In [50]:

```
import cv2
from skimage.feature import hog
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import numpy as np
```

In [158]:

```
import os
import cv2
import numpy as np

def load_dataset(datSet):
    images = []
    labels = []
    for root, dirs, files in os.walk(datSet):
        for file in files:
            if file.endswith(".jpg") or file.endswith(".png"):
                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

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 [159]:

```
import numpy as np
from skimage.feature import hog
from skimage.color import rgb2gray

# Convert the images to grayscale before computing HOG features
grayscale_images = [rgb2gray(img) for img in images]

# Compute HOG features for each grayscale image
data = np.array([hog(img, orientations=8, pixels_per_cell=(16, 16)
labels = np.array(labels)
```

In [160]:

```
X_train, X_test, y_train, y_test = train_test_split(data, labels,
```

In [161]:

```
svm = SVC()
svm.fit(X_train, y_train)
```

Out[161]:

```
SVC()
```

In [162]:

```
y_pred = svm.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.2893401015228426
```

Testing

In [179]:

```
import cv2

# Load the new image
new_image = cv2.imread('ragno1.jpg')
```

In [180]:

```
# Resize the image (if needed)
resized_image = cv2.resize(new_image, (32, 32))  # Specify width a

# Convert the image to grayscale
gray_image = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
```

In [181]:

```
from skimage.feature import hog

# Extract HOG features for the new image
hog_features = hog(gray_image, orientations=8, pixels_per_cell=(16
```

In [182]:

```
# Reshape the feature vector
hog_features = hog_features.reshape(1, -1)
```

OM SAI RAM

In [183]:

```
# Predict the label for the new image
predicted_label = svm.predict(hog_features)
predicted_label
```

Out[183]:

```
array(['pecora'], dtype='<U8')
```

In [ ]:

```

```