ccuds4s7s

February 14, 2024

0.1 2a>

[9]:
```
import matplotlib.pyplot as plt
import os
import cv2
from pathlib import Path
from skimage.color import rgb2gray
from skimage import io, exposure
import xml.etree.ElementTree as ET
from PIL import Image

myBreeds = [
    'n02099712-Labrador_retriever',
    'n02110185-Siberian_husky',
    'n02113799-standard_poodle',
    'n02113186-Cardigan'
    ]
```

[10]:
```
class AllRequiredPaths:
    myBreeds = [
    'n02099712-Labrador_retriever',
    'n02110185-Siberian_husky',
    'n02113799-standard_poodle',
    'n02113186-Cardigan'
    ]

    root_dir = ""
    imgs_folder = ""
    annotation_folder = ""
    imgs_folder_sub_folders = [f"images/{name}" for name in myBreeds]



    def __init__(self):
        self.root_dir = os.getcwd()
        self.imgs_folder = os.path.join(self.root_dir, "images")
        self.annotation_folder = os.path.join(self.root_dir, "Annotation")
```

```
class CropAndResizeImages:
    def __init__(self):
        self.all_paths = AllRequiredPaths()

    def _get_bbox(self, annot):
        xml = annot
        tree = ET.parse(xml)
        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

    def _get_images_from_classes(self, annot):
        image_name = f"{os.path.basename(annot)}.jpg"
        breed_name = os.path.basename(os.path.dirname(annot))
        return os.path.join(self.all_paths.imgs_folder, breed_name, image_name)

    def _crop_save_images(self, source_path, boxes):
        original_image = Image.open(source_path)
        for index, bounding_box in enumerate(boxes):
            cropped_img = original_image.crop(bounding_box).resize((128, 128),␣
 ↪Image.LANCZOS)
            original_filename = os.path.basename(source_path)
            new_file_name = f"{os.path.
 ↪splitext(original_filename)[0]}_section_{index}{os.path.
 ↪splitext(original_filename)[1]}"
            save_path = os.path.join("CroppedImagesFolder", new_file_name)
            Path(save_path).parent.mkdir(parents=True, exist_ok=True)
            cropped_img.convert('RGB').save(save_path)

    def process(self, myBreeds):
        annotation_files = []

        for myBreed in myBreeds:
            for file_name in os.listdir(os.path.join(self.all_paths.
 ↪annotation_folder, myBreed)):
                if not os.path.isdir(os.path.join(self.all_paths.
 ↪annotation_folder, myBreed, file_name)):
```

```
                    _path = os.path.join(self.all_paths.annotation_folder,␣
↪myBreed, file_name)
                    annotation_files.append(_path)

        for i, annot in enumerate(annotation_files, start = 1):
            bboxes = self._get_bbox(annot)
            img_file_path = self._get_images_from_classes(annot)

            if not os.path.exists(img_file_path):
                continue

            print(f"Processing: {i}/{len(annotation_files)}")
            self._crop_save_images(img_file_path, bboxes)


crop_and_resize_img = CropAndResizeImages()
crop_and_resize_img.process(myBreeds)
```

```
Processing: 1/677
Processing: 2/677
Processing: 3/677
Processing: 4/677
Processing: 5/677
Processing: 6/677
Processing: 7/677
Processing: 8/677
Processing: 9/677
Processing: 10/677
Processing: 11/677
Processing: 12/677
Processing: 13/677
Processing: 14/677
Processing: 15/677
Processing: 16/677
Processing: 17/677
Processing: 18/677
Processing: 19/677
Processing: 20/677
Processing: 21/677
Processing: 22/677
Processing: 23/677
Processing: 24/677
Processing: 25/677
Processing: 26/677
Processing: 27/677
Processing: 28/677
Processing: 29/677
```

```
Processing: 30/677
Processing: 31/677
Processing: 32/677
Processing: 33/677
Processing: 34/677
Processing: 35/677
Processing: 36/677
Processing: 37/677
Processing: 38/677
Processing: 39/677
Processing: 40/677
Processing: 41/677
Processing: 42/677
Processing: 43/677
Processing: 44/677
Processing: 45/677
Processing: 46/677
Processing: 47/677
Processing: 48/677
Processing: 49/677
Processing: 50/677
Processing: 51/677
Processing: 52/677
Processing: 53/677
Processing: 54/677
Processing: 55/677
Processing: 56/677
Processing: 57/677
Processing: 58/677
Processing: 59/677
Processing: 60/677
Processing: 61/677
Processing: 62/677
Processing: 63/677
Processing: 64/677
Processing: 65/677
Processing: 66/677
Processing: 67/677
Processing: 68/677
Processing: 69/677
Processing: 70/677
Processing: 71/677
Processing: 72/677
Processing: 73/677
Processing: 74/677
Processing: 75/677
Processing: 76/677
Processing: 77/677
```

```
Processing: 78/677
Processing: 79/677
Processing: 80/677
Processing: 81/677
Processing: 82/677
Processing: 83/677
Processing: 84/677
Processing: 85/677
Processing: 86/677
Processing: 87/677
Processing: 88/677
Processing: 89/677
Processing: 90/677
Processing: 91/677
Processing: 92/677
Processing: 93/677
Processing: 94/677
Processing: 95/677
Processing: 96/677
Processing: 97/677
Processing: 98/677
Processing: 99/677
Processing: 100/677
Processing: 101/677
Processing: 102/677
Processing: 103/677
Processing: 104/677
Processing: 105/677
Processing: 106/677
Processing: 107/677
Processing: 108/677
Processing: 109/677
Processing: 110/677
Processing: 111/677
Processing: 112/677
Processing: 113/677
Processing: 114/677
Processing: 115/677
Processing: 116/677
Processing: 117/677
Processing: 118/677
Processing: 119/677
Processing: 120/677
Processing: 121/677
Processing: 122/677
Processing: 123/677
Processing: 124/677
Processing: 125/677
```

```
Processing: 126/677
Processing: 127/677
Processing: 128/677
Processing: 129/677
Processing: 130/677
Processing: 131/677
Processing: 132/677
Processing: 133/677
Processing: 134/677
Processing: 135/677
Processing: 136/677
Processing: 137/677
Processing: 138/677
Processing: 139/677
Processing: 140/677
Processing: 141/677
Processing: 142/677
Processing: 143/677
Processing: 144/677
Processing: 145/677
Processing: 146/677
Processing: 147/677
Processing: 148/677
Processing: 149/677
Processing: 150/677
Processing: 151/677
Processing: 152/677
Processing: 153/677
Processing: 154/677
Processing: 155/677
Processing: 156/677
Processing: 157/677
Processing: 158/677
Processing: 159/677
Processing: 160/677
Processing: 161/677
Processing: 162/677
Processing: 163/677
Processing: 164/677
Processing: 165/677
Processing: 166/677
Processing: 167/677
Processing: 168/677
Processing: 169/677
Processing: 170/677
Processing: 171/677
Processing: 172/677
Processing: 173/677
```

```
Processing: 174/677
Processing: 175/677
Processing: 176/677
Processing: 177/677
Processing: 178/677
Processing: 179/677
Processing: 180/677
Processing: 181/677
Processing: 182/677
Processing: 183/677
Processing: 184/677
Processing: 185/677
Processing: 186/677
Processing: 187/677
Processing: 188/677
Processing: 189/677
Processing: 190/677
Processing: 191/677
Processing: 192/677
Processing: 193/677
Processing: 194/677
Processing: 195/677
Processing: 196/677
Processing: 197/677
Processing: 198/677
Processing: 199/677
Processing: 200/677
Processing: 201/677
Processing: 202/677
Processing: 203/677
Processing: 204/677
Processing: 205/677
Processing: 206/677
Processing: 207/677
Processing: 208/677
Processing: 209/677
Processing: 210/677
Processing: 211/677
Processing: 212/677
Processing: 213/677
Processing: 214/677
Processing: 215/677
Processing: 216/677
Processing: 217/677
Processing: 218/677
Processing: 219/677
Processing: 220/677
Processing: 221/677
```

```
Processing: 222/677
Processing: 223/677
Processing: 224/677
Processing: 225/677
Processing: 226/677
Processing: 227/677
Processing: 228/677
Processing: 229/677
Processing: 230/677
Processing: 231/677
Processing: 232/677
Processing: 233/677
Processing: 234/677
Processing: 235/677
Processing: 236/677
Processing: 237/677
Processing: 238/677
Processing: 239/677
Processing: 240/677
Processing: 241/677
Processing: 242/677
Processing: 243/677
Processing: 244/677
Processing: 245/677
Processing: 246/677
Processing: 247/677
Processing: 248/677
Processing: 249/677
Processing: 250/677
Processing: 251/677
Processing: 252/677
Processing: 253/677
Processing: 254/677
Processing: 255/677
Processing: 256/677
Processing: 257/677
Processing: 258/677
Processing: 259/677
Processing: 260/677
Processing: 261/677
Processing: 262/677
Processing: 263/677
Processing: 264/677
Processing: 265/677
Processing: 266/677
Processing: 267/677
Processing: 268/677
Processing: 269/677
```

```
Processing: 270/677
Processing: 271/677
Processing: 272/677
Processing: 273/677
Processing: 274/677
Processing: 275/677
Processing: 276/677
Processing: 277/677
Processing: 278/677
Processing: 279/677
Processing: 280/677
Processing: 281/677
Processing: 282/677
Processing: 283/677
Processing: 284/677
Processing: 285/677
Processing: 286/677
Processing: 287/677
Processing: 288/677
Processing: 289/677
Processing: 290/677
Processing: 291/677
Processing: 292/677
Processing: 293/677
Processing: 294/677
Processing: 295/677
Processing: 296/677
Processing: 297/677
Processing: 298/677
Processing: 299/677
Processing: 300/677
Processing: 301/677
Processing: 302/677
Processing: 303/677
Processing: 304/677
Processing: 305/677
Processing: 306/677
Processing: 307/677
Processing: 308/677
Processing: 309/677
Processing: 310/677
Processing: 311/677
Processing: 312/677
Processing: 313/677
Processing: 314/677
Processing: 315/677
Processing: 316/677
Processing: 317/677
```

```
Processing: 318/677
Processing: 319/677
Processing: 320/677
Processing: 321/677
Processing: 322/677
Processing: 323/677
Processing: 324/677
Processing: 325/677
Processing: 326/677
Processing: 327/677
Processing: 328/677
Processing: 329/677
Processing: 330/677
Processing: 331/677
Processing: 332/677
Processing: 333/677
Processing: 334/677
Processing: 335/677
Processing: 336/677
Processing: 337/677
Processing: 338/677
Processing: 339/677
Processing: 340/677
Processing: 341/677
Processing: 342/677
Processing: 343/677
Processing: 344/677
Processing: 345/677
Processing: 346/677
Processing: 347/677
Processing: 348/677
Processing: 349/677
Processing: 350/677
Processing: 351/677
Processing: 352/677
Processing: 353/677
Processing: 354/677
Processing: 355/677
Processing: 356/677
Processing: 357/677
Processing: 358/677
Processing: 359/677
Processing: 360/677
Processing: 361/677
Processing: 362/677
Processing: 363/677
Processing: 364/677
Processing: 365/677
```

```
Processing: 366/677
Processing: 367/677
Processing: 368/677
Processing: 369/677
Processing: 370/677
Processing: 371/677
Processing: 372/677
Processing: 373/677
Processing: 374/677
Processing: 375/677
Processing: 376/677
Processing: 377/677
Processing: 378/677
Processing: 379/677
Processing: 380/677
Processing: 381/677
Processing: 382/677
Processing: 383/677
Processing: 384/677
Processing: 385/677
Processing: 386/677
Processing: 387/677
Processing: 388/677
Processing: 389/677
Processing: 390/677
Processing: 391/677
Processing: 392/677
Processing: 393/677
Processing: 394/677
Processing: 395/677
Processing: 396/677
Processing: 397/677
Processing: 398/677
Processing: 399/677
Processing: 400/677
Processing: 401/677
Processing: 402/677
Processing: 403/677
Processing: 404/677
Processing: 405/677
Processing: 406/677
Processing: 407/677
Processing: 408/677
Processing: 409/677
Processing: 410/677
Processing: 411/677
Processing: 412/677
Processing: 413/677
```

```
Processing: 414/677
Processing: 415/677
Processing: 416/677
Processing: 417/677
Processing: 418/677
Processing: 419/677
Processing: 420/677
Processing: 421/677
Processing: 422/677
Processing: 423/677
Processing: 424/677
Processing: 425/677
Processing: 426/677
Processing: 427/677
Processing: 428/677
Processing: 429/677
Processing: 430/677
Processing: 431/677
Processing: 432/677
Processing: 433/677
Processing: 434/677
Processing: 435/677
Processing: 436/677
Processing: 437/677
Processing: 438/677
Processing: 439/677
Processing: 440/677
Processing: 441/677
Processing: 442/677
Processing: 443/677
Processing: 444/677
Processing: 445/677
Processing: 446/677
Processing: 447/677
Processing: 448/677
Processing: 449/677
Processing: 450/677
Processing: 451/677
Processing: 452/677
Processing: 453/677
Processing: 454/677
Processing: 455/677
Processing: 456/677
Processing: 457/677
Processing: 458/677
Processing: 459/677
Processing: 460/677
Processing: 461/677
```

```
Processing: 462/677
Processing: 463/677
Processing: 464/677
Processing: 465/677
Processing: 466/677
Processing: 467/677
Processing: 468/677
Processing: 469/677
Processing: 470/677
Processing: 471/677
Processing: 472/677
Processing: 473/677
Processing: 474/677
Processing: 475/677
Processing: 476/677
Processing: 477/677
Processing: 478/677
Processing: 479/677
Processing: 480/677
Processing: 481/677
Processing: 482/677
Processing: 483/677
Processing: 484/677
Processing: 485/677
Processing: 486/677
Processing: 487/677
Processing: 488/677
Processing: 489/677
Processing: 490/677
Processing: 491/677
Processing: 492/677
Processing: 493/677
Processing: 494/677
Processing: 495/677
Processing: 496/677
Processing: 497/677
Processing: 498/677
Processing: 499/677
Processing: 500/677
Processing: 501/677
Processing: 502/677
Processing: 503/677
Processing: 504/677
Processing: 505/677
Processing: 506/677
Processing: 507/677
Processing: 508/677
Processing: 509/677
```

```
Processing: 510/677
Processing: 511/677
Processing: 512/677
Processing: 513/677
Processing: 514/677
Processing: 515/677
Processing: 516/677
Processing: 517/677
Processing: 518/677
Processing: 519/677
Processing: 520/677
Processing: 521/677
Processing: 522/677
Processing: 523/677
Processing: 524/677
Processing: 525/677
Processing: 526/677
Processing: 527/677
Processing: 528/677
Processing: 529/677
Processing: 530/677
Processing: 531/677
Processing: 532/677
Processing: 533/677
Processing: 534/677
Processing: 535/677
Processing: 536/677
Processing: 537/677
Processing: 538/677
Processing: 539/677
Processing: 540/677
Processing: 541/677
Processing: 542/677
Processing: 543/677
Processing: 544/677
Processing: 545/677
Processing: 546/677
Processing: 547/677
Processing: 548/677
Processing: 549/677
Processing: 550/677
Processing: 551/677
Processing: 552/677
Processing: 553/677
Processing: 554/677
Processing: 555/677
Processing: 556/677
Processing: 557/677
```

```
Processing: 558/677
Processing: 559/677
Processing: 560/677
Processing: 561/677
Processing: 562/677
Processing: 563/677
Processing: 564/677
Processing: 565/677
Processing: 566/677
Processing: 567/677
Processing: 568/677
Processing: 569/677
Processing: 570/677
Processing: 571/677
Processing: 572/677
Processing: 573/677
Processing: 574/677
Processing: 575/677
Processing: 576/677
Processing: 577/677
Processing: 578/677
Processing: 579/677
Processing: 580/677
Processing: 581/677
Processing: 582/677
Processing: 583/677
Processing: 584/677
Processing: 585/677
Processing: 586/677
Processing: 587/677
Processing: 588/677
Processing: 589/677
Processing: 590/677
Processing: 591/677
Processing: 592/677
Processing: 593/677
Processing: 594/677
Processing: 595/677
Processing: 596/677
Processing: 597/677
Processing: 598/677
Processing: 599/677
Processing: 600/677
Processing: 601/677
Processing: 602/677
Processing: 603/677
Processing: 604/677
Processing: 605/677
```

```
Processing: 606/677
Processing: 607/677
Processing: 608/677
Processing: 609/677
Processing: 610/677
Processing: 611/677
Processing: 612/677
Processing: 613/677
Processing: 614/677
Processing: 615/677
Processing: 616/677
Processing: 617/677
Processing: 618/677
Processing: 619/677
Processing: 620/677
Processing: 621/677
Processing: 622/677
Processing: 623/677
Processing: 624/677
Processing: 625/677
Processing: 626/677
Processing: 627/677
Processing: 628/677
Processing: 629/677
Processing: 630/677
Processing: 631/677
Processing: 632/677
Processing: 633/677
Processing: 634/677
Processing: 635/677
Processing: 636/677
Processing: 637/677
Processing: 638/677
Processing: 639/677
Processing: 640/677
Processing: 641/677
Processing: 642/677
Processing: 643/677
Processing: 644/677
Processing: 645/677
Processing: 646/677
Processing: 647/677
Processing: 648/677
Processing: 649/677
Processing: 650/677
Processing: 651/677
Processing: 652/677
Processing: 653/677
```

```
Processing: 654/677
Processing: 655/677
Processing: 656/677
Processing: 657/677
Processing: 658/677
Processing: 659/677
Processing: 660/677
Processing: 661/677
Processing: 662/677
Processing: 663/677
Processing: 664/677
Processing: 665/677
Processing: 666/677
Processing: 667/677
Processing: 668/677
Processing: 669/677
Processing: 670/677
Processing: 671/677
Processing: 672/677
Processing: 673/677
Processing: 674/677
Processing: 675/677
Processing: 676/677
Processing: 677/677
```

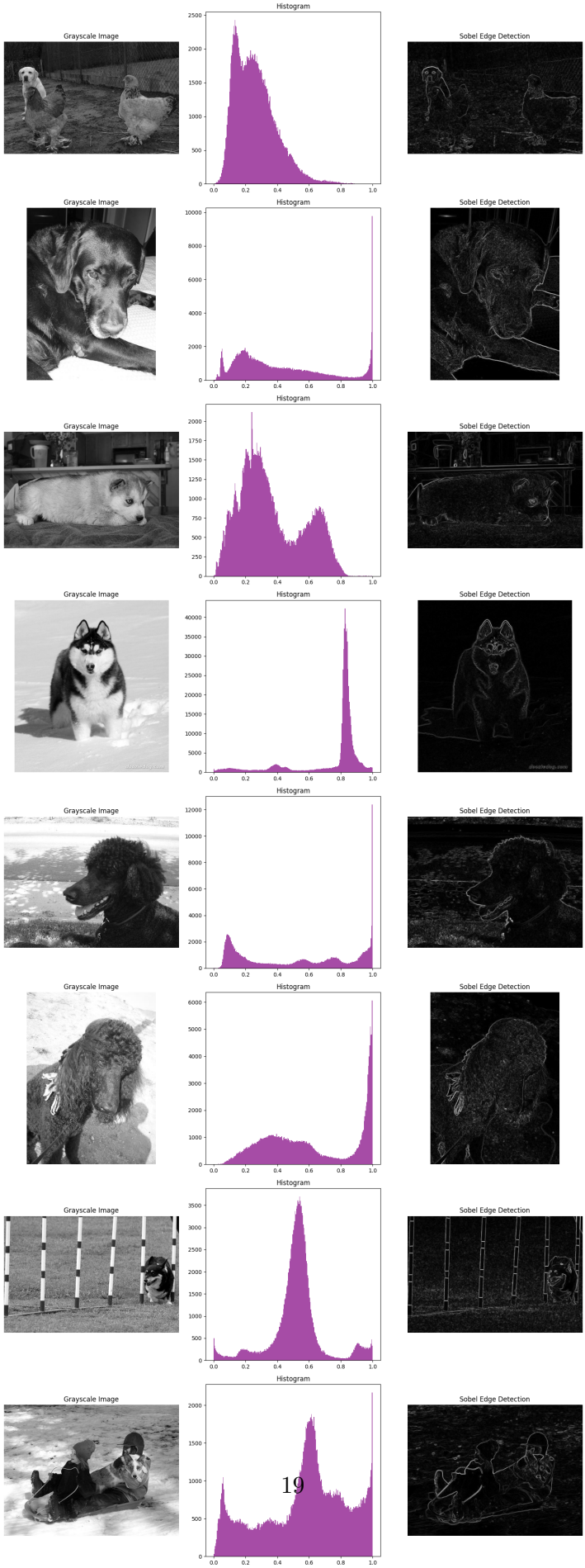
0.2 2b>

[4]:
```
import numpy as np
from skimage import io, color, filters, exposure
import matplotlib.pyplot as plt
```

[12]:
```
class ImageProcess:
    def __init__(self):
        self.all_paths = AllRequiredPaths()

    def _fetch_two_images_from_each_class(self, class_dir):
        img_files = []
        for file in os.listdir(class_dir):
            if file.endswith(".jpg"):
                img_files.append(file)

        images = []
        for img in img_files[3:5]:
            images.append(os.path.join(class_dir, img))
        return images

    def _get_and_process_img(self):
```

```
        all_images = []
        for img_dir in self.all_paths.imgs_folder_sub_folders:
            for img in self._fetch_two_images_from_each_class(img_dir):
                all_images.append(img)

        original_imgs = [io.imread(path) for path in all_images]
        gray_scale_imgs = [color.rgb2gray(img) for img in original_imgs]
        edge_imgs = [filters.sobel(img) for img in gray_scale_imgs]
        return original_imgs, gray_scale_imgs, edge_imgs

    def show_results(self):
        original_imgs, grayscale_imgs, edge_imgs = self._get_and_process_img()

        total_imgs = len(grayscale_imgs)
        fig, axes = plt.subplots(total_imgs, 3, figsize = (15, 5 * total_imgs))

        for i in range(total_imgs):
            # show grayscale images
            axGray = axes[i, 0] if total_imgs > 1 else axes[0]
            axGray.imshow(grayscale_imgs[i], cmap="gray")
            axGray.set_title("Grayscale Image")
            axGray.axis("off")

            # show histrogram graph for each images
            axHistogram = axes[i, 1] if total_imgs > 1 else axes[1]
            axHistogram.hist(grayscale_imgs[i].ravel(), bins=256, range=[0,␣
↪1],color='purple', alpha=0.7)
            axHistogram.set_title("Histogram")

            # show edge detection results
            axEdge = axes[i, 2] if total_imgs > 1 else axes[2]
            axEdge.imshow(edge_imgs[i], cmap="gray")
            axEdge.set_title("Sobel Edge Detection")
            axEdge.axis("off")

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

[13]:
```
image_proecess = ImageProcess()
image_proecess.show_results()
```


Grayscale Image
Histogram
Sobel Edge Detection
Grayscale Image
Histogram
Sobel Edge Detection
Grayscale Image
Histogram
Sobel Edge Detection
Grayscale Image
Histogram
Sobel Edge Detection
Grayscale Image
Histogram
Sobel Edge Detection
Grayscale Image
Histogram
Sobel Edge Detection
Grayscale Image
Histogram
Sobel Edge Detection
Grayscale Image
Histogram
Sobel Edge Detection

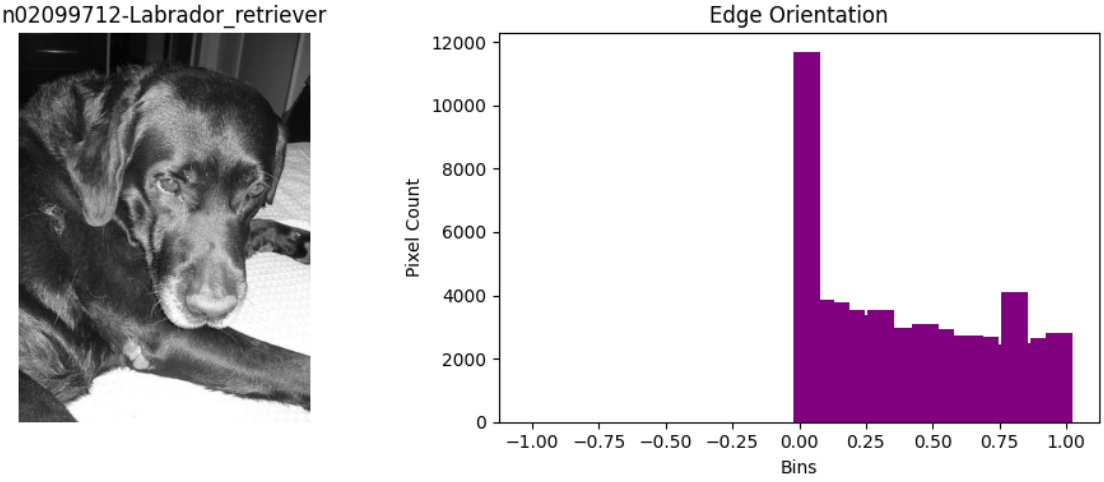
0.3 2c>

[7]:
```
import os
from skimage import io, color, filters, exposure
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from PIL import Image
```

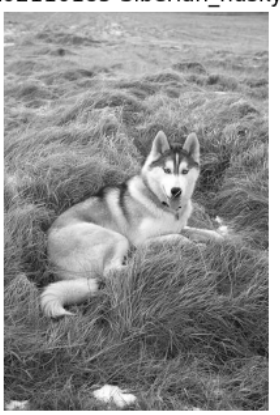
[8]:
```
class EdgeHistogramAnalysis:
    def __init__(self, all_images):
        self.all_paths = AllRequiredPaths()
        self.all_images = all_images

    def _angle(self, dx, dy):
        return np.mod(np.arctan2(dx, dy), np.pi)

    def show_result(self):
        imgs_subdir = self.all_paths.imgs_folder

        for catg, img_name in self.all_images:
            img_path = os.path.join(imgs_subdir, catg, img_name)
            img = io.imread(img_path)

            # grayscale image
            grayscale_img = color.rgb2gray(img)

            # edge orientation angles
            dx = filters.sobel_h(grayscale_img)
            dy = filters.sobel_v(grayscale_img)
            ang_img = self._angle(dx,dy)

            hist, hist_centers = exposure.histogram(ang_img, nbins=36,␣
 ↪source_range="dtype")

            plt.figure(figsize=(10,4))
            plt.subplot(1,2,1)
            plt.imshow(grayscale_img, cmap="gray")
            plt.title(f"{catg}")
            plt.axis("off")

            plt.subplot(1,2,2)
            plt.bar(hist_centers, hist, width=0.1, align="center", color =␣
 ↪"purple")
```

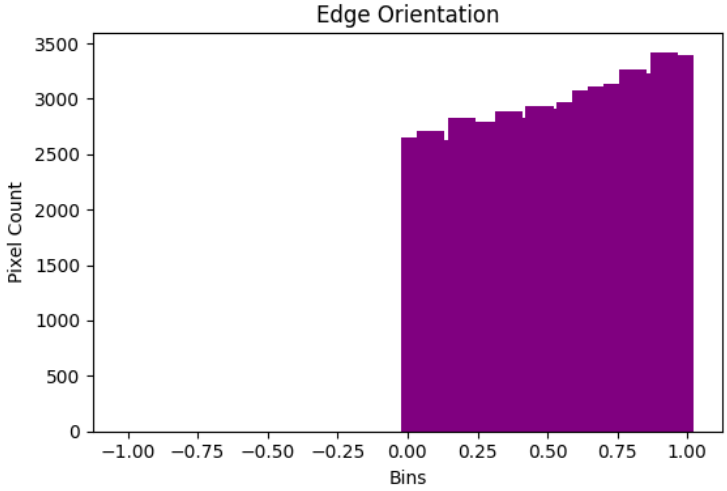
```
            plt.xlabel("Bins")
            plt.ylabel("Pixel Count")
            plt.title("Edge Orientation")

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

[11]:
```
selected_images = [
    ("n02099712-Labrador_retriever", "n02099712_129.jpg"),
    ("n02110185-Siberian_husky", "n02110185_10047.jpg"),
    ("n02113799-standard_poodle", "n02113799_1155.jpg"),
    ("n02113186-Cardigan", "n02113186_1030.jpg")
]

edge_hist = EdgeHistogramAnalysis(selected_images)
edge_hist.show_result()
```

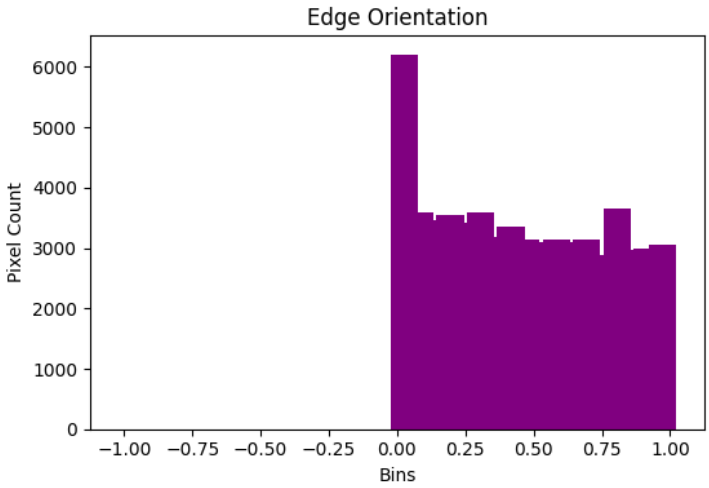


n02110185-Siberian_husky

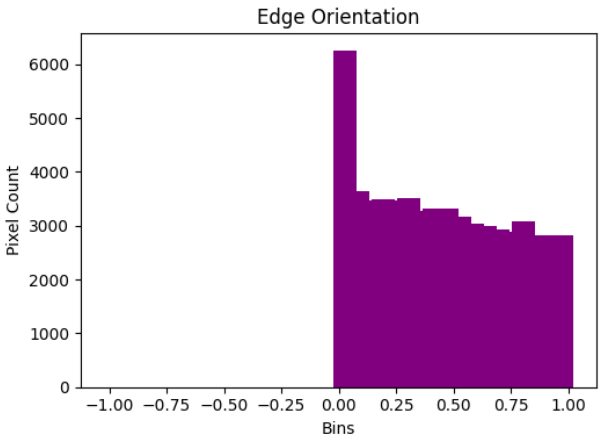




n02113799-standard_poodle



n02113186-Cardigan



0.4 2d>

[15]:
```
import numpy as np
import matplotlib.pyplot as plt
from skimage import io, color, filters, exposure
from sklearn.metrics.pairwise import euclidean_distances,␣
↪manhattan_distances,cosine_distances
```

[17]:
```
class HistorgramComparison:
    def __init__(self, selected_classes):
        self.all_paths = AllRequiredPaths()
        self.selected_classes = selected_classes

    def _angle(self, dx, dy):
        return np.mod(np.arctan2(dx,dy), np.pi)

    def _edge_histogram(self, image):
        gray_img = color.rgb2gray(image)
        angles = self._angle(filters.sobel_h(gray_img), filters.
↪sobel_v(gray_img))
        hist, _ = np.histogram(angles, bins=36, range=(-np.pi, np.pi))
        return hist

    def _process_images(self):
        edge_hist = {}
        for catg, img_name in self.selected_classes:
            img_path = os.path.join(self.all_paths.imgs_folder, catg, img_name)
            img = io.imread(img_path)
            hist = self._edge_histogram(img)
            edge_hist[img_name] = hist
        return edge_hist

    def compare_hist(self, title):
        edge_hist = self._process_images()
        histos = list(edge_hist.values())
        img_names = list(edge_hist.keys())

        euclidean_dist = euclidean_distances(histos)
        manhattan_dist = manhattan_distances(histos)
        cosine_dist = cosine_distances(histos)

        print(title)
        for i in range(len(img_names)):
            for j in range(i + 1, len(img_names)):
```

```
                print(f"Euclidean: {euclidean_dist[i][j]}")
                print(f"Manhattan: {manhattan_dist[i][j]}")
                print(f"Cosine: {cosine_dist[i][j]}")


histogram_compare1 = HistorgramComparison([
    ("n02110185-Siberian_husky", "n02110185_10047.jpg"),
    ("n02110185-Siberian_husky", "n02110185_10844.jpg")
])
histogram_compare1.compare_hist("Comparison between two images of same class")

print("")
histogram_compare2 = HistorgramComparison([
    ("n02110185-Siberian_husky", "n02110185_10047.jpg"),
    ("n02113186-Cardigan", "n02113186_1030.jpg")
])
histogram_compare2.compare_hist("Comparison between two images of different␣
 ↪class")
```

```
Comparison between two images of same class
Euclidean: 4678.233213511272
Manhattan: 16230.0
Cosine: 0.005141347337101587

Comparison between two images of different class
Euclidean: 10356.438383923307
Manhattan: 35606.0
Cosine: 0.024073880505602663
```

0.5 2e>

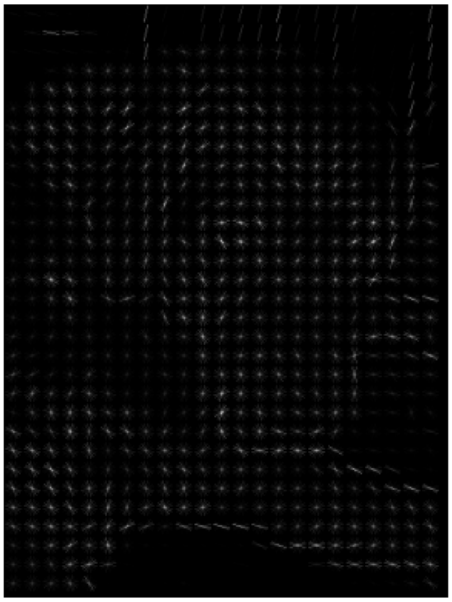
[18]:
```
import matplotlib.pyplot as plt
from skimage import io, exposure
from skimage.transform import resize
from skimage.feature import hog
```

[19]:
```
class HOGAnalysis:
    def __init__(self, images):
        self.all_path = AllRequiredPaths()
        self.images = images

    def _fetchAndPlot(self, img_path, orientations = 8, pixels_per_cell =␣
 ↪(16,16), cells_per_block = (1,1)):
        img = io.imread(img_path)
```

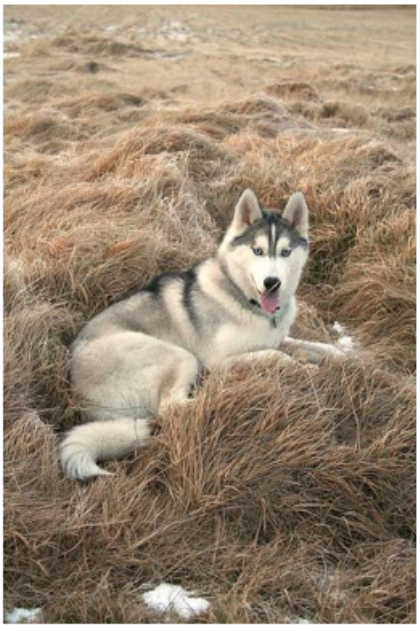
```
        fd, hog_img = hog(img, orientations=orientations,␣
 ↪pixels_per_cell=pixels_per_cell, cells_per_block=cells_per_block,␣
 ↪visualize=True, channel_axis=-1)

        hog_img_rescaled = exposure.rescale_intensity(hog_img,␣
 ↪out_range=(0,225))

        plt.figure(figsize=(10,5))

        plt.subplot(1,2,1)
        plt.imshow(img, cmap=plt.cm.gary if img.ndim == 2 else None)
        plt.title("Original Image")
        plt.axis('off')

        plt.subplot(1,2,2)
        plt.imshow(hog_img_rescaled, cmap="gray")
        plt.title("HOG Image")
        plt.axis('off')

        plt.show()

    def show_result(self):
        for catg, img_name in self.images:
            img_path = f"images/{catg}/{img_name}"
            self._fetchAndPlot(img_path)



images = [
    ("n02099712-Labrador_retriever", "n02099712_129.jpg"),
    ("n02110185-Siberian_husky", "n02110185_10047.jpg"),
    ("n02113799-standard_poodle", "n02113799_1155.jpg"),
    ("n02113186-Cardigan", "n02113186_1030.jpg")
]


hogAnalysis = HOGAnalysis(images)
hogAnalysis.show_result()
```

Original Image

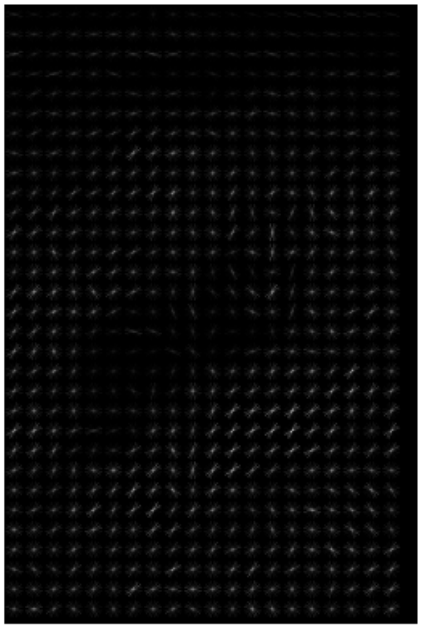
HOG Image

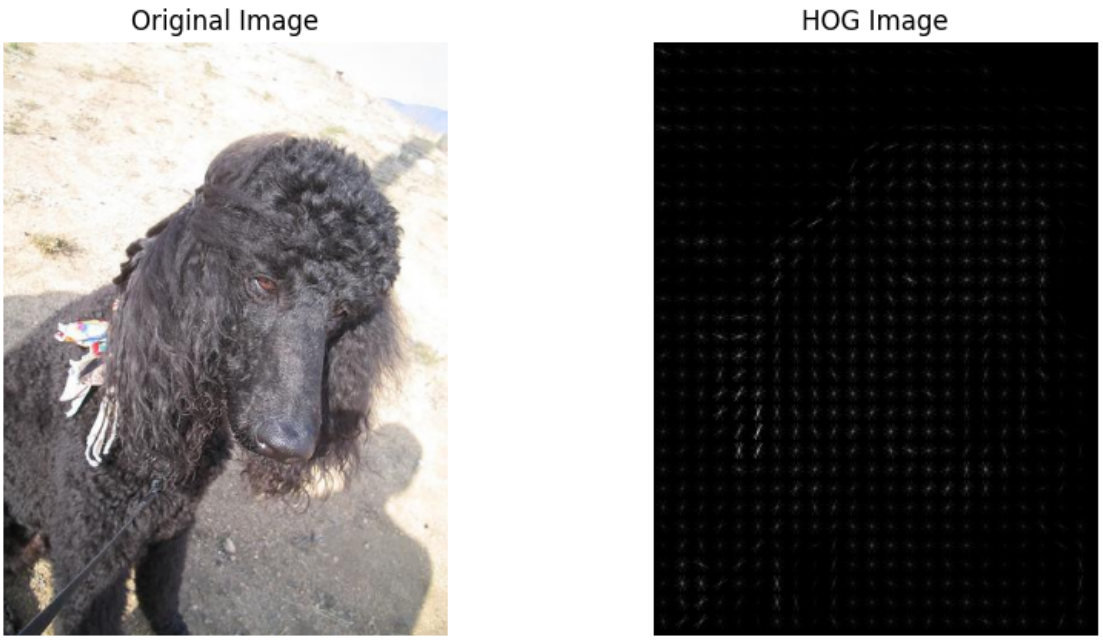


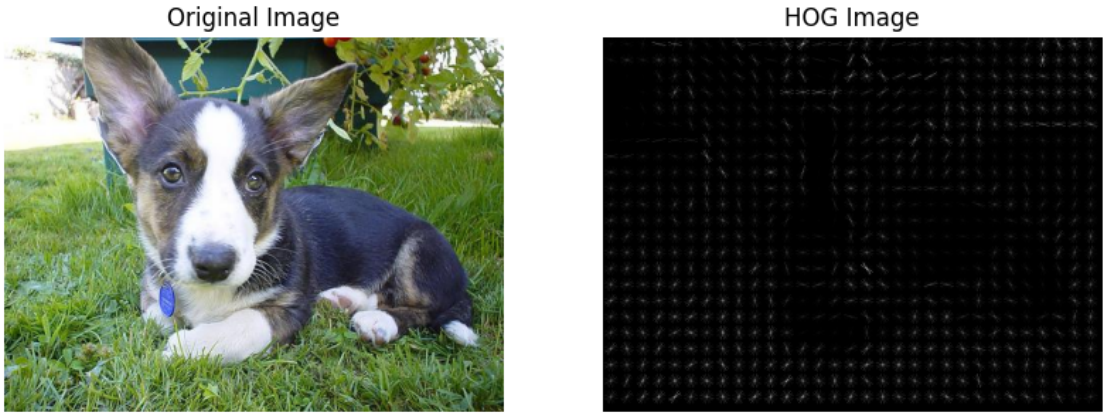
Original Image



HOG Image







0.6 2f>

[1]:
```
import os
import numpy as np
import matplotlib.pyplot as plt
from skimage import io, color, filters
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

[20]:
```
class PCAAnalysis:
    def __init__(self):
        self.all_paths = AllRequiredPaths()

    def _angle(self, dx, dy):
        return np.mod(np.arctan2(dx,dy), np.pi)

    def _edge_histogram(self, image):
        gray_img = color.rgb2gray(image)
        angles = self._angle(filters.sobel_h(gray_img), filters.
 ↪sobel_v(gray_img))
        hist, _ = np.histogram(angles, bins=36, range=(-np.pi, np.pi))
        return hist

    def fetch_images(self, folder_name):
        allImage = []
        for img_name in os.listdir(folder_name):
            if img_name.lower().endswith(".jpg"):
                allImage.append(os.path.join(folder_name, img_name))
        return allImage

    def perform_pca_and_show_result(self, class1, class2):
        hists = []
        labels = []
        classes = class1 + class2

        for img_path in classes:
            img = io.imread(img_path)
            hist = self._edge_histogram(img)
            hists.append(hist)
            labels.append("Class 1" if img_path in class1 else "Class 2")

        histogram = np.array(hists)

        std_scalar = StandardScaler()
        hist_std = std_scalar.fit_transform(histogram)

        pca = PCA(n_components=2)
        hist_pca = pca.fit_transform(hist_std)

        pca1 = hist_pca[:len(class1)]
        pca2 = hist_pca[:len(class2)]

        plt.scatter(pca1[:, 0], pca1[:, 1], color = "red", label = "Class-1")
        plt.scatter(pca2[:, 0], pca2[:, 1], color = "blue", label = "Class-2")

        plt.xlabel("Principal Component 1")
```

```
        plt.ylabel("Principal Component 2")
        plt.legend(["Class 1", "Class 2"])
        plt.title("PCA of Image")
        plt.show()


pca_analysis = PCAAnalysis()
pca1_folder = pca_analysis.fetch_images("images/n02110185-Siberian_husky")
pca2_folder = pca_analysis.fetch_images("images/n02113186-Cardigan")

pca_analysis.perform_pca_and_show_result(pca1_folder, pca2_folder)
```