Example of using image_dataset_analysis to analyze image dataset

First of all, let's import **ImageDataset**, which is the main class, representing essential information about dataset along with useful tools to discover it.

In [1]:

```python
from image_dataset_analysis import ImageDataset
```

Create an instance of this class by passing image dataset's directory path as an argument. After running the cell below, it will automatically analyze the content of the given dataset.

The following structure of dataset directory is required to successfully analyze it:

```
image_dataset:
    - class 1:
        - image 1;
        - image 2;
         ....
    - class 2
    - class 3
      .....
```

Note: The LFW (Labelled Faces in the Wild) face dataset is utilized in this example.

In [2]:

```python
lfw = ImageDataset("C:\\Users\\Samandar\\Desktop\\lfw_aligned_128\\lfw_aligned_128")
```

```
Analyzing dataset's content...
100%|██████████| 5734/5734 [00:18<00:00, 317.98it/s]
```

Get maximum number of images per class

In [3]:

```python
max_value = lfw.max_images_per_class()
print(f"Maximum number of images per class: {max_value}")
```

```
Maximum number of images per class: 530
```

Get minimum number of images per class

In [4]:

```python
min_value = lfw.min_images_per_class()
print(f"Minimum number of images per class: {min_value}")
```

```
Minimum number of images per class: 1
```

Get mean number of images per class

In [5]:

```python
mean_value = lfw.mean_images_per_class()
print(f"Mean number of images per class: {mean_value}")
```

```
Mean number of images per class: 2
```

Get class info with name, for instance, "Ben_Curtis"

In [9]:

```python
class_info = lfw.get_class_info(name='Ben_Curtis')
print(class_info)
```

```
{'images_number': 3, 'images': ['Ben_Curtis_0001.jpg', 'Ben_Curtis_0002.jpg', 'Ben_Curtis_
0003.jpg']}
```

Get classes with number of images equal to n

In [10]:

```
classes_with_n_images = lfw.get_classes_with_n_images(n=530)
print(classes_with_n_images)
```

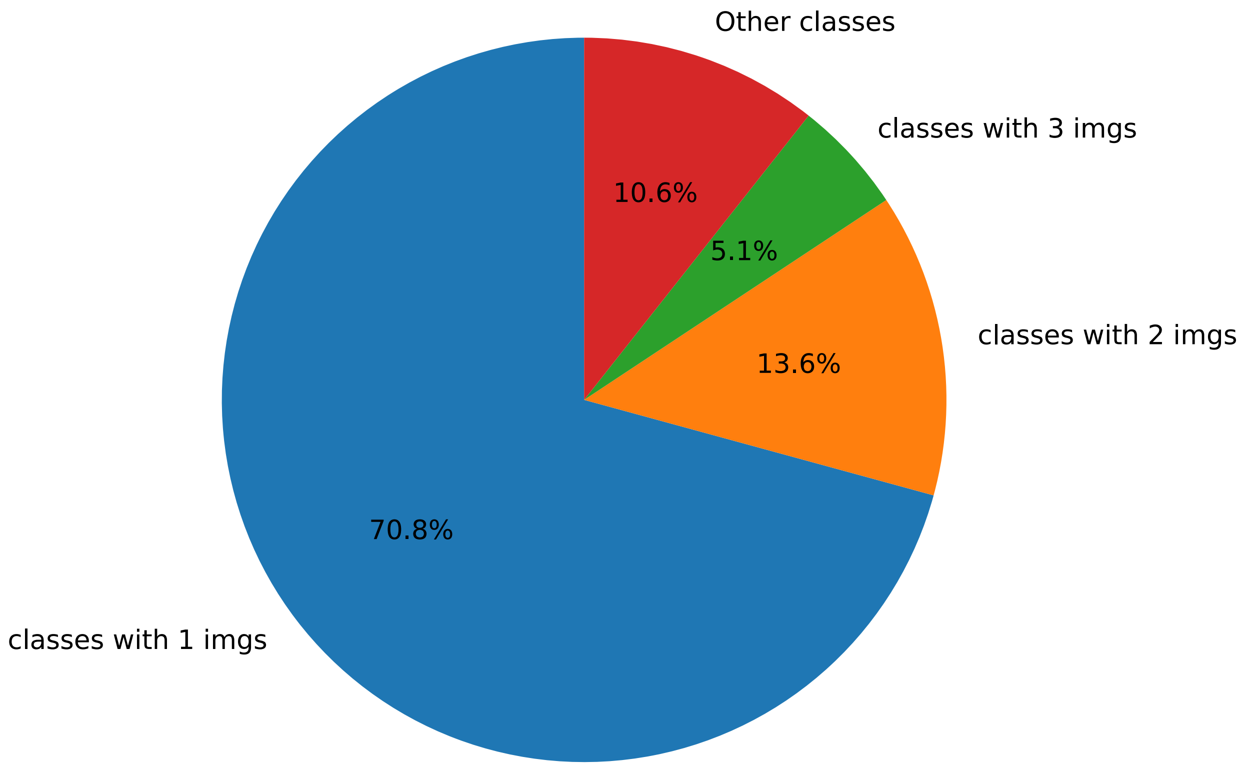
```
['George_W_Bush']
```

It might be interesting to analyze the proportion of classes with specified amount(s) of images in the given dataset. The following method provides with the innformation about the quantity of the aforementioned type of classes and plots a pie chart of proportions to visualize relations.

In [11]:

```
lfw.proportion_of_classes_with_n_images(images_number=[1, 2, 3], plot_pie_chart=True)
```

```
Number of classes with only 1 images : 4057
Number of classes with only 2 images : 777
Number of classes with only 3 images : 290
Remaining number of classes : 610
```



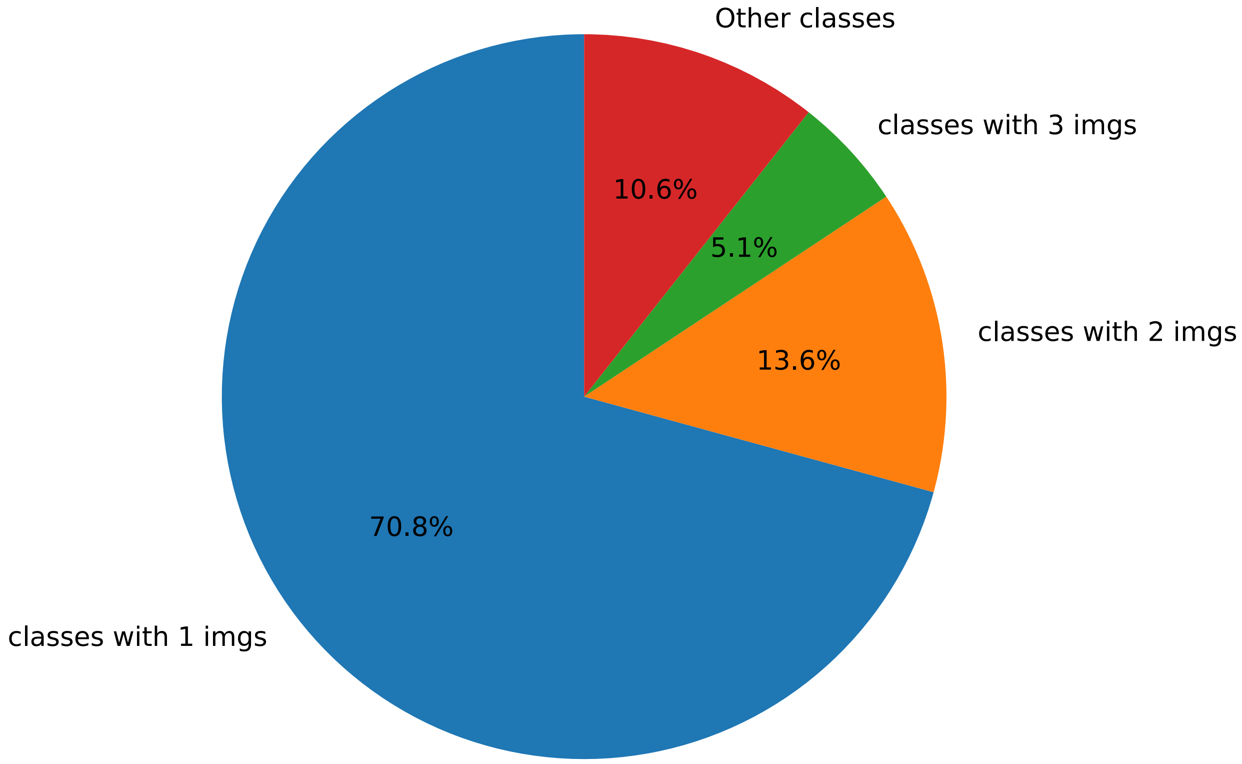
Method to perform basic analysis.

In [18]:

```
lfw.analyze()
```

```
Number of images in image dataset: 13214
Number of classes in image dataset: 5734
Mean number of images per class: 2
Minimum number of images per class: 1
Maximum number of images per class: 530
```

```
Number of classes with only 1 images : 4057
Number of classes with only 2 images : 777
Number of classes with only 3 images : 290
Remaining number of classes : 610
```



In ImageDataset class, beyond the methods of analysis, there is a method, which might be useful to split the dataset into train & test parts and save them into text files, given as parameters, in the following format:

'class_name/image_name' label

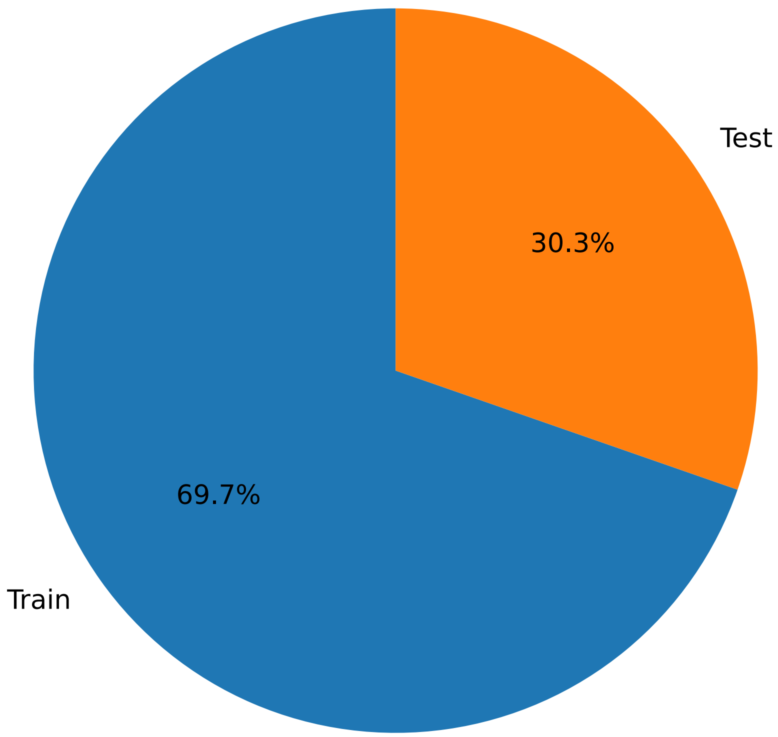
The n parameter is utilized to define the number of test images in a certain class using the following formula: [number of images in class / n], where [] means whole part.

Example: n = 3, a certain class contains following: image1, image2, image3, image4, image5, image6, image7. The number of images in class equal to 7, [number of images in class / n] = [7/3] = 2. Consequently, test images: image1, image2 & train images: image3, image4, image5, image6, image7.

In [24]:

```
train, test, full = lfw.create_train_test_list(n=3,
                                               train_list_file_path="lfw_train.txt",
                                               test_list_file_path="lfw_test.txt",
                                               full_list_file_path="lfw_full.txt",
                                               plot_relation=True)
```

```
Number of classes: 900
Number of train images: 5297
Number of test images: 2306
Total number of images: 7603
```


Test
30.3%
69.7%
Train