CourseName: Computer Vision Lab **Course Code:** CSP-422

Experiment:1.1

Aim: Write a program to implement various feature extraction techniques for image classification.

Software Required: Any Python IDE

Description:

The experiment involves implementing different feature extraction techniques for image classification using Python and relevant libraries. The steps involved in this experiment are as follows:

1. Import necessary libraries scikit-learn.
2. Load the dataset of labeled images for training and testing.
3. Preprocess the images by resizing, normalizing, or applying any necessary transformations.
4. Extract features from the images using various techniques such as:
    a. Histogram of Oriented Gradients (HOG)
    b. Scale-Invariant Feature Transform (SIFT)
    c. Speeded-Up Robust Features (SURF), etc.
5. Split the dataset into training and testing sets.
6. Train a classifier using the extracted features and the corresponding labels.
7. Evaluate the performance of the classifier on the testing set by calculating metrics like accuracy, precision, recall, and F1-score.
8. Compare the performance of different feature extraction techniques by analyzing the evaluation results.
9. Repeat steps 4-8 for different combinations of feature extraction techniques and classifiers to explore the impact on classification performance.
10. Document the observations and conclusions drawn from the experiment.

Steps:

1. Import necessary libraries
2. Load the dataset
3. Preprocess the images
4. Extract features using a specific technique
5. Split the dataset into training and testing sets
6. Train a classifier using the extracted features and labels
7. Evaluate the performance of the classifier on the testing set
8. Compare the performance of different techniques

CourseName: Computer Vision Lab **Course Code:** CSP-422

9. Repeat steps 4-8 for other techniques
10. Document the observations and conclusions

Implementation/Output:

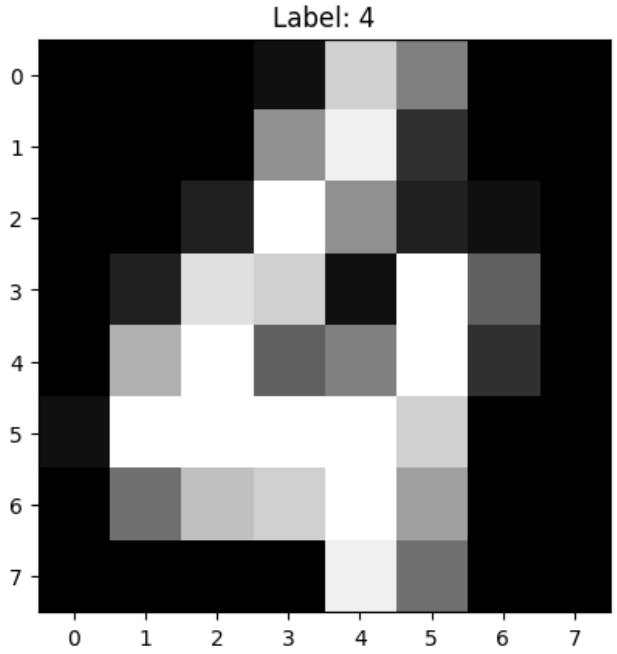
In [ ]:
```
pip install scikit-learn
```

In [82]:
```
import numpy as np
import matplotlib.pyplot as plt
from skimage.feature import hog
from skimage import exposure
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
```

In [83]:
```
# Load the digits dataset
digits = load_digits()

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target, test_size=0.7, random_state=42)
```

In [84]:
```
# Visualize the first digit in the dataset
plt.imshow(X_train[12].reshape(8, 8), cmap=plt.cm.gray)
plt.title(f"Label: {y_train[12]}")
plt.show()
```



CourseName: Computer Vision Lab **Course Code:** CSP-422

In [85]:
```
# Define HOG parameters
orientations = 8
pixels_per_cell = (4, 4)
cells_per_block = (2, 2)

# Extract HOG features for training and testing data
X_train_hog = []
for image in X_train:
    fd, hog_image = hog(image.reshape((8, 8)), orientations=orientations,
                        pixels_per_cell=pixels_per_cell, cells_per_block=cells_per_block,
                        block_norm='L2-Hys', visualize=True)
    X_train_hog.append(fd)
X_train_hog = np.array(X_train_hog)
```

In [86]:
```
# Train a Support Vector Machine classifier
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train_hog, y_train)
```

Out[86]:
```
SVC(kernel='linear')
```

In [87]:
```
# Make predictions on the test data
y_pred = svm_classifier.predict(X_test_hog)
```

In [88]:
```
# Calculate and print accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy*100:.3f}")
```

```
Accuracy: 84.340
```

In [92]:
```
Precision = precision_score(y_test, y_pred,average='micro')
print(f"Precision: {Precision*100:.3f}")
```

```
Precision: 84.340
```