Q1. What type of noise is salt & pepper noise? Is it linear or nonlinear noise?
Salt and pepper is a nonlinear noise. It is caused when the scanner settings notare not compatible with the compression algorithm. It is also called impulse noise.

Q2. What effect of the double filter size did you see?
There is no visual change in the resulting image. All of A, A5 and A7 are 216*316 double images. However, the values stored in the resulting image matrix are growing.

Q3. Did you succeed to remove or diminish the impulse noise from the image by this linear operation?
No, we did not succeed to remove or diminish the impulse noise from the image by this linear operation. This is because impulse noise is non-linear and conv2 is a linear filter. We need to use a median filter which is a nonlinear filter to deal with impulse noise.

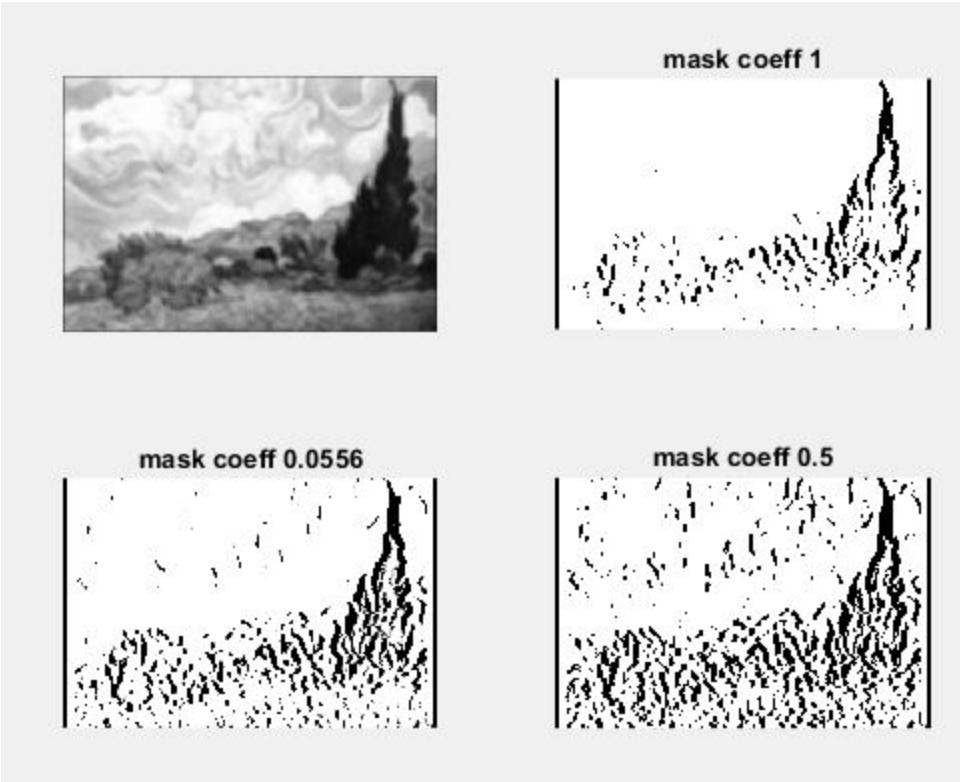
Q4. What was the offset value with size 7*7?
Offset value was 4.

Q5. How did sigma value affect the blurring of the image?
Higher sigma value increased the blurring of the image. Sigma is the standard deviation, in other words, width of the noise histogram. A higher sigma means we are introducing more deviation from the mean level of the noise.

Q6. Did the median filter succeed with the removal of impulse noise?
Yes median filter removed the impulse noise since it is a nonlinear filter.

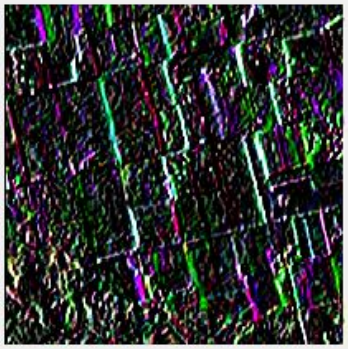
Q7. What filter did the best job? Why is it so?
Median filter did the best job. This is because it replaces the value of the central pixel with the median of intensity levels in the neighborhood of that pixel.

Q8. What filter gave you the satisfactory result (this is of course subjective) in sharpening up the image without significantly corrupting it?
See below results from different masking coefficients. Results depend on what we want to mask. If we mask the whole sky, masking coefficient 1 works best.



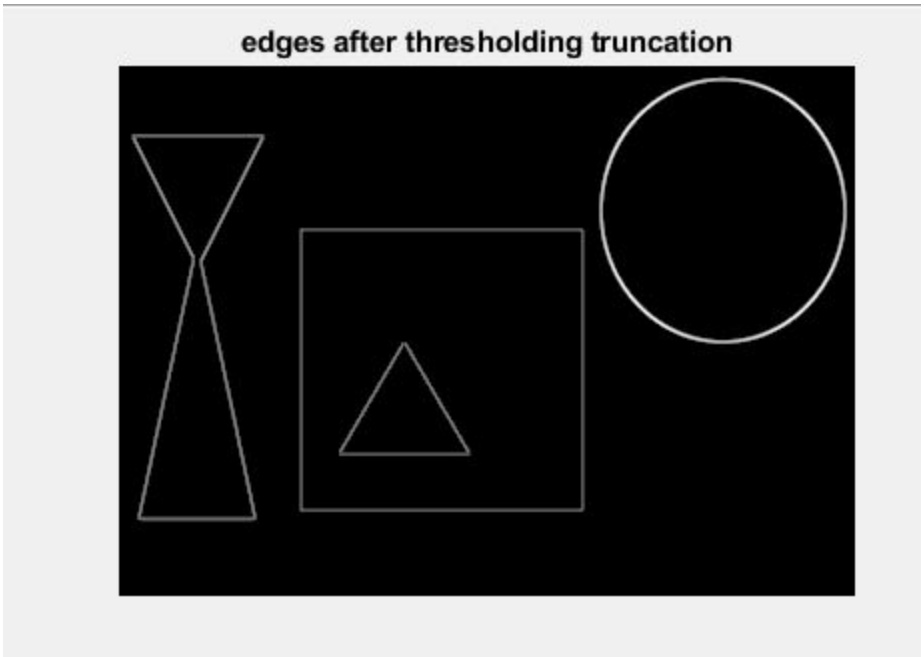
Q9. What direction of edges is detected? Why?
'g' finds edges in the y-axis or vertical edges since the zeros are across the y-axis in the 'g' matrix. It is infact a sobel edge detector.

Q10. Display the new image and compare with the original. What edges have been enhanced? What application can you think of for this kind of edge enhancement?

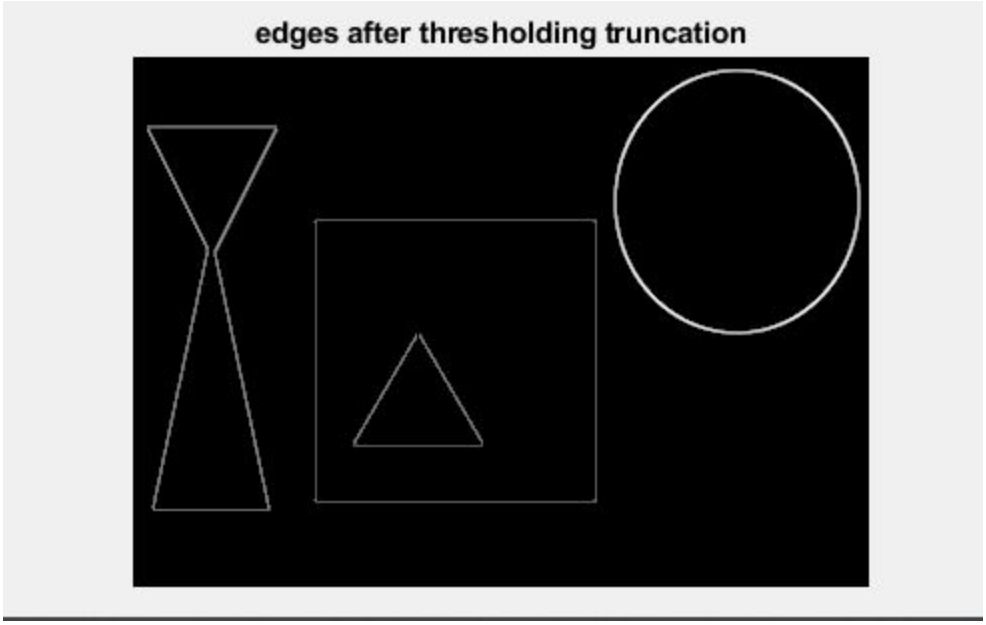


The vertical edges have been enhanced.
This kind of edge enhancement is used in industrial inspection, either to aid humans in detection of defects or as a pre-processing step in automated inspection.
It is used in modern television for sharpness control.

Q11. Change in the find() function by altering the threshold value and explain what happens with the found edges if you change the threshold.
First we see the result with threshold value 2M below,

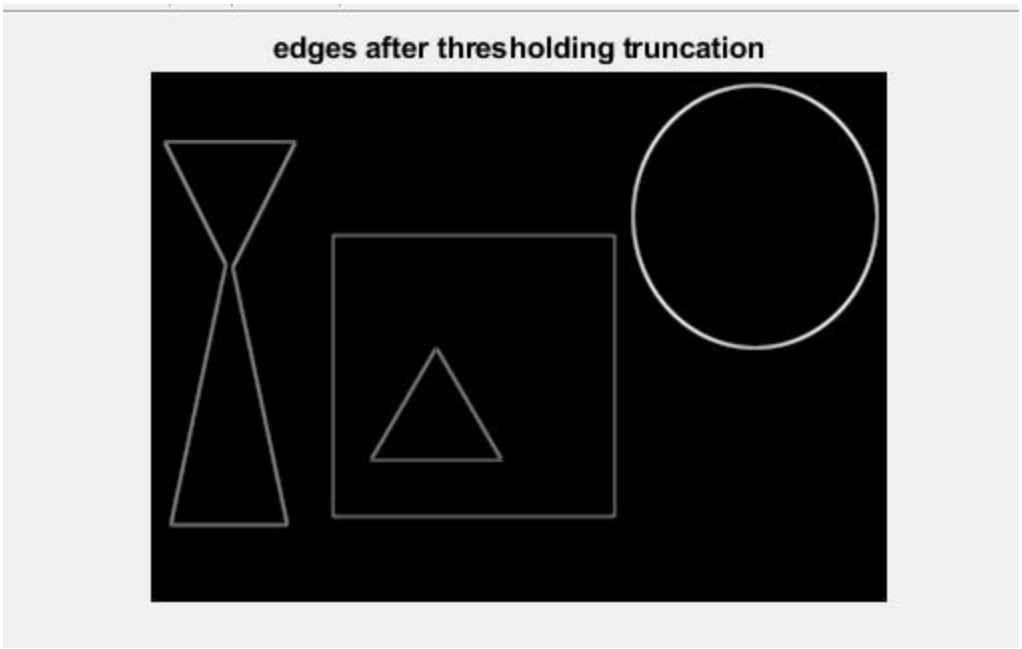


Changing 2M to 3M makes the edges blur as below



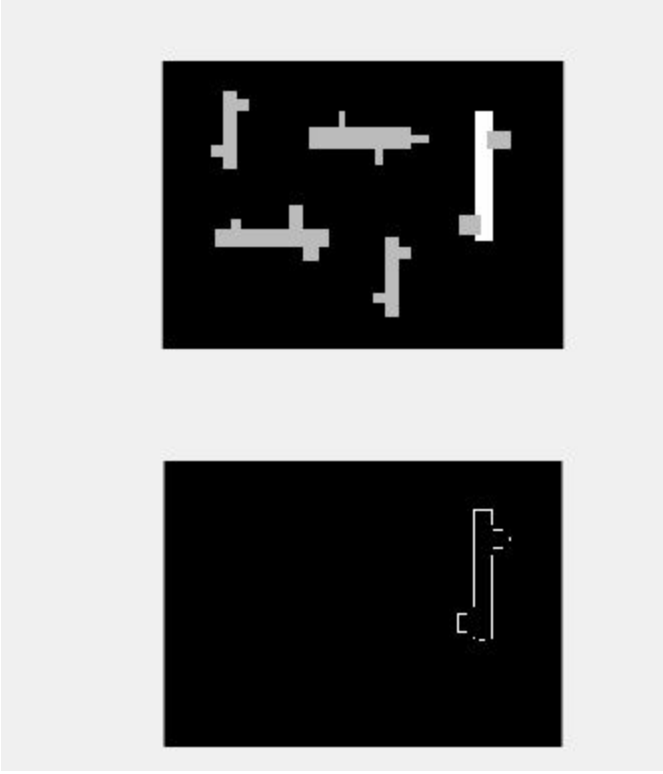
We can see the lines in the first vertical object are not exactly straight.

On the other side, changing 2M to 1M makes the edges sharper.



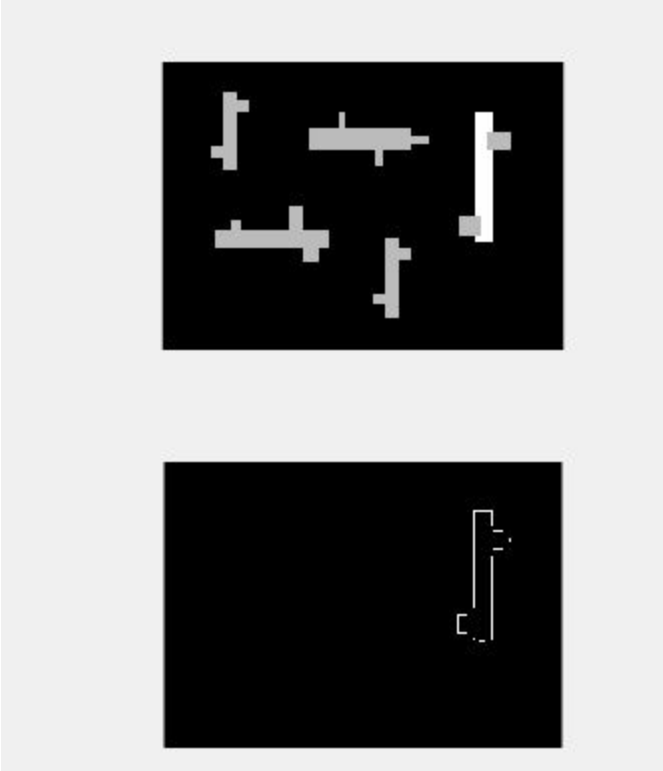
See the first vertical object, the lines are brighter than the lines with threshold 2M.

Q12. Explain shortly what is the aim of thresh parameter. What effect does the decreasing of this value have on the result?
The threshold parameter can have a value between 0 and 1. We first see the result with threshold parameter 0.9 below.



Since the threshold is very high, the algorithm detects only the darkest of the edges.

If we reduce the thresh parameter to 0.1, the algorithm does a good job and finds most of the edges, see below.
SE



Q13. Which edges were detected with each of these filters? Note! You should be able to answer this question without seeing the result, but by filter appearance.
Laplacian detects both horizontal and vertical edges from an image. First filter detects horizontal edges and the second one detects vertical edges.

Q14. What is the max pixel coordinate in the correlated image? You can find it by displaying the correct variable in the Command Window after running the code.
The maximum pixel coordinate in the image is 67382.
It is displayed by ‘ind’ variable in the code below

[v ind] = max(cor(:));

Q15. What happens in the steps 8 and 9? Why do you need to subtract k/2 from r and l/2 from c?
I think in steps 8 and 9 we are trying to move from the maximum correlation point to the midpoint of the target image. Since the target image has size k and l i.e.24X24, we are shifting 12X12 values from the maximum correlation point.

Q16. What are the coordinates of the found point in the original image?
The coordinates of the found point in the original image are 240X126.