Programming Assignment 1:
Data Preparation and Understanding

1. In this semester, we will be using the "Stanford Dogs" dataset (http://vision.stanford.edu/aditya86/ImageNetDogs/) for all our 4 programming assignments. There are a total of 120 classes (dog breeds). The number of images for each class ranges from 148 to 252.
Each student will
(a) be assigned 4 classes to work on the 4 assignments.
(b) download Images (and also Annotations - bounding boxes) datasets for the 4 classes to work on.
(c) create a Github account to share (as collaborator) their solution (Readme, Codes, Processed Dataset for Code to run correctly) with the grader.

2. Use XML processing modules(https://docs.python.org/3/library/xml.html) to obtain bounding box information from Annotations datasets and scikitImage (Reference: https://scikit-image. org/) to perform image processing and feature extraction.

```
import os
annotations_dir = 'Annotation.zip'

if os.path.exists(annotations_dir):
   print("Annotation directory exists. Listing
files...")     for root, dirs, files in
os.walk(annotations_dir):        print(f"Found
directory: {root}")        for file in files:
          print(f"File: {file}") else:
   print("Annotation directory not found!")
```

OUTPUT:

File: n02110185_6411

File: n02110185_3808

File: n02110185_1066

File: n02110185_632

File: n02110185_8708
File: n02110185_1446

File: n02110185_2614

File: n02110185_11783

File: n02110185_1497

File: n02110185_1338

File: n02110185_13187

File: n02110185_3039

File: n02110185_10849

File: n02110185_1534

File: n02110185_8360

File: n02110185_8749

File: n02110185_13158

File: n02110185_14650

File: n02110185_10844

File: n02110185_3328

File: n02110185_3302

File: n02110185_13423

File: n02110185_6351

File: n02110185_9846

File: n02110185_1130

File: n02110185_7210

File: n02110185_7762

File: n02110185_14766

File: n02110185_4906

File: n02110185_14479

File: n02110185_11773

File: n02110185_10597

File: n02110185_14523 File: n02110185_10967

File: n02110185_712

File: n02110185_5622

File: n02110185_10171

File: n02110185_13855

File: n02110185_5871

File: n02110185_4030

File: n02110185_7413

File: n02110185_2593

File: n02110185_9975

File: n02110185_10273

File: n02110185_6775

File: n02110185_698

File: n02110185_14283

File: n02110185_10175

File: n02110185_7044

File: n02110185_1439

File: n02110185_9461

File: n02110185_6473

File: n02110185_9429

File: n02110185_12656

File: n02110185_13942

File: n02110185_7379

File: n02110185_10360

File: n02110185_8216

File: n02110185_8564

File: n02110185_13821

File: n02110185_815 File: n02110185_11635

File: n02110185_1532

File: n02110185_10047

File: n02110185_6438

File: n02110185_699

File: n02110185_5495

File: n02110185_9712

File: n02110185_9194

File: n02110185_4522

File: n02110185_1614

File: n02110185_56

File: n02110185_1794

File: n02110185_353

File: n02110185_12380

File: n02110185_1289

File: n02110185_11131

File: n02110185_9177

File: n02110185_2446

File: n02110185_11396

File: n02110185_14056

File: n02110185_1164

File: n02110185_1598

File: n02110185_14061

File: n02110185_2820

File: n02110185_11580

File: n02110185_519

File: n02110185_6409

File: n02110185_2672 File: n02110185_12441

File: n02110185_14597

File: n02110185_2728

File: n02110185_2368

File: n02110185_13127

File: n02110185_14594

File: n02110185_1469

File: n02110185_7980

File: n02110185_14906

File: n02110185_2604

File: n02110185_11287

File: n02110185_5030

File: n02110185_9086

File: n02110185_9833

File: n02110185_5143

File: n02110185_4060

File: n02110185_8748

File: n02110185_8154

File: n02110185_1178

File: n02110185_4294

File: n02110185_6564

File: n02110185_12498

File: n02110185_12678

File: n02110185_3291

File: n02110185_12478

File: n02110185_13434

File: n02110185_1552

File: n02110185_3589 File: n02110185_9001

File: n02110185_3406

File: n02110185_15019

File: n02110185_3540

File: n02110185_4115

File: n02110185_2701

File: n02110185_7879

File: n02110185_4186

File: n02110185_8966

File: n02110185_8397

File: n02110185_6105

File: n02110185_2736

File: n02110185_8162

File: n02110185_11626

File: n02110185_11409

File: n02110185_5973

File: n02110185_10955

File: n02110185_2941

File: n02110185_5624

File: n02110185_6746

File: n02110185_10902

File: n02110185_9396

File: n02110185_14560

File: n02110185_13794

File: n02110185_931

File: n02110185_7117

File: n02110185_5392

File: n02110185_4133

File: n02110185_5159

File: n02110185_7329

File: n02110185_6263

File: n02110185_7246

File: n02110185_12748

File: n02110185_184

File: n02110185_6780

File: n02110185_10875

File: n02110185_8923

File: n02110185_7936

File: n02110185_6094

File: n02110185_725

File: n02110185_9334

File: n02110185_7564

File: n02110185_1748

File: n02110185_13704

File: n02110185_11138

File: n02110185_4677

File: n02110185_13197

File: n02110185_10116

File: n02110185_7888

File: n02110185_5628

File: n02110185_8600

File: n02110185_7594

File: n02110185_11445

File: n02110185_5172

File: n02110185_5716

File: n02110185_8327 File: n02110185_388

File: n02110185_13282

File: n02110185_6850

File: n02110185_12120

File: n02110185_8005

File: n02110185_15063

File: n02110185_58

File: n02110185_8860

File: n02110185_120

File: n02110185_10898

File: n02110185_14289

File: n02110185_3651

File: n02110185_248

File: n02110185_4694

File: n02110185_1511

File: n02110185_9855

File: n02110185_11636

File: n02110185_11841

File: n02110185_11114

Found directory: /content/sample_data/Annotation/n02094114-Norfolk_terrier **(a) Cropping and Resize Images in Your 4-class Images Dataset: Use the bounding box information in the Annotations dataset relevant to your 4-class Images Dataset to crop the images in your dataset and then resize each image to a 128×128 pixel image.**

```
import os
import numpy as np
from PIL import Image
import cv2
from sklearn.decomposition import PCA
from sklearn.preprocessing import normalize
from sklearn.metrics import pairwise
import matplotlib.pyplot as plt
import zipfile
```

```
images_folder_path = 'Annotation.zip'
extracted_images_path = 'Breeds'


with zipfile.ZipFile(images_folder_path, 'r') as zip_ref:
  zip_ref.extractall(extracted_images_path)


images_folder_path = extracted_images_path

def load_images(folder_path):
  """Loads images from the specified folder."""
  images = []
  for filename in os.listdir(folder_path):
    if filename.lower().endswith(('.jpg', '.jpeg', '.png')): # Add other image
extensions if needed and convert filename to lowercase for comparison
      img_path = os.path.join(folder_path, filename)
      img = cv2.imread(img_path)
      if img is not None:
        images.append(img)
  return images

dog_images = load_images(images_folder_path)

def crop_and_resize_images(images):
  resized_images = []
  for img in images:
    h, w, _ = img.shape
    center_h, center_w = h // 2, w // 2
    cropped_img = img[center_h-50:center_h+50, center_w-50:center_w+50]
    resized_img = cv2.resize(cropped_img, (128, 128),
interpolation=cv2.INTER_AREA)
    resized_images.append(resized_img)
  return np.array(resized_images)
```

```
cropped_resized_images = crop_and_resize_images(dog_images)

def compute_histograms(images):
    histograms = []
    for img in images:
        hist_r = cv2.calcHist([img], [0], None, [256], [0, 256])
        hist_g = cv2.calcHist([img], [1], None, [256], [0, 256])
        hist_b = cv2.calcHist([img], [2], None, [256], [0, 256])
        hist = np.concatenate((hist_r, hist_g, hist_b), axis=0)
        histograms.append(hist.flatten())
    return np.array(histograms)

histograms = compute_histograms(cropped_resized_images)

def compute_similarity_measurements(histograms):
    distances = {}
    for i in range(len(histograms)):
        for j in range(i + 1, len(histograms)):
            euclidean_dist = np.linalg.norm(histograms[i] - histograms[j])
            distances[(i, j)] = {'Euclidean': euclidean_dist}

            manhattan_dist = np.sum(np.abs(histograms[i] - histograms[j]))
            distances[(i, j)]['Manhattan'] = manhattan_dist

            cosine_dist = pairwise.cosine_distances(histograms[i].reshape(1, -1),
histograms[j].reshape(1, -1))[0][0]
            distances[(i, j)]['Cosine'] = cosine_dist
    return distances

similarity_measurements = compute_similarity_measurements(histograms)



for key, value in similarity_measurements.items():
    print(f"Images {key}: {value}")
```

```
def perform_pca(histograms):

    if histograms.size == 0:
        print("Histograms array is empty. Cannot perform PCA.")
        return


    if histograms.ndim == 1:
        histograms = histograms.reshape(-1, 1)


    histograms_normalized = normalize(histograms)
    pca = PCA(n_components=2)
    reduced_data = pca.fit_transform(histograms_normalized)


    plt.scatter(reduced_data[:, 0], reduced_data[:, 1])
    plt.title('PCA of Image Histograms')
    plt.xlabel('Principal Component 1')
    plt.ylabel('Principal Component 2')
    plt.show()


perform_pca(histograms)
```

```
Images (0, 1): {'Euclidean': 3457.4624, 'Manhattan': 60582.0, 'Cosine':
0.6825162}
Images (0, 2): {'Euclidean': 3984.1545, 'Manhattan': 65490.0, 'Cosine':
0.8253741}
Images (0, 3): {'Euclidean': 3899.4668, 'Manhattan': 71376.0, 'Cosine':
0.7919182}
Images (0, 4): {'Euclidean': 2996.233, 'Manhattan': 52486.0, 'Cosine':
0.5265511} Images (0, 5): {'Euclidean': 3713.0256, 'Manhattan': 69834.0,
'Cosine':
0.80765843}
```

Images (0, 6): {'Euclidean': 3549.726, 'Manhattan': 68584.0, 'Cosine':
0.7822863}
Images (0, 7): {'Euclidean': 4273.889, 'Manhattan': 83638.0, 'Cosine':
0.8985965} Images (0, 8): {'Euclidean': 4195.953, 'Manhattan': 82846.0,
'Cosine':
0.88892967}
Images (0, 9): {'Euclidean': 4797.943, 'Manhattan': 72312.0, 'Cosine':
0.91561556}
Images (0, 10): {'Euclidean': 3833.5942, 'Manhattan': 69682.0, 'Cosine':
0.69148505}
Images (0, 11): {'Euclidean': 4038.895, 'Manhattan': 75160.0, 'Cosine':
0.8729541}
Images (0, 12): {'Euclidean': 4304.3984, 'Manhattan': 79452.0, 'Cosine':
0.8892273}
Images (0, 13): {'Euclidean': 3132.434, 'Manhattan': 53988.0, 'Cosine':
0.57699823}
Images (0, 14): {'Euclidean': 3686.408, 'Manhattan': 71616.0, 'Cosine':
0.80207825}
Images (0, 15): {'Euclidean': 4196.2603, 'Manhattan': 82572.0, 'Cosine':
0.8932265}
Images (0, 16): {'Euclidean': 3907.0522, 'Manhattan': 76738.0, 'Cosine':
0.8767342}
Images (0, 17): {'Euclidean': 4066.2576, 'Manhattan': 62318.0, 'Cosine':
0.8289352}
Images (0, 18): {'Euclidean': 3911.8364, 'Manhattan': 74360.0, 'Cosine':
0.7885997}
Images (0, 19): {'Euclidean': 3661.05, 'Manhattan': 64994.0, 'Cosine':
0.7596242} Images (0, 20): {'Euclidean': 3752.5872, 'Manhattan': 65990.0,
'Cosine':
0.7929658}
Images (0, 21): {'Euclidean': 4070.2698, 'Manhattan': 72374.0, 'Cosine':
0.8442098}
Images (0, 22): {'Euclidean': 3581.0945, 'Manhattan': 68900.0, 'Cosine':
0.8061286}
Images (0, 23): {'Euclidean': 2477.4883, 'Manhattan': 36760.0, 'Cosine':
0.3320266}
Images (0, 24): {'Euclidean': 3523.3938, 'Manhattan': 60780.0, 'Cosine':
0.67376536}

Images (0, 25): {'Euclidean': 4243.425, 'Manhattan': 76318.0, 'Cosine': 0.8454045}
Images (0, 26): {'Euclidean': 4211.747, 'Manhattan': 53840.0, 'Cosine': 0.6044014}
Images (0, 27): {'Euclidean': 5275.997, 'Manhattan': 75268.0, 'Cosine': 0.85031015}
Images (0, 28): {'Euclidean': 3197.6758, 'Manhattan': 57088.0, 'Cosine': 0.6010748}
Images (0, 29): {'Euclidean': 3742.5112, 'Manhattan': 74444.0, 'Cosine': 0.8290365}
Images (0, 30): {'Euclidean': 3906.7087, 'Manhattan': 67662.0, 'Cosine': 0.8295082}
Images (0, 31): {'Euclidean': 4954.6445, 'Manhattan': 80432.0, 'Cosine': 0.8677642}
Images (1, 2): {'Euclidean': 2879.3455, 'Manhattan': 53638.0, 'Cosine': 0.5772694}
Images (1, 3): {'Euclidean': 2563.737, 'Manhattan': 51132.0, 'Cosine': 0.4572196} Images (1, 4): {'Euclidean': 1441.7198, 'Manhattan': 30092.0, 'Cosine': 0.18363315}
Images (1, 5): {'Euclidean': 1897.0409, 'Manhattan': 37262.0, 'Cosine': 0.301046} Images (1, 6): {'Euclidean': 1990.4417, 'Manhattan': 41460.0, 'Cosine':
0.36089873}
Images (1, 7): {'Euclidean': 3386.7417, 'Manhattan': 68690.0, 'Cosine': 0.73972344}
Images (1, 8): {'Euclidean': 2427.105, 'Manhattan': 48100.0, 'Cosine': 0.38683277}
Images (1, 9): {'Euclidean': 4036.9868, 'Manhattan': 63338.0, 'Cosine': 0.78991526}
Images (1, 10): {'Euclidean': 1791.3771, 'Manhattan': 34342.0, 'Cosine': 0.17047322}
Images (1, 11): {'Euclidean': 3264.496, 'Manhattan': 73478.0, 'Cosine': 0.77587277}
Images (1, 12): {'Euclidean': 2601.8496, 'Manhattan': 44322.0, 'Cosine': 0.40987408}
Images (1, 13): {'Euclidean': 2631.195, 'Manhattan': 58214.0, 'Cosine': 0.6135694}

Images (1, 14): {'Euclidean': 2209.939, 'Manhattan': 45746.0, 'Cosine': 0.41355544}
Images (1, 15): {'Euclidean': 3593.2905, 'Manhattan': 82528.0, 'Cosine': 0.87236464}
Images (1, 16): {'Euclidean': 2519.3972, 'Manhattan': 50648.0, 'Cosine': 0.51227725}
Images (1, 17): {'Euclidean': 3097.13, 'Manhattan': 48206.0, 'Cosine': 0.6334985} Images (1, 18): {'Euclidean': 1678.9229, 'Manhattan': 31052.0, 'Cosine':
0.18612045}
Images (1, 19): {'Euclidean': 1200.7955, 'Manhattan': 21454.0, 'Cosine': 0.114995}
Images (1, 20): {'Euclidean': 2618.249, 'Manhattan': 49120.0, 'Cosine': 0.5404015}
Images (1, 21): {'Euclidean': 2752.3206, 'Manhattan': 52292.0, 'Cosine': 0.5097543}
Images (1, 22): {'Euclidean': 2203.5845, 'Manhattan': 49184.0, 'Cosine': 0.45019388}
Images (1, 23): {'Euclidean': 2454.9343, 'Manhattan': 49432.0, 'Cosine': 0.51666117}
Images (1, 24): {'Euclidean': 1573.7338, 'Manhattan': 27678.0, 'Cosine': 0.18375003}
Images (1, 25): {'Euclidean': 1918.5499, 'Manhattan': 33426.0, 'Cosine': 0.19910103}
Images (1, 26): {'Euclidean': 3965.6023, 'Manhattan': 45658.0, 'Cosine': 0.6082397}
Images (1, 27): {'Euclidean': 4944.118, 'Manhattan': 75536.0, 'Cosine': 0.86597705}
Images (1, 28): {'Euclidean': 1548.5923, 'Manhattan': 29308.0, 'Cosine': 0.20845056}
Images (1, 29): {'Euclidean': 2893.024, 'Manhattan': 68878.0, 'Cosine': 0.7100404}
Images (1, 30): {'Euclidean': 3057.737, 'Manhattan': 64026.0, 'Cosine': 0.6970938}
Images (1, 31): {'Euclidean': 3509.2927, 'Manhattan': 54216.0, 'Cosine': 0.4658035}
Images (2, 3): {'Euclidean': 3285.9731, 'Manhattan': 61746.0, 'Cosine':

0.6657152}
Images (2, 4): {'Euclidean': 2594.4866, 'Manhattan': 44610.0, 'Cosine': 0.5016242}
Images (2, 5): {'Euclidean': 2830.9353, 'Manhattan': 50716.0, 'Cosine': 0.56850004}
Images (2, 6): {'Euclidean': 2132.5903, 'Manhattan': 37912.0, 'Cosine': 0.33022285}
Images (2, 7): {'Euclidean': 3127.8516, 'Manhattan': 57930.0, 'Cosine': 0.56141603}
Images (2, 8): {'Euclidean': 3054.716, 'Manhattan': 59122.0, 'Cosine': 0.5529129} Images (2, 9): {'Euclidean': 3929.1985, 'Manhattan': 48792.0, 'Cosine': 0.68377864}
Images (2, 10): {'Euclidean': 3483.2573, 'Manhattan': 66374.0, 'Cosine': 0.65995973}
Images (2, 11): {'Euclidean': 2441.729, 'Manhattan': 41954.0, 'Cosine': 0.37892938}
Images (2, 12): {'Euclidean': 3221.9133, 'Manhattan': 54446.0, 'Cosine': 0.57781684}
Images (2, 13): {'Euclidean': 2798.746, 'Manhattan': 49426.0, 'Cosine': 0.57975984}
Images (2, 14): {'Euclidean': 2268.7092, 'Manhattan': 41828.0, 'Cosine': 0.36269557}
Images (2, 15): {'Euclidean': 2729.3105, 'Manhattan': 48968.0, 'Cosine': 0.44380963}
Images (2, 16): {'Euclidean': 2661.9116, 'Manhattan': 49350.0, 'Cosine': 0.48789787}
Images (2, 17): {'Euclidean': 3214.084, 'Manhattan': 40416.0, 'Cosine': 0.6072681}
Images (2, 18): {'Euclidean': 3227.326, 'Manhattan': 61282.0, 'Cosine': 0.6338905}
Images (2, 19): {'Euclidean': 2804.4724, 'Manhattan': 52378.0, 'Cosine': 0.5382362}
Images (2, 20): {'Euclidean': 2400.746, 'Manhattan': 38830.0, 'Cosine': 0.3889438}
Images (2, 21): {'Euclidean': 3124.5874, 'Manhattan': 53926.0, 'Cosine': 0.5854156}
Images (2, 22): {'Euclidean': 2370.6711, 'Manhattan': 38452.0, 'Cosine': 0.42028904}

Images (2, 23): {'Euclidean': 2843.715, 'Manhattan': 46474.0, 'Cosine': 0.5845416}
Images (2, 24): {'Euclidean': 3081.309, 'Manhattan': 50580.0, 'Cosine': 0.6197362}
Images (2, 25): {'Euclidean': 3576.5728, 'Manhattan': 66196.0, 'Cosine': 0.6941053}
Images (2, 26): {'Euclidean': 4770.62, 'Manhattan': 74442.0, 'Cosine': 0.8690337} Images (2, 27): {'Euclidean': 4844.622, 'Manhattan': 60034.0, 'Cosine': 0.7707057}
Images (2, 28): {'Euclidean': 2833.753, 'Manhattan': 53998.0, 'Cosine': 0.5904722}
Images (2, 29): {'Euclidean': 2456.6226, 'Manhattan': 43942.0, 'Cosine': 0.42838317}
Images (2, 30): {'Euclidean': 2742.1465, 'Manhattan': 41196.0, 'Cosine': 0.48755598}
Images (2, 31): {'Euclidean': 4565.3247, 'Manhattan': 73574.0, 'Cosine': 0.8125695}
Images (3, 4): {'Euclidean': 2466.3167, 'Manhattan': 50974.0, 'Cosine': 0.45212758}
Images (3, 5): {'Euclidean': 2164.9988, 'Manhattan': 41610.0, 'Cosine': 0.32804644}
Images (3, 6): {'Euclidean': 2022.2324, 'Manhattan': 42574.0, 'Cosine': 0.29471135}
Images (3, 7): {'Euclidean': 2943.7708, 'Manhattan': 56268.0, 'Cosine': 0.4982103}
Images (3, 8): {'Euclidean': 2166.453, 'Manhattan': 39150.0, 'Cosine': 0.2784438} Images (3, 9): {'Euclidean': 3955.8533, 'Manhattan': 57856.0, 'Cosine': 0.6948316}
Images (3, 10): {'Euclidean': 3448.268, 'Manhattan': 64682.0, 'Cosine': 0.6481024}
Images (3, 11): {'Euclidean': 3592.3232, 'Manhattan': 77598.0, 'Cosine': 0.8232112}
Images (3, 12): {'Euclidean': 2829.6936, 'Manhattan': 50056.0, 'Cosine': 0.44569123}
Images (3, 13): {'Euclidean': 2817.2014, 'Manhattan': 59294.0, 'Cosine': 0.5900784}
Images (3, 14): {'Euclidean': 2309.8823, 'Manhattan': 47590.0, 'Cosine': 0.37813026}

Images (3, 15): {'Euclidean': 3807.6658, 'Manhattan': 83638.0, 'Cosine': 0.86649156}
Images (3, 16): {'Euclidean': 1883.9453, 'Manhattan': 37890.0, 'Cosine': 0.24115628}
Images (3, 17): {'Euclidean': 3515.8252, 'Manhattan': 60548.0, 'Cosine': 0.7286817}
Images (3, 18): {'Euclidean': 2389.0723, 'Manhattan': 42508.0, 'Cosine': 0.34825432}
Images (3, 19): {'Euclidean': 2476.9868, 'Manhattan': 49270.0, 'Cosine': 0.4197551}
Images (3, 20): {'Euclidean': 2818.3657, 'Manhattan': 53922.0, 'Cosine': 0.5400009}
Images (3, 21): {'Euclidean': 2439.4783, 'Manhattan': 43530.0, 'Cosine': 0.35762024}
Images (3, 22): {'Euclidean': 2142.0293, 'Manhattan': 45434.0, 'Cosine': 0.33692336}
Images (3, 23): {'Euclidean': 2656.3062, 'Manhattan': 54976.0, 'Cosine': 0.5095974}
Images (3, 24): {'Euclidean': 2924.5989, 'Manhattan': 53702.0, 'Cosine': 0.5597929}
Images (3, 25): {'Euclidean': 2957.4216, 'Manhattan': 50442.0, 'Cosine': 0.47346246}
Images (3, 26): {'Euclidean': 4312.438, 'Manhattan': 65836.0, 'Cosine': 0.69696975}
Images (3, 27): {'Euclidean': 5185.076, 'Manhattan': 81194.0, 'Cosine': 0.90157306}
Images (3, 28): {'Euclidean': 1908.535, 'Manhattan': 39050.0, 'Cosine': 0.25606334}
Images (3, 29): {'Euclidean': 2822.1843, 'Manhattan': 62686.0, 'Cosine': 0.57272196}
Images (3, 30): {'Euclidean': 3383.0288, 'Manhattan': 66656.0, 'Cosine': 0.7450859}
Images (3, 31): {'Euclidean': 3643.2742, 'Manhattan': 55428.0, 'Cosine': 0.49246466}
Images (4, 5): {'Euclidean': 1580.56, 'Manhattan': 30512.0, 'Cosine': 0.23059249} Images (4, 6): {'Euclidean': 1648.1517, 'Manhattan': 34492.0, 'Cosine':

0.28413463}
Images (4, 7): {'Euclidean': 2675.2256, 'Manhattan': 54626.0, 'Cosine': 0.4804126}
Images (4, 8): {'Euclidean': 2459.3083, 'Manhattan': 52046.0, 'Cosine': 0.42053813}
Images (4, 9): {'Euclidean': 3812.7488, 'Manhattan': 55962.0, 'Cosine': 0.73373425}
Images (4, 10): {'Euclidean': 2479.069, 'Manhattan': 50512.0, 'Cosine': 0.36714256}
Images (4, 11): {'Euclidean': 2948.2969, 'Manhattan': 65466.0, 'Cosine': 0.69005984}
Images (4, 12): {'Euclidean': 2761.207, 'Manhattan': 50808.0, 'Cosine': 0.49094784}
Images (4, 13): {'Euclidean': 2289.4675, 'Manhattan': 49704.0, 'Cosine': 0.5237316}
Images (4, 14): {'Euclidean': 1753.7417, 'Manhattan': 36798.0, 'Cosine': 0.28959978}
Images (4, 15): {'Euclidean': 3146.7917, 'Manhattan': 70010.0, 'Cosine': 0.7214392}
Images (4, 16): {'Euclidean': 2003.4625, 'Manhattan': 42984.0, 'Cosine': 0.3553053}
Images (4, 17): {'Euclidean': 2923.333, 'Manhattan': 43522.0, 'Cosine': 0.6046927}
Images (4, 18): {'Euclidean': 2218.5107, 'Manhattan': 48602.0, 'Cosine': 0.35129243}
Images (4, 19): {'Euclidean': 1933.1813, 'Manhattan': 42042.0, 'Cosine': 0.32556522}
Images (4, 20): {'Euclidean': 2483.4666, 'Manhattan': 48458.0, 'Cosine': 0.534603}
Images (4, 21): {'Euclidean': 2193.7024, 'Manhattan': 39070.0, 'Cosine': 0.33328873}
Images (4, 22): {'Euclidean': 1651.2814, 'Manhattan': 37030.0, 'Cosine': 0.2900083}
Images (4, 23): {'Euclidean': 2096.143, 'Manhattan': 39470.0, 'Cosine': 0.42136222}
Images (4, 24): {'Euclidean': 1477.7063, 'Manhattan': 23864.0, 'Cosine': 0.16601485}
Images (4, 25): {'Euclidean': 2722.2163, 'Manhattan': 54084.0, 'Cosine':

0.45458508}
Images (4, 26): {'Euclidean': 3986.9119, 'Manhattan': 53078.0, 'Cosine': 0.6371231}
Images (4, 27): {'Euclidean': 4808.4995, 'Manhattan': 73634.0, 'Cosine': 0.84622896}
Images (4, 28): {'Euclidean': 1513.9683, 'Manhattan': 31128.0, 'Cosine': 0.22582865}
Images (4, 29): {'Euclidean': 2303.5398, 'Manhattan': 52404.0, 'Cosine': 0.5023856}
Images (4, 30): {'Euclidean': 2746.9685, 'Manhattan': 53176.0, 'Cosine': 0.61396} Images (4, 31): {'Euclidean': 3856.9592, 'Manhattan': 66276.0, 'Cosine': 0.6096979}
Images (5, 6): {'Euclidean': 1579.9728, 'Manhattan': 32376.0, 'Cosine': 0.23326004}
Images (5, 7): {'Euclidean': 2081.384, 'Manhattan': 38864.0, 'Cosine': 0.26782048}
Images (5, 8): {'Euclidean': 2281.7485, 'Manhattan': 42856.0, 'Cosine': 0.3444031}
Images (5, 9): {'Euclidean': 3757.537, 'Manhattan': 50556.0, 'Cosine': 0.68107}
Images (5, 10): {'Euclidean': 3017.7136, 'Manhattan': 58924.0, 'Cosine': 0.55169463}
Images (5, 11): {'Euclidean': 3051.3652, 'Manhattan': 67268.0, 'Cosine': 0.69186586}
Images (5, 12): {'Euclidean': 2626.5312, 'Manhattan': 43208.0, 'Cosine': 0.42341816}
Images (5, 13): {'Euclidean': 2483.9043, 'Manhattan': 55500.0, 'Cosine': 0.56267804}
Images (5, 14): {'Euclidean': 1637.9353, 'Manhattan': 33472.0, 'Cosine': 0.23344076}
Images (5, 15): {'Euclidean': 3263.8872, 'Manhattan': 69760.0, 'Cosine': 0.7314582}
Images (5, 16): {'Euclidean': 1655.0457, 'Manhattan': 30656.0, 'Cosine': 0.22618556}
Images (5, 17): {'Euclidean': 3155.72, 'Manhattan': 51488.0, 'Cosine': 0.67083865}
Images (5, 18): {'Euclidean': 2175.0398, 'Manhattan': 42216.0, 'Cosine': 0.32458985}

Images (5, 19): {'Euclidean': 2109.1055, 'Manhattan': 44324.0, 'Cosine': 0.36362302}
Images (5, 20): {'Euclidean': 2562.0657, 'Manhattan': 49402.0, 'Cosine': 0.5297445}
Images (5, 21): {'Euclidean': 2177.3516, 'Manhattan': 38484.0, 'Cosine': 0.31775331}
Images (5, 22): {'Euclidean': 1324.3625, 'Manhattan': 27674.0, 'Cosine': 0.16307133}
Images (5, 23): {'Euclidean': 2358.9866, 'Manhattan': 47946.0, 'Cosine': 0.49043715}
Images (5, 24): {'Euclidean': 2106.374, 'Manhattan': 32198.0, 'Cosine': 0.33800125}
Images (5, 25): {'Euclidean': 2680.8108, 'Manhattan': 49732.0, 'Cosine': 0.42517674}
Images (5, 26): {'Euclidean': 4339.8877, 'Manhattan': 63678.0, 'Cosine': 0.7659661}
Images (5, 27): {'Euclidean': 4908.069, 'Manhattan': 74788.0, 'Cosine': 0.8603437}
Images (5, 28): {'Euclidean': 1710.1702, 'Manhattan': 35130.0, 'Cosine': 0.26325428}
Images (5, 29): {'Euclidean': 2137.2295, 'Manhattan': 45846.0, 'Cosine': 0.3979597}
Images (5, 30): {'Euclidean': 2925.442, 'Manhattan': 57742.0, 'Cosine': 0.65196776}
Images (5, 31): {'Euclidean': 3754.8962, 'Manhattan': 60850.0, 'Cosine': 0.556604}
Images (6, 7): {'Euclidean': 2293.849, 'Manhattan': 46528.0, 'Cosine': 0.34063888}
Images (6, 8): {'Euclidean': 1963.5442, 'Manhattan': 41980.0, 'Cosine': 0.25239253}
Images (6, 9): {'Euclidean': 3102.6333, 'Manhattan': 37818.0, 'Cosine': 0.43773448}
Images (6, 10): {'Euclidean': 2969.2363, 'Manhattan': 61464.0, 'Cosine': 0.5615693}
Images (6, 11): {'Euclidean': 2268.3086, 'Manhattan': 49294.0, 'Cosine': 0.40244687}
Images (6, 12): {'Euclidean': 2622.307, 'Manhattan': 47256.0, 'Cosine': 0.43962717}

Images (6, 13): {'Euclidean': 2039.8215, 'Manhattan': 42498.0, 'Cosine': 0.4236027}
Images (6, 14): {'Euclidean': 1099.9264, 'Manhattan': 22488.0, 'Cosine': 0.11281431}
Images (6, 15): {'Euclidean': 2459.0364, 'Manhattan': 52920.0, 'Cosine': 0.42798865}
Images (6, 16): {'Euclidean': 1435.5446, 'Manhattan': 29460.0, 'Cosine': 0.17968196}
Images (6, 17): {'Euclidean': 2737.6357, 'Manhattan': 40680.0, 'Cosine': 0.530151}
Images (6, 18): {'Euclidean': 2185.6938, 'Manhattan': 46616.0, 'Cosine': 0.34198558}
Images (6, 19): {'Euclidean': 1898.5131, 'Manhattan': 40574.0, 'Cosine': 0.3173778}
Images (6, 20): {'Euclidean': 2061.1978, 'Manhattan': 42042.0, 'Cosine': 0.36940712}
Images (6, 21): {'Euclidean': 1993.8987, 'Manhattan': 35200.0, 'Cosine': 0.26880467}
Images (6, 22): {'Euclidean': 995.4647, 'Manhattan': 20860.0, 'Cosine': 0.10770154}
Images (6, 23): {'Euclidean': 2206.57, 'Manhattan': 45404.0, 'Cosine': 0.47540486}
Images (6, 24): {'Euclidean': 2323.1375, 'Manhattan': 40404.0, 'Cosine': 0.4430353}
Images (6, 25): {'Euclidean': 2654.6873, 'Manhattan': 52444.0, 'Cosine': 0.43122447}
Images (6, 26): {'Euclidean': 4365.753, 'Manhattan': 70824.0, 'Cosine': 0.8138406}
Images (6, 27): {'Euclidean': 4354.813, 'Manhattan': 60550.0, 'Cosine': 0.6450757}
Images (6, 28): {'Euclidean': 1766.5175, 'Manhattan': 38304.0, 'Cosine': 0.31314796}
Images (6, 29): {'Euclidean': 1605.7783, 'Manhattan': 34232.0, 'Cosine': 0.24607193}
Images (6, 30): {'Euclidean': 2178.8887, 'Manhattan': 39902.0, 'Cosine': 0.3828361}

Images (6, 31): {'Euclidean': 3635.5142, 'Manhattan': 58734.0, 'Cosine': 0.5199179}
Images (7, 8): {'Euclidean': 3184.2512, 'Manhattan': 55878.0, 'Cosine': 0.5630572}
Images (7, 9): {'Euclidean': 3889.209, 'Manhattan': 54588.0, 'Cosine': 0.6404163} Images (7, 10): {'Euclidean': 4203.893, 'Manhattan': 84348.0, 'Cosine': 0.9081155}
Images (7, 11): {'Euclidean': 3100.5813, 'Manhattan': 60338.0, 'Cosine': 0.56831175}
Images (7, 12): {'Euclidean': 3545.1846, 'Manhattan': 58010.0, 'Cosine': 0.6598404}
Images (7, 13): {'Euclidean': 2978.2495, 'Manhattan': 64128.0, 'Cosine': 0.59928036}
Images (7, 14): {'Euclidean': 2179.8591, 'Manhattan': 39166.0, 'Cosine': 0.29748195}
Images (7, 15): {'Euclidean': 3233.133, 'Manhattan': 59634.0, 'Cosine': 0.58343816}
Images (7, 16): {'Euclidean': 2004.4716, 'Manhattan': 34886.0, 'Cosine': 0.24447823}
Images (7, 17): {'Euclidean': 3695.815, 'Manhattan': 67096.0, 'Cosine': 0.7531369}
Images (7, 18): {'Euclidean': 3547.174, 'Manhattan': 68466.0, 'Cosine': 0.7159155}
Images (7, 19): {'Euclidean': 3478.9324, 'Manhattan': 72972.0, 'Cosine': 0.76909065}
Images (7, 20): {'Euclidean': 3013.662, 'Manhattan': 59242.0, 'Cosine': 0.5677335}
Images (7, 21): {'Euclidean': 2801.8865, 'Manhattan': 48662.0, 'Cosine': 0.44039434}
Images (7, 22): {'Euclidean': 1787.43, 'Manhattan': 37394.0, 'Cosine': 0.18158942}
Images (7, 23): {'Euclidean': 3015.633, 'Manhattan': 64298.0, 'Cosine': 0.60194266}
Images (7, 24): {'Euclidean': 3224.8845, 'Manhattan': 60576.0, 'Cosine': 0.6297437}
Images (7, 25): {'Euclidean': 4024.1794, 'Manhattan': 76078.0, 'Cosine': 0.82984024}
Images (7, 26): {'Euclidean': 5015.1943, 'Manhattan': 85488.0, 'Cosine':

0.926374}
Images (7, 27): {'Euclidean': 5182.0874, 'Manhattan': 76894.0, 'Cosine': 0.8651019}
Images (7, 28): {'Euclidean': 2925.3115, 'Manhattan': 62412.0, 'Cosine': 0.5728577}
Images (7, 29): {'Euclidean': 2191.3757, 'Manhattan': 41370.0, 'Cosine': 0.30105114}
Images (7, 30): {'Euclidean': 3158.0205, 'Manhattan': 60328.0, 'Cosine': 0.60036325}
Images (7, 31): {'Euclidean': 4689.475, 'Manhattan': 75538.0, 'Cosine': 0.82500875}
Images (8, 9): {'Euclidean': 4221.9775, 'Manhattan': 67802.0, 'Cosine': 0.7752793}
Images (8, 10): {'Euclidean': 2982.44, 'Manhattan': 54304.0, 'Cosine': 0.46833462}
Images (8, 11): {'Euclidean': 3649.9648, 'Manhattan': 77542.0, 'Cosine': 0.81571996}
Images (8, 12): {'Euclidean': 2558.0024, 'Manhattan': 39874.0, 'Cosine': 0.35239542}
Images (8, 13): {'Euclidean': 3027.5037, 'Manhattan': 66966.0, 'Cosine': 0.6489675}
Images (8, 14): {'Euclidean': 1766.4178, 'Manhattan': 34046.0, 'Cosine': 0.19822127}
Images (8, 15): {'Euclidean': 3917.429, 'Manhattan': 82810.0, 'Cosine': 0.8830281}
Images (8, 16): {'Euclidean': 1824.9379, 'Manhattan': 31904.0, 'Cosine': 0.2112422}
Images (8, 17): {'Euclidean': 3553.8633, 'Manhattan': 65856.0, 'Cosine': 0.7173929}
Images (8, 18): {'Euclidean': 2027.6982, 'Manhattan': 30408.0, 'Cosine': 0.24110818}
Images (8, 19): {'Euclidean': 2192.489, 'Manhattan': 43144.0, 'Cosine': 0.30942512}
Images (8, 20): {'Euclidean': 3083.026, 'Manhattan': 63430.0, 'Cosine': 0.61776894}
Images (8, 21): {'Euclidean': 2063.8987, 'Manhattan': 36584.0, 'Cosine': 0.24649185}

Images (8, 22): {'Euclidean': 2117.3809, 'Manhattan': 44810.0, 'Cosine': 0.30198723}
Images (8, 23): {'Euclidean': 3014.7449, 'Manhattan': 66066.0, 'Cosine': 0.62779015}
Images (8, 24): {'Euclidean': 3043.2786, 'Manhattan': 57490.0, 'Cosine': 0.58023006}
Images (8, 25): {'Euclidean': 2501.835, 'Manhattan': 41292.0, 'Cosine': 0.3276536}
Images (8, 26): {'Euclidean': 4701.136, 'Manhattan': 73146.0, 'Cosine': 0.8229915}
Images (8, 27): {'Euclidean': 5127.803, 'Manhattan': 80960.0, 'Cosine': 0.85997474}
Images (8, 28): {'Euclidean': 2134.6362, 'Manhattan': 45044.0, 'Cosine': 0.30382776}
Images (8, 29): {'Euclidean': 2949.1687, 'Manhattan': 62310.0, 'Cosine': 0.5948967}
Images (8, 30): {'Euclidean': 3485.8472, 'Manhattan': 71660.0, 'Cosine': 0.7583636}
Images (8, 31): {'Euclidean': 3618.3235, 'Manhattan': 52210.0, 'Cosine': 0.47776186}
Images (9, 10): {'Euclidean': 4629.5166, 'Manhattan': 79778.0, 'Cosine': 0.88173735}
Images (9, 11): {'Euclidean': 3156.7014, 'Manhattan': 41774.0, 'Cosine': 0.43411028}
Images (9, 12): {'Euclidean': 4377.083, 'Manhattan': 61886.0, 'Cosine': 0.80002093}
Images (9, 13): {'Euclidean': 3489.947, 'Manhattan': 47744.0, 'Cosine': 0.5867516}
Images (9, 14): {'Euclidean': 3628.3945, 'Manhattan': 51376.0, 'Cosine': 0.6318337}
Images (9, 15): {'Euclidean': 3118.943, 'Manhattan': 45432.0, 'Cosine': 0.41118336}
Images (9, 16): {'Euclidean': 3605.15, 'Manhattan': 46246.0, 'Cosine': 0.60916185}
Images (9, 17): {'Euclidean': 3741.9766, 'Manhattan': 43700.0, 'Cosine': 0.5999307}
Images (9, 18): {'Euclidean': 4328.54, 'Manhattan': 70942.0, 'Cosine': 0.83181643}

Images (9, 19): {'Euclidean': 4000.2883, 'Manhattan': 61726.0, 'Cosine': 0.7647997}
Images (9, 20): {'Euclidean': 3637.6582, 'Manhattan': 46148.0, 'Cosine': 0.615077}
Images (9, 21): {'Euclidean': 4082.1367, 'Manhattan': 57686.0, 'Cosine': 0.7286396}
Images (9, 22): {'Euclidean': 3255.2922, 'Manhattan': 37708.0, 'Cosine': 0.49883044}
Images (9, 23): {'Euclidean': 3914.812, 'Manhattan': 53258.0, 'Cosine': 0.7578218}
Images (9, 24): {'Euclidean': 4130.2656, 'Manhattan': 57570.0, 'Cosine': 0.7917797}
Images (9, 25): {'Euclidean': 4575.1685, 'Manhattan': 73536.0, 'Cosine': 0.85907686}
Images (9, 26): {'Euclidean': 5480.71, 'Manhattan': 80318.0, 'Cosine': 0.942769}
Images (9, 27): {'Euclidean': 4764.36, 'Manhattan': 52358.0, 'Cosine': 0.6322529} Images (9, 28): {'Euclidean': 3796.017, 'Manhattan': 57646.0, 'Cosine': 0.7142314}
Images (9, 29): {'Euclidean': 3173.482, 'Manhattan': 37788.0, 'Cosine': 0.45875037}
Images (9, 30): {'Euclidean': 3265.665, 'Manhattan': 35920.0, 'Cosine': 0.4722281}
Images (9, 31): {'Euclidean': 5271.889, 'Manhattan': 75658.0, 'Cosine': 0.8900457}
Images (10, 11): {'Euclidean': 4069.2688, 'Manhattan': 89022.0, 'Cosine': 0.93059826}
Images (10, 12): {'Euclidean': 2932.4536, 'Manhattan': 52674.0, 'Cosine': 0.43046248}
Images (10, 13): {'Euclidean': 3467.8708, 'Manhattan': 74944.0, 'Cosine': 0.7720734}
Images (10, 14): {'Euclidean': 3138.3433, 'Manhattan': 62794.0, 'Cosine': 0.6044569}
Images (10, 15): {'Euclidean': 4321.034, 'Manhattan': 96142.0, 'Cosine': 0.99123335}
Images (10, 16): {'Euclidean': 3243.3499, 'Manhattan': 67392.0, 'Cosine': 0.6284763}

Images (10, 17): {'Euclidean': 3625.7056, 'Manhattan': 63454.0, 'Cosine': 0.688881}
Images (10, 18): {'Euclidean': 2481.234, 'Manhattan': 42762.0, 'Cosine': 0.329404}
Images (10, 19): {'Euclidean': 1988.7404, 'Manhattan': 36670.0, 'Cosine': 0.21765333}
Images (10, 20): {'Euclidean': 3320.311, 'Manhattan': 66052.0, 'Cosine': 0.6499665}
Images (10, 21): {'Euclidean': 3616.5757, 'Manhattan': 69136.0, 'Cosine': 0.6969707}
Images (10, 22): {'Euclidean': 3147.995, 'Manhattan': 68186.0, 'Cosine': 0.6465856}
Images (10, 23): {'Euclidean': 3183.5383, 'Manhattan': 64842.0, 'Cosine': 0.62805486}
Images (10, 24): {'Euclidean': 2574.4336, 'Manhattan': 46634.0, 'Cosine': 0.37212992}
Images (10, 25): {'Euclidean': 2330.2544, 'Manhattan': 37432.0, 'Cosine': 0.26569957}
Images (10, 26): {'Euclidean': 4025.7065, 'Manhattan': 42054.0, 'Cosine': 0.56306833}
Images (10, 27): {'Euclidean': 5493.161, 'Manhattan': 89750.0, 'Cosine': 0.95244396}
Images (10, 28): {'Euclidean': 2415.8254, 'Manhattan': 48584.0, 'Cosine': 0.3433782}
Images (10, 29): {'Euclidean': 3803.09, 'Manhattan': 86980.0, 'Cosine': 0.9086858}
Images (10, 30): {'Euclidean': 3818.6785, 'Manhattan': 79020.0, 'Cosine': 0.83291006}
Images (10, 31): {'Euclidean': 4156.975, 'Manhattan': 65974.0, 'Cosine': 0.61985683}
Images (11, 12): {'Euclidean': 3713.9585, 'Manhattan': 70002.0, 'Cosine': 0.7926738}
Images (11, 13): {'Euclidean': 2053.0518, 'Manhattan': 39162.0, 'Cosine': 0.31863028}
Images (11, 14): {'Euclidean': 2595.228, 'Manhattan': 54998.0, 'Cosine': 0.50281787}
Images (11, 15): {'Euclidean': 1270.0709, 'Manhattan': 22946.0, 'Cosine':

0.09747714}
Images (11, 16): {'Euclidean': 2906.347, 'Manhattan': 61660.0, 'Cosine': 0.6080811}
Images (11, 17): {'Euclidean': 2502.0886, 'Manhattan': 40958.0, 'Cosine': 0.37855244}
Images (11, 18): {'Euclidean': 3674.1382, 'Manhattan': 79562.0, 'Cosine': 0.8496264}
Images (11, 19): {'Euclidean': 3188.6294, 'Manhattan': 69990.0, 'Cosine': 0.72639406}
Images (11, 20): {'Euclidean': 2279.5337, 'Manhattan': 44008.0, 'Cosine': 0.36579794}
Images (11, 21): {'Euclidean': 3397.829, 'Manhattan': 67360.0, 'Cosine': 0.7153319}
Images (11, 22): {'Euclidean': 2270.849, 'Manhattan': 49276.0, 'Cosine': 0.40681577}
Images (11, 23): {'Euclidean': 2920.9084, 'Manhattan': 59996.0, 'Cosine': 0.6463469}
Images (11, 24): {'Euclidean': 3417.6213, 'Manhattan': 68514.0, 'Cosine': 0.79144216}
Images (11, 25): {'Euclidean': 3962.6453, 'Manhattan': 81828.0, 'Cosine': 0.87930894}
Images (11, 26): {'Euclidean': 4946.007, 'Manhattan': 86788.0, 'Cosine': 0.9612534}
Images (11, 27): {'Euclidean': 3941.566, 'Manhattan': 40786.0, 'Cosine': 0.47255486}
Images (11, 28): {'Euclidean': 3239.699, 'Manhattan': 74842.0, 'Cosine': 0.81453115}
Images (11, 29): {'Euclidean': 1746.3494, 'Manhattan': 32310.0, 'Cosine': 0.22197306}
Images (11, 30): {'Euclidean': 1583.4387, 'Manhattan': 28046.0, 'Cosine': 0.16862494}
Images (11, 31): {'Euclidean': 4696.198, 'Manhattan': 80002.0, 'Cosine': 0.88301116}
Images (12, 13): {'Euclidean': 3186.6165, 'Manhattan': 61238.0, 'Cosine': 0.6644294}
Images (12, 14): {'Euclidean': 2688.886, 'Manhattan': 46630.0, 'Cosine': 0.44813728}

Images (12, 15): {'Euclidean': 3980.0723, 'Manhattan': 74618.0, 'Cosine': 0.85966253}
Images (12, 16): {'Euclidean': 2637.8105, 'Manhattan': 40614.0, 'Cosine': 0.41964436}
Images (12, 17): {'Euclidean': 3650.5142, 'Manhattan': 58708.0, 'Cosine': 0.7143079}
Images (12, 18): {'Euclidean': 2716.1528, 'Manhattan': 44256.0, 'Cosine': 0.40611666}
Images (12, 19): {'Euclidean': 2340.1099, 'Manhattan': 42092.0, 'Cosine': 0.3236668}
Images (12, 20): {'Euclidean': 3200.7537, 'Manhattan': 55968.0, 'Cosine': 0.6198238}
Images (12, 21): {'Euclidean': 3146.1135, 'Manhattan': 53006.0, 'Cosine': 0.53923374}
Images (12, 22): {'Euclidean': 2616.8132, 'Manhattan': 46270.0, 'Cosine': 0.43980217}
Images (12, 23): {'Euclidean': 3277.5308, 'Manhattan': 64144.0, 'Cosine': 0.68977594}
Images (12, 24): {'Euclidean': 3215.3518, 'Manhattan': 52632.0, 'Cosine': 0.6053064}
Images (12, 25): {'Euclidean': 2567.103, 'Manhattan': 39664.0, 'Cosine': 0.32895553}
Images (12, 26): {'Euclidean': 4636.1587, 'Manhattan': 66944.0, 'Cosine': 0.7701571}
Images (12, 27): {'Euclidean': 5327.7993, 'Manhattan': 76372.0, 'Cosine': 0.903914}
Images (12, 28): {'Euclidean': 2403.291, 'Manhattan': 44186.0, 'Cosine': 0.35432148}
Images (12, 29): {'Euclidean': 3245.216, 'Manhattan': 59920.0, 'Cosine': 0.6703122}
Images (12, 30): {'Euclidean': 3608.498, 'Manhattan': 64952.0, 'Cosine': 0.76139736}
Images (12, 31): {'Euclidean': 4197.4985, 'Manhattan': 61716.0, 'Cosine': 0.6409745}
Images (13, 14): {'Euclidean': 2258.6333, 'Manhattan': 49544.0, 'Cosine': 0.47168916}
Images (13, 15): {'Euclidean': 2167.2578, 'Manhattan': 45254.0, 'Cosine':

0.31849372}
Images (13, 16): {'Euclidean': 2479.9924, 'Manhattan': 54006.0, 'Cosine': 0.5375451}
Images (13, 17): {'Euclidean': 2302.1025, 'Manhattan': 38574.0, 'Cosine': 0.3551538}
Images (13, 18): {'Euclidean': 2980.535, 'Manhattan': 66546.0, 'Cosine': 0.65171087}
Images (13, 19): {'Euclidean': 2578.3933, 'Manhattan': 57354.0, 'Cosine': 0.5745109}
Images (13, 20): {'Euclidean': 2287.874, 'Manhattan': 45768.0, 'Cosine': 0.44390893}
Images (13, 21): {'Euclidean': 2912.064, 'Manhattan': 56284.0, 'Cosine': 0.60788095}
Images (13, 22): {'Euclidean': 1924.9426, 'Manhattan': 41472.0, 'Cosine': 0.38317746}
Images (13, 23): {'Euclidean': 1875.6093, 'Manhattan': 37758.0, 'Cosine': 0.32975924}
Images (13, 24): {'Euclidean': 2834.2705, 'Manhattan': 56144.0, 'Cosine': 0.6482644}
Images (13, 25): {'Euclidean': 3293.027, 'Manhattan': 67910.0, 'Cosine': 0.68758786}
Images (13, 26): {'Euclidean': 4070.196, 'Manhattan': 66650.0, 'Cosine': 0.66806877}
Images (13, 27): {'Euclidean': 4068.1042, 'Manhattan': 52106.0, 'Cosine': 0.5242312}
Images (13, 28): {'Euclidean': 2384.1494, 'Manhattan': 52478.0, 'Cosine': 0.5476088}
Images (13, 29): {'Euclidean': 1860.6907, 'Manhattan': 38856.0, 'Cosine': 0.32014835}
Images (13, 30): {'Euclidean': 2212.9832, 'Manhattan': 38580.0, 'Cosine': 0.38593197}
Images (13, 31): {'Euclidean': 4138.123, 'Manhattan': 73008.0, 'Cosine': 0.7258345}
Images (14, 15): {'Euclidean': 2908.4966, 'Manhattan': 58370.0, 'Cosine': 0.58188426}
Images (14, 16): {'Euclidean': 1376.9117, 'Manhattan': 26396.0, 'Cosine': 0.15782213}

Images (14, 17): {'Euclidean': 3029.7627, 'Manhattan': 51430.0, 'Cosine': 0.6220802}
Images (14, 18): {'Euclidean': 2239.9995, 'Manhattan': 45614.0, 'Cosine': 0.34752488}
Images (14, 19): {'Euclidean': 2231.751, 'Manhattan': 47900.0, 'Cosine': 0.41188478}
Images (14, 20): {'Euclidean': 2391.6182, 'Manhattan': 50132.0, 'Cosine': 0.46653688}
Images (14, 21): {'Euclidean': 1681.4363, 'Manhattan': 28984.0, 'Cosine': 0.18206233}
Images (14, 22): {'Euclidean': 1125.9565, 'Manhattan': 24128.0, 'Cosine': 0.11878419}
Images (14, 23): {'Euclidean': 2363.6057, 'Manhattan': 51108.0, 'Cosine': 0.49900633}
Images (14, 24): {'Euclidean': 2502.8235, 'Manhattan': 45936.0, 'Cosine': 0.48431456}
Images (14, 25): {'Euclidean': 2862.9443, 'Manhattan': 55282.0, 'Cosine': 0.49367416}
Images (14, 26): {'Euclidean': 4494.925, 'Manhattan': 75756.0, 'Cosine': 0.8385184}
Images (14, 27): {'Euclidean': 4607.8125, 'Manhattan': 67956.0, 'Cosine': 0.7355304}
Images (14, 28): {'Euclidean': 2030.1769, 'Manhattan': 45356.0, 'Cosine': 0.37695158}
Images (14, 29): {'Euclidean': 1865.5509, 'Manhattan': 39410.0, 'Cosine': 0.30722523}
Images (14, 30): {'Euclidean': 2548.3115, 'Manhattan': 48774.0, 'Cosine': 0.49818206}
Images (14, 31): {'Euclidean': 3795.6375, 'Manhattan': 60420.0, 'Cosine': 0.573967}
Images (15, 16): {'Euclidean': 3117.431, 'Manhattan': 62286.0, 'Cosine': 0.64743376}
Images (15, 17): {'Euclidean': 2471.8171, 'Manhattan': 44596.0, 'Cosine': 0.3488413}
Images (15, 18): {'Euclidean': 4002.9402, 'Manhattan': 88754.0, 'Cosine': 0.94577456}
Images (15, 19): {'Euclidean': 3474.0535, 'Manhattan': 78540.0, 'Cosine':

0.8005284}
Images (15, 20): {'Euclidean': 2435.2039, 'Manhattan': 48064.0, 'Cosine': 0.38439667}
Images (15, 21): {'Euclidean': 3587.5293, 'Manhattan': 72322.0, 'Cosine': 0.74893504}
Images (15, 22): {'Euclidean': 2473.1736, 'Manhattan': 50790.0, 'Cosine': 0.43651527}
Images (15, 23): {'Euclidean': 3134.4395, 'Manhattan': 68286.0, 'Cosine': 0.68578744}
Images (15, 24): {'Euclidean': 3636.3718, 'Manhattan': 74014.0, 'Cosine': 0.83551615}
Images (15, 25): {'Euclidean': 4269.8315, 'Manhattan': 90580.0, 'Cosine': 0.9651731}
Images (15, 26): {'Euclidean': 5111.0537, 'Manhattan': 94788.0, 'Cosine': 0.98804355}
Images (15, 27): {'Euclidean': 3865.8862, 'Manhattan': 41126.0, 'Cosine': 0.44525826}
Images (15, 28): {'Euclidean': 3456.8252, 'Manhattan': 80372.0, 'Cosine': 0.8557493}
Images (15, 29): {'Euclidean': 1714.7961, 'Manhattan': 31270.0, 'Cosine': 0.18772542}
Images (15, 30): {'Euclidean': 1853.6661, 'Manhattan': 34106.0, 'Cosine': 0.21303737}
Images (15, 31): {'Euclidean': 4975.861, 'Manhattan': 88808.0, 'Cosine': 0.95812535}
Images (16, 17): {'Euclidean': 3166.245, 'Manhattan': 52990.0, 'Cosine': 0.6573682}
Images (16, 18): {'Euclidean': 2525.4353, 'Manhattan': 48744.0, 'Cosine': 0.43238854}
Images (16, 19): {'Euclidean': 2460.4622, 'Manhattan': 50678.0, 'Cosine': 0.4783665}
Images (16, 20): {'Euclidean': 2413.5522, 'Manhattan': 48532.0, 'Cosine': 0.45436758}
Images (16, 21): {'Euclidean': 1664.0896, 'Manhattan': 30510.0, 'Cosine': 0.17724097}
Images (16, 22): {'Euclidean': 1181.3204, 'Manhattan': 23628.0, 'Cosine': 0.117762804}

Images (16, 23): {'Euclidean': 2565.9412, 'Manhattan': 55056.0, 'Cosine': 0.5576949}
Images (16, 24): {'Euclidean': 2751.9316, 'Manhattan': 50402.0, 'Cosine': 0.5627589}
Images (16, 25): {'Euclidean': 3035.4688, 'Manhattan': 56590.0, 'Cosine': 0.5447461}
Images (16, 26): {'Euclidean': 4535.6953, 'Manhattan': 74518.0, 'Cosine': 0.83570415}
Images (16, 27): {'Euclidean': 4818.512, 'Manhattan': 69768.0, 'Cosine': 0.8085482}
Images (16, 28): {'Euclidean': 1858.092, 'Manhattan': 40624.0, 'Cosine': 0.2971362}
Images (16, 29): {'Euclidean': 1957.4769, 'Manhattan': 39456.0, 'Cosine': 0.3205483}
Images (16, 30): {'Euclidean': 2746.2415, 'Manhattan': 54362.0, 'Cosine': 0.55625284}
Images (16, 31): {'Euclidean': 3944.949, 'Manhattan': 63458.0, 'Cosine': 0.62179774}
Images (17, 18): {'Euclidean': 3403.873, 'Manhattan': 59156.0, 'Cosine': 0.6746981}
Images (17, 19): {'Euclidean': 3064.0898, 'Manhattan': 47062.0, 'Cosine': 0.6100645}
Images (17, 20): {'Euclidean': 3024.2976, 'Manhattan': 41922.0, 'Cosine': 0.58840024}
Images (17, 21): {'Euclidean': 3494.4233, 'Manhattan': 55118.0, 'Cosine': 0.70129144}
Images (17, 22): {'Euclidean': 2720.0613, 'Manhattan': 40358.0, 'Cosine': 0.5272474}
Images (17, 23): {'Euclidean': 3158.1768, 'Manhattan': 42700.0, 'Cosine': 0.68433976}
Images (17, 24): {'Euclidean': 3276.6921, 'Manhattan': 47160.0, 'Cosine': 0.66754603}
Images (17, 25): {'Euclidean': 3671.3408, 'Manhattan': 62156.0, 'Cosine': 0.70445085}
Images (17, 26): {'Euclidean': 4718.4136, 'Manhattan': 65262.0, 'Cosine': 0.82550895}
Images (17, 27): {'Euclidean': 4156.4717, 'Manhattan': 47230.0, 'Cosine':

0.5287998}
Images (17, 28): {'Euclidean': 3082.5374, 'Manhattan': 52096.0, 'Cosine': 0.66168976}
Images (17, 29): {'Euclidean': 2759.7837, 'Manhattan': 43946.0, 'Cosine': 0.5122415}
Images (17, 30): {'Euclidean': 2661.707, 'Manhattan': 33506.0, 'Cosine': 0.4363172}
Images (17, 31): {'Euclidean': 4540.296, 'Manhattan': 70626.0, 'Cosine': 0.7813821}
Images (18, 19): {'Euclidean': 1375.3073, 'Manhattan': 24988.0, 'Cosine': 0.12131542}
Images (18, 20): {'Euclidean': 3007.715, 'Manhattan': 58500.0, 'Cosine': 0.6063659}
Images (18, 21): {'Euclidean': 2749.4014, 'Manhattan': 50434.0, 'Cosine': 0.4488051}
Images (18, 22): {'Euclidean': 2406.3484, 'Manhattan': 52728.0, 'Cosine': 0.42641944}
Images (18, 23): {'Euclidean': 2874.09, 'Manhattan': 61150.0, 'Cosine': 0.58922327}
Images (18, 24): {'Euclidean': 2419.0527, 'Manhattan': 43664.0, 'Cosine': 0.3766333}
Images (18, 25): {'Euclidean': 1674.6528, 'Manhattan': 26312.0, 'Cosine': 0.14653468}
Images (18, 26): {'Euclidean': 4311.072, 'Manhattan': 57708.0, 'Cosine': 0.69167036}
Images (18, 27): {'Euclidean': 5215.513, 'Manhattan': 83556.0, 'Cosine': 0.90678656}
Images (18, 28): {'Euclidean': 2174.7712, 'Manhattan': 43192.0, 'Cosine': 0.3319394}
Images (18, 29): {'Euclidean': 3198.4536, 'Manhattan': 72524.0, 'Cosine': 0.7280865}
Images (18, 30): {'Euclidean': 3508.3386, 'Manhattan': 73444.0, 'Cosine': 0.7903552}
Images (18, 31): {'Euclidean': 2862.03, 'Manhattan': 41152.0, 'Cosine': 0.2770183}
Images (19, 20): {'Euclidean': 2422.1223, 'Manhattan': 45522.0, 'Cosine': 0.4531864}

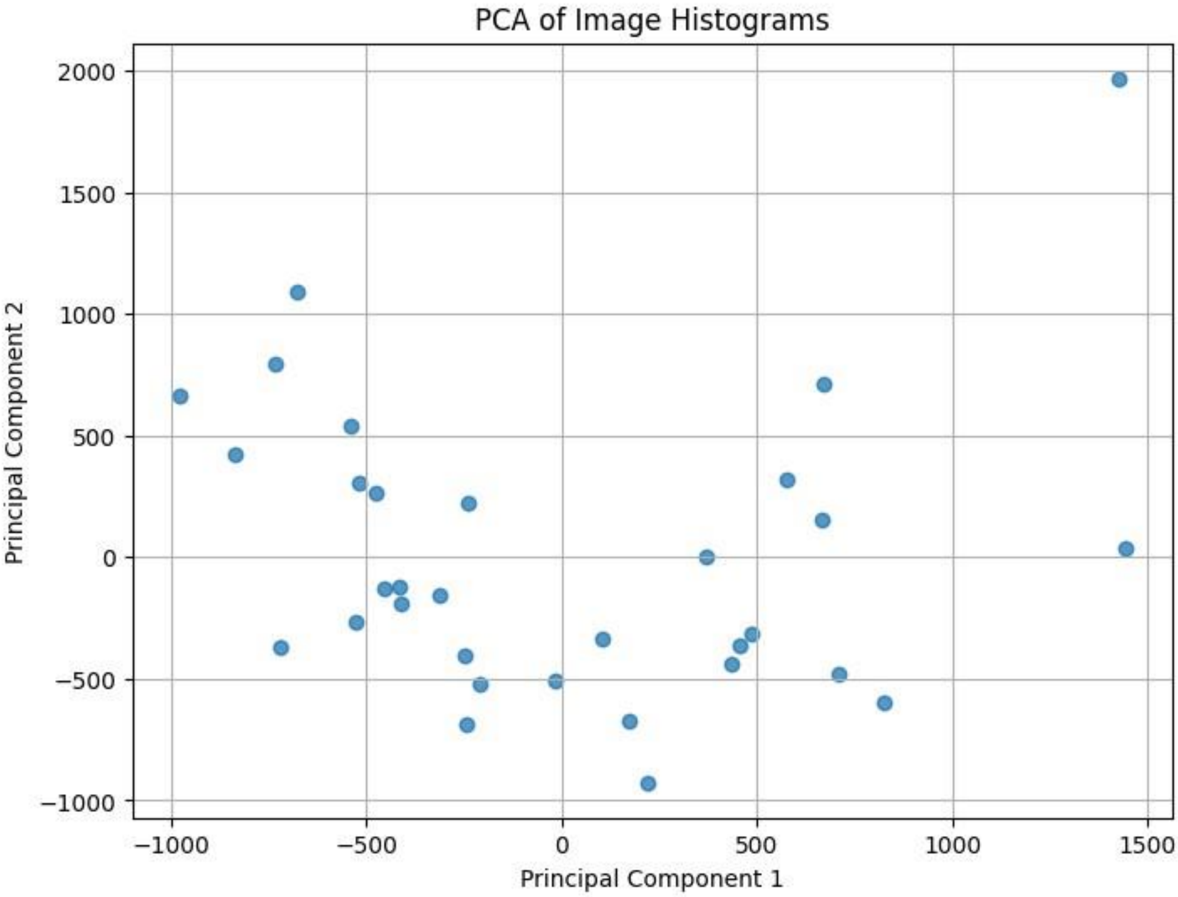
Images (19, 21): {'Euclidean': 2869.4617, 'Manhattan': 55080.0, 'Cosine': 0.5466618}
Images (19, 22): {'Euclidean': 2213.1553, 'Manhattan': 49302.0, 'Cosine': 0.44097453}
Images (19, 23): {'Euclidean': 2566.2163, 'Manhattan': 51064.0, 'Cosine': 0.55151534}
Images (19, 24): {'Euclidean': 2204.166, 'Manhattan': 40152.0, 'Cosine': 0.35719997}
Images (19, 25): {'Euclidean': 1352.5295, 'Manhattan': 24450.0, 'Cosine': 0.085092545}
Images (19, 26): {'Euclidean': 4065.847, 'Manhattan': 50070.0, 'Cosine': 0.64178646}
Images (19, 27): {'Euclidean': 4883.876, 'Manhattan': 71228.0, 'Cosine': 0.8321354}
Images (19, 28): {'Euclidean': 1795.2036, 'Manhattan': 37966.0, 'Cosine': 0.27334332}
Images (19, 29): {'Euclidean': 2876.6077, 'Manhattan': 67432.0, 'Cosine': 0.6862135}
Images (19, 30): {'Euclidean': 3041.815, 'Manhattan': 61904.0, 'Cosine': 0.6768765}
Images (19, 31): {'Euclidean': 3053.1577, 'Manhattan': 48340.0, 'Cosine': 0.32131565}
Images (20, 21): {'Euclidean': 3125.0706, 'Manhattan': 58516.0, 'Cosine': 0.64387494}
Images (20, 22): {'Euclidean': 2158.867, 'Manhattan': 44032.0, 'Cosine': 0.41119283}
Images (20, 23): {'Euclidean': 2491.471, 'Manhattan': 44318.0, 'Cosine': 0.51221406}
Images (20, 24): {'Euclidean': 2864.4238, 'Manhattan': 48266.0, 'Cosine': 0.5974923}
Images (20, 25): {'Euclidean': 3211.1624, 'Manhattan': 59114.0, 'Cosine': 0.6042227}
Images (20, 26): {'Euclidean': 4465.4663, 'Manhattan': 66224.0, 'Cosine': 0.7966759}
Images (20, 27): {'Euclidean': 4672.066, 'Manhattan': 60086.0, 'Cosine': 0.74189407}
Images (20, 28): {'Euclidean': 2452.6003, 'Manhattan': 47684.0, 'Cosine':

0.50661635}
Images (20, 29): {'Euclidean': 2015.8125, 'Manhattan': 43948.0, 'Cosine': 0.3309847}
Images (20, 30): {'Euclidean': 2343.7231, 'Manhattan': 42128.0, 'Cosine': 0.39671928}
Images (20, 31): {'Euclidean': 4224.7935, 'Manhattan': 69270.0, 'Cosine': 0.7244612}
Images (21, 22): {'Euclidean': 1953.8102, 'Manhattan': 34050.0, 'Cosine': 0.25609374}
Images (21, 23): {'Euclidean': 2902.0571, 'Manhattan': 53582.0, 'Cosine': 0.58919066}
Images (21, 24): {'Euclidean': 3032.0808, 'Manhattan': 51748.0, 'Cosine': 0.58355004}
Images (21, 25): {'Euclidean': 3373.5815, 'Manhattan': 61068.0, 'Cosine': 0.60420966}
Images (21, 26): {'Euclidean': 4805.41, 'Manhattan': 73960.0, 'Cosine': 0.8690965}
Images (21, 27): {'Euclidean': 4940.2285, 'Manhattan': 73472.0, 'Cosine': 0.7957158}
Images (21, 28): {'Euclidean': 2338.7192, 'Manhattan': 48854.0, 'Cosine': 0.37862557}
Images (21, 29): {'Euclidean': 2507.716, 'Manhattan': 49374.0, 'Cosine': 0.4310236}
Images (21, 30): {'Euclidean': 3233.9158, 'Manhattan': 59522.0, 'Cosine': 0.6605123}
Images (21, 31): {'Euclidean': 4155.053, 'Manhattan': 64040.0, 'Cosine': 0.6523162}
Images (22, 23): {'Euclidean': 2230.5981, 'Manhattan': 46178.0, 'Cosine': 0.49334317}
Images (22, 24): {'Euclidean': 2397.8691, 'Manhattan': 44444.0, 'Cosine': 0.4786229}
Images (22, 25): {'Euclidean': 2927.3228, 'Manhattan': 58408.0, 'Cosine': 0.5450809}
Images (22, 26): {'Euclidean': 4362.8145, 'Manhattan': 71924.0, 'Cosine': 0.81844413}
Images (22, 27): {'Euclidean': 4425.0894, 'Manhattan': 60174.0, 'Cosine': 0.67852694}

Images (22, 28): {'Euclidean': 1819.8434, 'Manhattan': 42812.0, 'Cosine': 0.3374588}
Images (22, 29): {'Euclidean': 1327.8344, 'Manhattan': 28050.0, 'Cosine': 0.16810566}
Images (22, 30): {'Euclidean': 2200.015, 'Manhattan': 40144.0, 'Cosine': 0.39425266}
Images (22, 31): {'Euclidean': 3861.2004, 'Manhattan': 62996.0, 'Cosine': 0.6168684}
Images (23, 24): {'Euclidean': 2588.3337, 'Manhattan': 45990.0, 'Cosine': 0.5237572}
Images (23, 25): {'Euclidean': 3224.3174, 'Manhattan': 64800.0, 'Cosine': 0.6429625}
Images (23, 26): {'Euclidean': 3435.0276, 'Manhattan': 50426.0, 'Cosine': 0.4196993}
Images (23, 27): {'Euclidean': 4703.3467, 'Manhattan': 65078.0, 'Cosine': 0.7797121}
Images (23, 28): {'Euclidean': 2163.033, 'Manhattan': 44602.0, 'Cosine': 0.43407965}
Images (23, 29): {'Euclidean': 2439.019, 'Manhattan': 53636.0, 'Cosine': 0.53199774}
Images (23, 30): {'Euclidean': 2861.7876, 'Manhattan': 52014.0, 'Cosine': 0.6369977}
Images (23, 31): {'Euclidean': 4179.7275, 'Manhattan': 70840.0, 'Cosine': 0.73331565}
Images (24, 25): {'Euclidean': 2856.295, 'Manhattan': 49684.0, 'Cosine': 0.46002567}
Images (24, 26): {'Euclidean': 4140.062, 'Manhattan': 52984.0, 'Cosine': 0.6542809}
Images (24, 27): {'Euclidean': 5160.528, 'Manhattan': 78122.0, 'Cosine': 0.92227244}
Images (24, 28): {'Euclidean': 1933.1534, 'Manhattan': 32426.0, 'Cosine': 0.29239058}
Images (24, 29): {'Euclidean': 2982.5693, 'Manhattan': 62130.0, 'Cosine': 0.6913474}
Images (24, 30): {'Euclidean': 3192.7314, 'Manhattan': 57270.0, 'Cosine': 0.7061615}
Images (24, 31): {'Euclidean': 3694.3025, 'Manhattan': 59220.0, 'Cosine':

0.5192515}
Images (25, 26): {'Euclidean': 4450.2256, 'Manhattan': 55906.0, 'Cosine': 0.6955904}
Images (25, 27): {'Euclidean': 5406.5405, 'Manhattan': 82512.0, 'Cosine': 0.9189471}
Images (25, 28): {'Euclidean': 2497.2786, 'Manhattan': 44730.0, 'Cosine': 0.36882383}
Images (25, 29): {'Euclidean': 3642.2993, 'Manhattan': 78016.0, 'Cosine': 0.8269383}
Images (25, 30): {'Euclidean': 3795.966, 'Manhattan': 76076.0, 'Cosine': 0.82034886}
Images (25, 31): {'Euclidean': 3037.2756, 'Manhattan': 42356.0, 'Cosine': 0.31487715}
Images (26, 27): {'Euclidean': 6028.8496, 'Manhattan': 85678.0, 'Cosine': 0.9278016}
Images (26, 28): {'Euclidean': 4121.448, 'Manhattan': 55404.0, 'Cosine': 0.68738985}
Images (26, 29): {'Euclidean': 4646.285, 'Manhattan': 84714.0, 'Cosine': 0.9079743}
Images (26, 30): {'Euclidean': 4785.735, 'Manhattan': 79810.0, 'Cosine': 0.90592605}
Images (26, 31): {'Euclidean': 5240.5454, 'Manhattan': 72922.0, 'Cosine': 0.78661585}
Images (27, 28): {'Euclidean': 4889.6294, 'Manhattan': 77536.0, 'Cosine': 0.87128586}
Images (27, 29): {'Euclidean': 4071.5151, 'Manhattan': 46514.0, 'Cosine': 0.52423733}
Images (27, 30): {'Euclidean': 4144.6807, 'Manhattan': 46690.0, 'Cosine': 0.5397894}
Images (27, 31): {'Euclidean': 6032.643, 'Manhattan': 81578.0, 'Cosine': 0.94439805}
Images (28, 29): {'Euclidean': 2574.736, 'Manhattan': 62932.0, 'Cosine': 0.60749936}
Images (28, 30): {'Euclidean': 2948.8228, 'Manhattan': 61410.0, 'Cosine': 0.6907653}
Images (28, 31): {'Euclidean': 3806.3484, 'Manhattan': 63772.0, 'Cosine': 0.5836951}

Images (29, 30): {'Euclidean': 1842.8809, 'Manhattan': 34418.0, 'Cosine': 0.25703955}
Images (29, 31): {'Euclidean': 4433.9263, 'Manhattan': 76704.0, 'Cosine': 0.84089255}
Images (30, 31): {'Euclidean': 4567.0713, 'Manhattan': 77030.0, 'Cosine': 0.8412553}



(b) Feature Extraction: Edge histogram AND Similarity Measurements

i. Choose 1 image from each class. ii. Convert the color images to grayscale images.

```
import os
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from sklearn.decomposition import PCA
from sklearn.metrics import pairwise
import cv2
import zipfile

images_zip_path = 'Image dataset.zip'
images_folder_path = 'Breeds'
cropped_images_path = 'sample_data/Cropped_Images'

def unzip_files(zip_file_path, extract_path):
    with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
        zip_ref.extractall(extract_path)

# Unzip the images
unzip_files(images_zip_path, images_folder_path)

def load_images(folder):
    images = []
    for dirpath, _, files in os.walk(folder):
        for filename in files:
            if filename.endswith(('.png', '.jpg', '.jpeg')):
                img_path = os.path.join(dirpath, filename)
                img = Image.open(img_path).convert('RGB')
                images.append(np.array(img))
    return images


# Function to crop and resize images
def crop_and_resize_images(images, size=(128, 128)):
    cropped_resized_images = []
    for img in images:

        width, height = img.shape[1], img.shape[0]
```

```
        left = (width - size[0]) / 2
        top = (height - size[1]) / 2
        right = (width + size[0]) / 2
        bottom = (height + size[1]) / 2

        cropped_img = img[int(top):int(bottom), int(left):int(right)]
        resized_img = cv2.resize(cropped_img, size)
        cropped_resized_images.append(resized_img)
    return cropped_resized_images


def compute_histograms(images):
    histograms = []
    for img in images:
        if len(img.shape) == 2:
            hist = cv2.calcHist([img], [0], None, [256], [0, 256])
        else:
            hist = cv2.calcHist([img], [0, 1, 2], None, [256, 256, 256], [0, 256, 0, 256,
0, 256])
        histograms.append(hist.flatten())
    return histograms

def compute_similarity_measurements(histograms):
    for i in range(len(histograms)):
        for j in range(i + 1, len(histograms)):
            euclidean_distance = np.linalg.norm(histograms[i] - histograms[j])
            manhattan_distance = np.sum(np.abs(histograms[i] - histograms[j]))
            cosine_distance = 1 - pairwise.cosine_similarity([histograms[i]],
[histograms[j]])[0][0]
            print(f'Images ({i}, {j}): {{\'Euclidean\': {euclidean_distance},
\'Manhattan\': {manhattan_distance}, \'Cosine\': {cosine_distance}}}')


def perform_pca(histograms):
    pca = PCA(n_components=2)
    reduced_data = pca.fit_transform(histograms)

    plt.figure(figsize=(20,15))
    plt.scatter(reduced_data[:, 0], reduced_data[:, 1], alpha=0.5)
    plt.title('PCA of Image Histograms')
```

```
    plt.xlabel('Principal Component 1')
    plt.ylabel('Principal Component 2')
    plt.grid()
    plt.show()

def plot_images(images, titles=None, cols=4):
    n_images = len(images)
    rows = (n_images + cols - 1) // cols
    plt.figure(figsize=(250, 250))
    for i in range(n_images):
        plt.subplot(rows, cols, i + 1)
        plt.imshow(images[i].astype(np.uint8))
        plt.axis('off')
        if titles is not None:
            plt.title(titles[i], fontsize=2)
    plt.show()

if __name__ == "__main__":

    dog_images = load_images(images_folder_path)

    plot_images(dog_images, titles=[f'Image {i+1}' for i in range(len(dog_images))])

    cropped_resized_images = crop_and_resize_images(dog_images)

    histograms = compute_histograms(cropped_resized_images)

    if histograms:
        compute_similarity_measurements(histograms)

        perform_pca(histograms)
```

```
plot_images(cropped_resized_images, titles=[f'Cropped Image {i+1}' for i in range(len(cropped_resized_images))])
```

3. Next, we perform some text processing steps on a tweet (i.e., text) dataset. The dataset file is in json format and each dataset consists of

- **Training Set: 3,000 records**
- **Test Set: 1,500 records • Validation Set: 400 records**

```
import json

file_path =' train.json' data = []
with open(file_path, 'r') as file:
for line in file:
    data.append(json.loads(line))

print(json.dumps(data[0], indent=4))
```

<u>OUTPUT:</u>

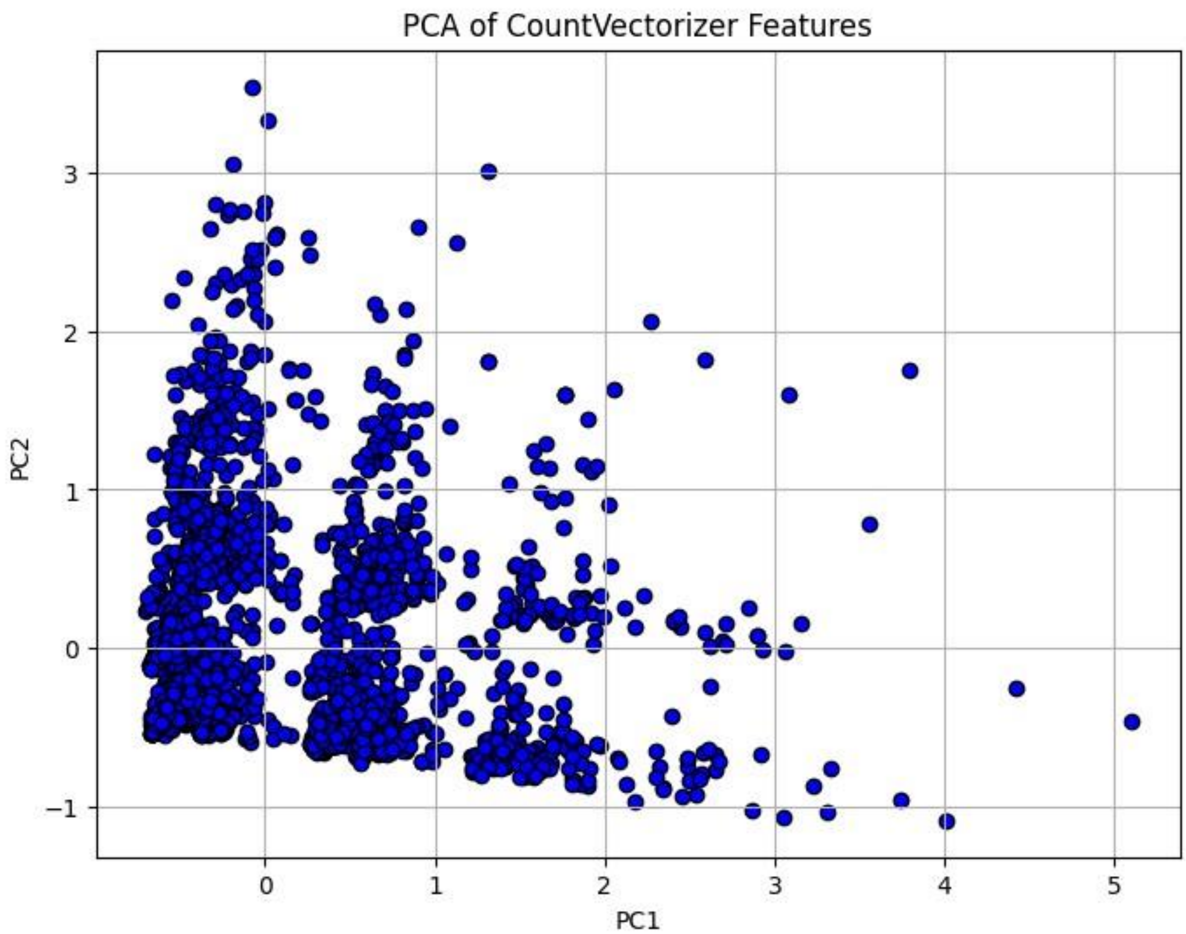

```
{
   "ID": "2017-En-21529",
   "Tweet": "Follow this amazing Australian author @KristyBerridge #fiction
#zombies #angels #demons #vampires #werewolves #follow #authorlove",
   "anger": false,
   "anticipation": true,
   "disgust": false,
   "fear": false,
   "joy": true,
   "love": true,
   "optimism": true,
   "pessimism": false,
   "sadness": false,
   "surprise": false,
   "trust": true
}
```

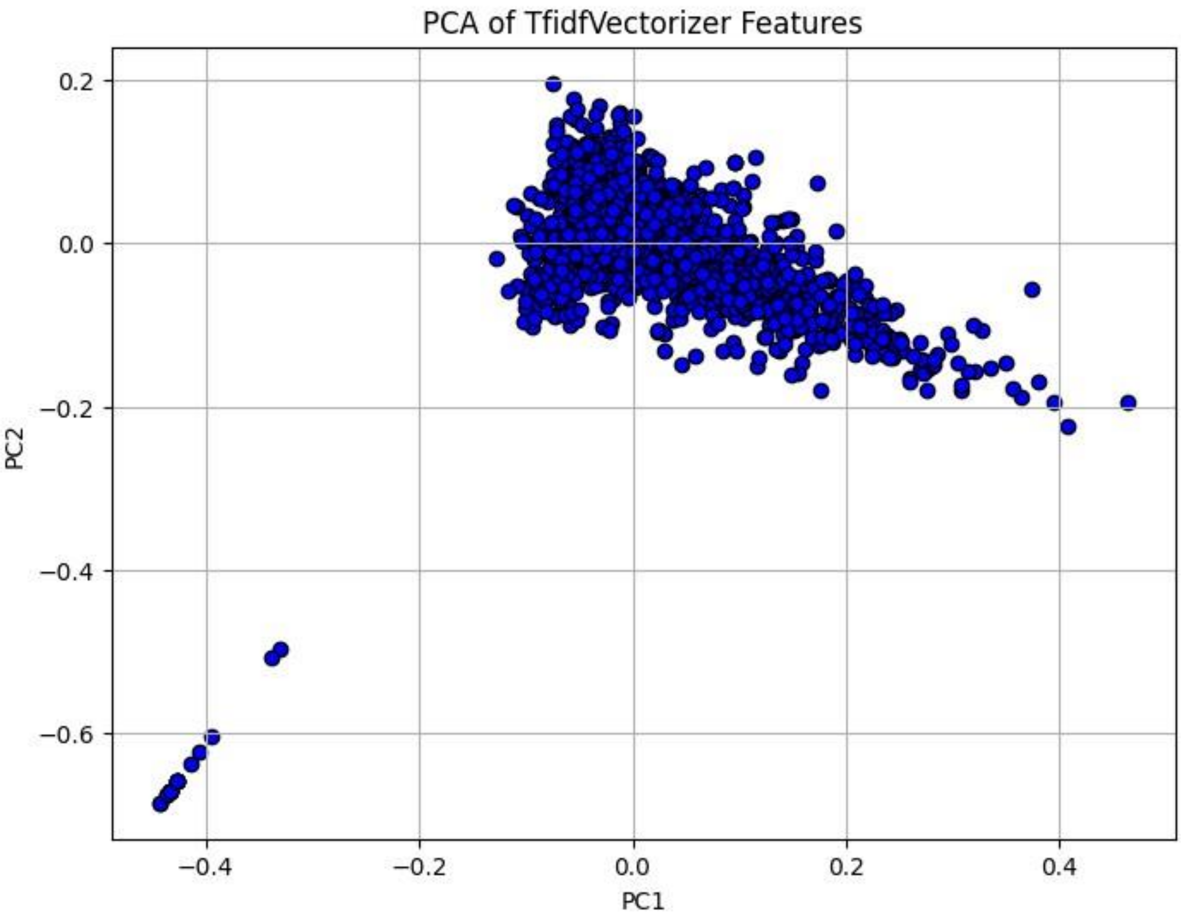
4. You will use the simple countvectorizer and tfidfvectorizer in https://scikitlearn.org/stable/api/sklearn.feature_extraction.html#modulesklearn.feature_extraction.text to extract (1) token (feature) counts, and (2) TF-IDF feature (counts), respectively

```
import json
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

file_path = 'train.json'
data = []
with open(file_path, 'r') as file:
    for line in file:
        data.append(json.loads(line))

texts = [entry['Tweet'] for entry in data]

count_vectorizer = CountVectorizer()
count_vectors = count_vectorizer.fit_transform(texts)

tfidf_vectorizer = TfidfVectorizer()
tfidf_vectors = tfidf_vectorizer.fit_transform(texts)

pca = PCA(n_components=2)
count_pca = pca.fit_transform(count_vectors.toarray())
tfidf_pca = pca.fit_transform(tfidf_vectors.toarray())

def plot_pca(pca_result, title):
    plt.figure(figsize=(8, 6))
    plt.scatter(pca_result[:, 0], pca_result[:, 1], c='blue', marker='o',
edgecolor='k')
    plt.title(title)
    plt.xlabel('PC1')
    plt.ylabel('PC2')
    plt.grid(True)
```

```
    plt.show()

plot_pca(count_pca, 'PCA of CountVectorizer Features')

plot_pca(tfidf_pca, 'PCA of TfidfVectorizer Features')

count_dimensionality = count_vectors.shape
tfidf_dimensionality = tfidf_vectors.shape

print(f"Dimensionality of CountVectorizer representation:
{count_dimensionality}")
print(f"Dimensionality of TfidfVectorizer representation:
{tfidf_dimensionality}")
```

OUTPUT:





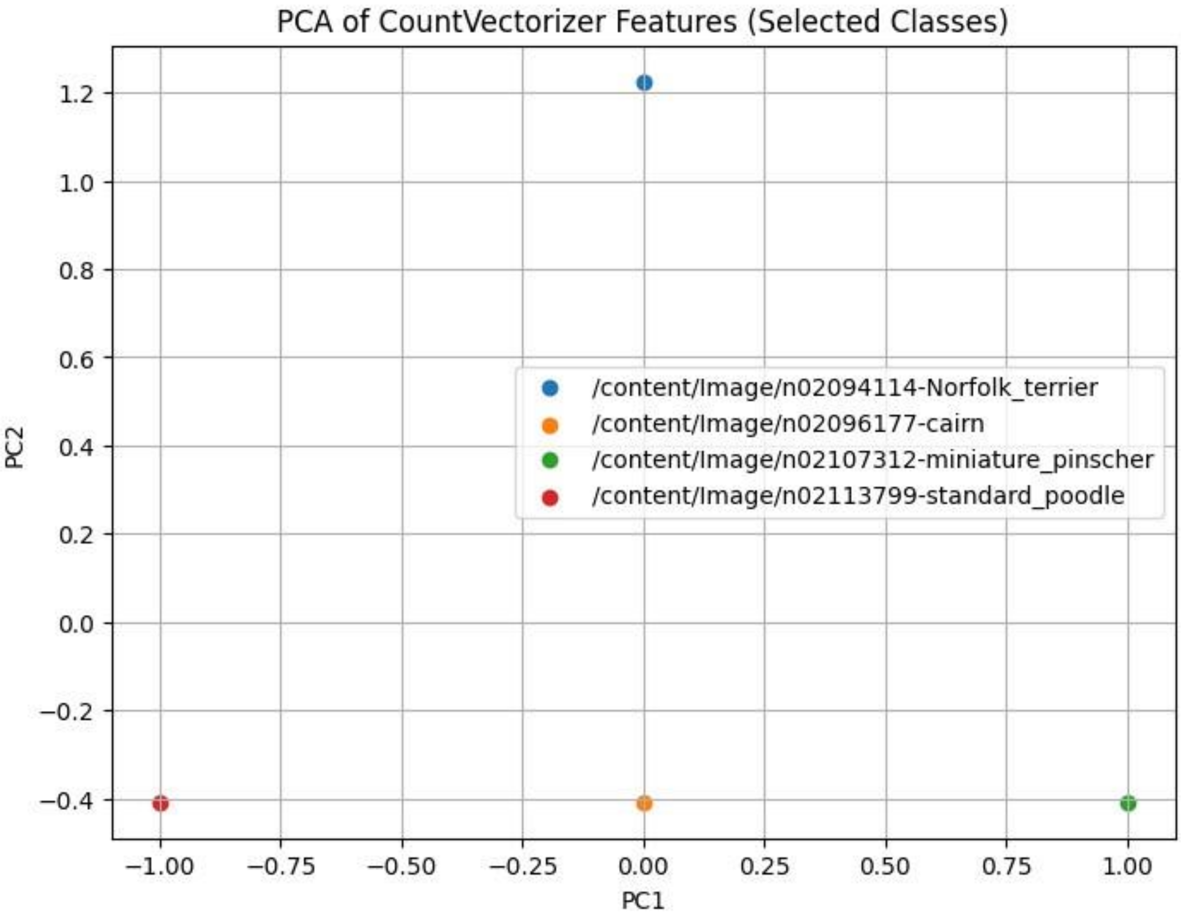
5. Using the two sets of processed text data in Item 4,

- **Pick four classes which you think will be separable. State the four classes.**
- **Perform dimensionality reduction similar to 2(d) with reduced to**
- **Plot the 2D points using four different colors for data from the four classes for both token count features and tf-idf features in two separate plots.**
- **How many classes are visually separable (i.e., non-overlapping) for both plots?**

```
# Import necessary libraries
import os
import tarfile
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
from skimage import filters, exposure
from skimage.feature import hog
```

```
from sklearn.decomposition import PCA
from sklearn.metrics import pairwise
from sklearn.feature_extraction.text import CountVectorizer


data = [
   {'Tweet': 'Image/n02094114-Norfolk_terrier', 'Class': 'Image/n02094114-
Norfolk_terrier'},
   {'Tweet': 'Image/n02096177-cairn', 'Class': 'Image/n02096177-cairn'},
   {'Tweet': 'Image/n02107312-miniature_pinscher', 'Class': 'Image/n02107312-
miniature_pinscher'},
   {'Tweet': 'Image/n02113799-standard_poodle', 'Class': 'Image/n02113799-
standard_poodle'},
]

selected_classes = ['Image/n02094114-Norfolk_terrier', 'Image/n02096177-
cairn', 'Image/n02107312-miniature_pinscher', 'Image/n02113799-
standard_poodle']  # Replace with your actual class labels

filtered_data = [entry for entry in data if entry['Class'] in selected_classes]

filtered_texts = [entry['Tweet'] for entry in filtered_data]
filtered_classes = [entry['Class'] for entry in filtered_data]


count_vectorizer = CountVectorizer()
count_vectors = count_vectorizer.fit_transform(filtered_texts)

count_vectors_filtered = count_vectorizer.transform(filtered_texts)
pca_count_filtered =
PCA(n_components=2).fit_transform(count_vectors_filtered.toarray())


plt.figure(figsize=(8, 6))
for class_label in selected_classes:
  indices = [i for i, cls in enumerate(filtered_classes) if cls == class_label]
```

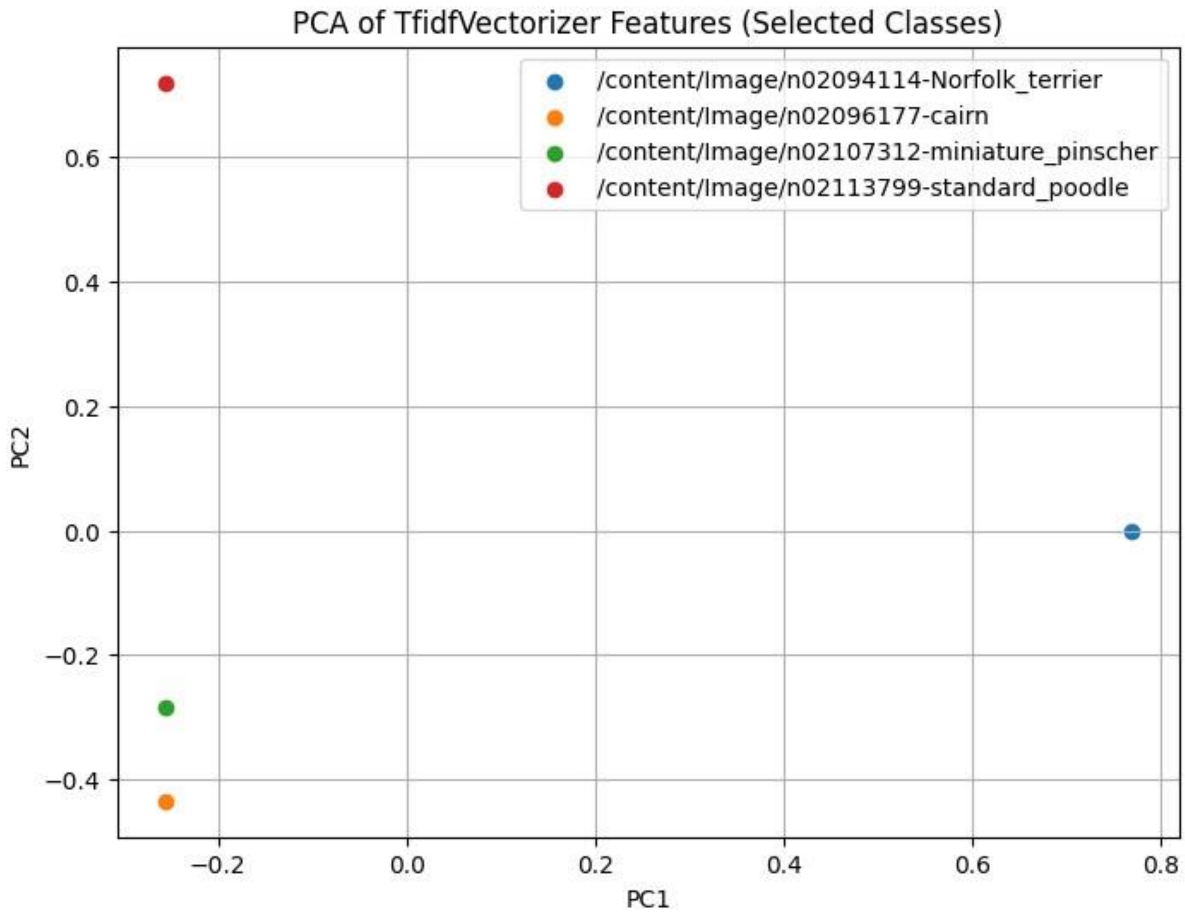
```
  plt.scatter(pca_count_filtered[indices, 0], pca_count_filtered[indices, 1],
label=class_label)

plt.title('PCA of CountVectorizer Features (Selected Classes)')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()
plt.grid(True)
plt.show()
```




```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

data = [
  {'Tweet': 'Image/n02094114-Norfolk_terrier', 'Class': 'Image/n02094114-Norfolk_terrier'},
  {'Tweet': 'Image/n02096177-cairn', 'Class': 'Image/n02096177-cairn'},
```

```
  {'Tweet': 'Image/n02107312-miniature_pinscher', 'Class': 'Image/n02107312-miniature_pinscher'},
  {'Tweet': 'Image/n02113799-standard_poodle', 'Class': 'Image/n02113799-standard_poodle'},
]

selected_classes = ['Image/n02094114-Norfolk_terrier', 'Image/n02096177-cairn', 'Image/n02107312-miniature_pinscher', 'Image/n02113799-standard_poodle']  # Replace with your actual class labels


filtered_data = [entry for entry in data if entry['Class'] in selected_classes]


filtered_texts = [entry['Tweet'] for entry in filtered_data]
filtered_classes = [entry['Class'] for entry in filtered_data]

tfidf_vectorizer = TfidfVectorizer()
tfidf_vectors = tfidf_vectorizer.fit_transform(filtered_texts)


tfidf_vectors_filtered = tfidf_vectorizer.transform(filtered_texts)
pca_tfidf_filtered = PCA(n_components=2).fit_transform(tfidf_vectors_filtered.toarray())


plt.figure(figsize=(8, 6))
for class_label in selected_classes:
  indices = [i for i, cls in enumerate(filtered_classes) if cls == class_label]
  plt.scatter(pca_tfidf_filtered[indices, 0], pca_tfidf_filtered[indices, 1], label=class_label)

plt.title('PCA of TfidfVectorizer Features (Selected Classes)')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()
plt.grid(True)
```

plt.show()


PCA of TfidfVectorizer Features (Selected Classes)
/content/Image/n02094114-Norfolk_terrier
/content/Image/n02096177-cairn
/content/Image/n02107312-miniature_pinscher
/content/Image/n02113799-standard_poodle
0.6
0.4
0.2
0.0
−0.2
−0.4
PC2
−0.2
0.0
0.2
0.4
0.6
0.8
PC1