Q1

a)



b)



c)



d)



e)



Q2

a)

function [J] = func1( I, M)

    [rows, cols] = size(I);

    fprintf('%d by %d\n', rows, cols);

    J = rand(rows,cols);

    N = (sqrt( M) - 1) / 2;

    for row=1:rows

        for col=1:cols

            sumTemp = 0.0;

            for r=(row-N):(row+N)

                for c=(col-N):(col+N)

                    if( r > 0) && (c > 0) && ( r <= rows) && ( c <= cols)

                        sumTemp = double(sumTemp) + double( I(r, c));

                    end

                end

            end

            J(row, col) = double(sumTemp) / double(M) / 255.0;

        end

    end

end

b)



c)

9



25



121



d)

9



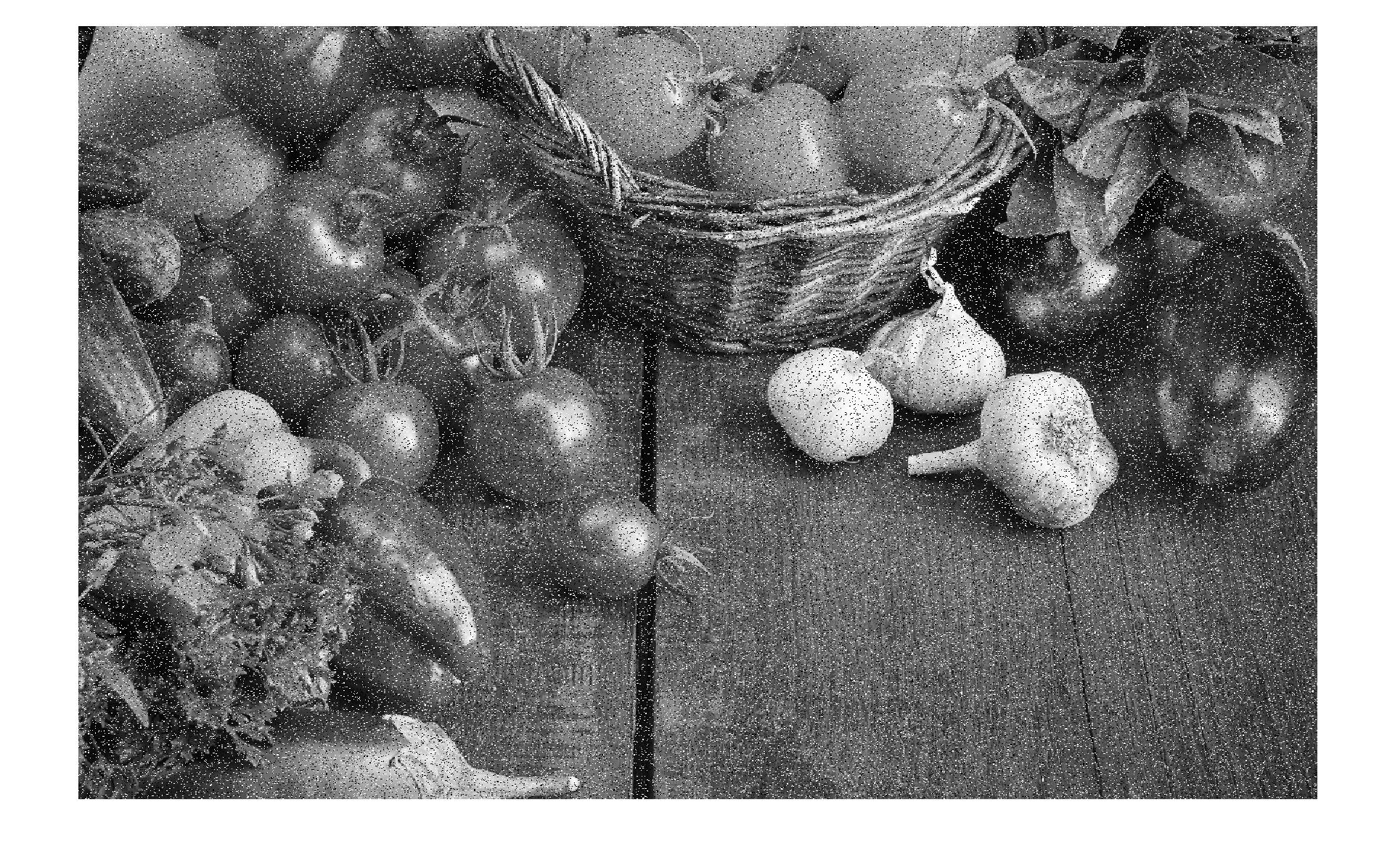
25



121



e)



f)

9



25



121



Q3:

a)

function [J] = func2( I, M)

    [rows, cols] = size(I);

    fprintf('%d by %d\n', rows, cols);

    J = rand(rows,cols);

    N = (sqrt( M) - 1) / 2;

    for row=1:rows

        for col=1:cols

            %Making a temporary matrix with 0 values

            len = (N \* 2) + 1;

            tempMatrix = zeros(len,len);

            if (row - N) < 1; Xstart = 1; else; Xstart = row - N; end

            if (row + N) > rows; Xend = rows; else; Xend = row + N; end

            if (col - N) < 1; Ystart = 1; else; Ystart = col - N; end

            if (col + N) > cols; Yend = cols; else; Yend = col + N; end

            xTemp = 1;

            for a=Xstart:Xend

                yTemp = 1;

                for b=Ystart:Yend

                    tempMatrix(xTemp, yTemp) = I(a, b);

                    yTemp = yTemp + 1;

                end

                xTemp = xTemp + 1;

            end

            %Finding median from temporary matrix

            medRow = double(median( tempMatrix(1:len,1:len)));

            medFinal = median( medRow);

            fprintf("%f, ", medFinal);

            J(row, col) = double(medFinal) / 255.0;

        end

    end

end

b)



c)

9



25



121



d)

e)

f)