



Machine Vision (CSE480)

Lab 2 Report

Name	ID	Section
Mahmoud Elsayd Abdelqader Labib Eldwakhly	21P0017	1

Submitted to Dr Hossam Hassan & Eng. Dina Zakaria

Fall 2025

Python Code

```
1 # =====
2 # TASK 1
3 # =====
4
5
6 import cv2
7 import numpy as np
8 from google.colab.patches import cv2_imshow
9
10 # =====
11 # 1. Load image
12 # =====
13 image_path = "/content/Mosalah.jpg"
14 img = cv2.imread(image_path)
15
16 if img is None:
17     raise ValueError("Image not found!")
18
19 print("◆ Original Image:")
20 cv2_imshow(img)
21
22 # =====
23 # 2. Create elliptical mask
24 # =====
25 h, w = img.shape[:2]
26
27 # Create empty mask (all black)
28 mask = np.zeros((h, w), dtype=np.uint8)
29
30 # Ellipse parameters (centerX, centerY), (width, height), angle
31 center = (w // 2, h // 2)
32 axes = (w // 3, h // 2)    # bigger ellipse = more sharp area
33 angle = 0
34
35 # Draw white ellipse on mask
36 cv2.ellipse(mask, center, axes, angle, 0, 360, 255, -1)
37
38 print("◆ Elliptical Mask (White = Keep Sharp):")
39 cv2_imshow(mask)
40
41 # Invert mask → inside ellipse black, outside white
42 mask_inv = cv2.bitwise_not(mask)
43
44 print("◆ Inverted Mask (White = Blur Background):")
45 cv2_imshow(mask_inv)
46
47 # =====
48 # 3. Create strong blur
49 # =====
50 blurred = cv2.GaussianBlur(img, (81, 81), 0)
51
52 print("◆ Blurred Image:")
```

```

53 cv2_imshow(blurred)
54
55 # =====
56 # 4. Bitwise operations to combine
57 # =====
58
59 # Subject remains sharp (ellipse area)
60 sharp_subject = cv2.bitwise_and(img, img, mask=mask)
61
62 print("◆ Extracted Sharp Subject:")
63 cv2_imshow(sharp_subject)
64
65 # Background blurred (mask_inv selects outside ellipse)
66 blurred_background = cv2.bitwise_and(blurred, blurred, mask=mask_inv)
67
68 print("◆ Extracted Blurred Background:")
69 cv2_imshow(blurred_background)
70
71 # Combine: OR = merges them
72 portrait_mode = cv2.bitwise_or(sharp_subject, blurred_background)
73
74 print("◆ Final Portrait Mode Effect:")
75 cv2_imshow(portrait_mode)
76
77 # =====
78 # 5. Save result
79 # =====
80 cv2.imwrite("/portrait_mode_result.jpg", portrait_mode)
81 print("Saved as /portrait_mode_result.jpg")
82
83 # =====
84
85
86 # =====
87 # TASK 2
88 # =====
89
90
91 import cv2
92 import numpy as np
93 from google.colab.patches import cv2_imshow
94
95 # =====
96 # 1. Load color image
97 # =====
98 img = cv2.imread("/content/task2-2.jpg") # CHANGE THIS!
99 if img is None:
100     raise ValueError("Image not found!")
101
102 print("◆ Original Image:")
103 cv2_imshow(img)
104
105 # =====
106 # 2. Make grayscale, invert, and blur
107 # =====

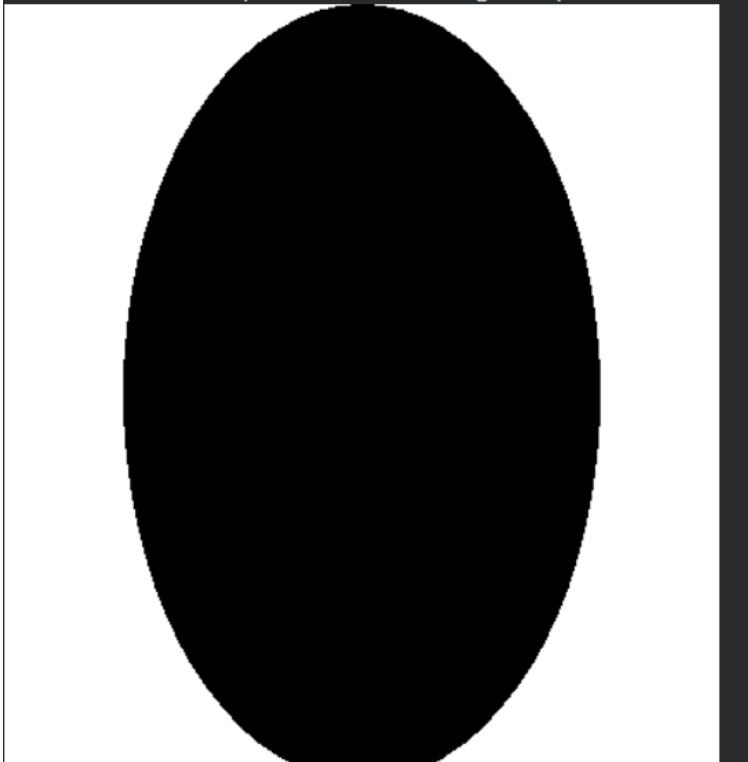
```

```
108 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
109 inverted = cv2.bitwise_not(gray) # 255-gray
110 blur = cv2.GaussianBlur(inverted, (25, 25), 0) # Try (15,15), (25,25), (51,51)
111
112 print("◆ Grayscale Image:")
113 cv2_imshow(gray)
114
115 print("◆ Inverted Gray Image:")
116 cv2_imshow(inverted)
117
118 print("◆ Blurred Inverted Image:")
119 cv2_imshow(blur)
120
121 # =====
122 # 3. Color Dodge = the pencil sketch
123 # =====
124 sketch = cv2.divide(gray, 255 - blur, scale=256)
125
126 print("👉 Pencil Sketch (Color Dodge):")
127 cv2_imshow(sketch)
128
129 # Save result
130 cv2.imwrite("/pencil_sketch_colordodge.jpg", sketch)
131
132 # =====
133 # 4. Alternative Method – Edge Detection (Canny)
134 # =====
135 edges = cv2.Canny(gray, 50, 150)
136
137 print("📐 Canny Edge Sketch:")
138 cv2_imshow(edges)
139
140 cv2.imwrite("/pencil_sketch_edges.jpg", edges)
141
142 print("\nSaved:")
143 print("/pencil_sketch_colordodge.jpg")
144 print("/pencil_sketch_edges.jpg")
145
146
```

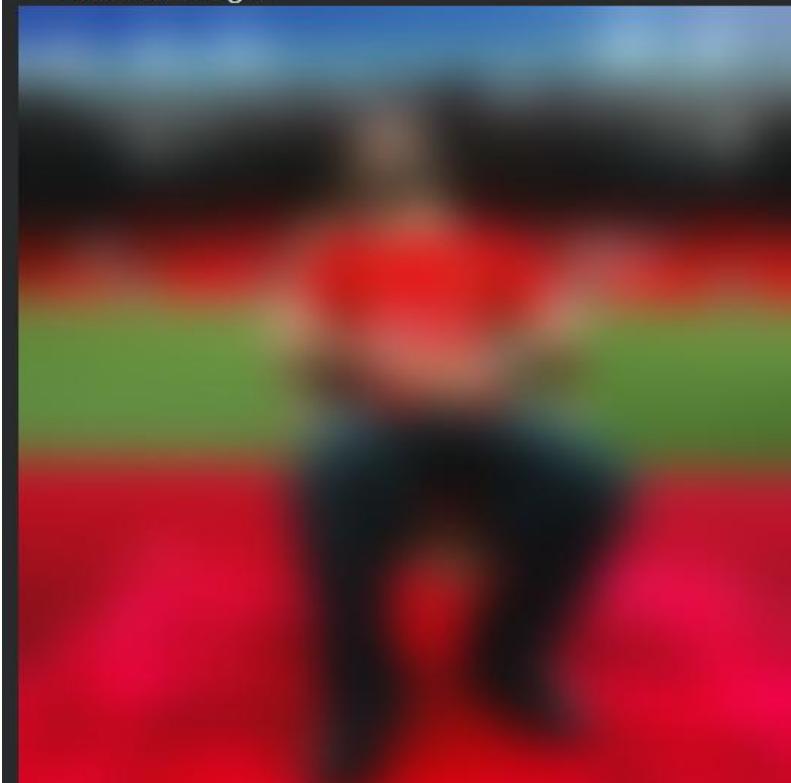
Output:
Task 1 :



• Inverted Mask (White = Blur Background):



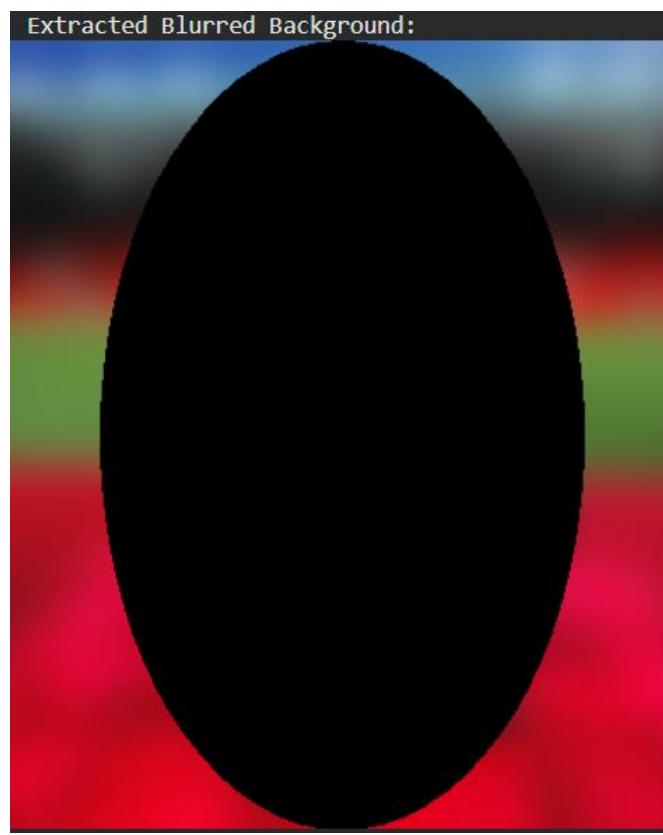
• Blurred Image:



Extracted Sharp Subject:



Extracted Blurred Background:



◆ Final Portrait Mode Effect:



Task 2 :



Inverted Gray Image:



• Blurred Inverted Image:



