Importing all required libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import cv2
import os
import xml.etree.ElementTree as ET
from PIL import Image
from pathlib import Path
import random
import warnings
warnings.filterwarnings("ignore")

img_dir = r'C:\Users\V Varunkumar\Desktop\dm programming\images'
annotation_dir = r'C:\Users\V Varunkumar\Desktop\dm programming\
annotation'

# Function to List all the directories inside the path
def list_directories (path):
    return [d for d in os.listdir(path) if
os.path.isdir(os.path.join(path, d))]

images_subdirs = list_directories (img_dir)
annotations_subdirs = list_directories (annotation_dir)
print("Directories in Images folder:", images_subdirs)
print("\nDirectories in Annotations folder:", annotations_subdirs)

Directories in Images folder: ['n02091831-Saluki', 'n02093859-
Kerry_blue_terrier', 'n02108551-Tibetan_mastiff', 'n02111277-
Newfoundland']

Directories in Annotations folder: ['n02091831-Saluki', 'n02093859-
Kerry_blue_terrier', 'n02108551-Tibetan_mastiff', 'n02111277-
Newfoundland']
```

(a) Cropping and Resize Images in Your 4-class Images Dataset

```
def get_bounding_boxes (annot_path):
    tree = ET.parse(annot_path)
    root = tree.getroot()
    objects = root.findall('object')
    bbox = []
    for o in objects:
        bndbox = o.find('bndbox')
        xmin = int(bndbox.find('xmin').text)
        ymin = int(bndbox.find('ymin').text)
        xmax = int(bndbox.find('xmax').text)
```

```
        ymax = int(bndbox.find('ymax').text)
        bbox.append((xmin, ymin, xmax, ymax))
    return bbox

for subdir in images_subdirs:
    #Path to the subdirectories of image and annotation
    img_subdir_path = img_dir + "\\" + subdir
    annot_subdir_path = annotation_dir +"\\"+ subdir
    # Getting all xml files in the annotation subdirectory
    images = [img_subdir_path +"\\" + f for f in
os.listdir(img_subdir_path)]
    annotations = [annot_subdir_path +"\\" + f for f in
os.listdir(annot_subdir_path)]
    for i, annot in enumerate(annotations):
        bbox = get_bounding_boxes(annot)
        dog_image_path = images[i]
        im = Image.open(dog_image_path)
        for j, box in enumerate (bbox):
            im2 = im.crop (box)
            im2 = im2.resize((128, 128))
            new_path = str(dog_image_path).replace(str(img_dir),
'./Cropped').replace('.jpg', f'-{j}.jpg')
            head, tail = os.path.split(new_path)
            Path(head).mkdir(parents=True, exist_ok=True)
            im2.save(new_path)


cropped_dir = Path('./Cropped')
```

(b) Feature Extraction: Edge histogram AND Similarity Measurements

```
selected_images = {}
for subdir in cropped_dir.iterdir():
    if subdir.is_dir():
        # List all jpg files in the subdirectory
        image_files = list(subdir.glob('*.jpg'))
        # Choose 2 images if available
        if len(image_files) >= 2:
            selected_images[subdir.name] = image_files[:1] #selecting
1 image
        else:
            print(f"Warning: Less than 2 images found for class
{subdir.name}")

for class_name, images in selected_images.items():
    print(f"Class: {class_name}")
    for img in images:
        print(img)
    print('-'* 50)
```

```
Class: n02091831-Saluki
Cropped\n02091831-Saluki\n02091831_10215-0.jpg
--------------------------------------------------
Class: n02093859-Kerry_blue_terrier
Cropped\n02093859-Kerry_blue_terrier\n02093859_10-0.jpg
--------------------------------------------------
Class: n02108551-Tibetan_mastiff
Cropped\n02108551-Tibetan_mastiff\n02108551_10182-0.jpg
--------------------------------------------------
Class: n02111277-Newfoundland
Cropped\n02111277-Newfoundland\n02111277_1008-0.jpg
--------------------------------------------------
```

```python
for class_name, images in selected_images.items():
    # Opening the selected image
    img = Image.open(images[0])
    # Display image size
    img_size = img.size  # This gives (width, height)
    print(f"Image Size: {img_size[0]} x {img_size[1]} pixels (Width x Height)")
```

```
Image Size: 128 x 128 pixels (Width x Height)
Image Size: 128 x 128 pixels (Width x Height)
Image Size: 128 x 128 pixels (Width x Height)
Image Size: 128 x 128 pixels (Width x Height)
```

```python
def display_images (image_path):
    img = Image.open(image_path)
    gray_img = img.convert('L')

    plt.figure(figsize=(10, 5))
    # Display colored image
    plt.subplot(1, 2, 1)
    plt.imshow(img)
    plt.title('Coloured')
    plt.axis('off')
    # Display grayscale image
    plt.subplot(1, 2, 2)
    plt.imshow(gray_img, cmap='gray')
    plt.title('Grayscale')
    plt.axis('off')
    plt.show()

for class_name, images in selected_images.items():
    print(f"Class: {class_name}")
    display_images (images[0])
```

```
Class: n02091831-Saluki
```

Coloured

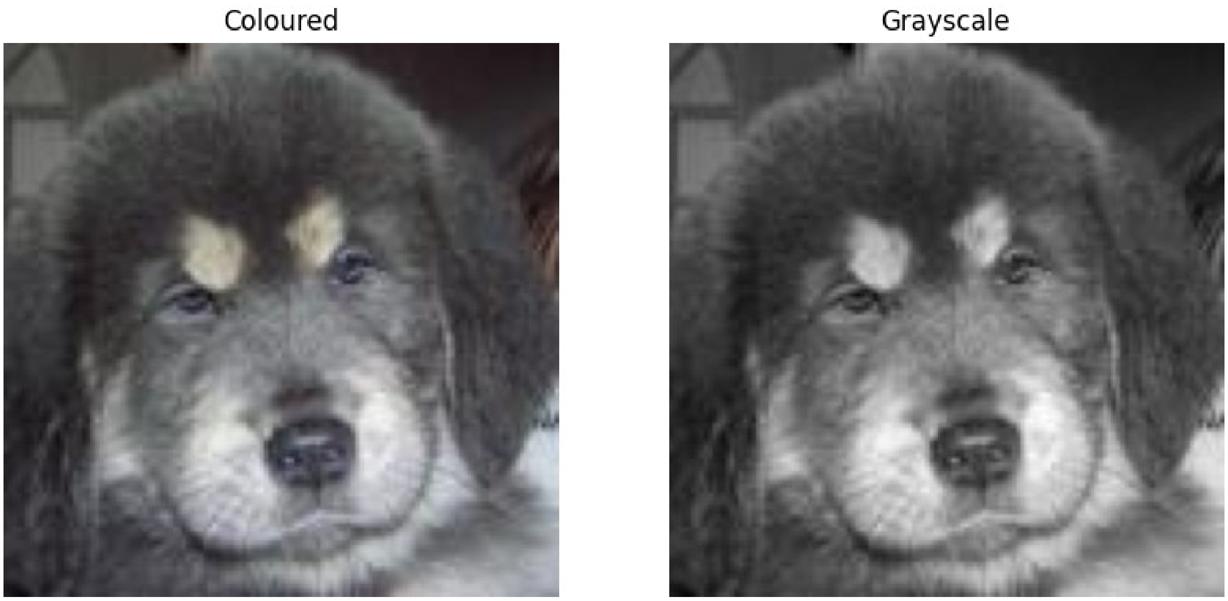
Grayscale

```
Class: n02093859-Kerry_blue_terrier
```

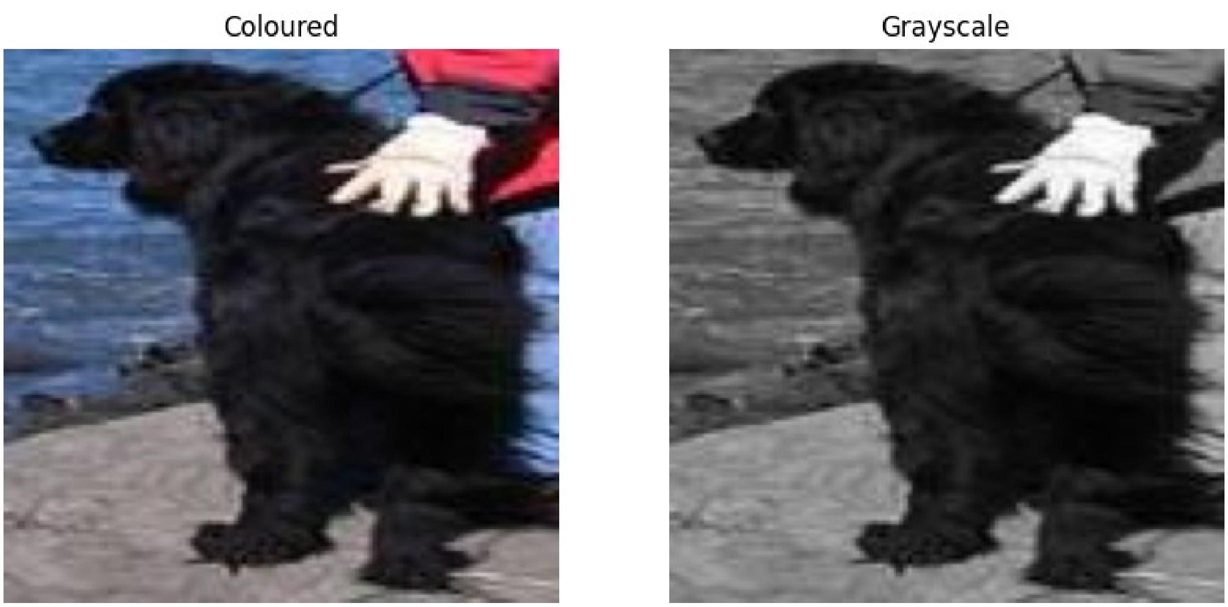
Coloured

Grayscale

```
Class: n02108551-Tibetan_mastiff
```



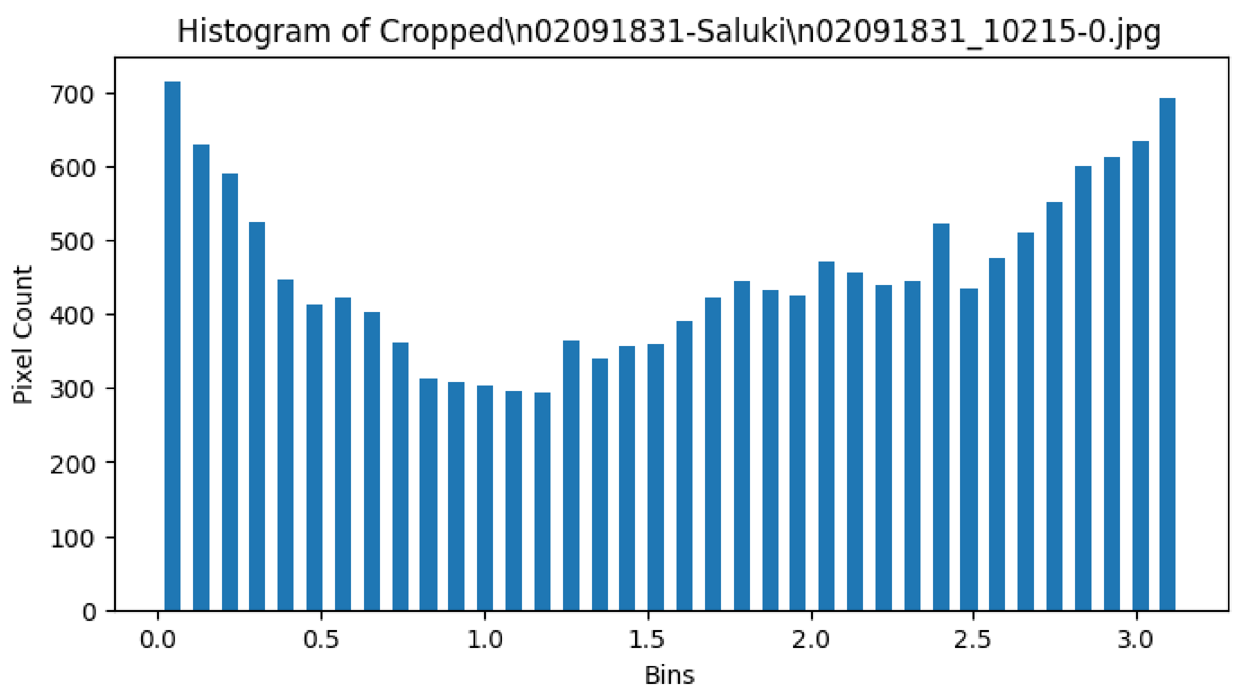

```
Class: n02111277-Newfoundland
```




```python
import numpy as np
from skimage import filters, color, exposure
from PIL import Image
import matplotlib.pyplot as plt

# Function for calculating gradient angle
def angle(dx, dy):
    return np.mod(np.arctan2(dy, dx), np.pi)
```

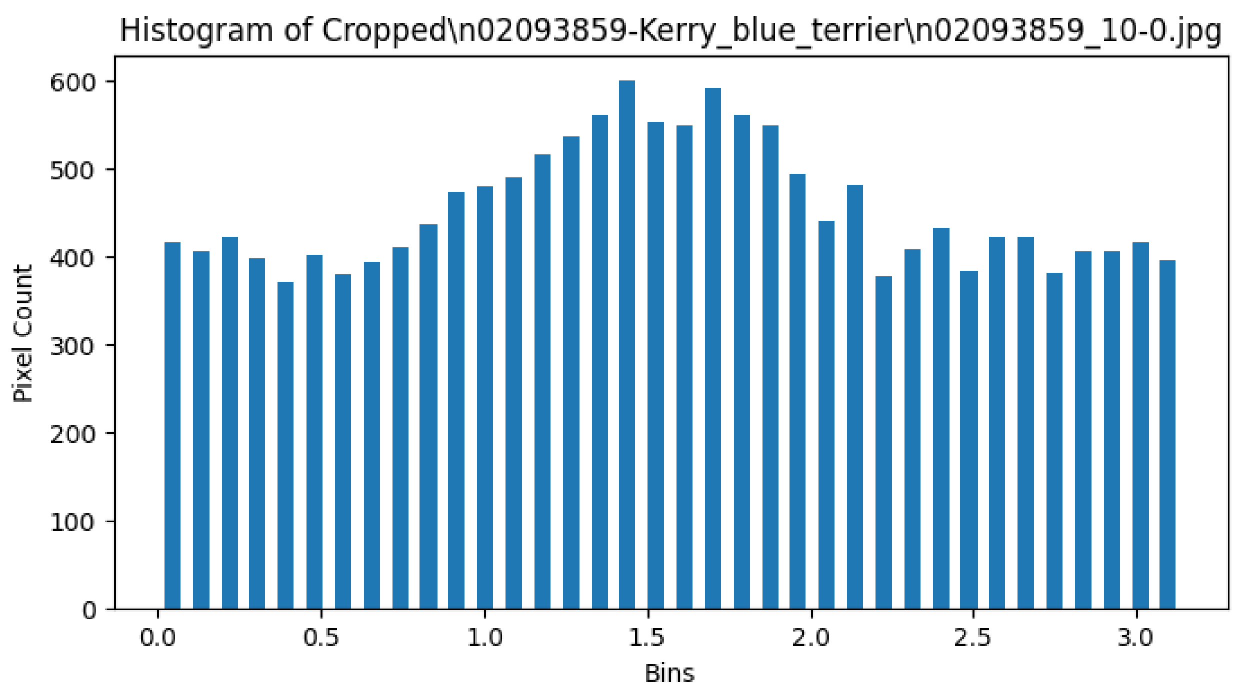
```
# Function for computing the histogram with 36 bins
def compute_histogram(gradient_angles, bins=36):
    hist, hist_centers = exposure.histogram(gradient_angles,
nbins=bins)
    return hist, hist_centers

# Dictionary to store the gradient angle results and histograms
edge = {}
histograms = {}
hist_value= []

# Processing the images by class
for class_name, images in selected_images.items():
    print(f"Class: {class_name}")
    for img_path in images:
        img = Image.open(img_path)

        # Convert to grayscale
        img_np = np.array(img)
        if len(img_np.shape) == 3:  # If the image is RGB, convert to
grayscale
            image = color.rgb2gray(img_np)
        else:
            image = img_np  # If already grayscale, use it

        # Calculate Sobel gradients and angles
        dx = filters.sobel_h(image)  # Horizontal gradient
        dy = filters.sobel_v(image)  # Vertical gradient
        angle_sobel = angle(dx, dy)  # Compute gradient angle

        # Store the angle result in edge dictionary
        edge[img_path] = angle_sobel
        hists= compute_histogram(angle_sobel, bins=36)
        hist_value.append(hists)
        # Compute the histogram with 36 bins for gradient angles
        hist, hist_centers = hists

        # Store the histogram and centers in histograms dictionary
        histograms[img_path] = {'hist': hist, 'centers': hist_centers}
        print(histograms)

        print(f"Processed {img_path}, gradient angle and histogram
computed.")

        plt.figure(figsize=(8, 4))
        plt.bar(hist_centers, hist, width=0.05)
        plt.title(f"Histogram of {img_path}")
        plt.xlabel('Bins')
        plt.ylabel('Pixel Count')
```

```
        plt.show()


Class: n02091831-Saluki
{WindowsPath('Cropped/n02091831-Saluki/n02091831_10215-0.jpg'):
{'hist': array([713, 628, 589, 525, 447, 413, 422, 403, 362, 313, 308,
304, 296,
       293, 364, 339, 356, 358, 390, 423, 445, 433, 425, 470, 456,
439,
       444, 521, 435, 476, 510, 550, 599, 611, 633, 691],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])}}
Processed Cropped\n02091831-Saluki\n02091831_10215-0.jpg, gradient
angle and histogram computed.
```

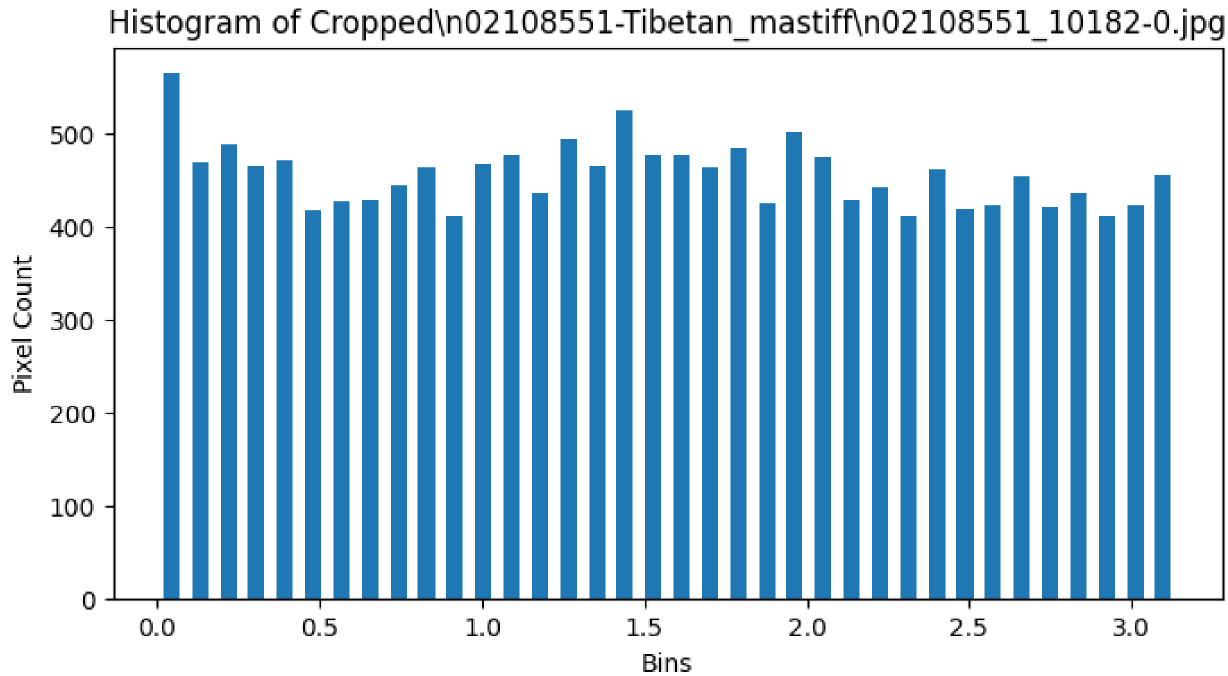



```
Class: n02093859-Kerry_blue_terrier
{WindowsPath('Cropped/n02091831-Saluki/n02091831_10215-0.jpg'):
{'hist': array([713, 628, 589, 525, 447, 413, 422, 403, 362, 313, 308,
304, 296,
       293, 364, 339, 356, 358, 390, 423, 445, 433, 425, 470, 456,
439,
```

```
       444, 521, 435, 476, 510, 550, 599, 611, 633, 691],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])}, WindowsPath('Cropped/n02093859-
Kerry_blue_terrier/n02093859_10-0.jpg'): {'hist': array([417, 406,
422, 399, 372, 402, 380, 395, 411, 438, 475, 480, 491,
       516, 538, 562, 600, 553, 549, 592, 562, 550, 494, 442, 483,
377,
       409, 433, 383, 423, 422, 382, 406, 406, 417, 397],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])}}
Processed Cropped\n02093859-Kerry_blue_terrier\n02093859_10-0.jpg,
gradient angle and histogram computed.
```

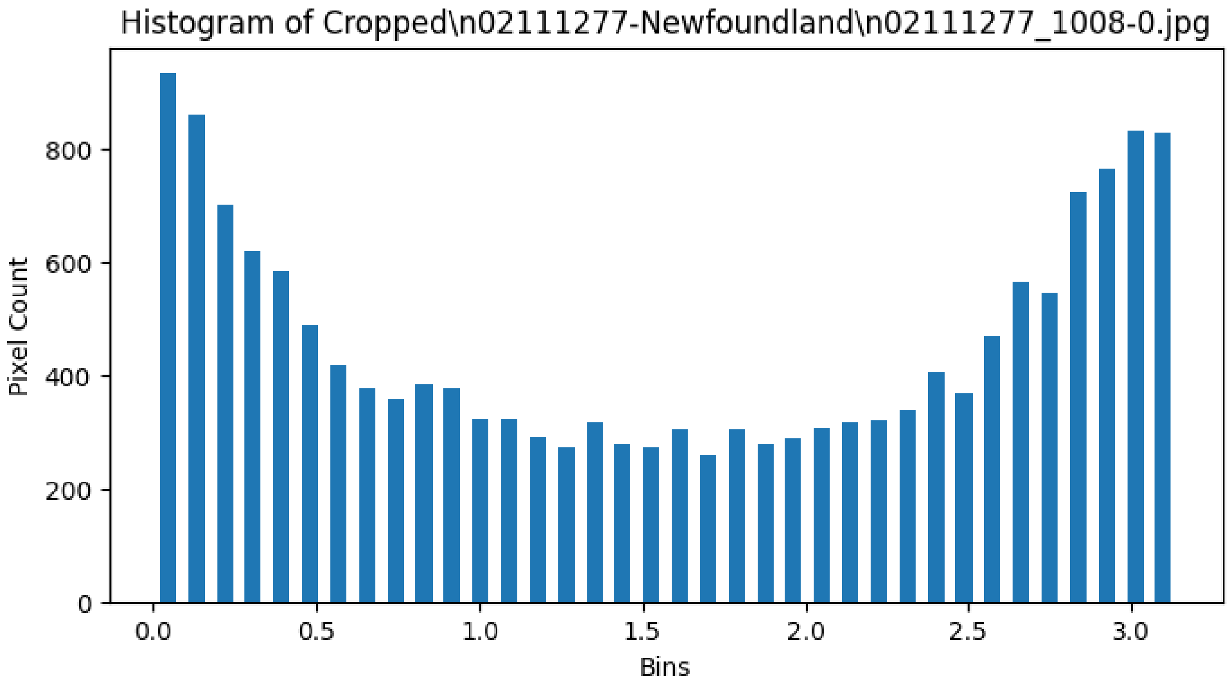


```
Class: n02108551-Tibetan_mastiff
{WindowsPath('Cropped/n02091831-Saluki/n02091831_10215-0.jpg'):
{'hist': array([713, 628, 589, 525, 447, 413, 422, 403, 362, 313, 308,
304, 296,
       293, 364, 339, 356, 358, 390, 423, 445, 433, 425, 470, 456,
439,
       444, 521, 435, 476, 510, 550, 599, 611, 633, 691],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])}, WindowsPath('Cropped/n02093859-
Kerry_blue_terrier/n02093859_10-0.jpg'): {'hist': array([417, 406,
422, 399, 372, 402, 380, 395, 411, 438, 475, 480, 491,
       516, 538, 562, 600, 553, 549, 592, 562, 550, 494, 442, 483,
377,
       409, 433, 383, 423, 422, 382, 406, 406, 417, 397],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])},
WindowsPath('Cropped/n02108551-Tibetan_mastiff/n02108551_10182-
0.jpg'): {'hist': array([564, 469, 488, 465, 470, 416, 427, 429, 443,
462, 411, 467, 477,
       436, 493, 465, 524, 477, 477, 462, 484, 425, 501, 474, 428,
441,
       411, 460, 418, 423, 453, 420, 435, 411, 422, 456],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])}}
Processed Cropped\n02108551-Tibetan_mastiff\n02108551_10182-0.jpg,
gradient angle and histogram computed.
```




```
Class: n02111277-Newfoundland
{WindowsPath('Cropped/n02091831-Saluki/n02091831_10215-0.jpg'):
{'hist': array([713, 628, 589, 525, 447, 413, 422, 403, 362, 313, 308,
304, 296,
       293, 364, 339, 356, 358, 390, 423, 445, 433, 425, 470, 456,
439,
       444, 521, 435, 476, 510, 550, 599, 611, 633, 691],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])}, WindowsPath('Cropped/n02093859-
Kerry_blue_terrier/n02093859_10-0.jpg'): {'hist': array([417, 406,
422, 399, 372, 402, 380, 395, 411, 438, 475, 480, 491,
       516, 538, 562, 600, 553, 549, 592, 562, 550, 494, 442, 483,
377,
       409, 433, 383, 423, 422, 382, 406, 406, 417, 397],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
```

```
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])},
WindowsPath('Cropped/n02108551-Tibetan_mastiff/n02108551_10182-
0.jpg'): {'hist': array([564, 469, 488, 465, 470, 416, 427, 429, 443,
462, 411, 467, 477,
       436, 493, 465, 524, 477, 477, 462, 484, 425, 501, 474, 428,
441,
       411, 460, 418, 423, 453, 420, 435, 411, 422, 456],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])},
WindowsPath('Cropped/n02111277-Newfoundland/n02111277_1008-0.jpg'):
{'hist': array([931, 859, 702, 619, 584, 488, 417, 378, 358, 384, 377,
323, 322,
       291, 273, 316, 280, 271, 304, 258, 304, 280, 288, 308, 315,
320,
       339, 404, 367, 470, 563, 546, 722, 765, 831, 827],
dtype=int64), 'centers': array([0.04363323, 0.13089969, 0.21816616,
0.30543262, 0.39269908,
       0.47996554, 0.56723201, 0.65449847, 0.74176493, 0.82903139,
       0.91629786, 1.00356432, 1.09083078, 1.17809725, 1.26536371,
       1.35263017, 1.43989663, 1.5271631 , 1.61442956, 1.70169602,
       1.78896248, 1.87622895, 1.96349541, 2.05076187, 2.13802833,
       2.2252948 , 2.31256126, 2.39982772, 2.48709418, 2.57436065,
       2.66162711, 2.74889357, 2.83616003, 2.9234265 , 3.01069296,
       3.09795942])}}
Processed Cropped\n02111277-Newfoundland\n02111277_1008-0.jpg,
gradient angle and histogram computed.
```

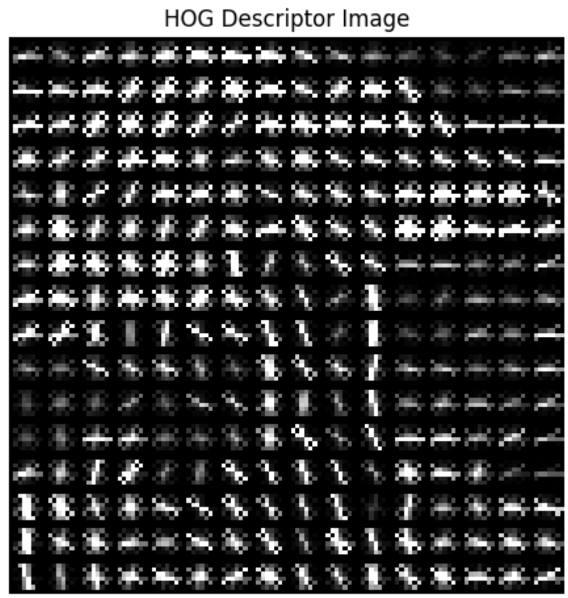



```python
def compute_grayscale_histogram(image_path):
    img = Image.open(image_path).convert('L')
    img_np = np.asarray(img)
    hist = cv2.calcHist([img_np], [0], None, [256], [0, 256])
    return hist.flatten()

def euclidean_distance (histA, histB):
    return np.linalg.norm(histA - histB)
def manhattan_distance (histA, histB):
    return np.sum (np.abs(histA - histB))
def cosine_distance(histA, histB):
    dot_product = np.dot(histA.flatten(), histB.flatten())
    normA = np.linalg.norm(histA.flatten())
    normB = np.linalg.norm(histB.flatten())
    cosine_similarity = dot_product / (normA * normB)
    return 1 - cosine_similarity

two_class_distances = {
    'Euclidean Distance': euclidean_distance (np.array(hist_value[0]),
np.array(hist_value[1])),
    'Manhattan Distance': manhattan_distance
(np.array(hist_value[0]) , np.array(hist_value[1])),
    'Cosine Distance': cosine_distance(np.array(hist_value[0]),
np.array(hist_value[1]))}

print("\nDistance Metrics for Two Images from Different Classes:")
for metric, value in two_class_distances.items():
    print(f"{metric}): {value:.4f}")
```

```
Distance Metrics for Two Images from Different Classes:
Euclidean Distance): 940.5339
Manhattan Distance): 4858.0000
Cosine Distance): 0.0568
```

(c) Histogram of Oriented Gradient (HOG) feature descriptor

```
random_class = random.choice (list(selected_images.keys()))
random_im_path = random.choice(selected_images [random_class])
img = Image.open(random_im_path)
gray_img = img.convert('L')
gray_img_np = np.array(gray_img)

from skimage.feature import hog
hog_features, hog_image = hog(gray_img, orientations=9,
pixels_per_cell=(8, 8),
                              cells_per_block=(2, 2), visualize=True,
channel_axis=None)

from skimage import exposure
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6), sharex=True,
sharey=True)

# Original image
ax1.imshow(gray_img, cmap='gray')
ax1.set_title('Original Image')
ax1.axis('off')

# HOG visualization
hog_image_rescaled = exposure.rescale_intensity(hog_image,
in_range=(0, 10))
ax2.imshow(hog_image_rescaled, cmap='gray')
ax2.set_title('HOG Descriptor Image')
ax2.axis('off')

plt.show()
```



(d) Dimensionality reduction (using Principal Component Analysis, PCA)

```
all_images = {}
for subdir in cropped_dir.iterdir():
    if subdir.is_dir():
        # List all jpg files in the subdirectory
        image_files = list(subdir.glob('*.jpg'))
        if len(image_files) >= 2:
            all_images[subdir.name] = image_files[:-1]
        else:
            print(f"Warning: Less than 2 images found for class
{subdir.name}")


# Processing the images by class
all_edges = {}
for class_name, images in all_images.items():
    print(f"Class: {class_name}")
    for img_path in images:
        # Open image using PIL
        img = Image.open(img_path)

        # Convert to grayscale
        img_np = np.array(img)
        if len(img_np.shape) == 3:  # If the image is RGB then convert
it into grayscale
            image = color.rgb2gray(img_np)
        else:
            image = img_np  # If already grayscale, use it

        # Calculate Sobel gradients and angles
```

```
        dx = filters.sobel_h(image)  # Horizontal gradient
        dy = filters.sobel_v(image)  # Vertical gradient
        angle_sobel = angle(dx, dy)  # Compute gradient angle

        # Store the angle result in edge dictionary
        all_edges[img_path] = angle_sobel
```

```
Class: n02091831-Saluki
Class: n02093859-Kerry_blue_terrier
Class: n02108551-Tibetan_mastiff
Class: n02111277-Newfoundland
```

```
selected_classes = images_subdirs[:4]
selected_images = {}
for class_name in selected_classes:
    class_dir = img_dir +"\\" +class_name
    image_files = [f for f in os.listdir(class_dir) if
f.endswith(('.jpg'))]
    selected_images [class_name] = [class_dir+"\\" +img_file for
img_file in image_files]

for class_name, images in selected_images.items():
    print(f"Number of images from {class_name}: {len(images)}")
```

```
Number of images from n02091831-Saluki: 200
Number of images from n02093859-Kerry_blue_terrier: 179
Number of images from n02108551-Tibetan_mastiff: 152
Number of images from n02111277-Newfoundland: 195
```

```
# Compute normalized histograms for all the selected images
histograms = {}
for class_name, image_paths in all_images.items():
    histograms [class_name] = [compute_grayscale_histogram(img_path)
for img_path in image_paths]


from sklearn.decomposition import PCA
#Converting histograms dictionary into a flat List
hist_list = [hist for class_hists in histograms.values() for hist in
class_hists]
pca = PCA(n_components=2)
hist_reduced = pca.fit_transform(hist_list)

for idx, class_name in enumerate(histograms):
    num_hists = len(histograms[class_name])
    print(f"Reduced Histograms for {class_name}: {hist_reduced[idx *
num_hists:(idx + 1) * num_hists]}")
```

```
Reduced Histograms for n02091831-Saluki: [[-4.22974322e+02
1.27966531e+02]
 [-7.88489305e+02 -3.18435609e+02]
```

```
 [ 2.52683083e+02 -1.73568615e+02]
 [ 2.18406446e+02  3.82427152e+01]
 [-3.32824445e+02  5.17376111e+02]
 [-2.63666061e+02  1.57574989e+02]
 [ 1.14678614e+03 -3.94468028e+02]
 [ 1.66848663e+02 -1.48733157e+02]
 [-2.49015779e+02  3.29973236e+02]
 [-3.75498329e+02  2.44051936e+02]
 [-4.05355344e+02  1.45201018e+02]
 [-6.53599429e+02 -2.35207303e+02]
 [-2.75609234e+02 -2.31884560e+02]
 [-1.55164260e+02 -2.79372040e+01]
 [-5.82608035e+02 -1.58107551e+02]
 [-4.89716323e+02 -1.14219988e+02]
 [-3.47403258e+02 -1.62216607e+02]
 [ 3.36331541e+02  2.80479360e+02]
 [-2.95633194e-01  1.53441602e+02]
 [ 3.28354166e+02 -1.78420897e+02]
 [-1.65595300e+00  3.58065814e+02]
 [ 3.15988494e+02  2.02367896e+02]
 [-6.44881095e+02 -2.65733529e+02]
 [ 1.85262339e+02  3.70673142e+02]
 [ 2.84347705e+02 -2.59044979e+02]
 [-2.39650281e+02  1.63009537e+02]
 [ 1.77012530e+02  9.20462068e+02]
 [-3.47057591e+02  1.16418673e+02]
 [-2.13260182e+02  3.23585517e+02]
 [ 5.96512908e+02 -3.57063632e+02]
 [ 4.49890815e+02 -6.60217792e+02]
 [-8.24426289e+02 -6.20739597e+02]
 [-4.94376879e+02 -1.76471474e+01]
 [-9.17995549e+02 -6.41863272e+02]
 [-6.67156579e+02 -4.09143861e+02]
 [-9.17046715e+02 -6.33177001e+02]
 [-6.98556325e+02 -1.02962664e+02]
 [-1.71606876e+02 -4.92621046e+01]
 [-6.07868949e+02 -4.02471028e+02]
 [ 3.30439899e+02  1.60190445e+02]
 [-8.92450248e+02 -3.56625637e+02]
 [-7.29544602e+02 -4.06230591e+02]
 [-5.04124417e+01  7.70455225e+01]
 [-2.53770220e+02  3.60047796e+01]
 [-3.89037576e+02 -8.31240097e+01]
 [-5.82339406e+02 -2.07633963e+02]
 [-7.87227862e+02 -4.20102159e+02]
 [-1.00427314e+02 -3.00749783e+02]
 [ 2.53328789e+02  3.79523526e+02]
 [-5.76470556e+02  6.64115120e+00]
 [-6.65207054e+02 -3.95034962e+02]
```

```
 [ 1.81017940e+02  8.00681009e+02]
 [-7.43495078e+02 -5.38246925e+02]
 [-4.15683163e+02  4.93999672e+01]
 [-8.20375435e+02 -4.76040748e+02]
 [ 3.77203355e+02  2.95443092e+02]
 [-4.75512511e+02  2.19410838e+02]
 [-8.59286394e+02 -5.24822558e+02]
 [-3.77892851e+02 -4.76906000e+02]
 [-1.53850667e+01 -9.85406524e+01]
 [-4.15746031e+02 -1.38783087e+02]
 [ 4.05745808e+02 -5.42474528e+02]
 [-5.14173991e+02 -2.82881634e+02]
 [-6.47858108e+02 -1.59181066e+01]
 [-4.93533859e+02 -2.59801902e+02]
 [ 3.39137225e+02 -2.22721342e+02]
 [ 1.29540935e+02  7.03535715e+01]
 [-6.17487578e+02 -1.49830225e+02]
 [-1.43499328e+02 -6.91218993e+01]
 [-4.46908558e+02 -1.33840095e+00]
 [-2.99367661e+02  3.87292367e+01]
 [-1.05493770e+02  1.57669290e+02]
 [ 2.95466185e+01 -1.96707588e+02]
 [-2.53565765e+01 -1.30956715e+02]
 [ 5.78790930e+02 -2.71691019e+02]
 [-3.80909837e+02  1.95964810e+02]
 [-7.10592500e+02 -4.39997490e+02]
 [-1.21616371e+02 -4.63222625e+02]
 [-6.99851998e+02 -1.75218207e+02]
 [ 1.13333099e+02  1.77168671e+02]
 [-4.48759064e+01  1.00105978e+03]
 [-1.84007644e+02 -7.01394715e+01]
 [-8.54766265e+02 -5.60057986e+02]
 [-3.47815430e+02  2.05818432e+02]
 [-6.31932928e+01  1.14435753e+02]
 [-6.44199509e+02 -4.73578074e+02]
 [-8.48376772e+02 -4.60902115e+02]
 [-5.24189125e+02 -2.91733436e+02]
 [-3.24998507e+02 -5.40310219e+02]
 [ 2.27231993e+02  6.07109597e+01]
 [ 1.49965603e+02  5.03718756e+01]
 [-4.05244717e+02  2.32994133e+02]
 [-4.83946110e+02  1.25412638e+02]
 [ 9.69684795e+00 -1.51283134e+01]
 [ 5.46191540e+02  1.73628332e+02]
 [-3.12505524e+02 -4.07577417e+01]
 [-7.80591924e+02 -1.61917401e+01]
 [-8.37348794e+02 -4.35291818e+02]
 [ 1.08666111e+02 -2.35038045e+02]
 [ 1.55287739e+02 -8.44895942e+01]
```

```
 [-8.39982809e+02 -3.66980873e+02]
 [-7.63327121e+02 -4.95007647e+02]
 [ 7.34673455e+01 -9.17880631e+01]
 [-8.65798793e+02 -3.14560407e+02]
 [-8.03167960e+02 -6.27873561e+02]
 [-4.71507698e+02 -4.52451205e+02]
 [-6.96571165e+02 -2.58362314e+02]
 [-2.68410981e+02 -3.10765891e+02]
 [-5.82334673e+02 -1.34715016e+02]
 [-2.79860406e+02  1.43691215e+02]
 [-8.37249023e+02 -6.93535055e+02]
 [-7.02719779e+02 -5.85683037e+02]
 [-7.30946468e+02 -5.88188261e+02]
 [-8.67824622e+02 -5.29888224e+02]
 [-1.64847290e+02  5.62131696e+02]
 [-7.52186633e+02 -3.90035345e+02]
 [ 5.31423841e+01 -5.40550116e+02]
 [-4.72277585e+02  3.81461454e+01]
 [-4.90387594e+02  8.65288337e+01]
 [-9.54619813e+02 -8.51848227e+02]
 [-4.59045388e+02 -6.38139434e+02]
 [ 3.65169558e+01  4.26084367e+01]
 [-2.81575248e+02 -1.36477006e+02]
 [-8.15586517e+02 -5.01034665e+02]
 [-4.70258644e+02  8.77170691e+01]
 [-3.27664099e+02  2.40654699e+02]
 [ 9.41817963e+00 -8.80082880e+02]
 [-3.35944363e+02 -2.28392304e+02]
 [ 1.17678103e+02  2.21179244e+02]
 [-3.81225790e+02  2.52501756e+02]
 [-8.49508206e+02 -6.14457344e+02]
 [-6.09336621e+02 -1.03987012e+02]
 [-1.59185583e+02 -1.77600871e+01]
 [-1.62255551e+02  1.48229693e+02]
 [ 4.18321131e+01 -1.57655943e+02]
 [ 6.56973016e+01  1.12750819e+02]
 [-6.00248210e+02 -1.13519182e+02]
 [-1.75482157e+02  6.08658023e+01]
 [-3.36585823e+02 -6.03647692e+01]
 [ 9.70370299e+00 -2.11838187e+02]
 [-8.32776819e+02 -7.68678532e+02]
 [-9.25398967e+02 -4.70265753e+02]
 [-3.40180446e+02  3.18421787e+02]
 [-1.43600331e+02 -3.19603600e+02]
 [-8.83331008e+01 -3.24951599e+02]
 [-5.94759297e+02 -2.04951710e+02]
 [-2.37597655e+02 -6.85683876e+01]
 [-1.58770062e+02 -8.60570087e+01]
 [-6.07740038e+02 -3.87902995e+02]
```

```
 [-5.78990351e+02 -2.62648790e+02]
 [-3.73766864e+02  1.83399635e+02]
 [-1.05399643e+01 -5.61092322e+02]
 [-4.96421890e+02  1.61028535e+02]
 [-1.89851419e+02 -1.16527472e+01]
 [-4.60605795e+02  2.58699253e+00]
 [-3.75916893e+02 -2.89477135e+02]
 [-2.58365594e+01  7.97726925e+02]
 [-6.32303934e+01  2.37237252e+02]
 [ 1.55128053e+02 -4.91775993e+02]
 [-6.96817374e+02 -5.91010670e+02]
 [-8.74988555e+02 -5.88963088e+02]
 [-4.38480306e+02  2.58627760e+02]
 [-6.41803108e+01  3.28381765e+02]
 [-7.73534012e+02 -1.73701439e+02]
 [-7.14931262e+02 -3.41145256e+02]
 [-2.00701974e+01 -4.79807677e+02]
 [-6.82228300e+02 -7.12404497e+01]
 [-6.71924146e+02 -4.20593911e+02]
 [-4.24871710e+02 -5.16231381e+02]
 [-6.91645541e+02 -2.13589333e+02]
 [-3.49016599e+02 -8.18647692e+01]
 [-5.78940227e+02  3.43210605e+02]
 [-5.01708414e+02 -7.68766077e+01]
 [-4.60301301e+02  2.33840712e+02]
 [-4.55251539e+02 -9.76432359e+01]
 [-6.65117823e+02 -5.27348609e+02]
 [-5.20076926e+02 -5.80054900e+01]
 [-5.25385942e+02 -2.04031932e+02]
 [-6.90474260e+02 -5.50346571e+02]
 [-8.14153729e+02 -7.10737503e+02]
 [-6.92808558e+02 -4.69834732e+02]
 [-2.92625690e+02  2.03637307e+02]
 [-6.93364695e+02 -4.92015766e+02]
 [ 5.12657536e+01 -6.32613893e+02]
 [-6.31257378e+02  1.38789084e+02]
 [-5.89071476e+02 -2.10764571e+02]
 [-4.70453594e+02 -4.82447084e+02]
 [ 7.90518726e+01 -3.21037259e+02]
 [ 4.96489268e+02 -4.20676282e+02]
 [-5.41957343e+02 -3.18409885e+02]
 [-5.86752331e+02 -3.57936917e+02]
 [-2.02010205e+01  3.67050576e+02]
 [ 1.17815333e+03 -5.69745665e+02]
 [-6.64141082e+02  2.36435894e+02]
 [-4.53886570e+02  6.17087275e+01]
 [-3.01011732e+02  5.19248139e+02]
 [-6.23220693e+02  4.13970125e+02]
 [ 2.42172217e+02  6.56585136e+02]
```

```
 [-5.42633980e+02  6.84194411e+01]
 [-2.16227765e+02  4.02132430e+01]
 [-2.66108595e+02 -1.82523316e+02]
 [-4.35391966e+02 -3.14082938e+02]
 [-4.01199166e+02  3.62979191e+02]
 [-2.55487852e+02  6.05992640e+02]
 [ 1.02751563e+02 -2.41971574e+02]
 [ 2.26237624e+01  5.13043272e+02]
 [-4.23783426e+02  1.18659356e+02]
 [-3.42381456e+02  3.15011004e+01]
 [-2.93366932e+02 -5.50505471e+01]
 [-8.37816763e+02 -3.71523956e+02]
 [-4.24427301e+02 -2.33738624e+02]
 [-1.59448138e+02  1.17088837e+02]
 [-6.61801339e+02 -4.22390275e+02]
 [-6.26780508e+02 -4.75792732e+02]
 [-8.47785258e+02 -4.95928911e+02]
 [-7.73127084e+02 -3.67315023e+02]
 [-6.22097542e+02 -6.11677129e+02]
 [ 1.62022929e+02 -2.32503420e+02]
 [-6.07230467e+02 -1.40099934e+01]
 [-3.46039535e+02 -1.04142574e+02]
 [-4.00166740e+02 -4.94988408e+02]
 [ 6.16430427e+01 -4.37274620e+02]]
Reduced Histograms for n02093859-Kerry_blue_terrier: [[-5.86752331e+02
-3.57936917e+02]
 [-2.02010205e+01  3.67050576e+02]
 [ 1.17815333e+03 -5.69745665e+02]
 [-6.64141082e+02  2.36435894e+02]
 [-4.53886570e+02  6.17087275e+01]
 [-3.01011732e+02  5.19248139e+02]
 [-6.23220693e+02  4.13970125e+02]
 [ 2.42172217e+02  6.56585136e+02]
 [-5.42633980e+02  6.84194411e+01]
 [-2.16227765e+02  4.02132430e+01]
 [-2.66108595e+02 -1.82523316e+02]
 [-4.35391966e+02 -3.14082938e+02]
 [-4.01199166e+02  3.62979191e+02]
 [-2.55487852e+02  6.05992640e+02]
 [ 1.02751563e+02 -2.41971574e+02]
 [ 2.26237624e+01  5.13043272e+02]
 [-4.23783426e+02  1.18659356e+02]
 [-3.42381456e+02  3.15011004e+01]
 [-2.93366932e+02 -5.50505471e+01]
 [-8.37816763e+02 -3.71523956e+02]
 [-4.24427301e+02 -2.33738624e+02]
 [-1.59448138e+02  1.17088837e+02]
 [-6.61801339e+02 -4.22390275e+02]
 [-6.26780508e+02 -4.75792732e+02]
```

```
 [-8.47785258e+02 -4.95928911e+02]
 [-7.73127084e+02 -3.67315023e+02]
 [-6.22097542e+02 -6.11677129e+02]
 [ 1.62022929e+02 -2.32503420e+02]
 [-6.07230467e+02 -1.40099934e+01]
 [-3.46039535e+02 -1.04142574e+02]
 [-4.00166740e+02 -4.94988408e+02]
 [ 6.16430427e+01 -4.37274620e+02]
 [-5.48839416e+02 -1.12373019e+02]
 [-6.13201589e+02 -2.61463697e+02]
 [-3.87343888e+01  2.80265672e+02]
 [ 7.16297808e+02 -3.09669043e+02]
 [ 7.00490345e+02  4.64651772e+02]
 [ 8.28197779e+01  2.42361309e+02]
 [ 1.90354249e+02  3.54356031e+02]
 [-2.30581213e+02  4.84480768e+02]
 [ 7.38630642e+02 -3.23764571e+01]
 [ 5.62940947e+02  2.24801117e+02]
 [-6.64880794e+02 -1.85517528e+02]
 [-5.65853990e+02 -1.11855351e+02]
 [ 1.20885055e+02  2.51870204e+02]
 [-1.89779539e+02 -2.35590598e+01]
 [-9.96530946e+01 -1.20276942e+02]
 [-1.83143524e+02  1.01628622e+02]
 [-4.87725100e+02  1.33654226e+02]
 [ 6.85235664e+02  8.19409007e+02]
 [-2.26808616e+02 -6.57660546e+01]
 [-2.72477486e+01  9.30686077e+00]
 [ 1.96879075e+02  6.35037470e+02]
 [-9.49764631e+01  2.35549279e+02]
 [-7.48488088e+02 -3.58192725e+02]
 [-6.88778437e+02 -2.39521492e+02]
 [-3.59701329e+01  6.91978062e+02]
 [ 2.02939812e+02  4.19431745e+02]
 [ 6.96740452e+02  4.50098975e+02]
 [ 1.00083299e+02  2.40284807e+02]
 [-2.51349121e+00 -2.84755766e+02]
 [ 1.31352242e+02  5.73900690e+02]
 [-5.18190605e+02 -2.50281491e+02]
 [ 4.53821501e+02 -3.99907077e+02]
 [-2.65668942e+02  5.57343333e+02]
 [-3.32748629e+02  1.76360938e+02]
 [ 6.33340356e+02  5.48891734e+02]
 [ 4.05104324e+02  2.80904709e+02]
 [ 7.77977648e+01  2.99688580e+02]
 [-8.52153657e+02 -4.70852388e+02]
 [-6.09551505e+02 -3.51065619e+00]
 [-4.75209877e+01  3.10505088e+02]
 [-6.58854417e+02  8.01650862e+00]
```

```
 [-1.04314190e+02  3.90592865e+02]
 [ 1.69870546e+01  5.36233211e+02]
 [ 8.59652398e+02 -2.29255195e+02]
 [ 6.13052034e+02  3.38518020e+02]
 [ 4.06674199e+02  7.40984844e+02]
 [ 7.95296463e+02  3.08265605e+02]
 [-5.12105961e+02  5.24834722e+01]
 [ 1.45153064e+02 -1.19727969e+02]
 [ 2.10424455e+02 -5.52785787e+02]
 [ 2.69442832e+02  2.56995996e+02]
 [-2.33693964e+02  7.79815075e+01]
 [-3.53204858e+02  4.80410139e+02]
 [-2.71353636e+02  6.39256178e+02]
 [-2.57894842e+02  9.92162341e+02]
 [ 1.62143689e+02 -4.17036635e+02]
 [ 3.18698799e+02  1.54948866e+02]
 [-1.86027776e+02  3.96976285e+02]
 [-3.75742562e+01  5.47187023e+01]
 [-2.05302912e+02 -1.31243436e+01]
 [-4.10654988e+02  9.01829344e+01]
 [-4.85269844e+02 -1.37087319e+02]
 [ 2.09276197e+02 -2.01181583e+02]
 [ 2.88572420e+02 -2.48920249e+02]
 [-5.62519733e+01  5.42638600e+02]
 [ 4.28390675e+02 -2.92337089e+02]
 [-5.93626206e+01  2.09233785e+02]
 [-5.47448892e+02  1.30690125e+02]
 [-5.64944334e+02  3.59340158e+02]
 [ 7.32647613e+02  1.15492643e+02]
 [ 7.10675603e+02  5.16686933e+01]
 [ 3.37446270e+02 -6.62968254e+02]
 [-1.75699283e+01  5.65171406e+01]
 [ 7.97146794e+02 -5.17094995e+02]
 [ 1.65439520e+02 -3.74982367e+02]
 [-1.35296592e+02 -4.35588142e+01]
 [-4.57910106e+02 -5.63285562e+01]
 [ 5.01865335e+02 -2.46670239e+02]
 [ 1.32741710e+02  4.35802053e+01]
 [ 1.91176482e+02 -7.92156340e+01]
 [-5.48270402e+02 -2.79325804e+01]
 [-8.97580045e+02 -4.58632432e+02]
 [-7.98967275e+01  3.03307134e+02]
 [-5.32874450e+02 -1.05309955e+01]
 [-7.06533131e+01 -1.46277136e+02]
 [-3.11558607e+02 -6.96558358e+01]
 [ 5.52127403e+01  3.32910987e+02]
 [ 8.99537912e+02 -1.33109183e+02]
 [ 6.16456521e+01  3.05815323e+02]
 [-1.49972024e+02  5.21068334e+02]
```

```
 [-7.70796721e+02 -3.18607488e+02]
 [ 1.73752905e+02 -1.59872848e+02]
 [-2.94523971e+02 -1.94479692e+02]
 [-5.39692194e+01 -2.94836327e+01]
 [ 2.89760025e+01  3.53658419e+02]
 [ 2.16022588e+02 -2.81053730e+01]
 [ 3.21587886e+02 -4.84774196e+02]
 [ 1.96203289e+02  2.77919487e+02]
 [-1.26104532e+02  8.73444186e+01]
 [ 1.52074444e+02  4.51362090e+02]
 [-1.66584206e+02  3.96778824e+02]
 [-7.20109796e+02 -2.53857783e+02]
 [-6.31908286e+02 -3.07847816e+02]
 [-5.03364872e+02  1.74251898e+02]
 [ 2.15250766e+02  4.41698630e+02]
 [-8.54093086e+01  3.87664265e+02]
 [-7.76108365e+01  1.96927263e+02]
 [-4.83342081e+02  2.44574294e+02]
 [ 9.67904422e+02 -6.57138470e+02]
 [-3.44274801e+02  4.49931090e+01]
 [-4.58955752e+02 -1.14558253e+00]
 [-3.24102814e+02  3.76797526e+02]
 [ 1.68027287e+02 -2.97245647e+02]
 [-2.35768028e+02  1.84617084e+01]
 [ 5.43607450e+02 -4.56357721e+02]
 [-1.77929000e+02  8.16531883e+01]
 [ 1.65108057e+02  1.25702263e+02]
 [-4.55781255e+02  1.18532428e+02]
 [-2.36864204e+02  6.20289768e+01]
 [ 3.07959225e+02 -8.79872871e+00]
 [ 3.39113195e+02 -6.45525095e+02]
 [-7.53445021e+02 -5.39811688e+02]
 [-3.27637735e+02  2.03351931e+02]
 [ 4.83486153e+02 -3.92347597e+02]
 [ 4.53741035e+02  2.60534936e+02]
 [ 6.13307108e+02  2.58596232e+02]
 [-1.44835318e+02 -1.75168084e+02]
 [ 5.98185977e+02 -1.52106186e+02]
 [ 4.74530424e+02  2.72064316e+02]
 [ 1.56288192e+02  2.74897474e+02]
 [ 1.09512346e+01  4.19223468e+01]
 [ 1.90330904e+02  1.11244925e+03]
 [-1.44171270e+02  1.09938902e+02]
 [ 1.48689253e+03 -2.12193042e+02]
 [ 1.30673118e+03 -2.59699540e+02]
 [-5.96861870e+02 -4.43565482e+00]
 [ 2.47076444e+02  2.02149975e+02]
 [-7.86170506e+01  1.31819169e+02]
 [ 8.76779906e+01  3.81091881e+02]
```

```
 [-4.41745679e+02  5.30596521e+02]
 [-6.12716291e+02  9.18454310e+01]
 [ 9.97247671e+02  2.35126284e+02]
 [-3.53450300e+02  8.90354609e+01]
 [-4.13640350e+02 -1.54220496e+02]
 [ 4.17284487e+02  3.07494644e+02]
 [ 1.35637164e+02 -3.35395311e+01]
 [-6.26184049e+02  2.71886301e+01]
 [-2.60462349e+02  1.59936500e+02]
 [-1.81697729e+02  5.91968569e+02]
 [ 5.38782617e+01  1.13392561e+02]
 [ 2.28638616e+02 -4.12369607e+02]
 [-6.33896995e+01  2.14824526e+03]
 [-7.29613708e+01  4.08980328e+02]
 [-5.00329510e+01  5.73386969e+02]
 [-5.45256721e+02 -4.55935371e+01]
 [ 6.83594334e+02 -1.75893813e+02]
 [ 1.48598468e+01  7.63754195e+01]
 [-2.56061762e+02 -8.55773187e+01]]
Reduced Histograms for n02108551-Tibetan_mastiff: [[-7.06533131e+01 -
1.46277136e+02]
 [-3.11558607e+02 -6.96558358e+01]
 [ 5.52127403e+01  3.32910987e+02]
 [ 8.99537912e+02 -1.33109183e+02]
 [ 6.16456521e+01  3.05815323e+02]
 [-1.49972024e+02  5.21068334e+02]
 [-7.70796721e+02 -3.18607488e+02]
 [ 1.73752905e+02 -1.59872848e+02]
 [-2.94523971e+02 -1.94479692e+02]
 [-5.39692194e+01 -2.94836327e+01]
 [ 2.89760025e+01  3.53658419e+02]
 [ 2.16022588e+02 -2.81053730e+01]
 [ 3.21587886e+02 -4.84774196e+02]
 [ 1.96203289e+02  2.77919487e+02]
 [-1.26104532e+02  8.73444186e+01]
 [ 1.52074444e+02  4.51362090e+02]
 [-1.66584206e+02  3.96778824e+02]
 [-7.20109796e+02 -2.53857783e+02]
 [-6.31908286e+02 -3.07847816e+02]
 [-5.03364872e+02  1.74251898e+02]
 [ 2.15250766e+02  4.41698630e+02]
 [-8.54093086e+01  3.87664265e+02]
 [-7.76108365e+01  1.96927263e+02]
 [-4.83342081e+02  2.44574294e+02]
 [ 9.67904422e+02 -6.57138470e+02]
 [-3.44274801e+02  4.49931090e+01]
 [-4.58955752e+02 -1.14558253e+00]
 [-3.24102814e+02  3.76797526e+02]
 [ 1.68027287e+02 -2.97245647e+02]
```

```
 [-2.35768028e+02  1.84617084e+01]
 [ 5.43607450e+02 -4.56357721e+02]
 [-1.77929000e+02  8.16531883e+01]
 [ 1.65108057e+02  1.25702263e+02]
 [-4.55781255e+02  1.18532428e+02]
 [-2.36864204e+02  6.20289768e+01]
 [ 3.07959225e+02 -8.79872871e+00]
 [ 3.39113195e+02 -6.45525095e+02]
 [-7.53445021e+02 -5.39811688e+02]
 [-3.27637735e+02  2.03351931e+02]
 [ 4.83486153e+02 -3.92347597e+02]
 [ 4.53741035e+02  2.60534936e+02]
 [ 6.13307108e+02  2.58596232e+02]
 [-1.44835318e+02 -1.75168084e+02]
 [ 5.98185977e+02 -1.52106186e+02]
 [ 4.74530424e+02  2.72064316e+02]
 [ 1.56288192e+02  2.74897474e+02]
 [ 1.09512346e+01  4.19223468e+01]
 [ 1.90330904e+02  1.11244925e+03]
 [-1.44171270e+02  1.09938902e+02]
 [ 1.48689253e+03 -2.12193042e+02]
 [ 1.30673118e+03 -2.59699540e+02]
 [-5.96861870e+02 -4.43565482e+00]
 [ 2.47076444e+02  2.02149975e+02]
 [-7.86170506e+01  1.31819169e+02]
 [ 8.76779906e+01  3.81091881e+02]
 [-4.41745679e+02  5.30596521e+02]
 [-6.12716291e+02  9.18454310e+01]
 [ 9.97247671e+02  2.35126284e+02]
 [-3.53450300e+02  8.90354609e+01]
 [-4.13640350e+02 -1.54220496e+02]
 [ 4.17284487e+02  3.07494644e+02]
 [ 1.35637164e+02 -3.35395311e+01]
 [-6.26184049e+02  2.71886301e+01]
 [-2.60462349e+02  1.59936500e+02]
 [-1.81697729e+02  5.91968569e+02]
 [ 5.38782617e+01  1.13392561e+02]
 [ 2.28638616e+02 -4.12369607e+02]
 [-6.33896995e+01  2.14824526e+03]
 [-7.29613708e+01  4.08980328e+02]
 [-5.00329510e+01  5.73386969e+02]
 [-5.45256721e+02 -4.55935371e+01]
 [ 6.83594334e+02 -1.75893813e+02]
 [ 1.48598468e+01  7.63754195e+01]
 [-2.56061762e+02 -8.55773187e+01]
 [-7.07151708e+02 -2.54463663e+02]
 [-3.75357957e+02  4.83420610e+02]
 [ 4.44080908e+02 -4.34659333e+02]
 [-5.11965385e+02 -3.35483428e+02]
 [-7.37294956e+01  8.41511619e+01]
```

```
 [-1.04141636e+02 -1.33425599e+02]
 [ 4.46826041e+01  3.01472719e+02]
 [ 3.19447401e+02 -1.51684965e+02]
 [ 1.59557482e+02 -1.74511234e+02]
 [ 1.17341961e+03 -8.79659484e+02]
 [ 8.10382401e+02 -8.93383832e+02]
 [-5.82094125e+01 -7.22823591e+01]
 [ 6.37738485e+02  4.09917783e+02]
 [-3.98418249e+00  2.77625166e+02]
 [-5.10028791e+02 -1.90928950e+01]
 [ 2.93638678e+02 -1.21395304e+02]
 [-3.18789482e+02  2.81122612e+02]
 [ 4.24076916e+02 -4.36317576e+02]
 [-5.51453419e+02 -3.49898634e+02]
 [ 1.32190854e+02  8.44733937e+01]
 [-4.59981664e+02  2.48532761e+02]
 [ 7.59601294e+02  1.09302361e+02]
 [-1.93159936e+02  1.20780907e+02]
 [ 1.71949975e+02  1.73873931e+02]
 [-8.00872509e+01  2.77597990e+02]
 [-3.43060773e+02  1.70464776e+02]
 [ 4.34044871e+02  1.42809100e+02]
 [ 1.84790963e+02  9.69106130e+01]
 [-3.36981721e+02  1.92938468e+02]
 [-2.94970157e+02  5.06089797e+02]
 [ 8.49086360e+01  4.85669792e+02]
 [ 2.84749096e+02  5.87440989e+02]
 [-2.85900035e+02  4.82575488e+02]
 [ 3.02903360e+02  1.13475172e+03]
 [-4.23195146e+02 -4.10182291e+02]
 [ 5.05982776e+02 -7.55836531e+02]
 [ 5.45780435e+02 -4.34393282e+02]
 [ 2.90589995e+02  7.26964017e+01]
 [ 6.37871561e+02 -3.11026489e+02]
 [ 9.99679352e+01  1.67821668e+02]
 [ 6.88144759e+00  7.80638707e+02]
 [-3.60745403e+02  3.45946863e+02]
 [ 2.04143819e+02 -8.20074308e-01]
 [-2.73842163e+01  3.69941344e+02]
 [ 4.16420028e+02  2.56886537e+02]
 [-1.64074014e+02 -5.17209111e+01]
 [ 7.25111540e+02  1.86738038e+02]
 [-3.16590678e+02  4.14248034e+02]
 [-1.45006598e+02  5.84175698e+02]
 [-3.88692472e+02 -1.57387256e+02]
 [ 3.37587130e+02 -1.18728316e+02]
 [ 4.42290480e+02 -4.25538689e+02]
 [-1.70839946e+02  1.85787156e+00]
 [ 6.13327233e+02 -2.67897953e+01]
```

```
 [ 1.47713361e+02 -2.02064432e+02]
 [ 6.57805797e+02  1.50046539e+02]
 [-1.81919153e+02  9.55194293e+02]
 [-2.11992269e+02  2.45094469e+02]
 [ 6.39483614e+01  3.93776699e+02]
 [ 3.75106563e+02  1.82533038e+02]
 [ 4.55042964e+02 -3.27850988e+02]
 [ 9.77205235e+02 -2.10994591e+02]
 [-3.69356634e+02  4.46386911e+01]
 [ 1.97901417e+02  2.17170289e+02]
 [ 3.54601242e+02  1.80579100e+02]
 [ 8.08133385e+01 -1.51100187e+02]
 [ 6.37642597e+02 -1.85102898e+02]
 [ 7.02829435e+01 -1.79718636e+02]
 [ 1.87272199e+02  7.70144815e+00]
 [ 4.71666118e+02  1.46308584e+02]
 [ 6.06233258e+02 -6.78017623e+02]
 [ 2.58583939e+02  9.55815512e+01]
 [ 2.23993654e+01  8.00480419e+01]
 [ 1.84286007e+02  2.48711674e+02]
 [ 6.59992445e+02  1.68211945e+02]
 [-3.75813503e+02  1.92533277e+02]
 [ 4.10481389e+02 -1.20080552e+01]
 [-3.67156562e+02  2.16603088e+02]
 [-5.70817013e+02  3.22454995e+01]]
Reduced Histograms for n02111277-Newfoundland: [[-6.45722874e+00
1.10550714e+02]
 [ 8.89334820e+02 -5.32208195e+02]
 [ 1.96080719e+02 -9.05953749e+02]
 [ 7.78174161e+02 -6.36961548e+02]
 [-3.21242148e+02  2.62820555e+01]
 [ 1.32281392e+02  2.80299216e+02]
 [ 7.50549271e+02 -3.23945959e+02]
 [ 3.82046641e+02 -1.72846387e+02]
 [ 5.76580486e+02 -4.20348193e+02]
 [ 1.62632786e+02  1.75614909e+02]
 [ 1.12125207e+03 -9.20724783e+02]
 [ 9.49737690e+02 -5.47107359e+02]
 [ 1.17273690e+03  1.59095330e+02]
 [-2.29176143e+02 -4.26017667e+01]
 [ 2.47328516e+02  2.80612445e+02]
 [ 6.74265233e+02  4.80980696e+02]
 [ 1.21827947e+03 -3.71961344e+02]
 [ 1.13970425e+03  3.39169220e+02]
 [ 9.01304555e+02  2.06766769e+02]
 [ 9.47852930e+02  1.62432497e+02]
 [-4.73513703e+02  4.48366866e+02]
 [-6.95237971e+02 -1.21473422e+02]
 [ 4.43585822e+02 -3.39985559e+02]
```

```
 [ 4.95590065e+02 -1.49741708e+02]
 [ 1.88996730e+02  6.75998419e+02]
 [-2.46122214e+02  1.23430848e+03]
 [ 5.46821512e+02  1.77354988e+02]
 [ 8.33040857e+02 -2.32243780e+02]
 [ 5.15899927e+02 -5.92563387e+02]
 [ 1.74527639e+02  3.16759825e+02]
 [ 3.52044354e+01  4.65876589e+02]
 [-1.19964902e+02  2.57738345e+00]
 [ 7.08926910e+02 -1.53451326e+03]
 [ 5.80279434e+02  2.96934998e+02]
 [-3.02457382e+02  3.67400134e+02]
 [ 1.17276700e+02  2.04600861e+02]
 [ 2.70133850e+02 -4.66262997e+02]
 [ 1.42016418e+02  4.60694689e+02]
 [-4.33115831e+02  8.47275842e+02]
 [ 5.81822675e+02 -3.37255756e+02]
 [ 5.37569412e+02  2.54932336e+02]
 [ 5.00251703e+02  2.01908880e+01]
 [ 8.92603135e+02 -7.26071438e+02]
 [ 9.21184784e+02  1.62490306e+02]
 [ 1.24223870e+02 -4.79711289e+02]
 [ 2.66739239e+02  2.83187541e+01]
 [ 5.14513874e+02  4.66867366e+02]
 [ 8.31416333e+01 -3.58403127e+02]
 [ 1.72028166e+02  9.52633760e+00]
 [-4.47089652e+02 -2.70074361e+01]
 [-5.30471177e+02 -3.63234702e+02]
 [ 3.52486567e+02  4.09754528e+02]
 [ 1.20209593e+03 -3.11510028e+02]
 [-2.87861921e+00  9.55149793e+01]
 [-4.81183388e+02 -2.05589503e+02]
 [ 4.76304764e+02 -1.29966409e+02]
 [ 3.79979655e+02 -3.94646079e+02]
 [ 3.69178864e+02 -5.19436265e+02]
 [ 2.81710188e+02  1.89909867e+02]
 [-3.54743411e+01 -2.59064926e+02]
 [-3.64785495e+02 -7.35034160e-01]
 [-7.12710326e+02  6.33234797e+00]
 [ 1.60352368e+03 -1.32204553e+03]
 [ 1.78599998e+02  1.16712778e+02]
 [-6.60293259e+02  3.84601281e+01]
 [-4.24951967e+02  9.44521489e+01]
 [ 1.62338517e+02  8.43899327e+01]
 [ 1.74150350e+03 -1.24777142e+03]
 [ 1.16780165e+03 -3.75262747e+02]
 [ 1.08438601e+03  2.30946769e+02]
 [ 1.16987033e+03 -6.08232860e+02]
 [-5.02931122e+02 -1.68306242e+02]
```

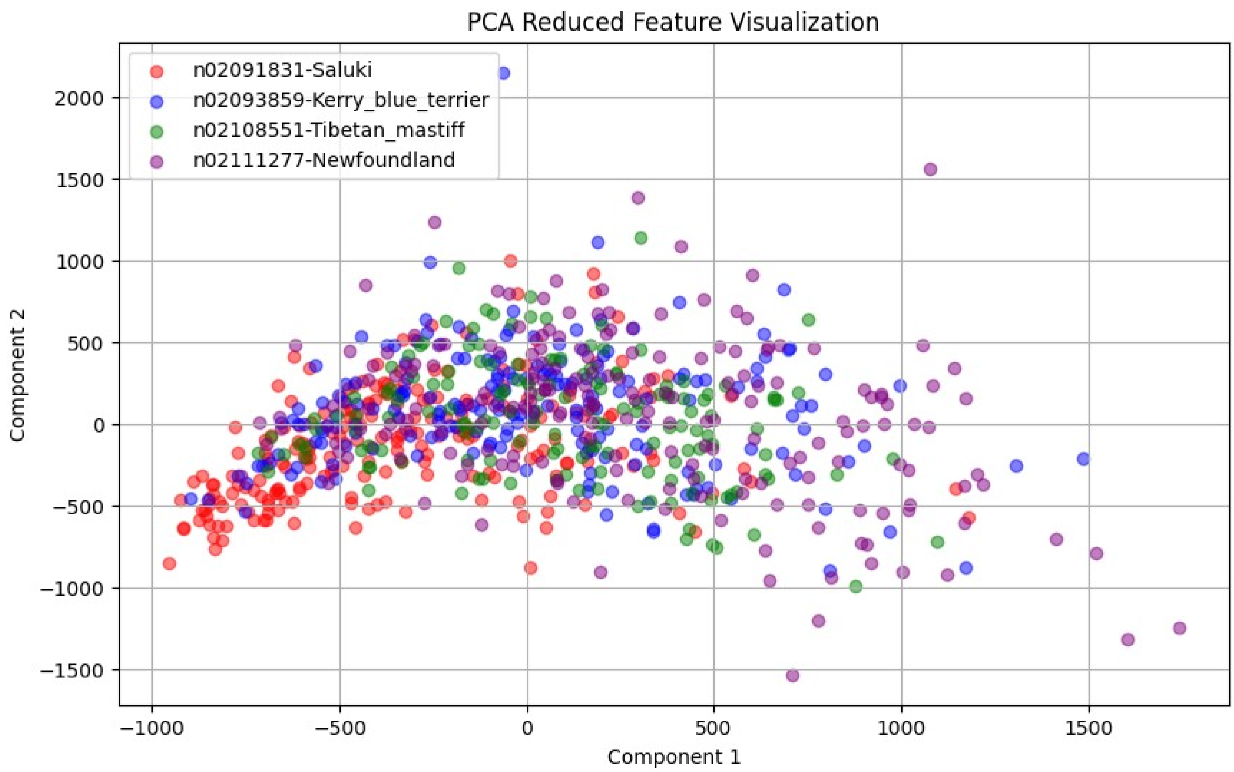
```
 [-2.96291028e+02  9.25721273e+01]
 [ 2.01372557e+02  8.23352633e+02]
 [-6.95682048e+01 -2.51956877e+02]
 [ 2.35232680e+02 -1.98015189e+02]
 [ 3.79486003e+02  6.17918431e+00]
 [ 1.00333372e+03 -9.03817866e+02]
 [ 3.26878201e+02  2.71606537e+02]
 [-1.21656354e+02 -6.11876776e+02]
 [ 6.67013472e+02  2.51171292e+02]
 [ 9.47460223e+02  1.85541353e+02]
 [ 3.56567570e+02  6.73812645e+02]
 [ 1.63683709e+01  3.14641522e+02]
 [ 7.50829524e+01  3.42085450e+02]
 [ 5.72876016e+01  2.11237722e+02]
 [ 1.03396696e+03 -7.22658432e-02]
 [-1.20676661e+02  1.45587838e+02]
 [-1.14287494e+02  1.94706970e+02]
 [-4.10722677e+02  3.54186466e+02]
 [ 8.53244179e+02 -4.26025724e+01]
 [-2.37785096e+02  4.79917438e+02]
 [ 8.00335559e+01 -3.43738360e+02]
 [ 7.65006439e+02  4.64919404e+02]
 [ 4.77682143e+02  1.72114703e+01]
 [ 6.15901164e+02  2.32695934e+02]
 [-2.51931764e+02  3.56549989e+02]
 [ 6.46475186e+02 -9.57111353e+02]
 [-7.62582305e+02 -3.22032324e+02]
 [-9.68790756e+01  4.82423331e+02]
 [ 3.24292441e+02  7.52649416e+01]
 [ 1.07286776e+03 -2.05160483e+01]
 [-1.76570310e+02 -3.18600144e+02]
 [-3.79588103e+02 -6.22736944e+01]
 [ 8.54475073e+01  4.96832150e+01]
 [-5.11408269e+02 -1.19164543e+02]
 [ 7.84614957e+00  1.41177070e+02]
 [ 5.95426045e+01  5.75428225e+02]
 [ 4.56164986e+01  2.08207306e+02]
 [ 8.12332460e+02 -9.44050388e+02]
 [ 1.07479279e+03  1.56134336e+03]
 [ 9.53689684e+02  4.40217326e-01]
 [ 5.59368334e+02  6.91899503e+02]
 [-3.24490622e+02 -2.62528261e+01]
 [ 6.38205404e+02 -7.73399185e+02]
 [ 4.12440432e+02  1.08586473e+03]
 [-5.19899793e+02  1.14892038e+02]
 [ 3.69446584e+02  1.94736013e+02]
 [ 1.10837097e+02  6.82187520e+02]
 [ 3.88558642e+02  1.62620419e+02]
 [ 8.97549690e+01  6.45204390e+01]
```

```
 [-2.75651969e+02 -4.86869786e+02]
 [ 5.55607613e+02  4.46160760e+02]
 [ 6.38747342e+02  4.52241175e+02]
 [ 2.81022747e+02  5.84523324e+02]
 [ 4.09135402e+01  5.40234553e+02]
 [-1.57820644e+02 -2.49508000e+02]
 [-3.49302935e+02  2.95793740e+02]
 [ 3.88680369e+01  3.73052581e+02]
 [-3.23942470e+02  3.54888987e+02]
 [ 6.01599281e+02  9.05401882e+02]
 [-2.06948124e+02 -1.81921469e+02]
 [ 2.95829757e+02  1.38249643e+03]
 [ 1.01970619e+03 -2.86381808e+02]
 [ 1.02143724e+03 -4.91609053e+02]
 [ 2.20897450e+02  6.85117174e+02]
 [ 7.79348219e+02 -1.19631152e+02]
 [-7.81833011e+01  8.17303661e+02]
 [ 7.34738035e+00  1.94144324e+02]
 [ 4.83007577e+01  2.05426083e+02]
 [ 2.10542430e+02  5.48210316e+02]
 [-4.23668816e+02  2.65207463e+02]
 [ 1.70379633e+02 -8.72854415e+01]
 [ 4.39459158e+02 -2.48613778e+02]
 [ 2.21842533e+02  5.74916883e+02]
 [ 3.15909207e+02  2.44158552e+00]
 [ 1.46222370e+01  6.85112513e+01]
 [ 6.66712930e+02 -4.91642221e+02]
 [ 4.71445245e+02  7.59120157e+02]
 [ 8.95519944e+02 -1.28946450e+01]
 [-3.41320574e+02 -2.10536647e+02]
 [ 1.41499172e+03 -7.08308484e+02]
 [ 8.59056634e+01  5.32964379e+02]
 [ 5.97461037e+02  1.34074272e+02]
 [ 1.53282907e+02  2.35351879e+02]
 [-3.83976972e+01 -1.91777676e+02]
 [ 1.77059850e+02  1.15577216e+02]
 [ 7.69353059e+01  8.72251163e+02]
 [-3.09044508e+02  3.20109693e+02]
 [-2.21001799e+02  4.91371074e+02]
 [ 1.39854256e+01  4.23875726e+02]
 [ 6.07035369e+02 -8.76695388e+01]
 [-6.57938786e+02 -2.47301153e+01]
 [ 4.25233119e+01  2.61214904e+02]
 [ 3.81033650e+01 -2.32643424e+02]]
```

```
histograms.keys()
```

```
dict_keys(['n02091831-Saluki', 'n02093859-Kerry_blue_terrier',
'n02108551-Tibetan_mastiff', 'n02111277-Newfoundland'])
```

```
colors = ['red', 'blue', 'green', 'purple']
labels = list(histograms.keys())

plt.figure(figsize=(10, 6))
start_idx = 0
for idx, (class_name, class_hists) in enumerate(histograms.items()):
    end_idx = start_idx + len(class_hists)
    plt.scatter(hist_reduced[start_idx:end_idx, 0],
hist_reduced[start_idx:end_idx, 1],
                c=colors[idx], label=class_name, alpha=0.5)
    start_idx = end_idx

plt.xlabel('Component 1')
plt.ylabel('Component 2')
plt.title('PCA Reduced Feature Visualization')
plt.grid(True)
plt.legend()
plt.show()
```



From the above plot, we can observe that all the classes are intersecting and cannot be separable. But the red points belong to the class "n02091831-Saluki", can be visually set apart in lesser amount comparing to the others at the bottom left. Where the other classes are mostly overlap each other in the center of the plot and hence they cannot visually separable

Reading the training set

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

# Loading the dataset
file_path = r'C:\Users\V Varunkumar\Desktop\dm programming\student_31\
train.json'
with open(file_path, 'r') as file:
    data = [json.loads(line) for line in file]

# Extracting the tweets
tweets = [entry['Tweet'] for entry in data]

# Initialize the vectorizers and transform the data
count_vectorizer = CountVectorizer()
tfidf_vectorizer = TfidfVectorizer()
count_matrix = count_vectorizer.fit_transform(tweets)
tfidf_matrix = tfidf_vectorizer.fit_transform(tweets)
#count_features = count_vectorizer.get_feature_names_out()

print("Token Feature Count Matrix for the first 10 documents:")
for i in range(min(10, len(tweets))):  # prints up to 10 for
verification
    print(f"\nDocument {i+1} token counts:")
    print(count_matrix.toarray()[i])  # Print the token counts for
selected document

print("\nTF-IDF Feature Count Matrix for the first 10 documents:")
for i in range(min(10, len(tweets))):  # prints up to 10 for
verification
    print(f"\nDocument {i+1} TF-IDF counts:")
    print(tfidf_matrix.toarray()[i])

print(f"\nCountVectorizer dimension: {count_matrix.shape}")
print(f"TfidfVectorizer dimension: {tfidf_matrix.shape}")
```

```
Token Feature Count Matrix for the first 10 documents:

Document 1 token counts:
[0 0 0 ... 0 0 0]

Document 2 token counts:
[0 0 0 ... 0 0 0]

Document 3 token counts:
[0 0 0 ... 0 0 0]
```

```
Document 4 token counts:
[0 0 0 ... 0 0 0]

Document 5 token counts:
[0 0 0 ... 0 0 0]

Document 6 token counts:
[0 0 0 ... 0 0 0]

Document 7 token counts:
[0 0 0 ... 0 0 0]

Document 8 token counts:
[0 0 0 ... 0 0 0]

Document 9 token counts:
[0 0 0 ... 0 0 0]

Document 10 token counts:
[0 0 0 ... 0 0 0]

TF-IDF Feature Count Matrix for the first 10 documents:

Document 1 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 2 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 3 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 4 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 5 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 6 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 7 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 8 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 9 TF-IDF counts:
[0. 0. 0. ... 0. 0. 0.]

Document 10 TF-IDF counts:
```

```
[0. 0. 0. ... 0. 0. 0.]

CountVectorizer dimension: (3000, 9619)
TfidfVectorizer dimension: (3000, 9619)
```

```
labels = [] #Creating four separable classes
for entry in data:
    if entry['joy']:
        labels.append('Joy')
    elif entry['anger']:
        labels.append('Anger')
    elif entry['sadness']:
        labels.append('Sadness')
    elif entry['surprise']:
        labels.append('Surprise')
    else:
        labels.append('Other')

df = pd.DataFrame({'Tweet': tweets, 'Emotion': labels})# Creating a
DataFrame

# Filter the DataFrame to separate the selected emotions
selected_emotions = ['Joy', 'Anger', 'Sadness', 'Surprise']
df_filtered = df[df['Emotion'].isin(selected_emotions)]

print("\nSelected Classes:")# Print the selected classes
for emotion in selected_emotions:
    print(f"- {emotion}")
```

```
Selected Classes:
- Joy
- Anger
- Sadness
- Surprise
```

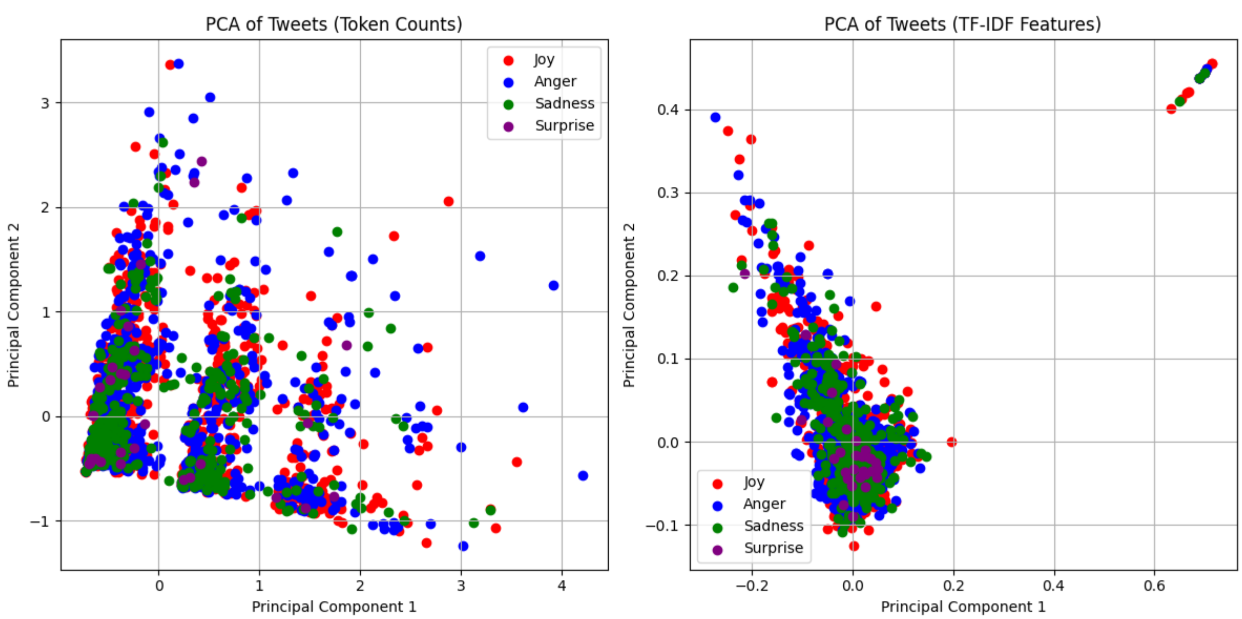
```
# Perform PCA to reduce dimensionality to 2 components
pca = PCA(n_components=2)
reduced_count_data = pca.fit_transform(count_matrix.toarray())
pca_tfidf = PCA(n_components=2)
reduced_tfidf_data = pca_tfidf.fit_transform(tfidf_matrix.toarray())
reduced_count_df = pd.DataFrame(data=reduced_count_data,
columns=['PCA1', 'PCA2'])
reduced_count_df['Emotion'] =
df_filtered['Emotion'].reset_index(drop=True)

reduced_tfidf_df = pd.DataFrame(data=reduced_tfidf_data,
columns=['PCA1', 'PCA2'])
reduced_tfidf_df['Emotion'] =
df_filtered['Emotion'].reset_index(drop=True)
print("\nReduced PCA Data:")# Print the reduced data
```

```
print(reduced_count_df)
print("\nReduced PCA Data:")
print(reduced_tfidf_df)
```

```
Reduced PCA Data:
          PCA1      PCA2  Emotion
0     0.325822 -0.332136      Joy
1    -0.682282 -0.431461      Joy
2    -0.397113 -0.019127    Anger
3    -0.695044 -0.453337    Anger
4    -0.631551 -0.361290  Sadness
...        ...       ...      ...
2995 -0.542532 -0.379673      NaN
2996 -0.317720  0.477656      NaN
2997  0.674851  0.208649      NaN
2998  0.689920  1.049253      NaN
2999  0.722780 -0.499146      NaN

[3000 rows x 3 columns]

Reduced PCA Data:
          PCA1      PCA2  Emotion
0     0.016749  0.005161      Joy
1     0.039115 -0.025404      Joy
2    -0.005184 -0.032009    Anger
3     0.046470 -0.027785    Anger
4     0.004575  0.000867  Sadness
...        ...       ...      ...
2995  0.029429 -0.020061      NaN
2996 -0.038815 -0.021252      NaN
2997 -0.003491 -0.052665      NaN
2998 -0.181667  0.206594      NaN
2999 -0.027730 -0.064546      NaN

[3000 rows x 3 columns]
```

```
colors = ['red', 'blue', 'green', 'purple']
# Plot for token counts
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
for i,emotion in enumerate(selected_emotions):
    plt.scatter(reduced_count_df[reduced_count_df['Emotion'] ==
emotion]['PCA1'],
                reduced_count_df[reduced_count_df['Emotion'] ==
emotion]['PCA2'],
                label=emotion, c=colors[i])

plt.title('PCA of Tweets (Token Counts)')
plt.xlabel('Principal Component 1')
```

```
plt.ylabel('Principal Component 2')
plt.legend()
plt.grid()

# Plot for TF-IDF
plt.subplot(1, 2, 2)
for i,emotion in enumerate(selected_emotions):
    plt.scatter(reduced_tfidf_df[reduced_tfidf_df['Emotion'] ==
emotion]['PCA1'],
                reduced_tfidf_df[reduced_tfidf_df['Emotion'] ==
emotion]['PCA2'],
                label=emotion, c=colors[i])

plt.title('PCA of Tweets (TF-IDF Features)')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
plt.grid()

plt.tight_layout()
plt.show()
```



In both the plots, no classes can visually separable because all the the classes are overlapped with each other.