

In [1]:

```
1 def Mask():
2     import cv2
3     import numpy as np
4     from IPython.display import Image
5     from PIL import Image as convert_to_image
6     import matplotlib.pyplot as plt
7     import os
8
9     replace_path = "/home/jovyan/shared/datasets/smaller_faces/"
10    source_path = "/home/jovyan/HW3/smaller_faces.txt"
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
60                if(0<=x<img.shape[0] and 0<=y<img.shape[1]): #boundary check to make sure co-ordinates are
61                    queue.append((x,y))
62
63        return np.transpose(image)
64
65    ### -----#
66
67    def lookup():
68        heading_list = []
69        n_faces = []
70        dim_list = []
71        heading_name = []
72
73        with open("smaller_faces_dup.txt") as file:
74            lines = file.readlines()
75            file.close()
76
77        lookup = "/shared/datasets/smaller_faces/"
78
79        for num, line in enumerate(lines, 0):
80            if lookup in line:
81                heading_name.append(line.strip())
82                heading_list.append(int(num))
83                n = int(lines[num+1])
84                n_faces.append(n)
85                for i in range(n):
86                    dim_list.append(list(map(int, lines[num+2+i].split()[0:4])))
87
88        return heading_list, n_faces, dim_list, heading_name
89
90    ### -----#
```

