In [1]:
```
# import libraires
import numpy as np
import cv2
import os
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

In [2]:
```
# Load images from folder
def load_images_from_folder(folder, label, size=(64, 64)):
    images = []
    labels = []
    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)
        img = cv2.imread(img_path)
        if img is not None:
            img = cv2.resize(img, size)
            img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            img = img.flatten()
            images.append(img)
            labels.append(label)
    return images, labels
```

In [3]:
```
# Load Dog images
dog_folder ='D:\Datasets\PetImages\Dog'
dog_images, dog_labels = load_images_from_folder(dog_folder, 0)
```

In [4]:
```
# Load Cat Images
cat_folder = 'D:\Datasets\PetImages\Cat'
cat_images, cat_labels = load_images_from_folder(cat_folder, 1)
```

In [5]:
```
# Split X and Y
X = np.array(dog_images + cat_images)
y = np.array(dog_labels + cat_labels)
```

In [38]:
```
# Split Into test and train
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, rando
```

In [39]:
```
# fit the model to SVM
clf = svm.SVC(kernel='linear')
clf.fit(X_train, y_train)
```

Out[39]:
```
SVC
SVC(kernel='linear')
```

In [40]:
```
# predict
y_pred = clf.predict(X_test)
```

In [41]:
```
# Check the Accuracy level and Classification Report
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
```

In [42]:
```
print("Accuracy:", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy: 0.5231788079470199
Classification Report:
               precision    recall  f1-score   support

           0       0.61      0.57      0.59        91
           1       0.41      0.45      0.43        60

    accuracy                           0.52       151
   macro avg       0.51      0.51      0.51       151
weighted avg       0.53      0.52      0.53       151
```

In [13]:
```
# Visualize the Classify
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

In [14]:
```
pca = PCA(n_components=2)
X_reduced = pca.fit_transform(X)
```

In [48]:
```
plt.scatter(X_reduced[y==0, 0], X_reduced[y==0, 1], label='Dog', alpha=0.6, c=
plt.scatter(X_reduced[y==1, 0], X_reduced[y==1, 1], label='Cat', alpha=0.6, c=
plt.title(' Dog and Cat Images')
plt.xlabel('Component 1')
plt.ylabel('Component 2')
plt.legend()
plt.show()
```