

```
In [6]: from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

In [7]: import os
import cv2 #using OpenCV for this task
import numpy as np
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from skimage.feature import hog # histogram of oriented gradients
import matplotlib.pyplot as plt

# Function to extract HOG features from an image and visualize them
def extract_features(image_path, visualize=False):
    image = cv2.imread(image_path) #reading the image using OpenCV
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) #converting to greyscale to simplify and reduce number of channels

    # Compute HOG features
    features, hog_image = hog(gray, orientations=9, pixels_per_cell=(8, 8),
                             cells_per_block=(2, 2), visualize=True, transform_sqrt=True) #hog= Histogram of Gradients

    # Ensure all feature vectors have the same length (e.g., by padding or resizing)
    max_length = 5000 # Example: Set a maximum Length for feature vectors
    if len(features) < max_length:
        features = np.pad(features, (0, max_length - len(features)), mode='constant')
    elif len(features) > max_length:
        features = features[:max_length] # Trim if too long

    if visualize:
        # Visualize HOG features
        plt.figure(figsize=(8, 4))
        plt.subplot(121).set_title('Image')
        plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
        plt.axis('off')
        plt.subplot(122).set_title('HOG Features')
        plt.imshow(hog_image, cmap='gray')
        plt.axis('off')
        plt.show()

    return features

# Function to Load dataset
def load_dataset(dataset_path):
    images = []
    labels = []
    image_files = [] # To store image file paths for result output

    for image_file in os.listdir(dataset_path): #iterating over images in dataset
        if image_file.endswith('.jpg') or image_file.endswith('.png'):
            image_path = os.path.join(dataset_path, image_file)
            xml_path = os.path.join(dataset_path, image_file.replace('.jpg', '.xml').replace('.png', '.xml'))

            if os.path.exists(xml_path) and os.stat(xml_path).st_size > 0:

                labels.append(1) #assign 1 to images with license plate
            else:

                labels.append(0) #assign 0 to images without license plate

        # Extract features for every image (with or without license plate)
        features = extract_features(image_path)
        images.append(features)
        image_files.append(image_file) # Store image file path

    # Convert Lists to numpy arrays
    images = np.array(images)
    labels = np.array(labels)

    return images, labels, image_files

#implementing the code:
dataset_path = '/content/drive/MyDrive/BaselineModel/' #google drive Link to dataset
images, labels, image_files = load_dataset(dataset_path)

#now, splitting the data into training,validation and testing sets. followed the 60= training, 20= validation, and 20= testi
X_train, X_test, y_train, y_test, train_files, test_files = train_test_split(images, labels, image_files, test_size=0.2, ran
svm_classifier = SVC(kernel='linear') #using the Linear kernel and initializaing the classifier
```

```

# Train SVM
svm_classifier.fit(X_train, y_train)

# Predict on test set and visualize results
y_pred = svm_classifier.predict(X_test)

#evaluating the model accuracy by comparing the pred vs test coordinate
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")

# for feature extraction visualization
print("\nResults:")
for i in range(len(y_pred)):
    result = "License Plate Detected" if y_pred[i] == 1 else "No License Plate Detected"
    print(f"{test_files[i]}: {result}")
    # Visualize HOG features for the current image
    extract_features(os.path.join(dataset_path, test_files[i]), visualize=True)

```

Accuracy: 0.9662921348314607

Results:

Cars417.png: License Plate Detected

Image



HOG Features



Cars212.png: License Plate Detected

Image



HOG Features



Cars150.png: License Plate Detected

Image



HOG Features



Cars50.png: License Plate Detected



Cars148.png: License Plate Detected



Cars350.png: License Plate Detected



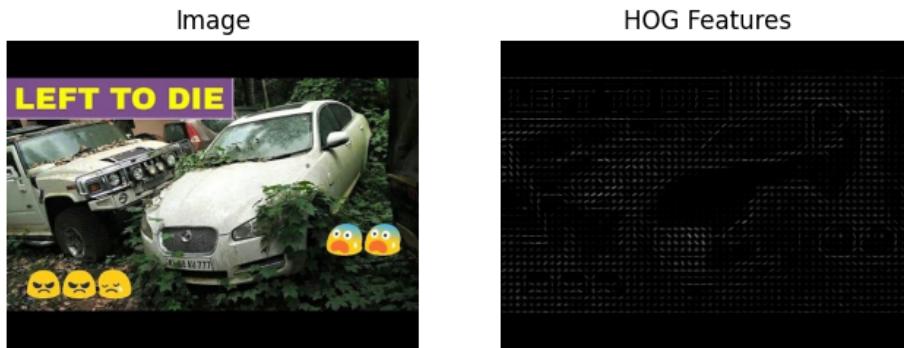
Cars84.png: License Plate Detected



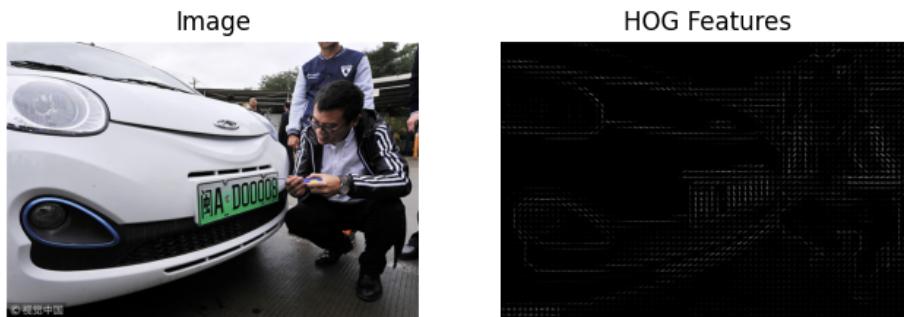
Cars279.png: License Plate Detected



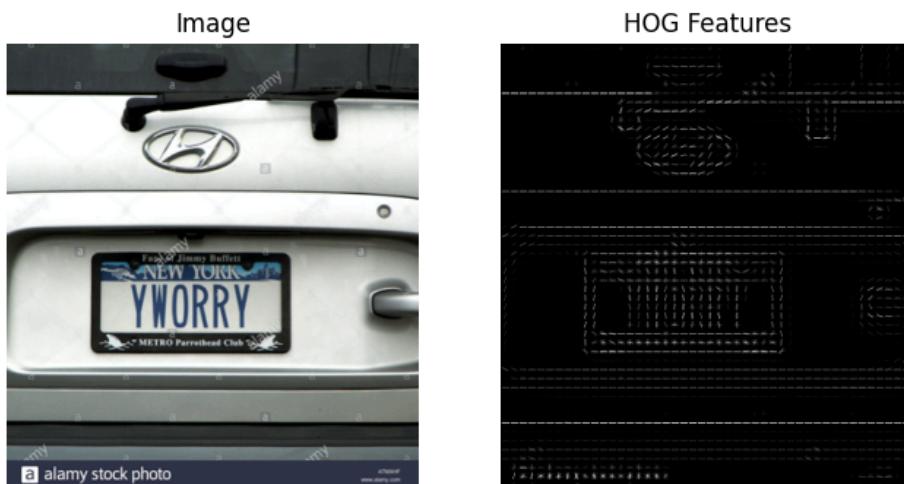
Cars224.png: License Plate Detected



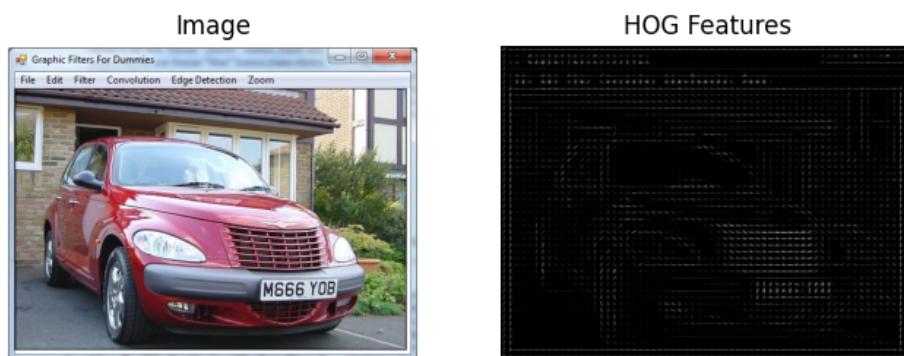
Cars92.png: License Plate Detected



Cars98.png: License Plate Detected



Cars161.png: License Plate Detected



Cars83.png: License Plate Detected



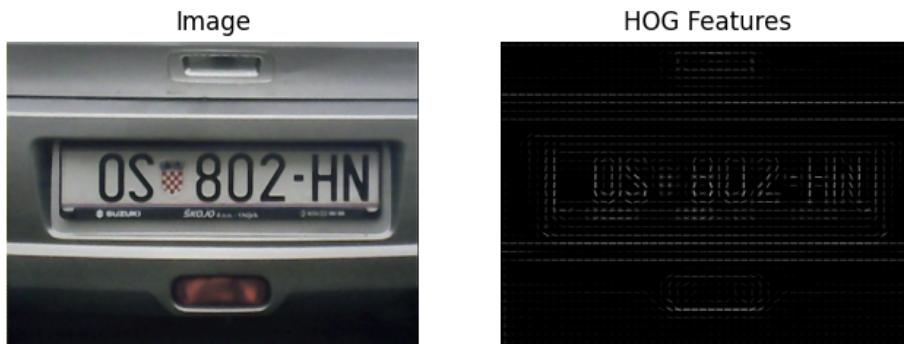
Cars431.png: License Plate Detected



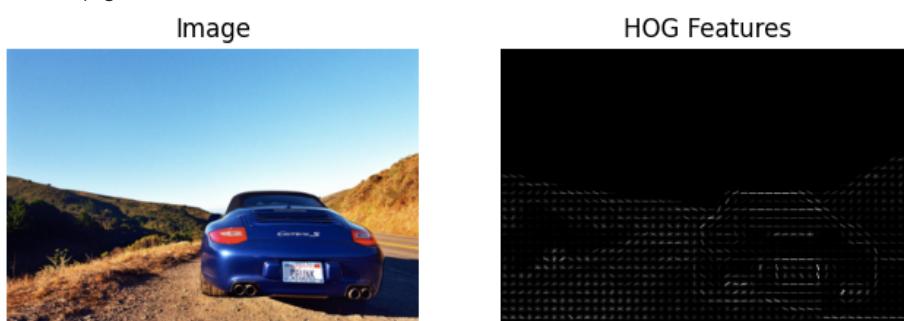
Cars87.png: License Plate Detected



Cars60.png: License Plate Detected



Cars414.png: License Plate Detected



Cars216.png: License Plate Detected



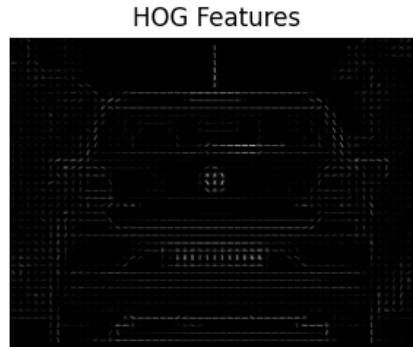
Cars149.png: License Plate Detected



Cars72.png: License Plate Detected



Cars108.png: License Plate Detected



Cars31.png: License Plate Detected



Cars165.png: License Plate Detected



Cars79.png: License Plate Detected



Cars144.png: License Plate Detected



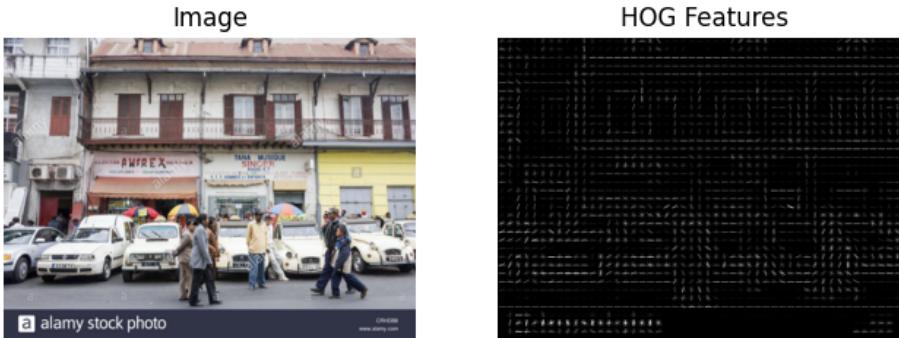
Cars336.png: License Plate Detected



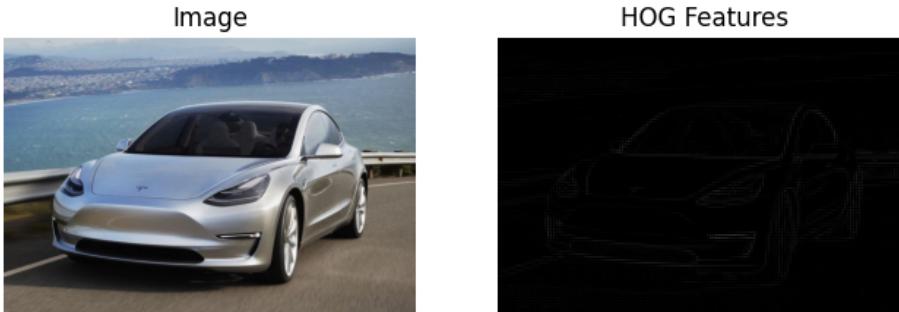
Cars401.png: License Plate Detected



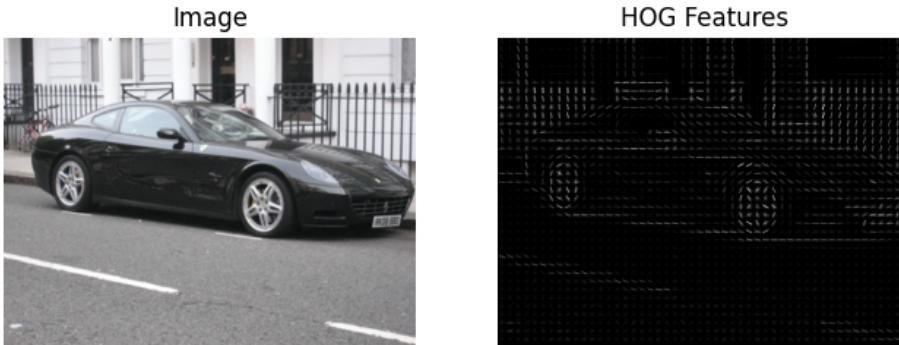
Cars277.png: License Plate Detected



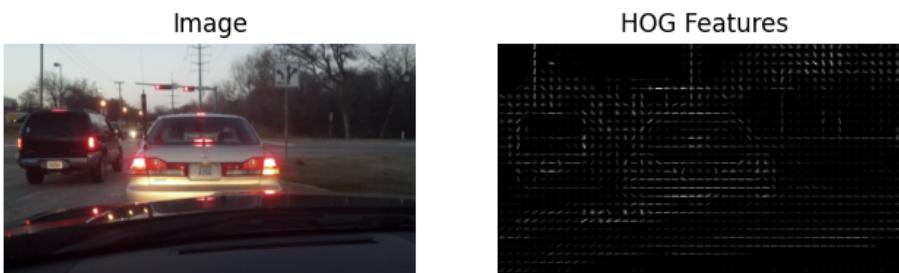
yrv39qi2rr141.jpg: License Plate Detected



Cars209.png: License Plate Detected



Cars70.png: License Plate Detected



Cars157.png: License Plate Detected



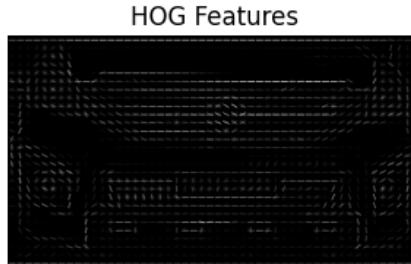
Cars142.png: License Plate Detected



Cars206.png: License Plate Detected



Cars373.png: License Plate Detected



Cars370.png: License Plate Detected



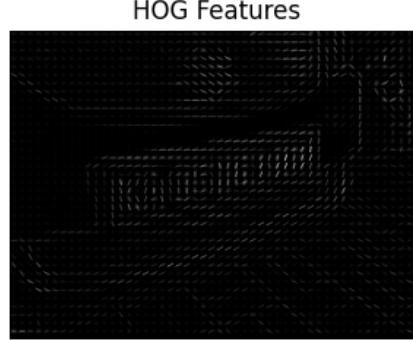
Cars146.png: License Plate Detected



Cars430.png: License Plate Detected



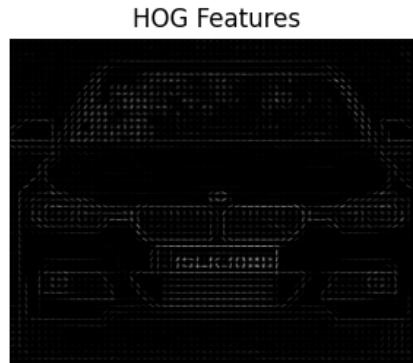
Cars97.png: License Plate Detected



Cars369.png: License Plate Detected



Cars167.png: License Plate Detected



Cars255.png: License Plate Detected



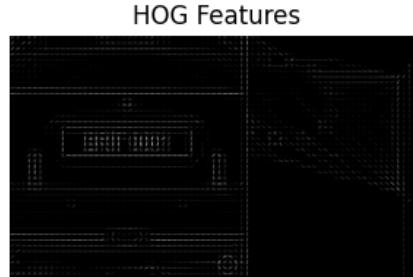
Cars353.png: License Plate Detected



Cars24.png: License Plate Detected



Cars273.png: License Plate Detected



Cars179.png: License Plate Detected



Cars16.png: License Plate Detected



Cars125.png: License Plate Detected



Cars365.png: License Plate Detected



Cars397.png: License Plate Detected



Cars327.png: License Plate Detected



Cars215.png: License Plate Detected



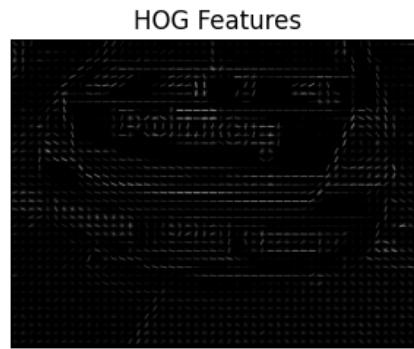
Cars106.png: License Plate Detected



Cars55.png: License Plate Detected



Cars118.png: License Plate Detected



Cars88.png: License Plate Detected



Cars219.png: License Plate Detected





w8bzulpiech91.jpg: License Plate Detected



Cars109.png: License Plate Detected



Cars356.png: License Plate Detected



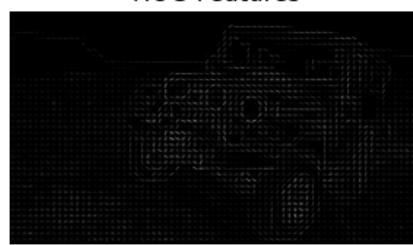
Cars96.png: License Plate Detected



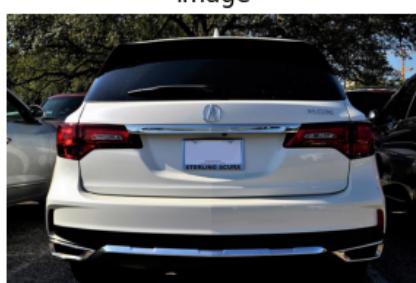
Cars162.png: License Plate Detected



Cars427.png: License Plate Detected



white-acura-mdx.jpg: License Plate Detected

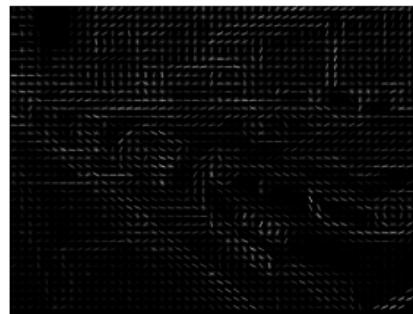


Cars299.png: License Plate Detected



Cars384.png: License Plate Detected





Cars222.png: License Plate Detected



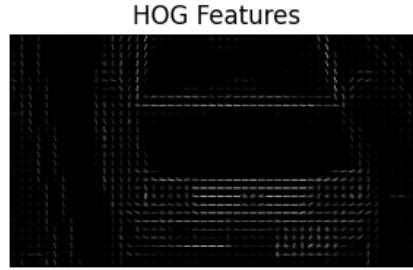
Cars428.png: License Plate Detected



Cars5.png: License Plate Detected



Cars135.png: License Plate Detected



Cars29.png: License Plate Detected



Cars287.png: License Plate Detected



Cars34.png: License Plate Detected



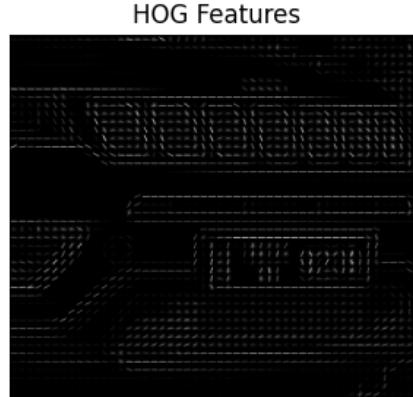
Cars127.png: License Plate Detected



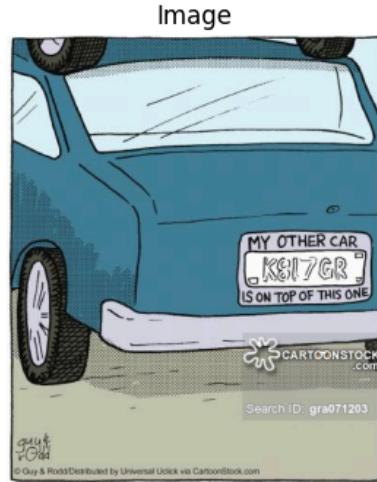
Cars263.png: License Plate Detected



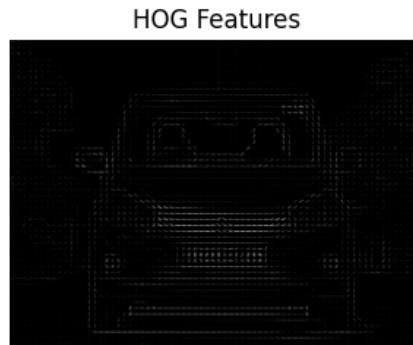
Cars46.png: License Plate Detected



Cars207.png: License Plate Detected



Cars372.png: License Plate Detected



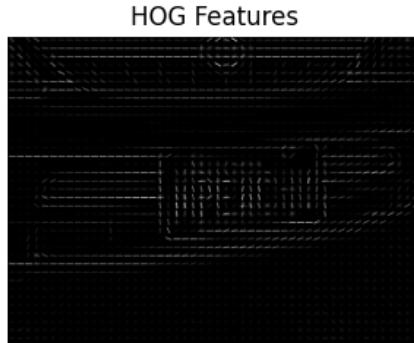
Cars307.png: License Plate Detected



Cars359.png: License Plate Detected



Cars113.png: License Plate Detected



Cars134.png: License Plate Detected



Cars189.png: License Plate Detected



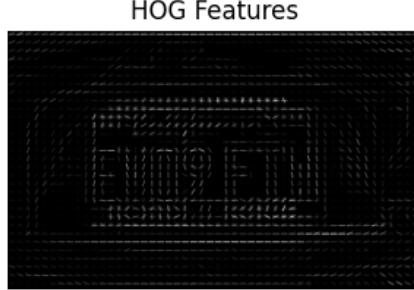
Cars416.png: License Plate Detected



Cars177.png: License Plate Detected



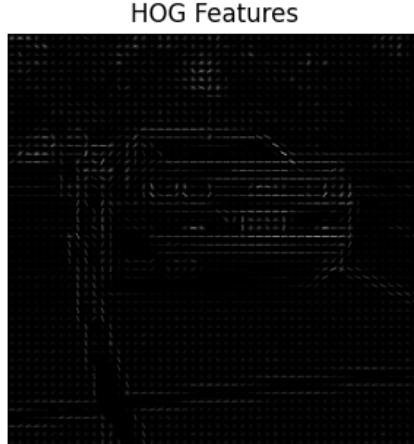
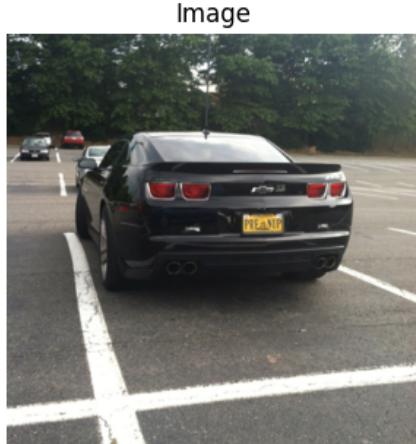
Cars176.png: License Plate Detected



Cars15.png: License Plate Detected



Cars242.png: License Plate Detected



Cars102.png: License Plate Detected

Image



HOG Features

