```
In [1]:
            def Mask():
          1
                import cv2
          2
          3
                import numpy as np
          4
                from IPython.display import Image
                from PIL import Image as convert_to_image
          5
          6
                import matplotlib.pyplot as plt
          7
                import os
          8
          9
                replace_path = "/home/jovyan/shared/datasets/smaller_faces/"
                source_path = "/home/jovyan/HW3/smaller_faces.txt"
         10
         11
         12
                with open(source_path, 'r') as file :
         13
                    filedata = file.read()
         14
         15
                filedata = filedata.replace("smaller_faces/", replace_path)
         16
         17
                with open('smaller_faces_dup.txt', 'w') as file:
         18
                    file.write(filedata)
         19
                ### -----
         20
         21
         22
                def create_folder():
         23
                    path = "/home/jovyan/HW3/Masks/"
         24
                    isExist = os.path.exists(path)
         25
         26
                    if not isExist:
         27
                        os.makedirs(path)
         28
                    return path
         29
                ### -----
         30
         31
         32
                def ellipse(box_coord, image_width, image_height, w_x, h_y):
         33
                    image = np.zeros((image_height, image_width), dtype=int)
         34
                    center = [box_coord[0]+w_x//2, box_coord[1]+h_y//2]
         35
                    x, y = center[1], center[0] # Rows and columns of the center
         36
         37
                    cx,cy = [x, y]
         38
                    queue = [[cx,cy]]
         39
                    neighbors = [[1, 0], [-1, 0], [0, 1], [0, -1]]
         40
         41
                    if w_x < h_y:
         42
                        min_ax = w_x//2
         43
                        maj_ax = h_y//2
         44
                    else:
         45
                        min_ax = h_y//2
         46
                        maj_ax = w_x//2
         47
         48
                    def dist(a,b):
         49
                        return np.sqrt((((a[0]-b[0])**2)*(min_ax**2)) + (((a[1]-b[1])**2)*(maj_ax**2)))
         50
         51
                    while(len(queue)>0):
         52
                        tx, ty = queue.pop(0)
         53
                        if(dist([cx,cy], [tx,ty])) >= (min_ax*maj_ax) or image[ty,tx]==1:
         54
                            continue # Do not need to fill this value, because outside the bounds
         55
                        image[ty,tx] = 1 # mark the pixel; we'll multiply the mask by 255 for visualization later
         56
         57
                        for i,j in neighbors:
         58
                            x = tx + i
         59
                            y = ty + j
                            if(0<=x<img.shape[0] and 0<=y<img.shape[1]): #boundary check to make sure co-ordinates are
         60
         61
                                queue.append((x,y))
         62
         63
                    return np.transpose(image)
         64
         65
         66
                def lookup():
         67
         68
                    heading_list = []
         69
                    n_{faces} = []
                    dim_list = []
         70
         71
                    heading_name = []
         72
                    with open("smaller faces dup.txt") as file:
         73
         74
                        lines = file.readlines()
         75
                    file.close()
         76
         77
                    lookup = "/shared/datasets/smaller_faces/"
         78
         79
                    for num, line in enumerate(lines, 0):
                        if lookup in line:
         80
                            heading name.append(line.strip())
         81
         82
                            heading_list.append(int(num))
         83
                            n = int(lines[num+1])
         84
                            n_faces.append(n)
         85
                            for i in range(n):
                                dim_list.append(list(map(int, lines[num+2+i].split()[0:4])))
         86
         87
         88
                    return heading_list, n_faces, dim_list, heading_name
         89
         90
```

```
91
 92
         def loops(dim_list, image_width, image_height):
 93
             box_coord = [dim_list[0], dim_list[1]]
                                                          # The coordinates of the bounding box [x-dir, y-dir]
 94
             w_x = dim_list[2] # Width of the bouning box
 95
             h_y = dim_list[3] # Height of the bounding box
 96
 97
             image_ellipse = ellipse(box_coord, image_width, image_height, w_x, h_y)
98
99
             return image_ellipse
100
101
102
103
         def print_to_file(image, image_width, image_height, heading_name, img):
104
             path = create_folder()
105
             print(path)
106
             image_ = np.zeros((image_width,image_height,1),dtype= np.uint8)
107
             image_[:,:,0] = image
108
109
110
             img_save = heading_name[42:]
111
             #dst = cv2.addWeighted(original_image, 0.5, image_*255, 0.5, 0.0)
112
             save_path = "/home/jovyan/HW3/Masks"+img_save
113
             cv2.imwrite(save_path, image_)
114
             # plt.imshow(image_)
115
116
117
118
        from itertools import islice
119
120
        heading_list, n_faces, dim_list, heading_name = lookup()
121
122
         temp = iter(dim_list)
123
         dim_list_grouped = [list(islice(temp, 0, i)) for i in n_faces]
124
125
        for i in range(len(heading_name)):
126
             path = heading_name[i]
127
             img = cv2.imread(path)
128
             image_width = img.shape[0]
129
             image_height = img.shape[1]
130
             image_ellipse = np.zeros((image_width,image_height), dtype = int)
131
             for j in range(n_faces[i]):
132
133
                 image_ellipse += loops(dim_list_grouped[i][j], image_width, image_height)
134
135
             print_to_file(image_ellipse, image_width, image_height, heading_name[i], img)
136
137
        return
138
139 | Mask()
```

```
/home/jovyan/HW3/Masks/
```