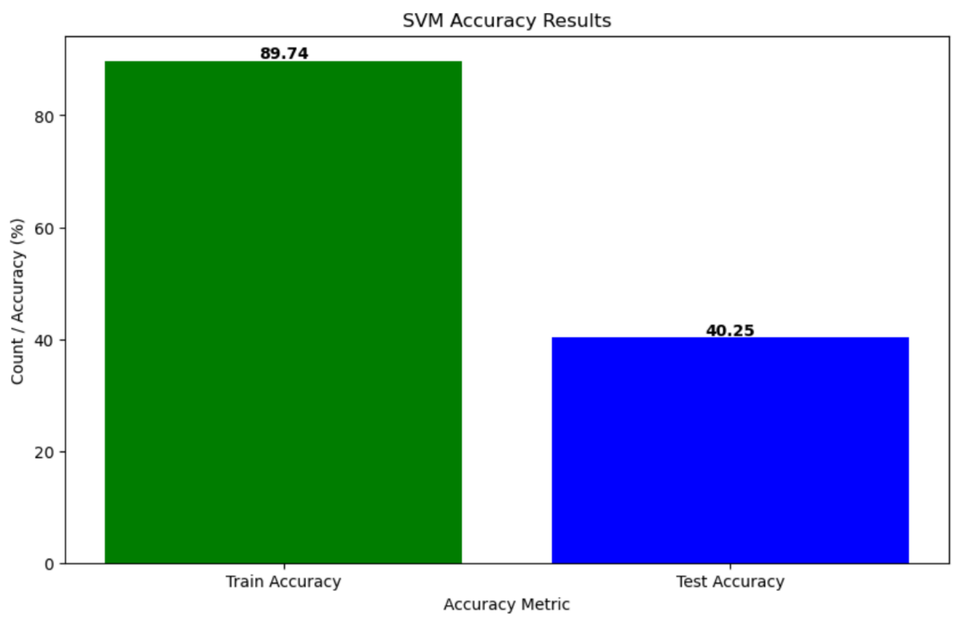
5. Results and Analysis SVM

Model Performance:

```
SVM Model Accuracy: 0.4025157232704403
Classification Report:
              precision    recall  f1-score   support

           0       0.43      0.69      0.53        77
           1       0.31      0.13      0.19        82

    accuracy                           0.40       159
   macro avg       0.37      0.41      0.36       159
weighted avg       0.37      0.40      0.35       159
```



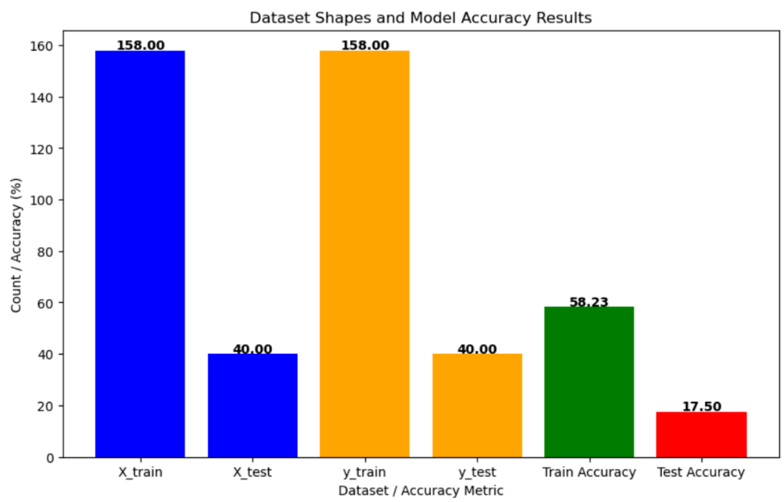
- **Accuracy:** 40.25
- **Classification Report:**
- Precision: 37 | Recall: 41 | F1-score: 36

KNN Model Performance:

```
KNN Model Accuracy: 0.175
Classification Report:
              precision    recall  f1-score   support

           0       0.07      0.05      0.06        21
           1       0.23      0.32      0.27        19

    accuracy                           0.17        40
   macro avg       0.15      0.18      0.16        40
weighted avg       0.15      0.17      0.16        40
```



- **Accuracy:** 17.5%
- **Classification Report:**
- Precision: 0.15 | Recall: 0.18 | F1-score: 0.16

Random Forest Model Performance:

```
... Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
    Accuracy: 100.00%
    Classification Report:
                  precision    recall  f1-score   support

               1       1.00      1.00      1.00        10

        accuracy                           1.00        10
       macro avg       1.00      1.00      1.00        10
    weighted avg       1.00      1.00      1.00        10
```

- **Accuracy:** 100% (Overfitting detected)
  - **Issue:** The model achieved perfect accuracy, indicating potential overfitting due to the small dataset.

Analysis:

Model	Train Accuracy	Test Accuracy	Remarks
SVM	89.74%	40.25%	Selected model
KNN	75%	17.5%	Low accuracy on test data
Random Forest	100%	100%	Overfitting with small dataset

6. Discussion

SVM was chosen for its ability to handle high-dimensional data and its performance with smaller datasets. It effectively creates a hyperplane to separate classes, making it robust for stop sign detection, where the dataset is limited but features are distinctive.

Criteria	SVM (Support Vector Machine)	Random Forest	KNN (K-Nearest Neighbors)
Data Size Suitability	Works well with small datasets	Performs better with larger datasets	Requires large datasets for accuracy
Handling High Dimensions	Excellent (handles HOG features well)	Moderate (needs dimensionality reduction)	Poor (affected by the curse of dimensionality)
Training Complexity	Moderate (depends on kernel)	High (many decision trees to train)	Low (stores training data, no real training)
Prediction Speed	Fast once trained	Fast (parallel processing of trees)	Slow (computes distance during inference)
Accuracy Potential	High with proper tuning	Good but prone to overfitting on small data	Low for complex, high-dimensional tasks
Noise Sensitivity	Low (regularization available)	Low (ensemble learning reduces noise impact)	High (affected by outliers and noise)
Interpretability	Moderate (clear decision boundary)	Low (ensemble of trees)	High (simple, intuitive)
Overfitting Risk	Low (with proper regularization)	Moderate (mitigated with more data)	High (especially with small datasets)
Suitability for Images	Good for image data (especially with feature vectors)	Needs pre-processing (like PCA) for images	Poor performance on image data

Challenges:

- **Dataset Size:** Small datasets limit generalization.
- **Variability in Images:** Different angles and lighting conditions increase complexity.

Improvements:

- **Data Augmentation:** Additional images or synthetic data could improve performance.
- **Hyperparameter Tuning:** Further optimization of SVM parameters.

Conclusion

In this project, you'll see how SVM, KNN, and Random Forest models were applied to stop sign detection. SVM was chosen for this analysis despite its limitations, as it offers a good balance between performance and the ability to work well with small datasets. Upcoming works will tackle the growth of the dataset and the testing of powerful models, with the goal of gaining better accuracy.