HW_05_Shah_Niyati

2a. fltr = [1 2 1 ; 0 0 0 ; -1 -2 -1] / 8
Does this filter have its highest response (highest positive value) at places where the image goes light to dark vertically, or diagonally, or is it dark to light? What input conditions cause the largest response

Solution:
Here it detects the horizontal edges. When the image goes from dark to light, a dark/black edge is formed and when the image goes from light to dark a white/light edge is formed.

2b. fltr = [1 0 -1 ; 2 0 -2 ; 1 0 -1] / 8
How does this filter's response differ from part A?

Solution:
Here it detects the vertical edges. Again there is a dark edge formed when the image goes from dark to light and a light edge is formed when it goes from light to dark. Unlike the filter A, the edges are thinner.

2C. fltr = [1 0 0 0 -1 ; 2 0 0 0 -2 ; 1 0 0 0 -1] / 16
How does this filter's response differ from part B?

Solution:
Here it detects the vertical edges. This image is darker than filter B and we can see more details in the grass area and even the walls. The edges formed are thicker than in filter B.

2D. fltr = [1 0 0 0 0 0 0 0 -1 ; 2 0 0 0 0 0 0 0 -2 ; 1 0 0 0 0 0 0 0 -1] /norm_constant
a. Find the correct normalization constant for norm_constant.
What is the value? Describe how you arrived at this and why.
b. How does this filter's response differ from part B?

Solution:
Normalization: 32
As taught by Prof. Kinsman, Normalization is calculated as follows:
(Number of columns of filter -1) * (sum of all positive weights in the filter)

Here the edge detected is much thicker than any of the filters used before. Also due to that at places where there is a small difference between the image going from light to dark and then dark again, these edges have overlapped. So the words on the sign board are not as clear as they were in the output images of filters used before.

2E. [0 -1 0 ; -1 4 -1 ; 0 -1 0]
How does this filter differ from part B? How does the response differ? What do you notice?
Solution:
Here instead of just detecting the horizontal or just vertical edges, this filter detects both the horizontal and the vertical edges. But now the edges are not clearly visible like in the other filters. The response to the change in the color is not clearly seen as seen in the above filters.

2F. fltr = fspecial('laplacian', 1)
Again, describe what you notice. Which other filter is the response of this filter most similar to?
Solution:

A very thin edge is detected when the image goes from light to dark or dark to light. These edges are not continuous and break in between. As the overall image is dark the edges are not as clear as the filters used above.

2G. fltr = fspecial('log')
Again, describe what you notice. Which other filter is the response of this filter most similar to?
What is the log filter?
Solution:

The log filter is also called as the Laplacian of Gaussian filter. It is also a second order derivative operator. It is most similar to filter E. Here the edge detection is not very clear. A very thin white and black edge is detected but as the image is too dark comparatively so the difference is not clearly visible like the others.

3a.

Filter	**Manual Time**	**Imfilter Time**	**Discussions and Observations**
A	0.4154	0.0639	The manual filter and the imfilter function detect the horizontal edges. The edges detected by imfilter was from when the image went from dark to light only. It gave sharper edges which were also seen in the grassy area and the brick area.
B	0.4314	0.0664	Vertical edges were detected when the image went from dark to light only. The edges were sharper and even small edges which were part of the grass are visible through the imfilter function
C	0.5777	0.0741	Here the vertical edges are detected even more sharply than in filter B, a lot of details from the grass, bricks and details from the reflection of things on the window panes are also detected and also it detects only when image goes from dark to light.
D	0.8718	0.0876	Here,we can see sharp vertical edges being detected and also the horizontal edges. The vertical edges are thicker than the horizontal detected edges. Here again the words on the sign board are not clear like they were in the other filters. More edges are detected in the grass, bricks and window panes. And there is a shade of red blue color which is visible in some parts of the image while detecting the edge.
E	0.4310	0.0663	Here we can see thin edges detected vertically and horizontally. But none of these are images detected are complete lines. Only parts of the complete edge is detected at once. Both image going from light to dark and image going from dark to light is detected by a single white line which is not continuous as seen in the other filters. It also detects small changes in the color which can be faintly seen in the grass and bricks

F	0.4225	0.0666	Here it detects both the horizontal and vertical edges when the image goes from both dark to light and light to dark. The edges are sharper with less breaks in between (more continuous line is seen). This filter does not detect small changes like filter E and so the grass and brick area does not have many edges detected.
G	0.9229	0.0789s	Here the output detects both horizontal and vertical edges very clearly and sharply. The minute changes in the grass, bricks and shadows in window planes are also detected and seen as white marks/dots by this filter. The lines here are more continuous than the other filters.

3b. Test to see if imfilter( ) is doing exactly what you think it should be doing? Does it do convolution or local weighting?

Solution:

Imfilter interchanges the filter (gives vertical edges instead of horizontal like in when processing through manual filter.)

It does a local weighting as convolution is a parameter that needs to be passed to the imfilter function.

4. Conclusions:

Solution:

After using the manual filters and the built in matlab imfilter function I learned that the imfilter works faster and efficiently then the manual filters that I wrote. Also it gave a much better result then the manual filtering by giving a sharper edge.
After using the seven filters, I understood how different filters work on the image and how different edges are detected using the different filters.

5. Challenge:

Always feel free to exceed expectations. Perhaps you have done image processing in the past. If you need a challenge you can write a Matlab program to use cell arrays ("doc cell") which collects all of the above timings and results automatically.

Solution:

Stored the image output and times in cells of 4 different arrays named :
output_image_manual, manual_elapsed_time, output_image_filter, filter_elapsed_time
which stores the output of manual filter, time taken by manual filter, output of imfilter and time taken by imfilter respectively.

They are then displayed with a pause to see the output and with the title showing the time taken and which filter used.