In [1]:

```
pip install scikit-image
```

```
Requirement already satisfied: scikit-image in /Users/jeevanreddy
ramireddy/opt/anaconda3/lib/python3.9/site-packages (0.19.2)
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,!=8.3.0,>=6.
1.0 in /Users/jeevanreddyramireddy/opt/anaconda3/lib/python3.9/si
te-packages (from scikit-image) (9.2.0)
Requirement already satisfied: scipy>=1.4.1 in /Users/jeevanreddy
ramireddy/opt/anaconda3/lib/python3.9/site-packages (from scikit-
image) (1.9.1)
Requirement already satisfied: packaging>=20.0 in /Users/jeevanre
ddyramireddy/opt/anaconda3/lib/python3.9/site-packages (from scik
it-image) (21.3)
Requirement already satisfied: imageio>=2.4.1 in /Users/jeevanred
dyramireddy/opt/anaconda3/lib/python3.9/site-packages (from sciki
t-image) (2.19.3)
Requirement already satisfied: PyWavelets>=1.1.1 in /Users/jeevan
reddyramireddy/opt/anaconda3/lib/python3.9/site-packages (from sc
ikit-image) (1.3.0)
Requirement already satisfied: tifffile>=2019.7.26 in /Users/jeev
anreddyramireddy/opt/anaconda3/lib/python3.9/site-packages (from
scikit-image) (2021.7.2)
Requirement already satisfied: networkx>=2.2 in /Users/jeevanredd
yramireddy/opt/anaconda3/lib/python3.9/site-packages (from scikit
-image) (2.8.4)
Requirement already satisfied: numpy>=1.17.0 in /Users/jeevanredd
yramireddy/opt/anaconda3/lib/python3.9/site-packages (from scikit
-image) (1.21.5)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /User
s/jeevanreddyramireddy/opt/anaconda3/lib/python3.9/site-packages
(from packaging>=20.0->scikit-image) (3.0.9)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
import cv2
import numpy as np
from skimage import transform
```

In [3]:

```
def spatial_pyramid_pooling(img, levels):
    height, width, _ = img.shape
    pooled_features = []
    for level in levels:
        grid_size = (2 ** level, 2 ** level)
        cell_size = (height // grid_size[0], width // grid_size[1]
        for i in range(grid_size[0]):
            for j in range(grid_size[1]):
                cell = img[i * cell_size[0]:(i + 1) * cell_size[0]
                pooled_features.append(np.mean(cell, axis=(0, 1)))
    return np.array(pooled_features)
```

In [4]:

```
import os
import cv2
import numpy as np

def load_dataset(dataset_path):
    images = []
    labels = []
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if file.endswith(".png") or file.endswith(".jpg"):
                image_path = os.path.join(root, file)
                label = os.path.basename(root)  # Assuming each su
                # Read and preprocess the image
                image = cv2.imread(image_path)
                image = cv2.resize(image, (32, 32))  # Resize imag
                # You can perform additional preprocessing steps h
                images.append(image)
                labels.append(label)
    return images, labels
```

In [5]:

```
dataset_path = 'datSet'
images, labels = load_dataset(dataset_path)
```

```
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
```

In [6]:

```
images = np.array(images)
```

In [7]:

```
levels = [1, 2, 4]  # Define the levels of the pyramid (e.g., 1x1,
pooled_features = [spatial_pyramid_pooling(img, levels) for img in
```

In [8]:
```
data = np.array(pooled_features)
```

In [9]:
```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

In [10]:
```
X_train, X_test, y_train, y_test = train_test_split(data, labels,
```

In [11]:
```
svm_classifier = SVC()
```

In [12]:
```
# Flatten the images before applying SPP
X_train_flattened = X_train.reshape(X_train.shape[0], -1)

# Train the SVM classifier
svm_classifier.fit(X_train_flattened, y_train)
```

Out[12]:
```
SVC()
```

In [13]:
```
X_test_flattened = X_test.reshape(X_test.shape[0], -1)

y_pred = svm_classifier.predict(X_test_flattened)
```

In [14]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.5253807106598984
```

In [15]:
```
import cv2
import numpy as np
```

In [16]:
```
new_image_path = 'pecora.jpeg'
new_image = cv2.imread(new_image_path)
```

In [17]:

```
preprocessed_image = cv2.resize(new_image, (32, 32))  # Resize to

# Apply Spatial Pyramid Pooling to the preprocessed image
levels = [1, 2, 4]  # Define the levels of the pyramid (e.g., 1x1,
pooled_features = spatial_pyramid_pooling(preprocessed_image, leve

# Reshape the feature vector to match the format expected by the m
pooled_features = pooled_features.reshape(1, -1)

# Predict the label for the new image
predicted_label = svm_classifier.predict(pooled_features)
```

In [ ]:

```
print("Predicted label:", predicted_label)
#cv2.imshow("New Image", new_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
Predicted label: ['pecora']
```

In [ ]:

In [ ]: