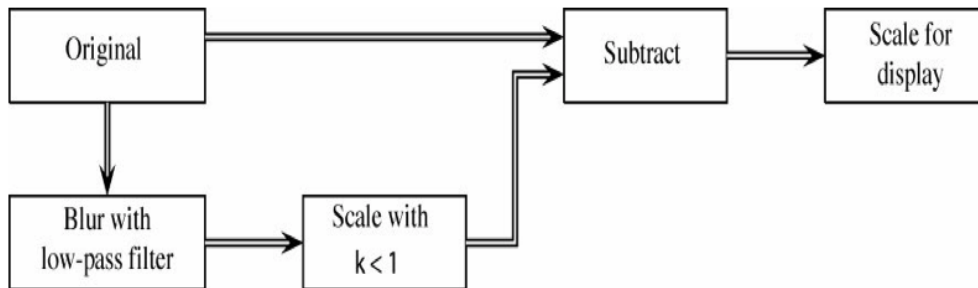


Homework 5



(1) Problem statement

1. Select an experimental image
2. Apply a 3 by 3 (a) average filter and (b) median filter to the image
3. Unsharp masking





(2.1) Experimental results

(i) Apply a 3 by 3 average filter and unsharp masking

Original RGB image	Experimental result
	

(ii) Apply a 3 by 3 median filter and unsharp masking

Original RGB image	Experimental result
	

(2.2) Source code

(i) Apply a 3 by 3 average filter and unsharp masking

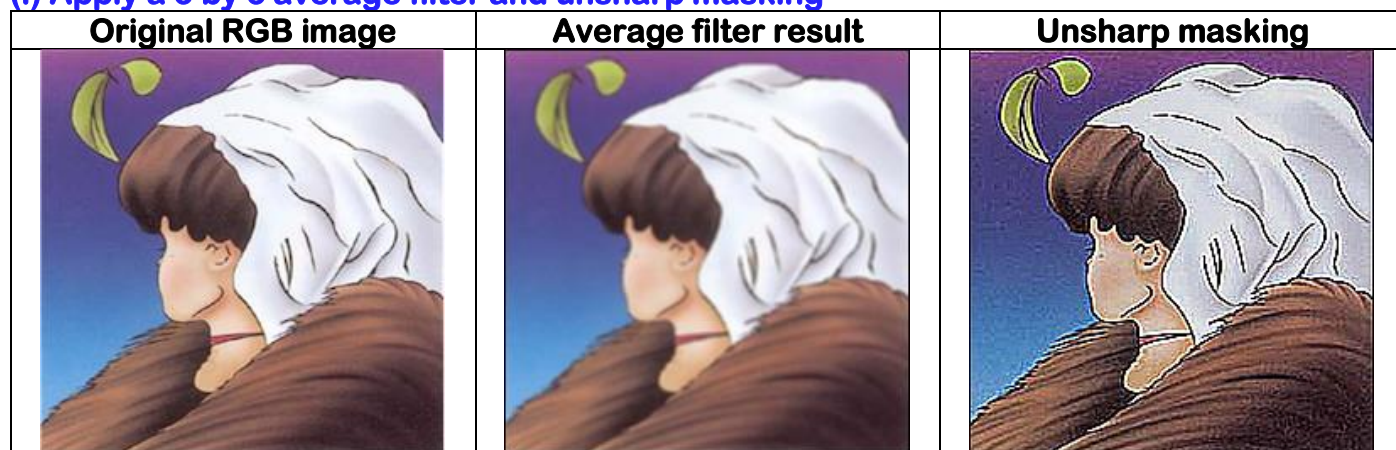
```
1  # HW5-a (Implement unsharp masking and use average filter)|
2
3  from PIL import Image, ImageDraw
4
5  # Input a color image (RGB
6  input_image = Image.open("W.E.Hill.png").convert("RGB")
7  input_pixels = input_image.load()
8
9  # (low-pass) Average Filter mask
10 mask = [[1 / 9, 1 / 9, 1 / 9],
11          [1 / 9, 1 / 9, 1 / 9],
12          [1 / 9, 1 / 9, 1 / 9]]
13 k = 0.9
14 val = 9 # scale for display
15
16 # Middle of the kernel
17 offset = len(mask) // 2
18
19 # Create output image
20 output_image = Image.new("RGB", input_image.size)
21 draw = ImageDraw.Draw(output_image)
22
23 for y in range(offset, input_image.height - offset):
24     for x in range(offset, input_image.width - offset):
25         original_pixel = input_pixels[x, y]
26         acc = [0, 0, 0]
27
28         # Compute convolution between intensity and mask
29         for a in range(len(mask)):
30             for b in range(len(mask)):
31                 xn = x + a - offset
32                 yn = y + b - offset
33                 pixel = input_pixels[xn, yn]
34                 acc[0] += pixel[0] * mask[a][b]
35                 acc[1] += pixel[1] * mask[a][b]
36                 acc[2] += pixel[2] * mask[a][b]
37
38         # Unsharp masking
39         new_pixel = (
40             int(original_pixel[0] - acc[0] * k) * val,
41             int(original_pixel[1] - acc[1] * k) * val,
42             int(original_pixel[2] - acc[2] * k) * val
43         )
44         draw.point((x, y), new_pixel)
45
46 # Save experimental image
47 output_image.save("unsharp(average filter).png")
48
```

(ii) Apply a 3 by 3 median filter and unsharp masking

```
1  # HW5-b (Implement unsharp masking and use median filter)
2
3  from PIL import Image, ImageFilter
4
5  # Input a color image (RGB)
6  input_image = Image.open("W.E.Hill.png").convert("RGB")
7  width, height = input_image.size
8
9  # Using Median filter (3X3)
10 filter_image = input_image.filter(ImageFilter.MedianFilter(3))
11
12 # Unsharp masking
13 k = 0.9
14 val = 9 # scale for display
15 for y in range(height):
16     for x in range(width):
17         original_pixel = input_image.getpixel((x, y))
18         filter_pixel = filter_image.getpixel((x, y))
19         new_pixel = (
20             int(original_pixel[0] - filter_pixel[0] * k) * val,
21             int(original_pixel[1] - filter_pixel[1] * k) * val,
22             int(original_pixel[2] - filter_pixel[2] * k) * val
23         )
24         input_image.putpixel((x,y), new_pixel)
25
26 # Save experimental image
27 input_image.save("unsharp(median filter).png")
28
```

(2.3) Comments

(i) Apply a 3 by 3 average filter and unsharp masking



(ii) Apply a 3 by 3 median filter and unsharp masking

