```
! gdown https://drive.google.com/uc?id=137RyRjvTBkBiIfeYBNZBtViDHQ6_Ewsp --output 101_ObjectC
! tar -xvf 101_ObjectCategories.tar.gz
! mv 101_ObjectCategories caltech101
! rm -rf caltech101/BACKGROUND_Google
```

```
    101_ObjectCategories/crayfish/image_0068.jpg
    101_ObjectCategories/crayfish/image_0070.jpg
    101_ObjectCategories/crayfish/image_0001.jpg
    101_ObjectCategories/crayfish/image_0002.jpg
    101_ObjectCategories/crayfish/image_0003.jpg
    101_ObjectCategories/crayfish/image_0009.jpg
    101_ObjectCategories/crayfish/image_0015.jpg
    101_ObjectCategories/crayfish/image_0021.jpg
    101_ObjectCategories/crayfish/image_0027.jpg
    101_ObjectCategories/crayfish/image_0033.jpg
    101_ObjectCategories/crayfish/image_0039.jpg
    101_ObjectCategories/crayfish/image_0045.jpg
    101_ObjectCategories/crayfish/image_0051.jpg
    101_ObjectCategories/crayfish/image_0057.jpg
    101_ObjectCategories/crayfish/image_0063.jpg
    101_ObjectCategories/crayfish/image_0069.jpg
    101_ObjectCategories/crocodile/
    101_ObjectCategories/crocodile/image_0004.jpg
    101_ObjectCategories/crocodile/image_0005.jpg
    101_ObjectCategories/crocodile/image_0006.jpg
    101_ObjectCategories/crocodile/image_0007.jpg
    101_ObjectCategories/crocodile/image_0008.jpg
    101_ObjectCategories/crocodile/image_0010.jpg
    101_ObjectCategories/crocodile/image_0011.jpg
    101_ObjectCategories/crocodile/image_0012.jpg
    101_ObjectCategories/crocodile/image_0013.jpg
    101_ObjectCategories/crocodile/image_0014.jpg
    101_ObjectCategories/crocodile/image_0016.jpg
    101_ObjectCategories/crocodile/image_0017.jpg
    101_ObjectCategories/crocodile/image_0018.jpg
    101_ObjectCategories/crocodile/image_0019.jpg
    101_ObjectCategories/crocodile/image_0020.jpg
    101_ObjectCategories/crocodile/image_0022.jpg
    101_ObjectCategories/crocodile/image_0023.jpg
    101_ObjectCategories/crocodile/image_0024.jpg
    101_ObjectCategories/crocodile/image_0025.jpg
    101_ObjectCategories/crocodile/image_0026.jpg
    101_ObjectCategories/crocodile/image_0028.jpg
    101_ObjectCategories/crocodile/image_0029.jpg
    101_ObjectCategories/crocodile/image_0030.jpg
    101_ObjectCategories/crocodile/image_0031.jpg
    101_ObjectCategories/crocodile/image_0032.jpg
    101_ObjectCategories/crocodile/image_0034.jpg
    101_ObjectCategories/crocodile/image_0035.jpg
    101_ObjectCategories/crocodile/image_0036.jpg
    101_ObjectCategories/crocodile/image_0037.jpg
    101_ObjectCategories/crocodile/image_0038.jpg
    101_ObjectCategories/crocodile/image_0040.jpg
    101_ObjectCategories/crocodile/image_0041.jpg
    101_ObjectCategories/crocodile/image_0042.jpg
```

```
101_ObjectCategories/crocodile/image_0043.jpg
101_ObjectCategories/crocodile/image_0044.jpg
101_ObjectCategories/crocodile/image_0046.jpg
101_ObjectCategories/crocodile/image_0047.jpg
101_ObjectCategories/crocodile/image_0048.jpg
101_ObjectCategories/crocodile/image_0049.jpg
101_ObjectCategories/crocodile/image_0050.jpg
101_ObjectCategories/crocodile/image_0001.jpg
101_ObjectCategories/crocodile/image_0002.jpg
```

```
import numpy as np
from numpy.linalg import norm
import pickle
from tqdm import tqdm, tqdm_notebook
import os
import time
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input


model = ResNet50(weights='imagenet', include_top=False,
                 input_shape=(224, 224, 3),pooling='max')
def extract_features(img_path, model):
    input_shape = (224, 224, 3)
    img = image.load_img(img_path, target_size=(
        input_shape[0], input_shape[1]))
    img_array = image.img_to_array(img)
    expanded_img_array = np.expand_dims(img_array, axis=0)
    preprocessed_img = preprocess_input(expanded_img_array)
    features = model.predict(preprocessed_img)
    flattened_features = features.flatten()
    normalized_features = flattened_features / norm(flattened_features)
    return normalized_features
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resne
94773248/94765736 [==============================] - 1s 0us/step
94781440/94765736 [==============================] - 1s 0us/step
```

```
features = extract_features('/content/caltech101/Faces/image_0002.jpg', model)
print(len(features))
```

```
2048
```

```
extensions = ['.jpg', '.JPG', '.jpeg', '.JPEG', '.png', '.PNG']
def get_file_list(root_dir):
    file_list = []
    counter = 1
    for root, directories, filenames in os.walk(root_dir):
        for filename in filenames:
            if any(ext in filename for ext in extensions):
                file_list.append(os.path.join(root, filename))
                counter += 1
    return file_list
```

```
# path to the datasets
root_dir = '/content/caltech101'
filenames = sorted(get_file_list(root_dir))
```

```
feature_list = []
for i in tqdm_notebook(range(len(filenames))):
    feature_list.append(extract_features(filenames[i], model))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: TqdmDeprecationWarning:
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
```

```
100%                                        8677/8677 [09:36<00:00, 14.41it/s]
```

```
pickle.dump(feature_list, open('features-caltech101-resnet.pickle', 'wb'))
pickle.dump(filenames, open('filenames-caltech101.pickle','wb'))
```

```
filenames = pickle.load(open('filenames-caltech101.pickle', 'rb'))
feature_list = pickle.load(open('features-caltech101-resnet.pickle', 'rb'))
```

```
from sklearn.neighbors import NearestNeighbors
neighbors = NearestNeighbors(n_neighbors=5, algorithm='brute',
metric='euclidean').fit(feature_list)
distances, indices = neighbors.kneighbors([feature_list[0]])
```

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline
plt.imshow(mpimg.imread(filenames[0]))
```

```
<matplotlib.image.AxesImage at 0x7f142a0596d0>
```

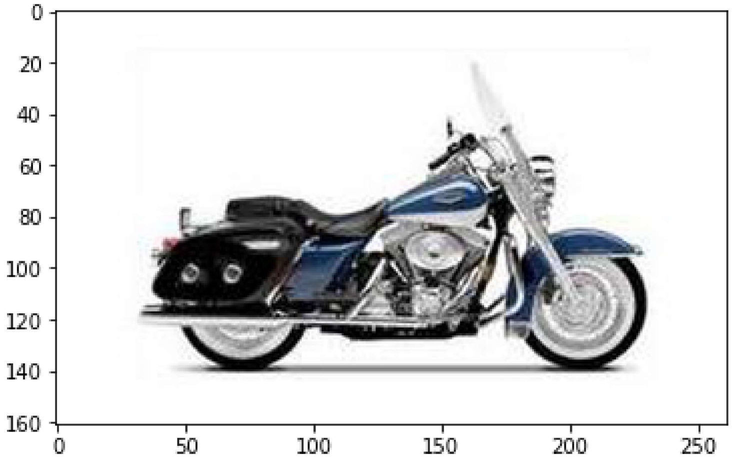
```
for i in range(5):
    print(distances[0][i])
```

```
4.2146848e-08
0.6032694
0.6071576
0.6297979
0.63568777
```

```
import random
```

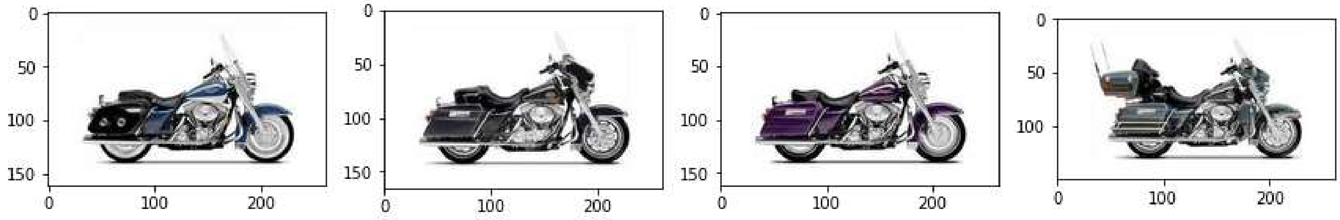
```
def similar_images(paths):
    plt.figure(figsize=(15,10), facecolor='white')
    plotnumber = 1
    for index in range(len(paths)):
        if plotnumber<=len(paths) :
            ax = plt.subplot(2,5,plotnumber)
            plt.imshow(mpimg.imread(paths[index]), interpolation='lanczos')
            plotnumber+=1
    plt.tight_layout()


def org_image(querry):
    plt.imshow(mpimg.imread(querry), interpolation='lanczos')
    plt.xlabel(querry.split('.')[0] + '_Original Image',fontsize=20)
    plt.show()


lenimg = len(filenames)


for i in range(6):
    random_image_index = random.randint(0,lenimg)
    distances, indices = neighbors.kneighbors([feature_list[random_image_index]])
    # don't take the first closest image as it will be the same image
    similar_image_paths = [filenames[random_image_index]] +[filenames[indices[0][i]] for i in
    print("******Original Image***********")
    org_image(filenames[random_image_index])
    print('********* Predictions ***********')
    similar_images(similar_image_paths)
    plt.show()
```

```
******Original Image***********
```

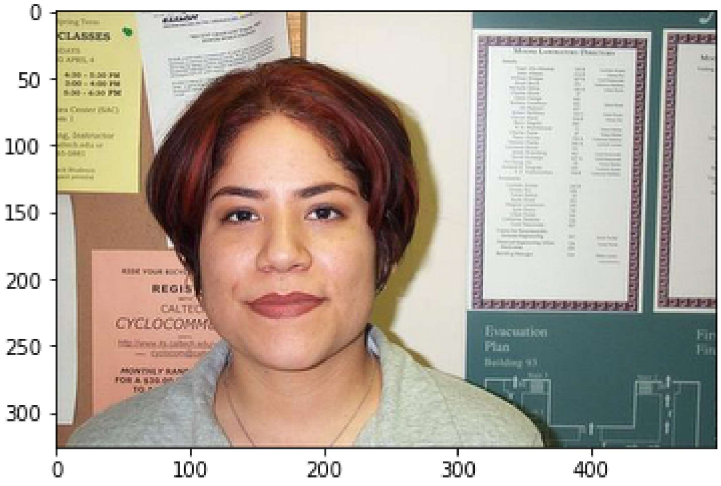


/content/caltech101/Motorbikes/image_0001_Original Image

```
********* Predictions ***********
```

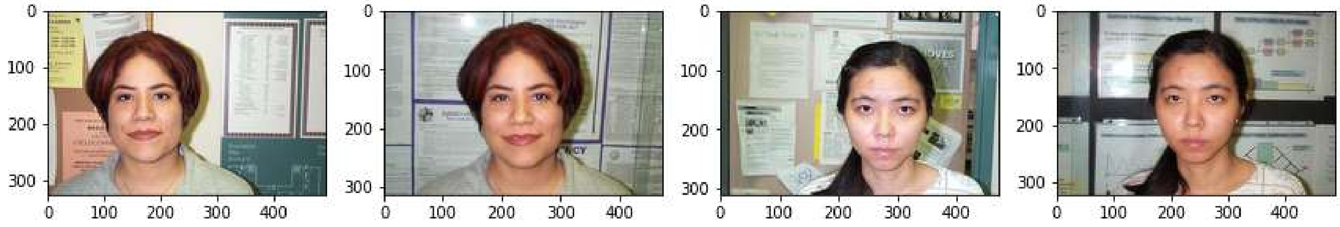


```
******Original Image***********
```

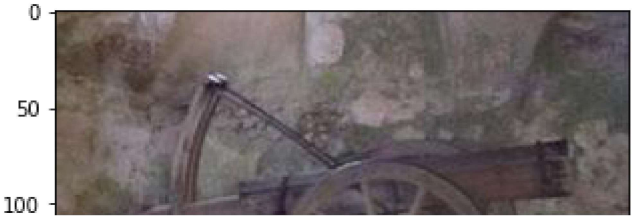


/content/caltech101/Faces/image_0165_Original Image

```
********* Predictions ***********
```

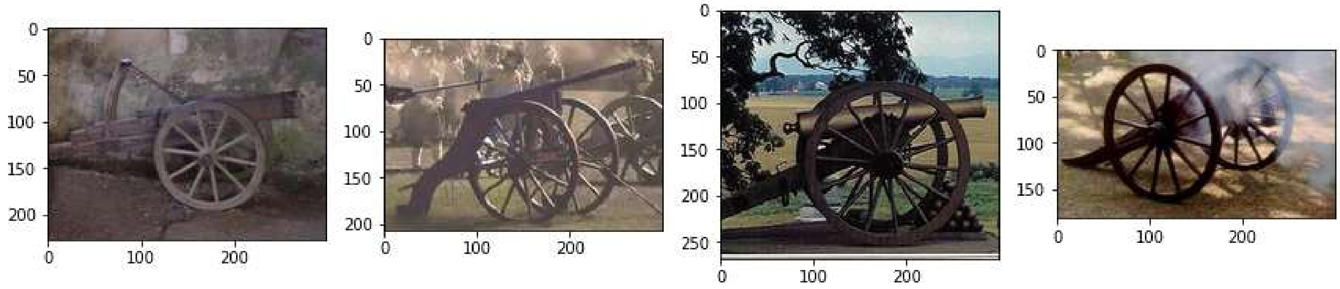


```
******Original Image***********
```

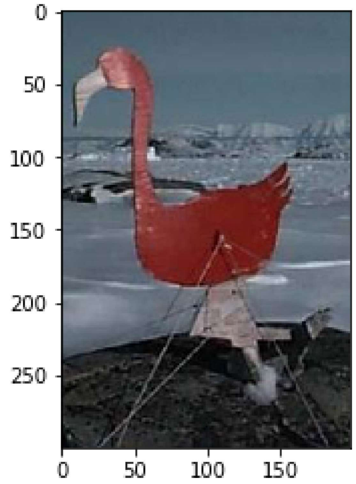


/content/caltech101/cannon/image_0028_Original Image

```
********* Predictions ***********
```

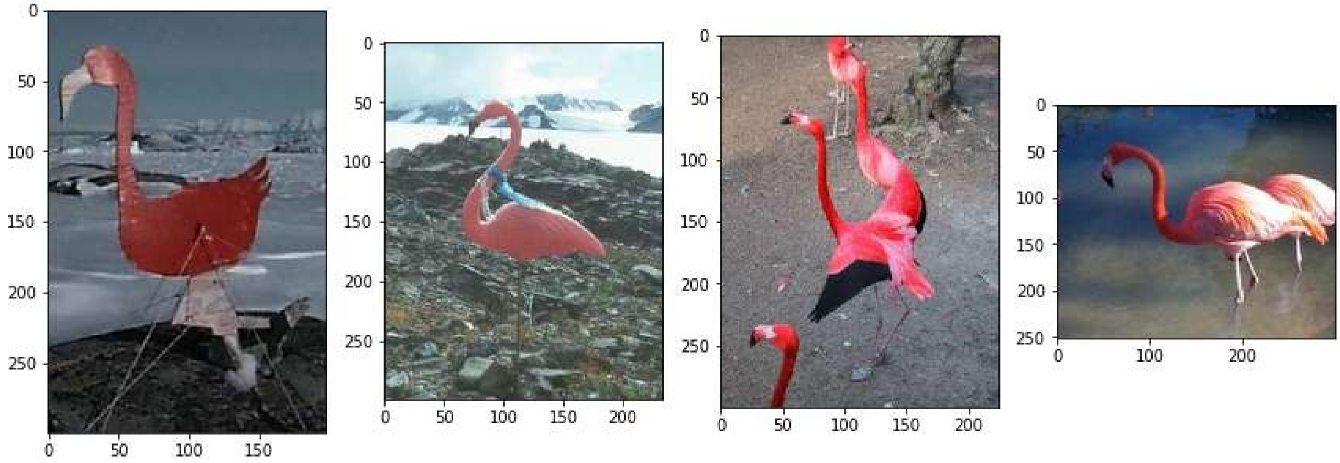


```
******Original Image***********
```

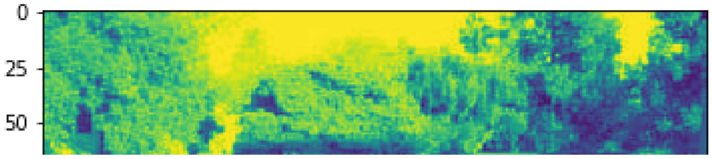


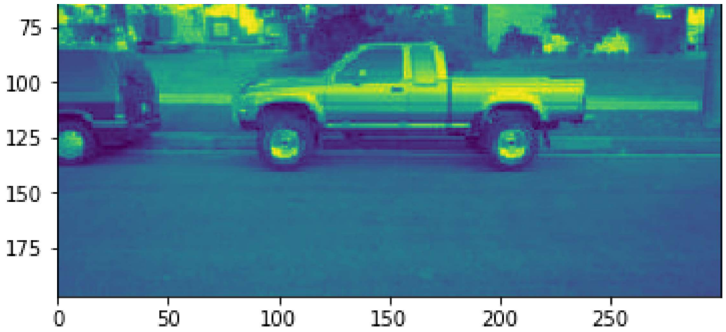
/content/caltech101/flamingo/image_0066_Original Image

```
********* Predictions ***********
```



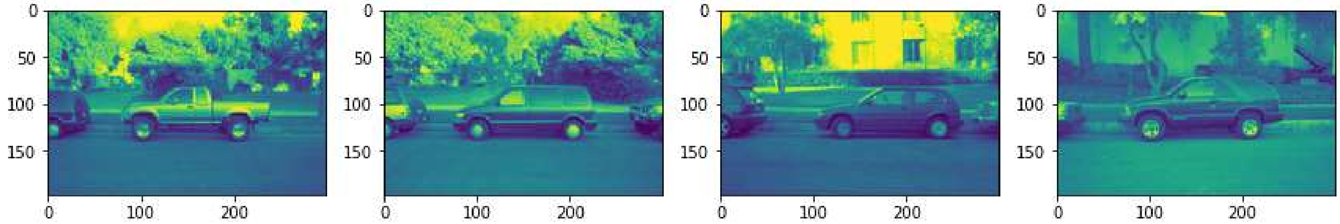
```
******Original Image***********
```



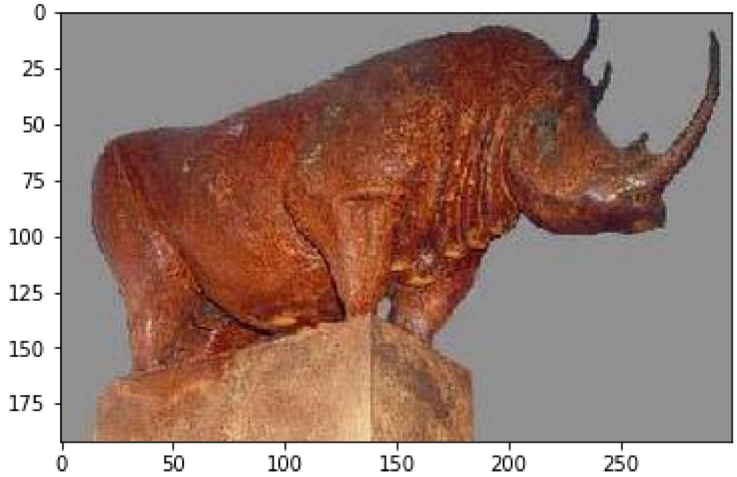


/content/caltech101/car_side/image_0109_Original Image

```
********* Predictions ***********
```

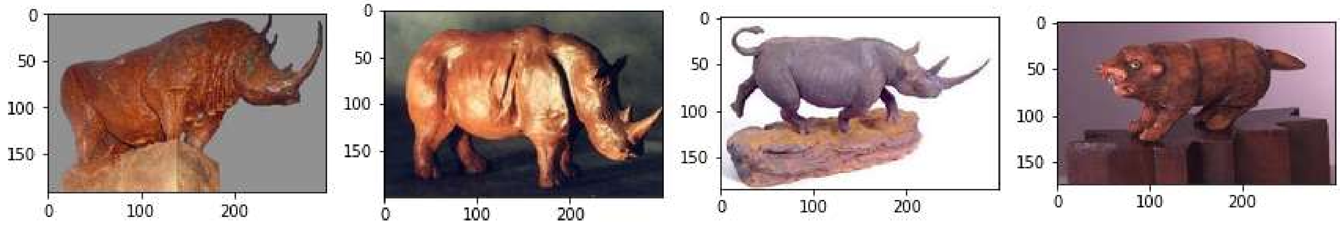


```
******Original Image***********
```



/content/caltech101/rhino/image_0041_Original Image

```
********* Predictions ***********
```



```
from sklearn.decomposition import PCA
import matplotlib
```

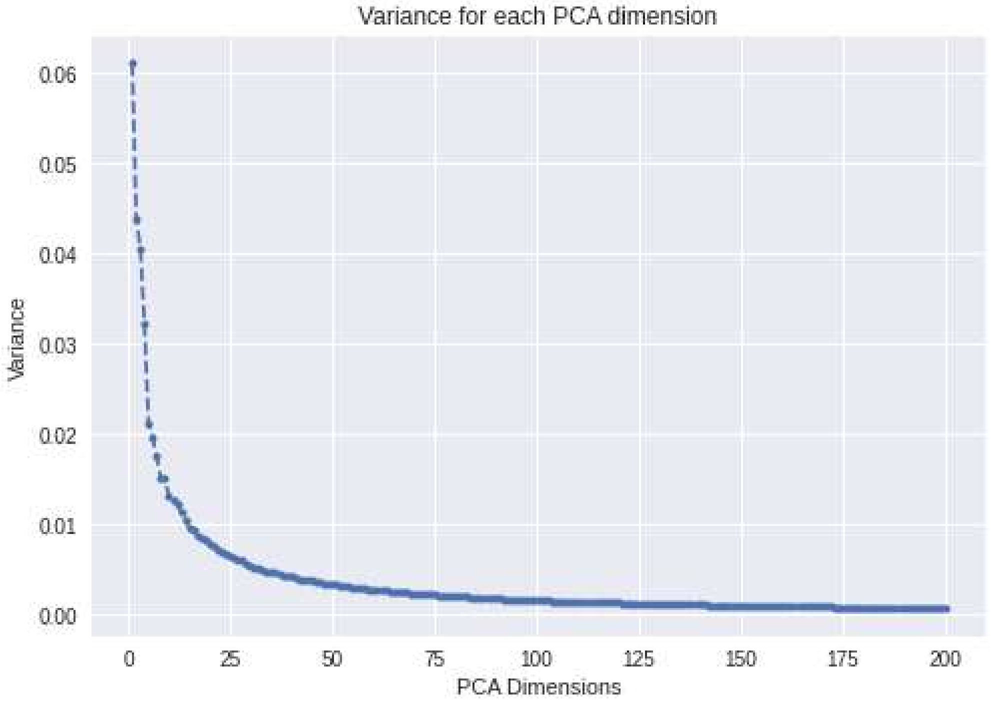
```
num_feature_dimensions=100
```

```
pca = PCA(n_components = num_feature_dimensions)
pca.fit(feature_list)
feature_list_compressed = pca.transform(feature_list)
```

```
print(pca.explained_variance_ratio_[0:20])
```

```
[0.0611019  0.04382469 0.04060571 0.0322854  0.02124297 0.01967339
 0.01750923 0.01519272 0.01506693 0.01313027 0.01261716 0.01226298
 0.01129625 0.01055882 0.00959002 0.0093974  0.00869047 0.00849483
 0.00836701 0.00772746]
```

```
pca = PCA(200)
pca.fit(feature_list)
matplotlib.style.use('seaborn')
plt.plot(range(1,201),pca.explained_variance_ratio_,'o--', markersize=4)
plt.title ('Variance for each PCA dimension')
plt.xlabel('PCA Dimensions')
plt.ylabel('Variance')
plt.grid(True)
plt.show()
```



```
plt.plot(range(1,201),pca.explained_variance_ratio_.cumsum(),'o--', markersize=4)
plt.title ('Cumulative Variance with each PCA dimension')
plt.xlabel('PCA Dimensions')
plt.ylabel('Variance')
plt.grid(True)
plt.show()
```