Project Report

Team Members:

- Harikrishnan K.P (150260026)
- Pranjal R.S (150260030)

AIM

Write the code for 2D convolution in MATLAB and using it pass different images through the given filters and make observations about the output.

PROCEDURE

A function for 2D convolution was written, which takes the input signal and the filter as input parameters and returns the convoluted signal. This file titled '*convolution_2d.m*' has been uploaded. Different images were then passed through the filters, both black and white and colour and observations were made.

The 2 filters given are:

- h1 = [1 1 1; 1 1 1; 1 1 1]/9

- h2 = [0 -1 0; -1 4 -1; 0 -1 0]/4

CODE

This function contains code for the convolution of x & h, where x is the input and h is the filter. This file has been uploaded, titled '*convolution_2d.m*'.

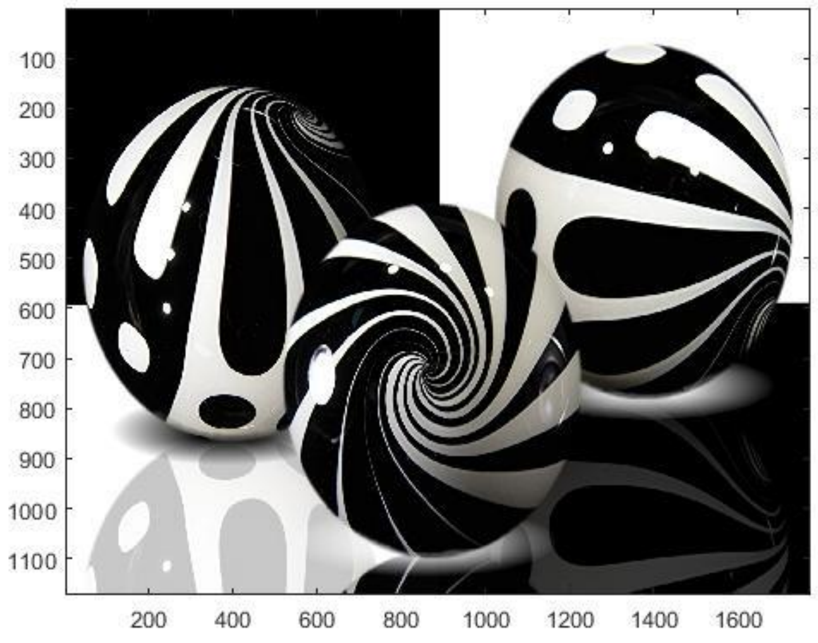
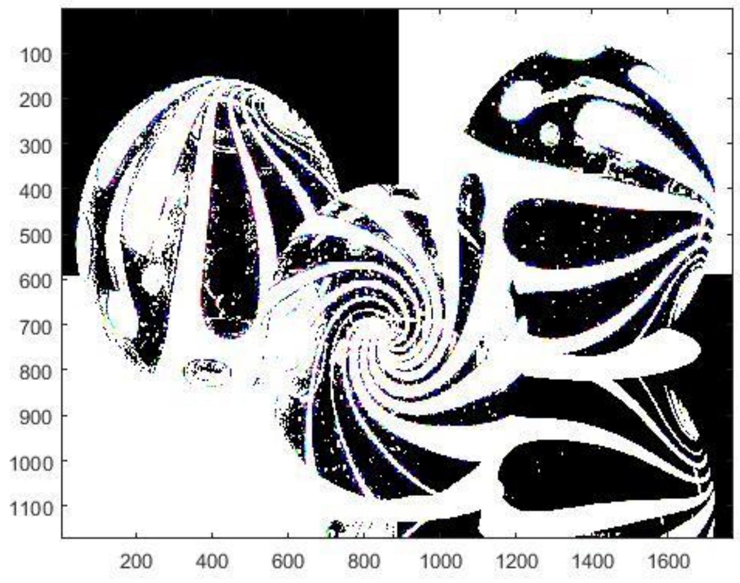
```
function z=convolution_2d(x,h)
% Receiving the image and filter as input
        z=zeros(size(x));
% Defining an empty matrix for output
        x=[zeros(size(x,1),1) x];
% Input array padded with column of zeros on left
        x=[x zeros(size(x,1),1)];
% Input array padded with column of zeros on right
        x=[zeros(1,size(x,2)); x];
% Input array padded with row of zeros on top
        x=[x;zeros(1,size(x,2))];
% Input array padded with row of zeros on bottom
        r=size(z,1);
% Number of rows in output matrix
        c=size(z,2);
% Number of columns in output matrix
        for m=1:r
% For loop covering every row
            for n=1:c
% For loop covering every column
                        z(m,n)=x(m+1-1,n+1-1)*h(3,3)+x(m+1-
1,n+1)*h(3,2)+x(m+1-1,n+1+1)*h(3,1)+x(m+1,n+1-
1)*h(2,3)+x(m+1,n+1)*h(2,2)+x(m+1,n+1+1)*h(2,1)+x(m+1+1,n+1-
1)*h(1,3)+x(m+1+1,n+1)*h(1,2)+x(m+1+1,n+1+1)*h(1,1);
% Convolution
            end
        end
```

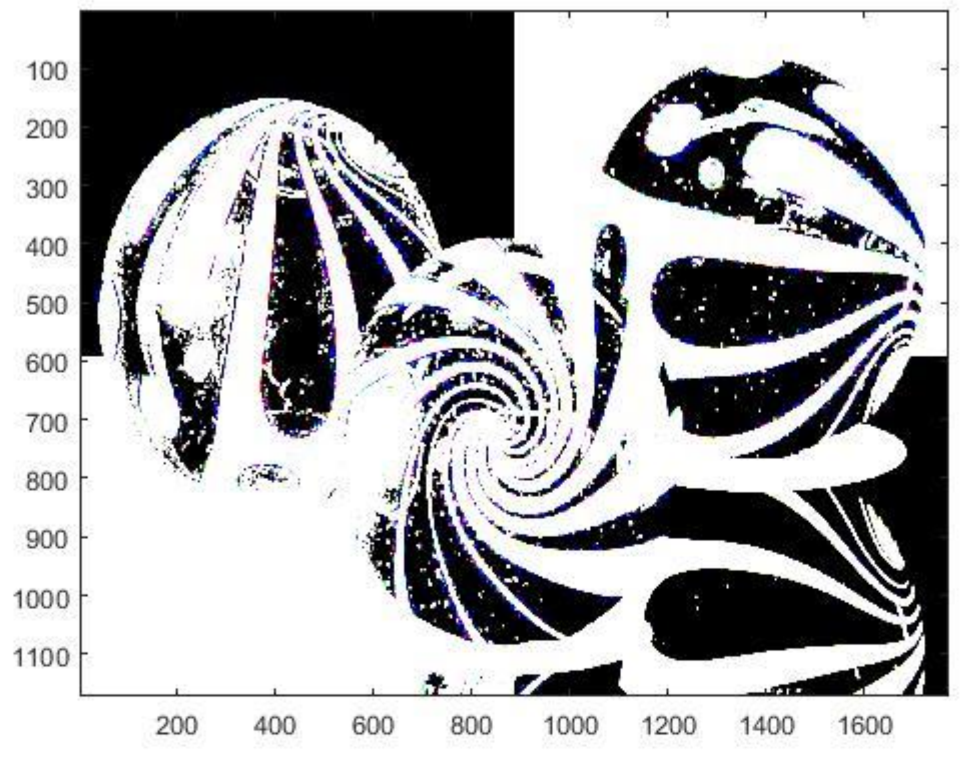
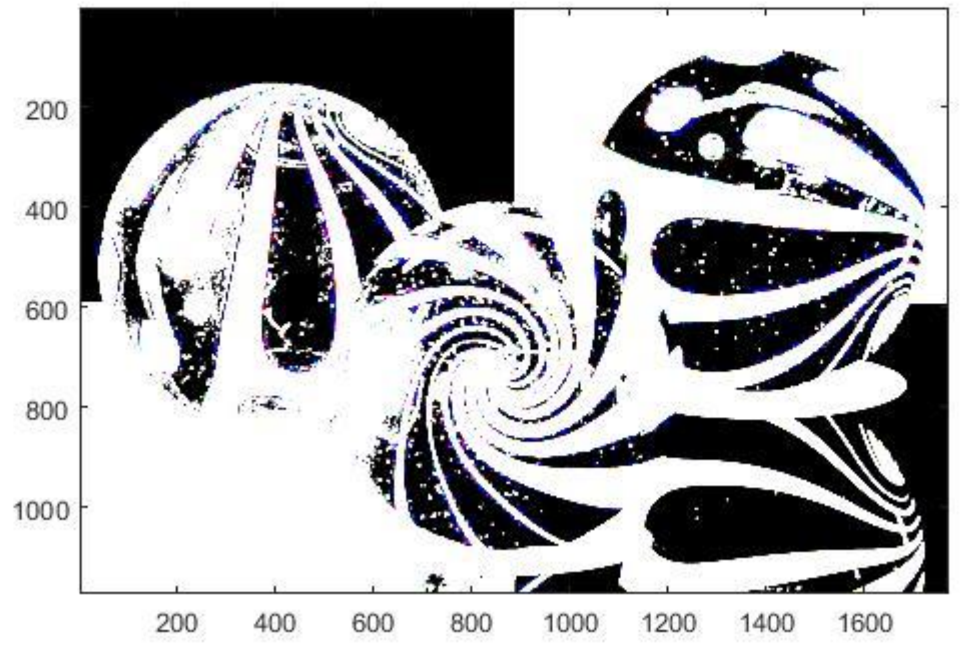
The following code reads an image file and using the '*convolution_2d'* function defined above, obtains the output when it is passed through the filter up to three times. This file has been uploaded titled '*image_processing.m*'.

```
% A black and white image has been filtered in this code, 
change image
%   files and filter required to view various results.
% The code for convolution has been separately written as a 
function named
% 'convolution_2d.m' and has been used in the code for 
filtering.
a=imread('bw.jpg');                  % reading the image
r=a(:,:,1);                           % splitting the RGB array 
into 3 separate 2D arrays
g=a(:,:,2);
b=a(:,:,3);
h=ones(3)*(1/9);                      % defining the filters
% h=[0 -1 0;-1 4 -1;0 -1 0]*(1/4);
r1=convolution_2d(r,h);               % passing the R,G,B arrays 
through the filter
g1=convolution_2d(g,h);
b1=convolution_2d(b,h);
res1=cat(3,r1,g1,b1);                 % Reforming the image by 
assembling the filtered 2D arrays
r2=convolution_2d(r1,h);              % Repeated filtering
g2=convolution_2d(g1,h);
b2=convolution_2d(b1,h);
res2=cat(3,r2,g2,b2);
r3=convolution_2d(r2,h);
g3=convolution_2d(g2,h);
b3=convolution_2d(b2,h);
res3=cat(3,r3,g3,b3);
figure(1);
image(a);                             % Original image
figure(2);
image(res1);                          % Image after filtering 
once
figure(3);
image(res2);                          % Image after filtering 
twice
figure(4);
image(res3);                          % Image after filtering 
thrice
```

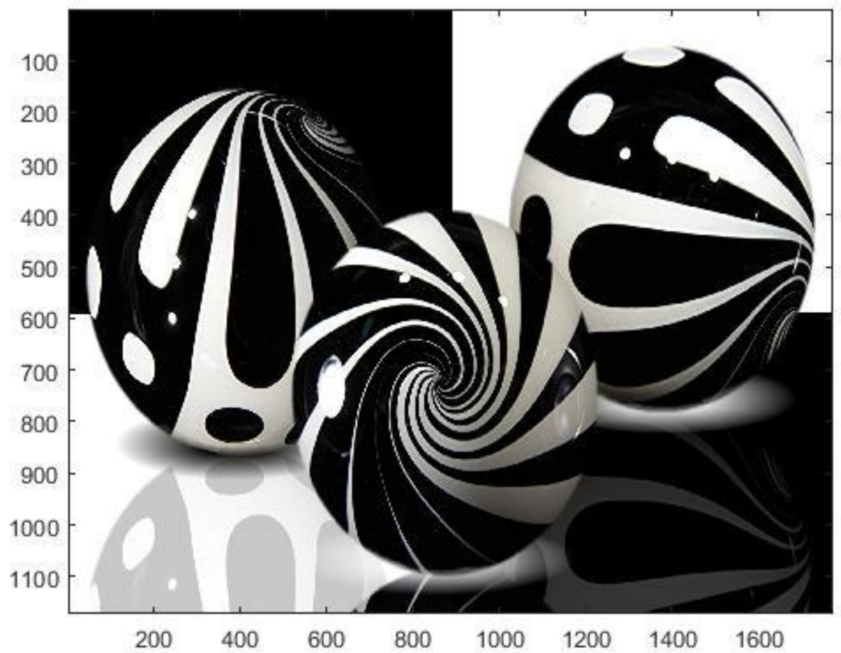
OBSERVATIONS

On passing a black and white image *‘bw.jpg’* through h1 filter, the following observations were made:

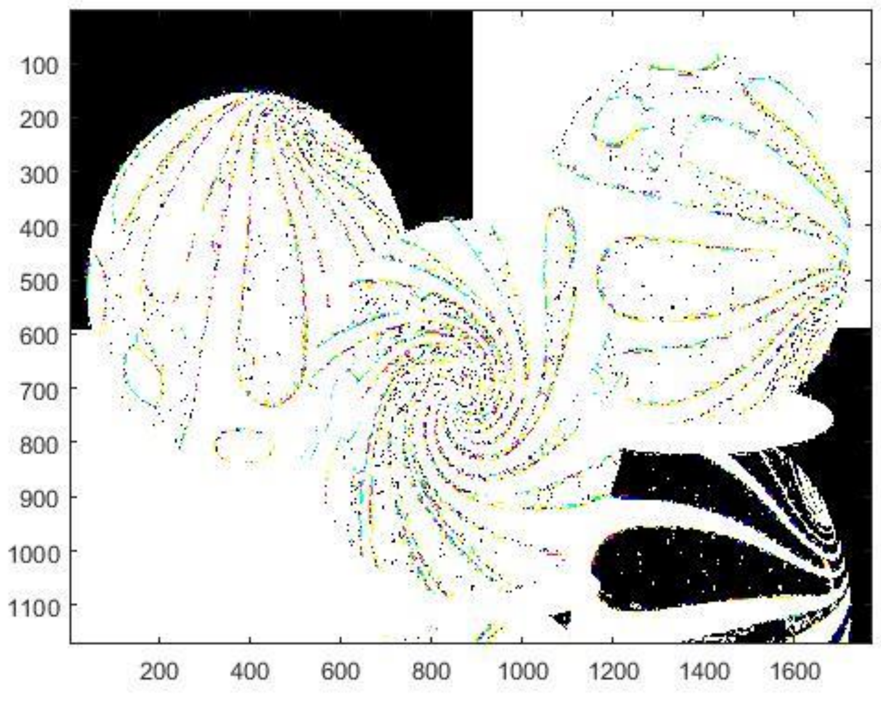

Original Image

After passing through filter once


After passing through filter twice

After passing through filter thrice

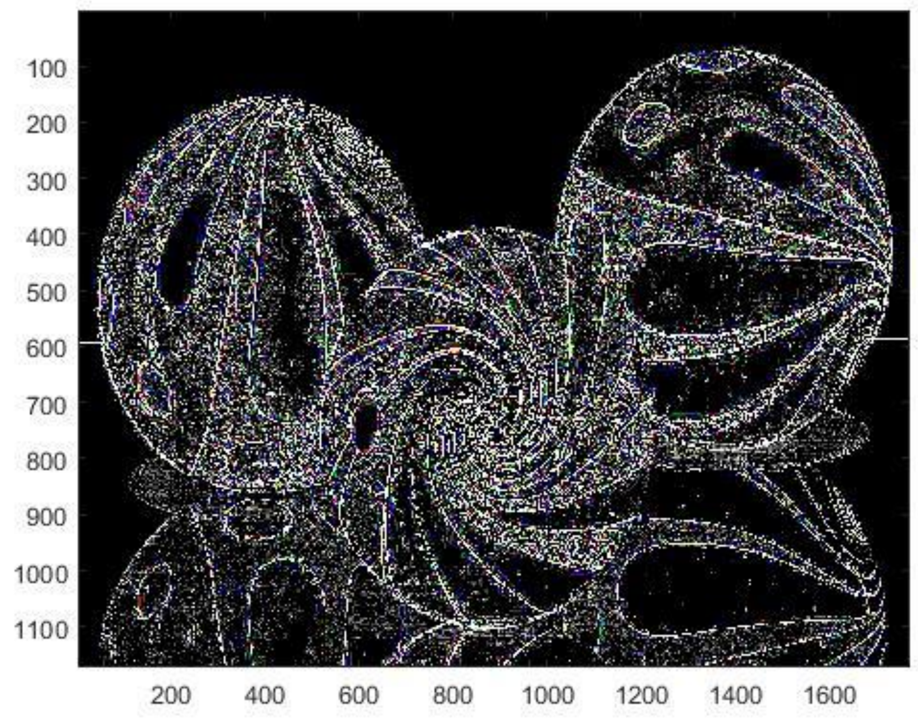
On passing the same image through **h2 filter**, we observed:



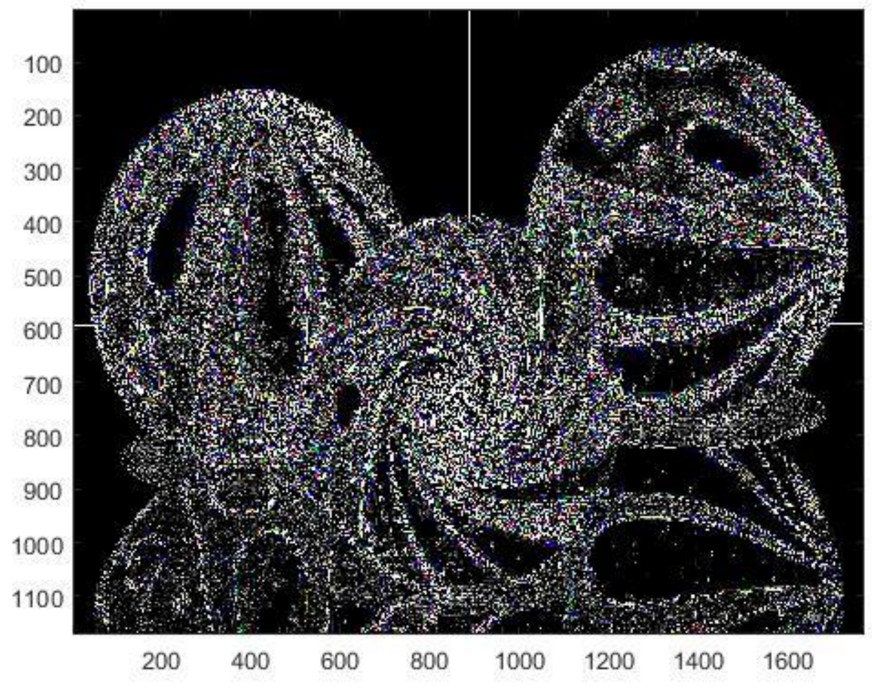
Original image



After passing through filter once

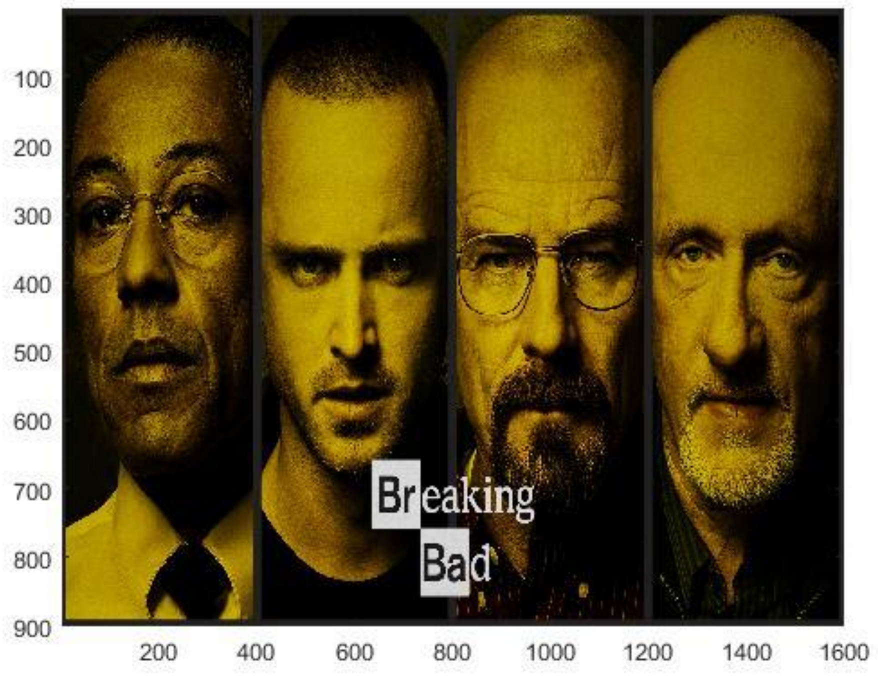


After passing through filter twice

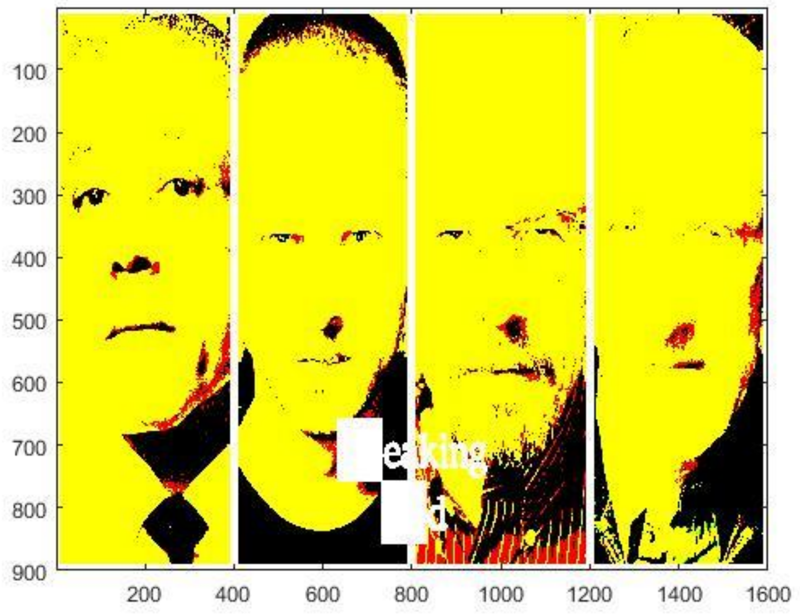


After passing through filter thrice

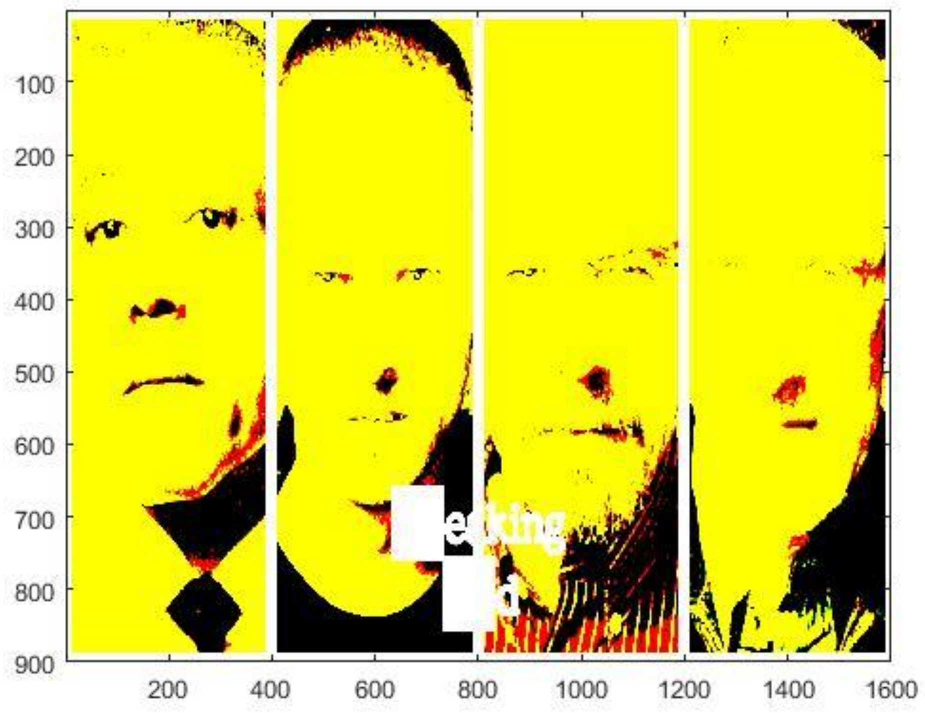
On passing the colour image *‘breaking.png’* through h1, we obtained these results.



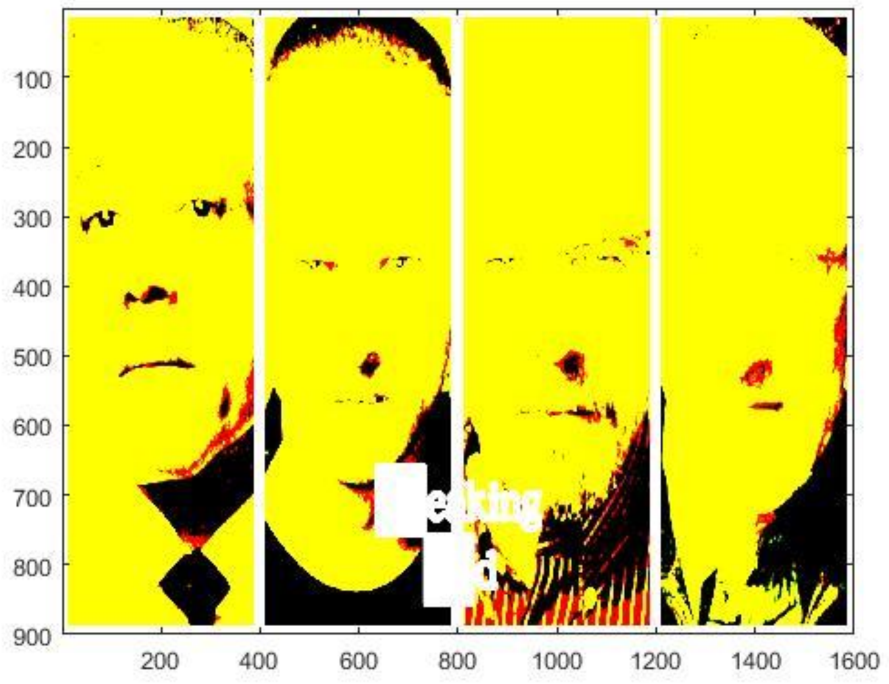
Original Image



After passing through filter once

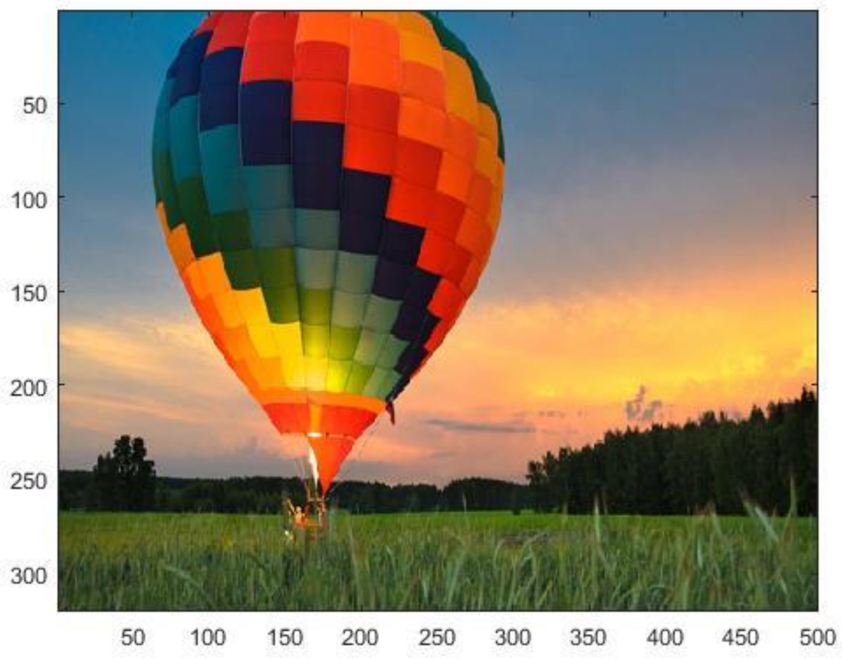


After passing through filter twice

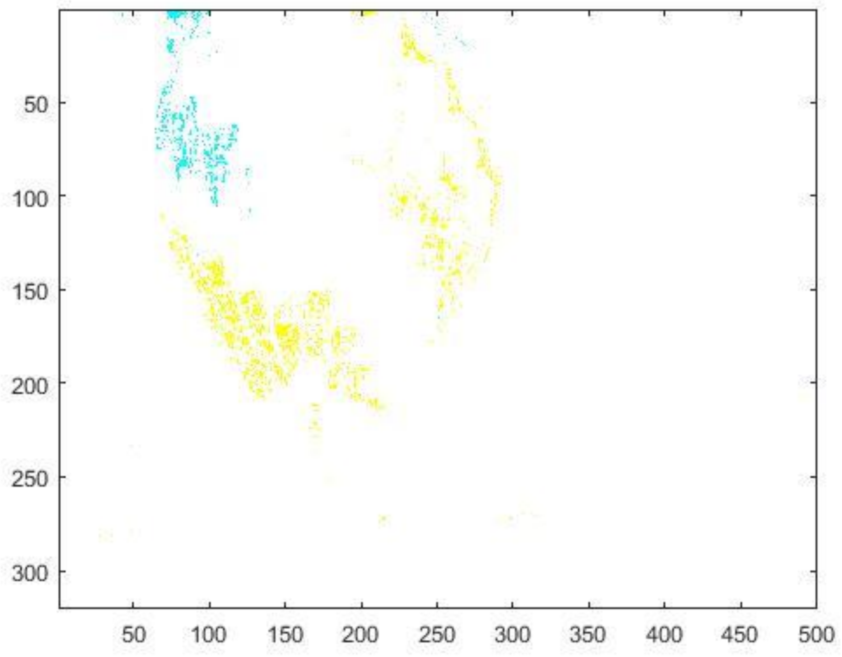


After passing through filter thrice

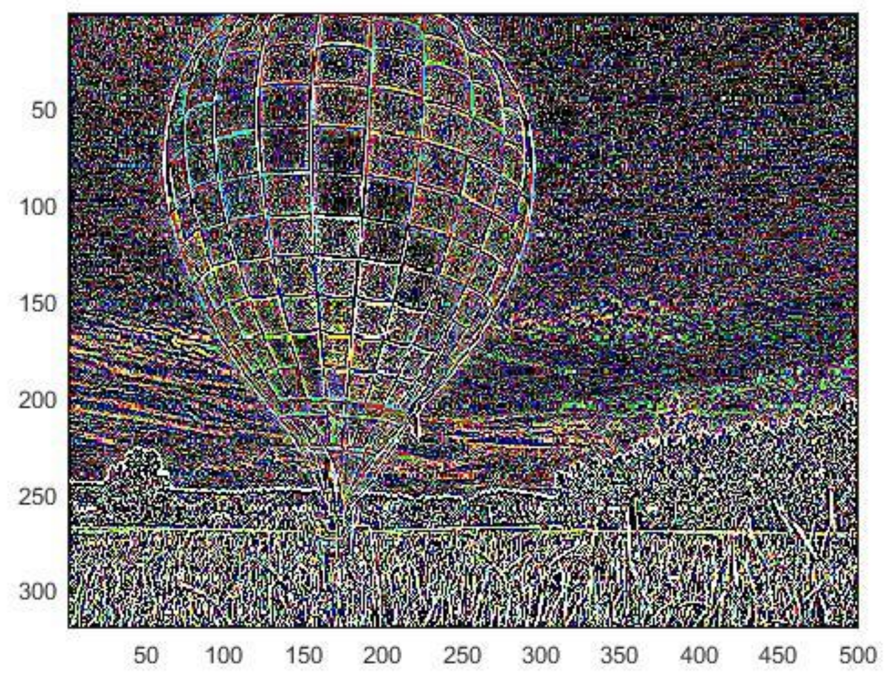
Now a different colour image *'para.jpg'* is passed through h2.



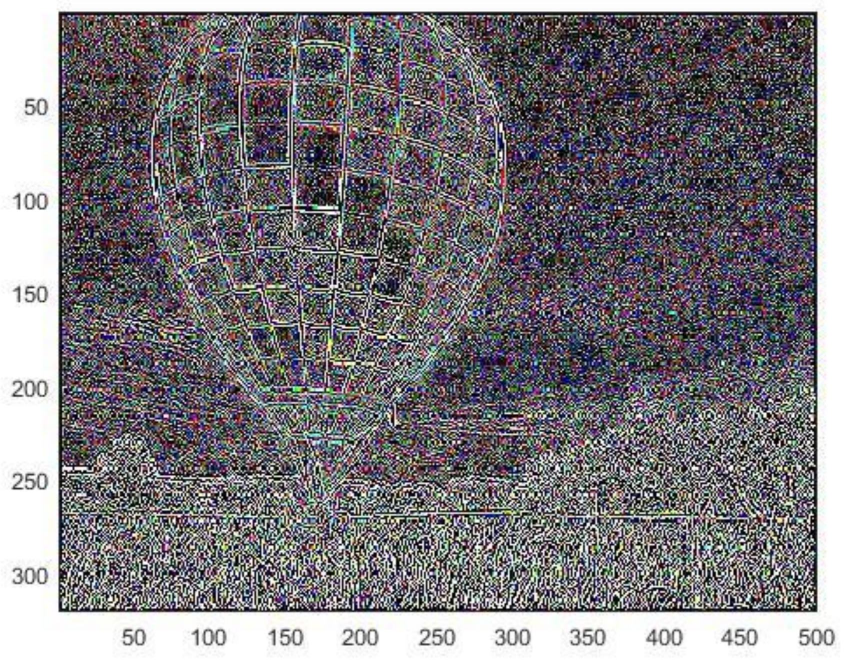
Original Image



After passing through filter once

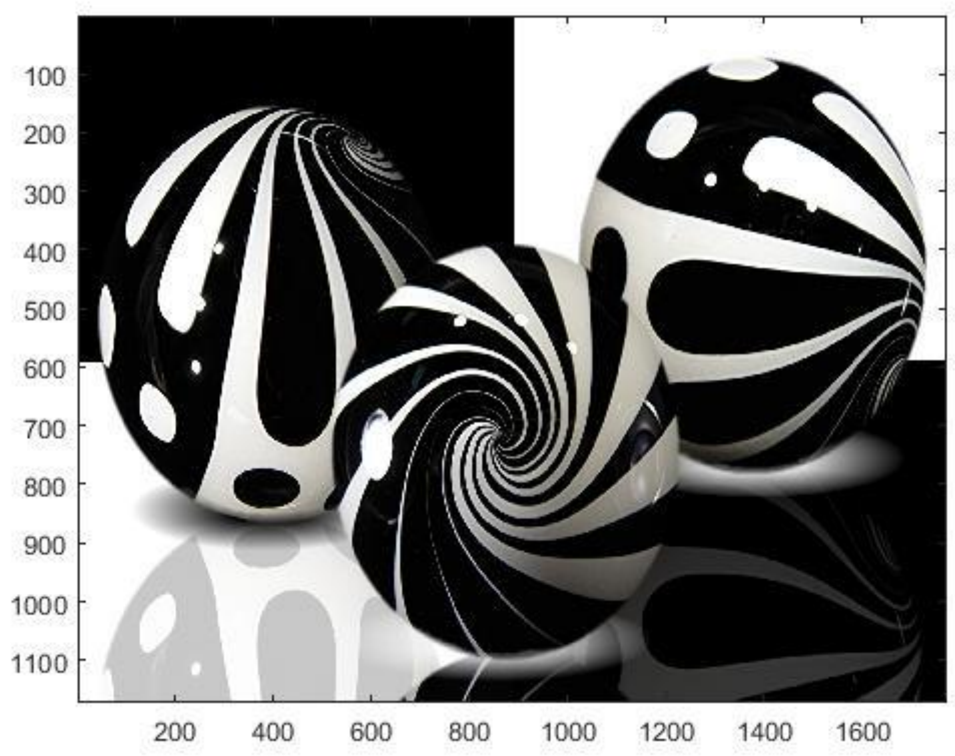


After passing through filter twice

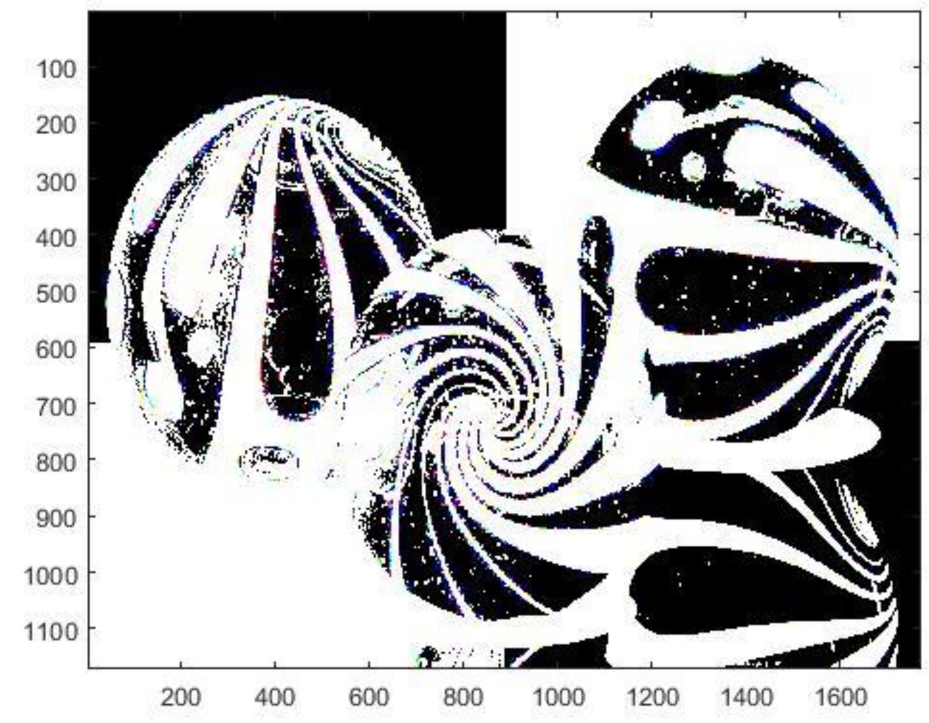


After passing through filter thrice

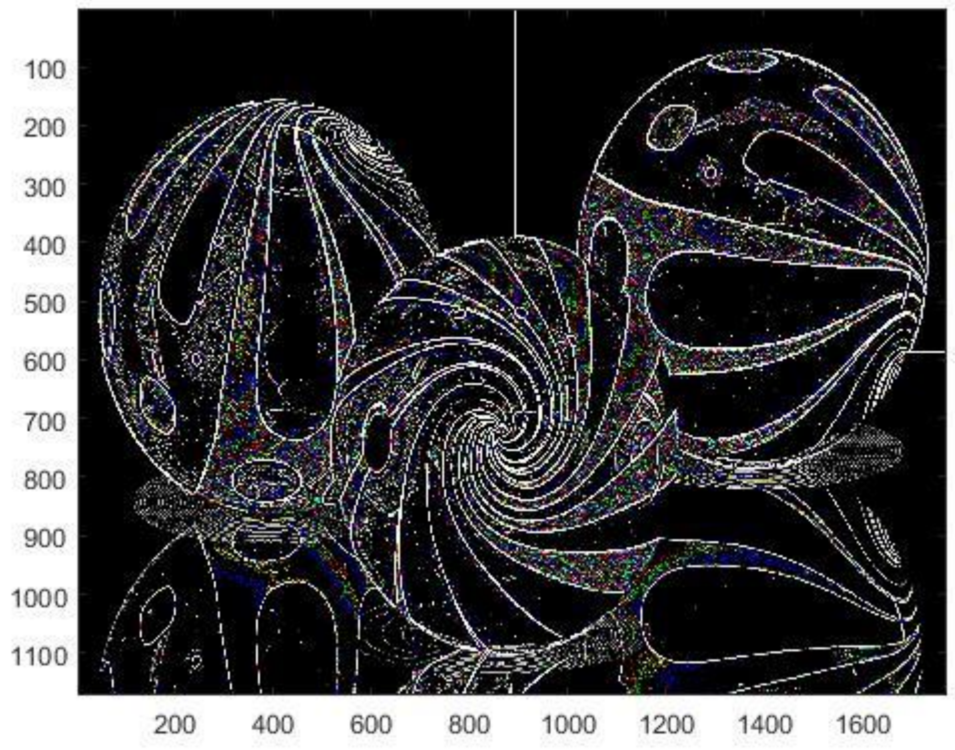
Now the black and white image is passed first through h1 and then through h2. Here are the results.



Original Image



After passing through filter h1



After passing through filter h1 and then h2

RESULTS

On passing the images through h1, they get blurred. Each pixel in the resulting image has a value equal to the average of the value of pixels around it. This is a simple **blurring filter**.

On passing images through h2, the output mainly contains the edges of the input picture. The image as a whole is blurred, but edges of various objects, where there are discontinuities can be observed clearly. This is an **edge detection filter**.