

## LAB - 6

Name	Keval D Gandevia
Roll Number	CE046
ID	19CEUEG017
Subject	Image Processing

**Aim:** Analysis of effect of applying different filters on the image to give a blur effect.

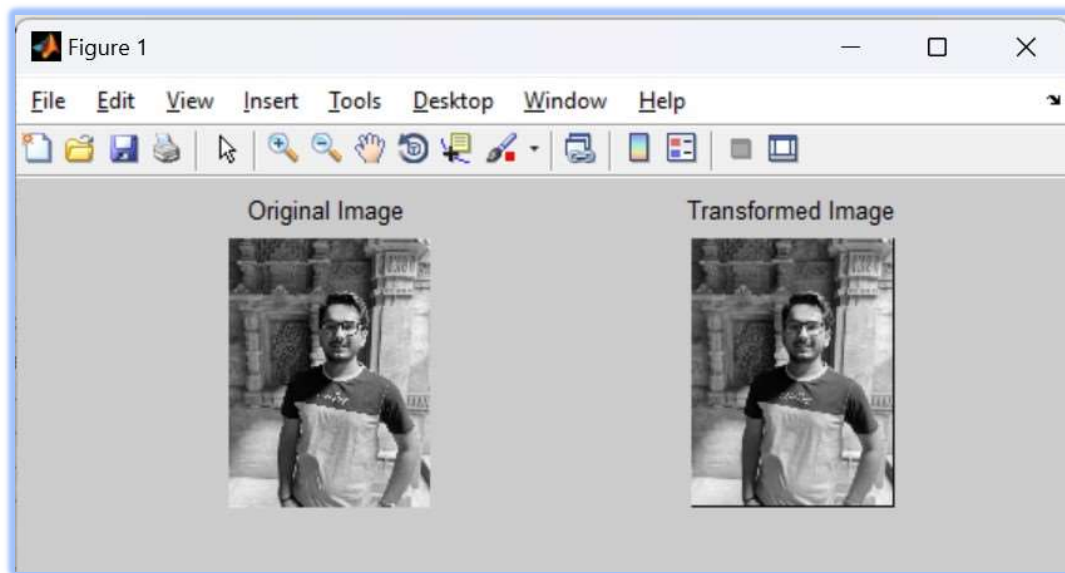
**Q. 1: Take any gray scale photo and blur it with standard box filter of size 3X3, 5X5, 7X7, 9X9.**

**a) Perform the blurring process without using built-in functions.**

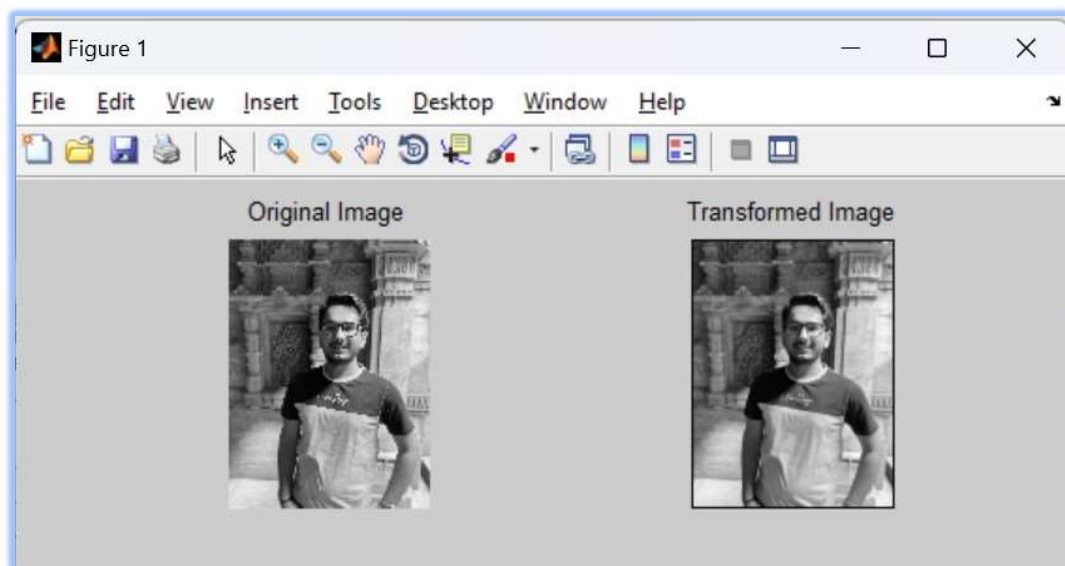
❖ **Code:**

```
a_1.m x lab_5_2.m x +
1      clear all;
2
3      img = imread("Keval_Image.jpg");
4      img = rgb2gray(img);
5      img = imresize(img, 0.2);
6      img = double(img);
7      [m, n] = size(img);
8
9      subplot(2, 2, 1);
10     imshow(img, []);
11     title("Original Image");
12
13     box_size = 15;
14     standard_box_filter = ones(box_size, box_size);
15     [s1, s2] = size(standard_box_filter);
16     new_img = zeros(m, n);
17
18     g = floor(box_size / 2);
19
20     for i = box_size : 1 : m - box_size
21         for j = box_size : 1 : n - box_size
22             temp = img((i - g) : (i + g), (j - g) : (j + g));
23             new_img(i, j) = sum(sum(temp .* standard_box_filter));
24         end
25     end
26
27     new_img = new_img / (s1 * s2);
28
29     subplot(2, 2, 2);
30     imshow(new_img, []);
31     title("Transformed Image");
```

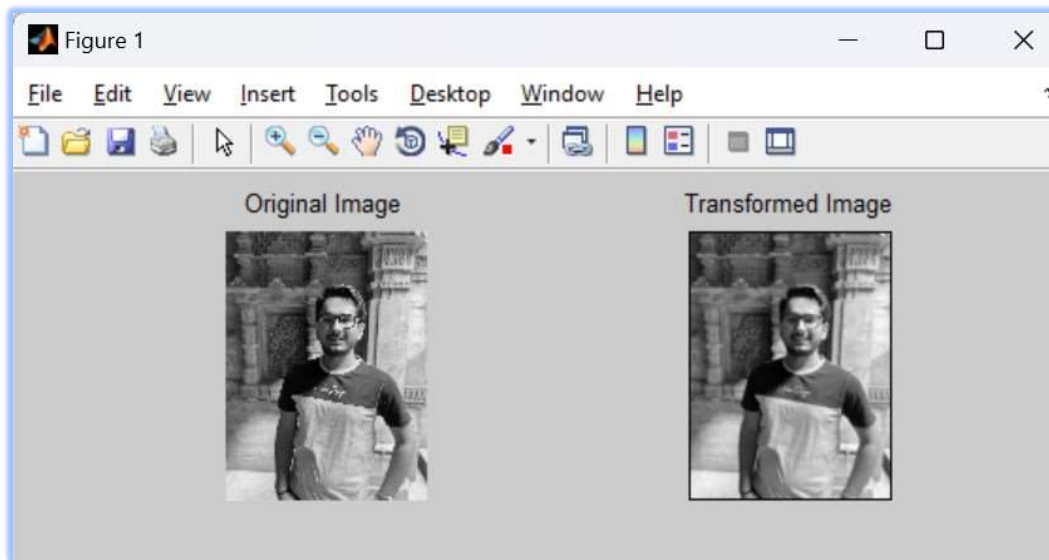
❖ Output of 3X3 filter:



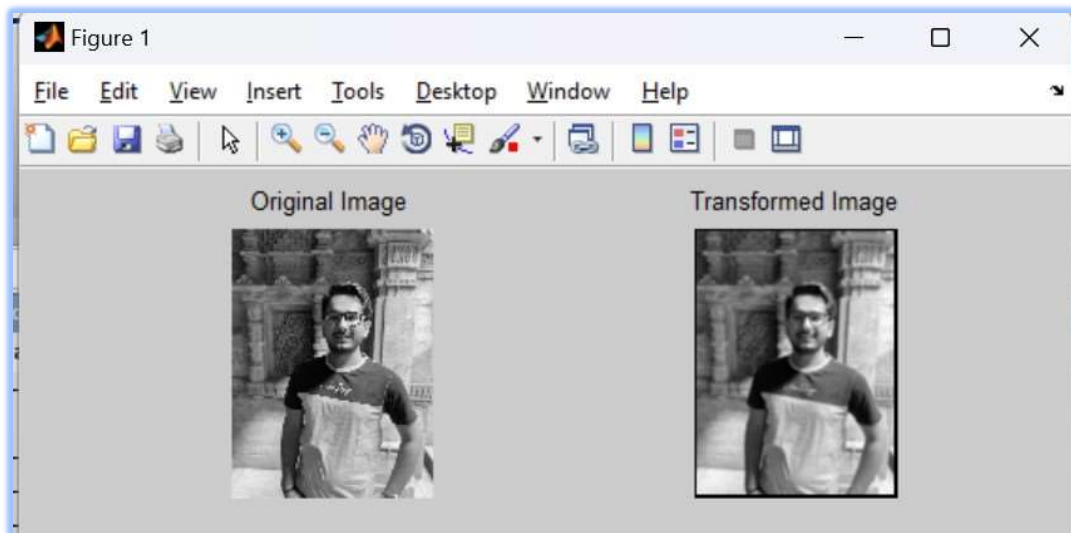
❖ Output of 5X5 filter:



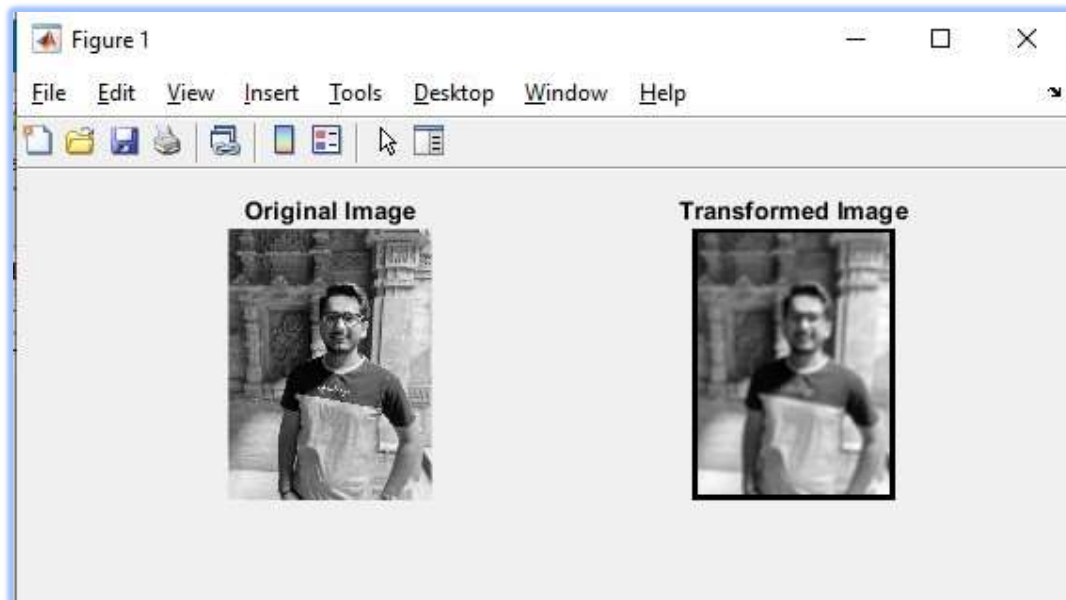
❖ Output of 7X7 filter:



❖ Output of 9X9 filter:



❖ Output of 15X15 filter:

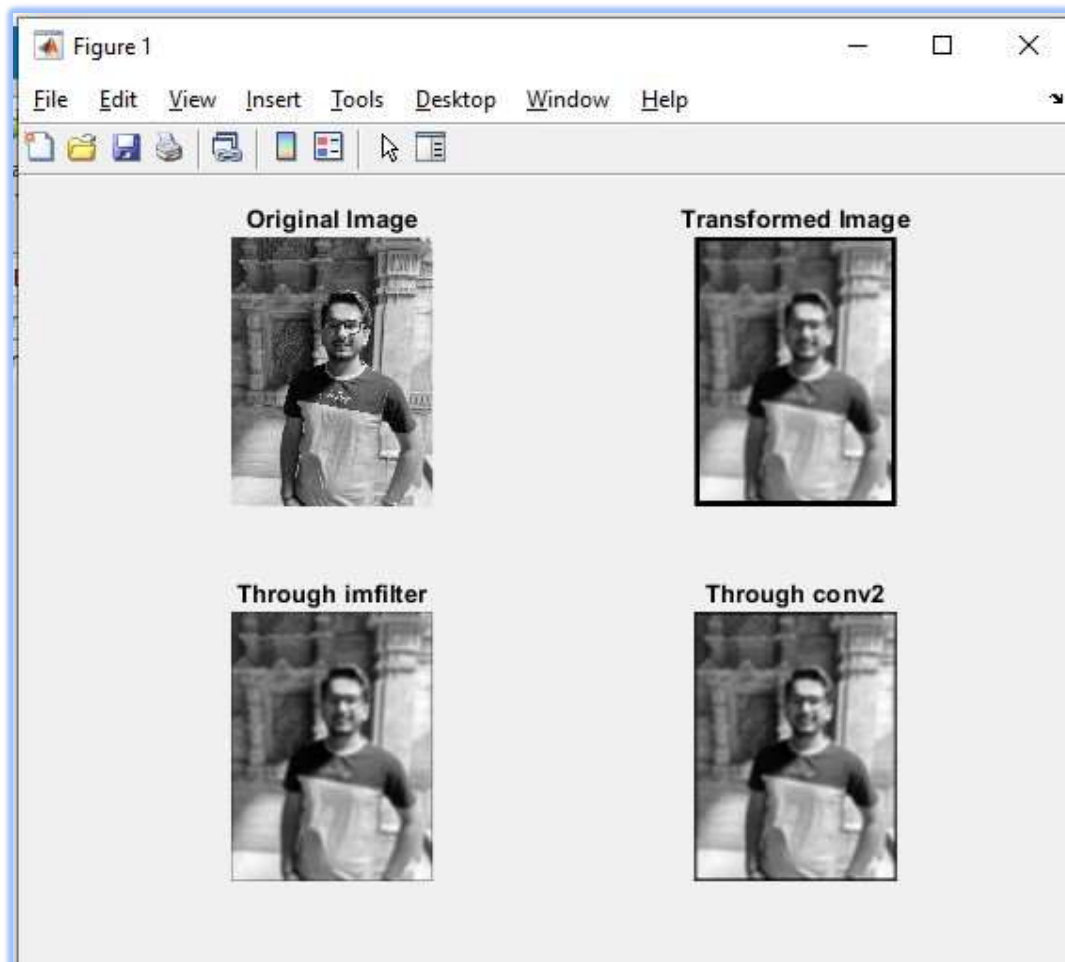


**b) Compare the results with the results obtained using in-built functions like imfilter and conv2.**

❖ Code:

```
33
34 - output_from_built = imfilter(img, standard_box_filter);
35 - subplot(2, 2, 3);
36 - imshow(output_from_built, []);
37 - title('Through imfilter');
38
39 - conv2image = conv2(img, standard_box_filter);
40 - subplot(2, 2, 4);
41 - imshow(conv2image, []);
42 - title('Through conv2');
43
```

❖ Output:



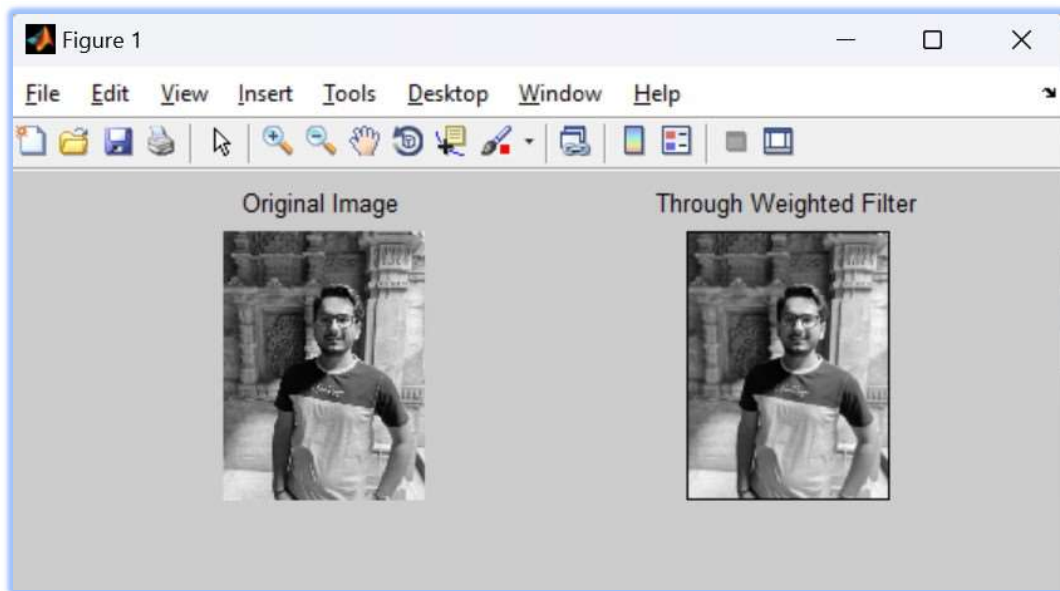
**Q. 2: Take any your gray scale photo and blur it with weighted average filter. Compare the amount of blurring with the standard box filter of the same size.**

❖ Code:

```
a_1.m x a_2.m x +
1      clear all;
2
3      img = imread("Keval_Image.jpg");
4      img = rgb2gray(img);
5      img = imresize(img, 0.1);
6      img = double(img);
7      [m, n] = size(img);
8
9      subplot(2, 2, 1);
10     imshow(img, []);
11     title("Original Image");
12
13     weighted_filter = [1, 2, 1; 2, 4, 2; 1, 2, 1];
14     new_img = zeros(m, n);
15     box_size = 3;
16     [s1, s2] = size(weighted_filter);
17
18     g = floor(box_size / 2);
19
```

```
19
20 - for i = box_size : 1 : m - box_size
21 -     for j = box_size : 1 : n - box_size
22 -         temp = img((i - g) : (i + g), (j - g) : (j + g));
23 -         new_img(i, j) = sum(sum(temp .* weighted_filter));
24 -     end
25 - end
26
27 - new_img = new_img / 16;
28
29 - subplot(2, 2, 2);
30 - imshow(new_img, []);
31 - title('Through Weighted Filter');
32
```

❖ Output:



**Q. 3: Show the impact of multiple passes of smoothing filter of the same size. Derive your conclusion of image quality and maximum number of passes of filter? What happens if infinite number of passes are applied? Will it change image quality?**

❖ **Conclusion:**

- ✓ If we apply smoothing filter multiple times then image quality becomes more and more blurred.
- ✓ Maximum number of passes of filter = 15
- ✓ If we apply the filters infinite number of times then at the end image will not be visible at all.



## ❖ Code:

```
1      clear all;
2
3      img = imread("Keval_Image.jpg");
4      img = rgb2gray(img);
5      img = imresize(img, 0.2);
6      img = double(img);
7      [m, n] = size(img);
8
9      subplot(3, 3, 1);
10     imshow(img, []);
11     title("Original Image");
12
13     box_size = 15;
14     standard_box_filter = ones(box_size, box_size);
15     [s1, s2] = size(standard_box_filter);
16     new_img = zeros(m, n);
17
18     g = floor(box_size / 2);
```

```
temp_img = img;
for k = 1 : 7
    for i = box_size : 1 : m - box_size
        for j = box_size : 1 : n - box_size
            temp = temp_img((i - g) : (i + g), (j - g) : (j + g));
            new_img(i, j) = sum(sum(temp .* standard_box_filter));
        end
    end

    new_img = new_img / (s1 * s2);
    temp_img = new_img;
    subplot(3, 3, k + 1);
    imshow(new_img, []);
end
```

## ❖ Output:

