

```

In [ ]: import os
import cv2
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
import matplotlib.pyplot as plt
from skimage import exposure

def extract_features(image_path, visualize=False):# Function to extract HOG features from an image and visualize them
    image = cv2.imread(image_path) # Read the image using OpenCV
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # Convert to grayscale

    features, hog_image = hog(gray, orientations=8, pixels_per_cell=(16, 16),
                              cells_per_block=(2, 2), visualize=True, transform_sqrt=True)# computing hog features

    hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10)) #increased exposure

    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] #this happens to trim

    if visualize:

        plt.figure(figsize=(15, 5))
        plt.subplot(131).set_title('Original Image')
        plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
        plt.axis('off')
        plt.subplot(132).set_title('HOG Features')
        plt.imshow(hog_image_rescaled, cmap='gray')
        plt.axis('off')

        height, width = gray.shape
        plate_region = gray[int(height * 0.6):int(height * 0.9), int(width * 0.1):int(width * 0.9)]
        plt.subplot(133).set_title('Zoomed-in Plate Region') #returning zoom-in picture of License plate
        plt.imshow(plate_region, cmap='gray')
        plt.axis('off')
        plt.show()

    return features

def load_dataset(dataset_path):#Loading dataset
    images = []
    labels = []
    image_files = []

    for folder_name in os.listdir(dataset_path):
        folder_path = os.path.join(dataset_path, folder_name)
        if os.path.isdir(folder_path):
            for image_file in os.listdir(folder_path):
                if image_file.endswith('.jpg') or image_file.endswith('.png'):
                    image_path = os.path.join(folder_path, image_file)

                    features = extract_features(image_path) #extracting features
                    images.append(features)
                    labels.append(folder_name) # Use folder name as Label
                    image_files.append(image_path) # Store image file path

    images = np.array(images) #converting to numpy array
    labels = np.array(labels) #converting to numpy array

    return images, labels, image_files

dataset_path = '/content/drive/MyDrive/worked_cars/'
images, labels, image_files = load_dataset(dataset_path)

from sklearn.preprocessing import LabelEncoder #using sklearn
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(labels)

X_train, X_test, y_train, y_test, train_files, test_files = train_test_split(images, y, image_files, test_size=0.2, random_state=42) #splitting train,test,

svm_classifier = SVC(kernel='linear') #initializing a SVM classifier

svm_classifier.fit(X_train, y_train)

y_pred = svm_classifier.predict(X_test)
y_pred_labels = label_encoder.inverse_transform(y_pred) # convert back to original Labels

# Evaluate the model accuracy

```

```

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

print("\nResults:")
for i in range(len(y_pred)):
    result = "License Plate Detected: " + y_pred_labels[i]
    print(f"{test_files[i]}: {result}")

    extract_features(test_files[i], visualize=True) #visualizing HOG feature

```

Accuracy: 0.00

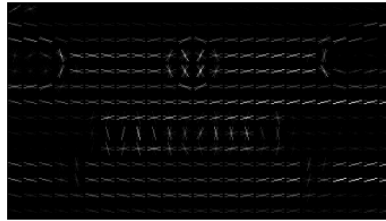
Results:

/content/drive/MyDrive/worked\_cars/ALR486/Cars187.png: License Plate Detected: IT20B0M

Original Image



HOG Features



Zoomed-in Plate Region

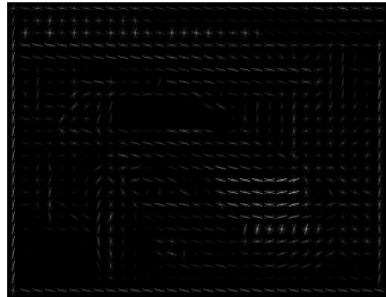


/content/drive/MyDrive/worked\_cars/M666Y0B/Cars115.png: License Plate Detected: LR33TEE

Original Image



HOG Features



Zoomed-in Plate Region

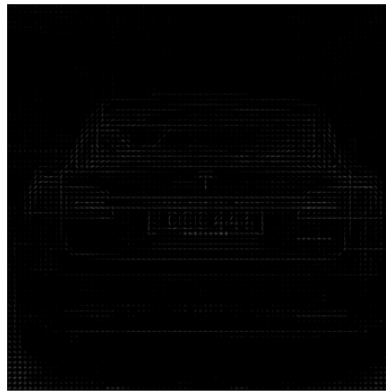


/content/drive/MyDrive/worked\_cars/CCC444/Copy of dataset-card.jpg: License Plate Detected: B58BP5

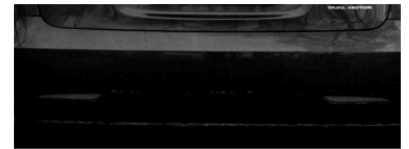
Original Image



HOG Features



Zoomed-in Plate Region



/content/drive/MyDrive/worked\_cars/AB44887/Cars386.png: License Plate Detected: GT

Original Image



HOG Features



Zoomed-in Plate Region

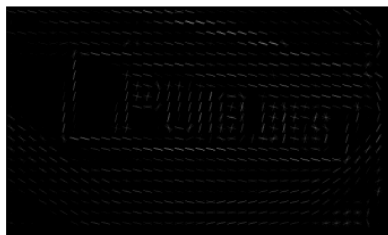


/content/drive/MyDrive/worked\_cars/PUI8BES/Cars50.png: License Plate Detected: MH20EE7598

Original Image



HOG Features



Zoomed-in Plate Region

