Student: Melis Kilic

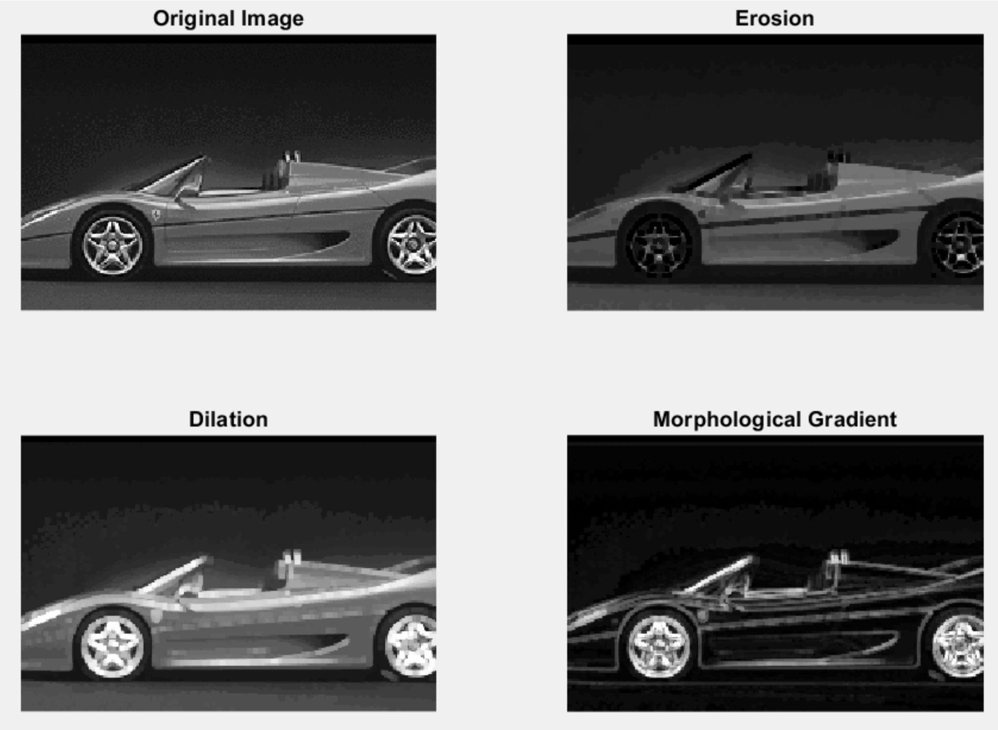
Morphological operations for grayscale images

MG-1. Morphological operations for greyscale images

Task 1a Erosion and dilation

```
%% Task 1a Erosion and dilation

clearvars;
close all;
clc;

image = imread('ferrari.bmp');
grayImage = rgb2gray(image);

se = strel('square', 3);

erosion = imerode(grayImage, se);
dilation = imdilate(grayImage, se);

morphGradient = dilation - erosion;

figure;

subplot(2, 2, 1);
imshow(grayImage);
title('Original Image');

subplot(2, 2, 2);
imshow(erosion);
title('Erosion');

subplot(2, 2, 3);
imshow(dilation);
title('Dilation');

subplot(2, 2, 4);
imshow(morphGradient);
title('Morphological Gradient');
```

Result of the code:



What changes after dilation and what changes after erosion?

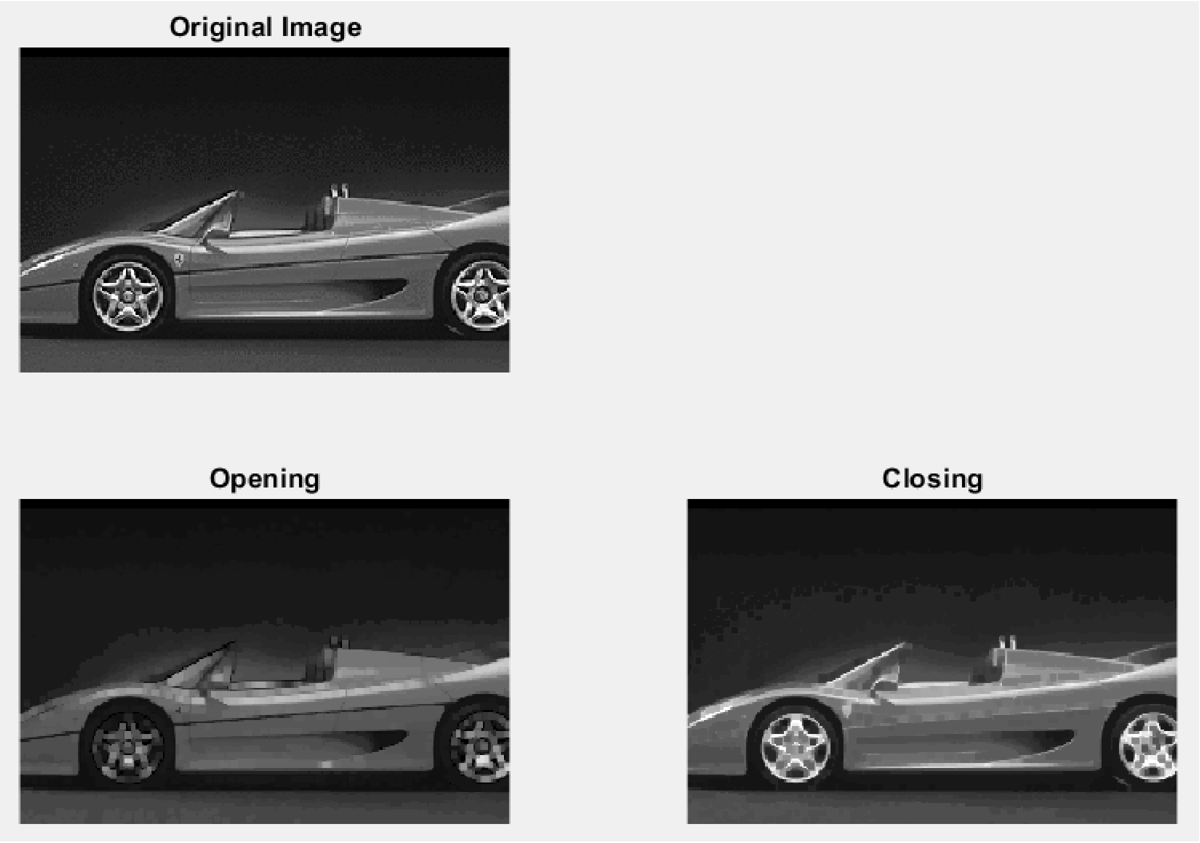
After dilation, bright regions expand, and small dark gaps are filled. After erosion, bright regions shrink, and small dark spots are removed. So, dilation makes bright areas larger, while erosion makes them smaller.

Task 1b Opening and closing

```
%% Task 1b Opening and closing

clearvars;
close all;
clc;

image = imread('ferrari.bmp');
grayImage = rgb2gray(image);

se = strel('square', 3);

opening = imopen(grayImage, se);
closing = imclose(grayImage, se);

figure;

subplot(2, 2, 1);
imshow(grayImage);
title('Original Image');

subplot(2, 2, 3);
imshow(opening);
title('Opening');
```

```
subplot(2, 2, 4);
imshow(closing);
title('Closing');
```

Result of the code:

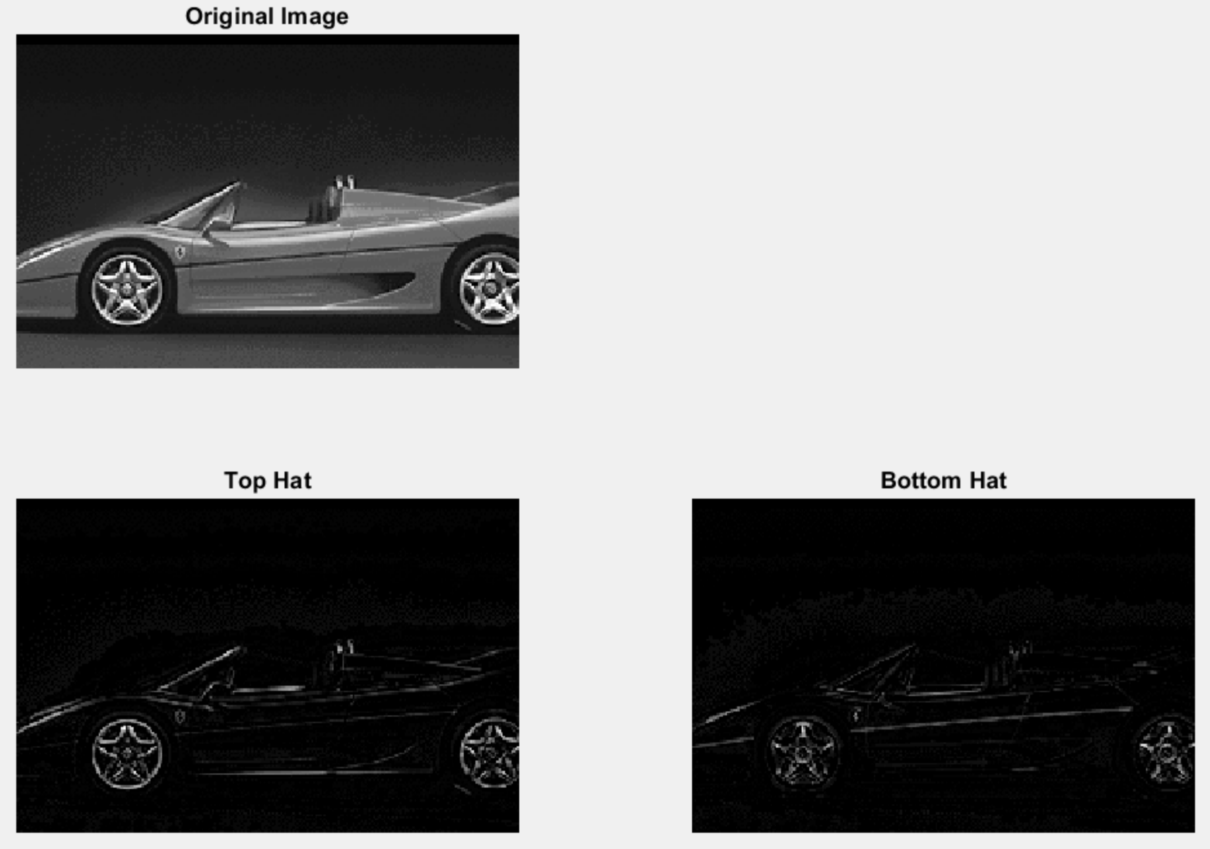


Task 1c Top hat and bottom hat

```
%% Task 1c Top hat and bottom hat

clearvars;
close all;
clc;

image = imread('ferrari.bmp');
grayImage = rgb2gray(image);

se = strel('square', 3);

topHat = imtophat(grayImage, se);
bottomHat = imbothat(grayImage, se);

figure;

subplot(2, 2, 1);
imshow(grayImage);
title('Original Image');
```

```
subplot(2, 2, 3);
imshow(topHat);
title('Top Hat');

subplot(2, 2, 4);
imshow(bottomHat);
title('Bottom Hat');
```

Result of the code:

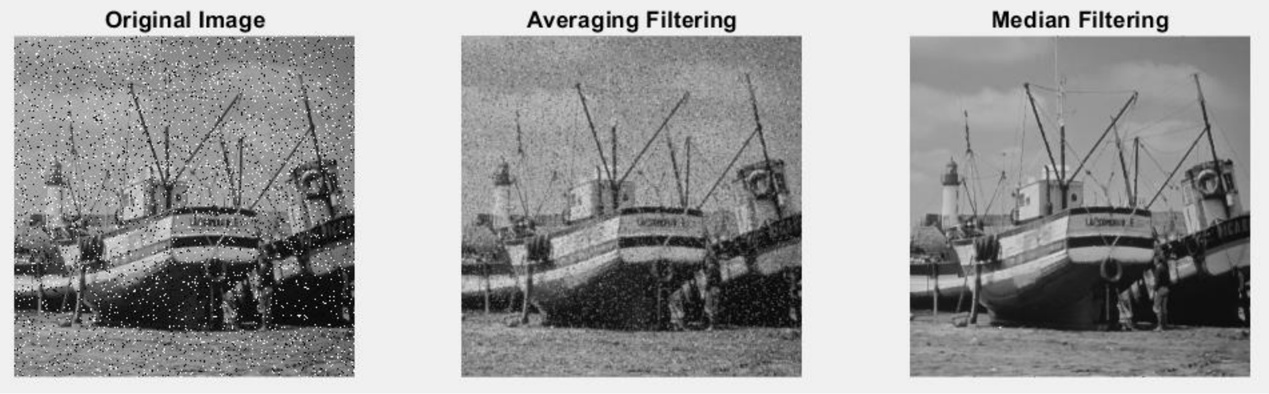


MG-2. Median filtering

```
close all;
clearvars;
clc;

imageWithNoise = imread('boatsBWNoise.bmp');

averageFilter = fspecial('average');
averagingResult = filter2(averageFilter, imageWithNoise);

median = medfilt2(imageWithNoise);

figure;

subplot(1, 3, 1);
imshow(imageWithNoise);
title('Original Image');
```

```
subplot(1, 3, 2);
imshow(uint8(averagingResult));
title('Averaging Filtering');

subplot(1, 3, 3);
imshow(median);
title('Median Filtering');
```

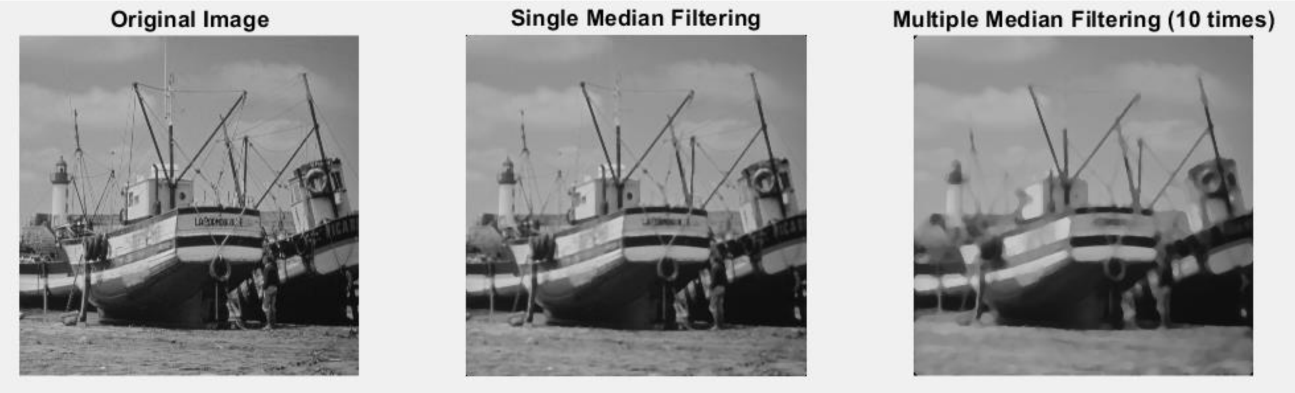
Result of the code:



MG-3. Posterization

```
close all;
clearvars;
clc;

image = imread('boatsBW.bmp');

singleMedian = medfilt2(image, [5, 5]);

% Perform median filtering 10 times with context size 5×5
multipleMedianResult = image;
for i = 1:10
    multipleMedianResult = medfilt2(multipleMedianResult, [5, 5]);
end

figure;

subplot(1, 3, 1);
imshow(image);
title('Original Image');

subplot(1, 3, 2);
imshow(singleMedian);
title('Single Median Filtering');

subplot(1, 3, 3);
imshow(multipleMedianResult);
title('Multiple Median Filtering (10 times)');
```

Result of the code:



Original Image

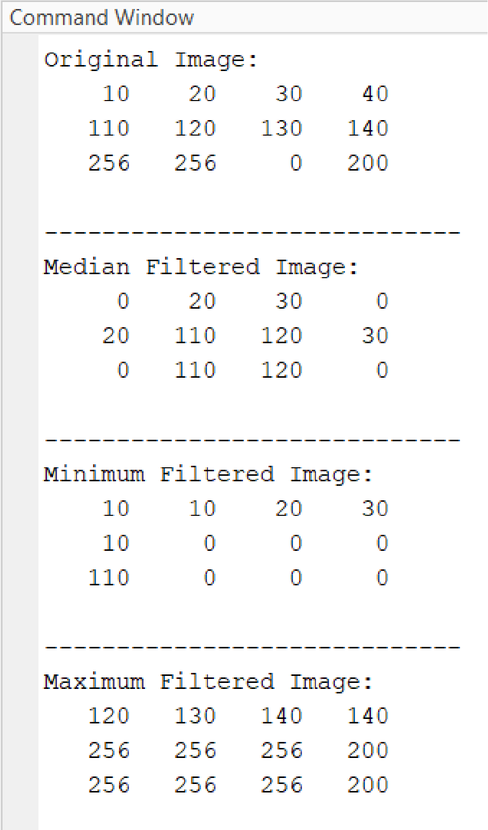
Single Median Filtering

Multiple Median Filtering (10 times)

MG-4. Non-linear filtering of an artificial image

```
close all;
clearvars;
clc;

A = [ 10  20  30  40;
     110 120 130 140;
     256 256   0 200];

se = strel('rectangle', [3, 3]);

medianFilteredA = medfilt2(A, [3, 3]);
minFilteredA = imerode(A, se);
maxFilteredA = imdilate(A, se);

disp('Original Image:');
disp(A);
disp('-----------------------------');
disp('Median Filtered Image:');
disp(medianFilteredA);
disp('-----------------------------');
disp('Minimum Filtered Image:');
disp(minFilteredA);
disp('-----------------------------');
disp('Maximum Filtered Image:');
disp(maxFilteredA);
```

Result of the code:



```
Original Image:
    10    20    30    40
   110   120   130   140
   256   256     0   200

-----------------------------
Median Filtered Image:
     0    20    30     0
    20   110   120    30
     0   110   120     0

-----------------------------
Minimum Filtered Image:
    10    10    20    30
    10     0     0     0
   110     0     0     0

-----------------------------
Maximum Filtered Image:
   120   130   140   140
   256   256   256   200
   256   256   256   200
```

MG-5. Morphology – case study

```
close all;
clearvars;
clc;

image = imread('calculator.bmp');

% Method 1: Using Classical Opening
se1 = strel('line', 71, 0); % Horizontal line
openingResult1 = imopen(image, se1);
topHatResult1 = image - openingResult1;

% Method 2: Using Opening by Reconstruction
erosionResult2 = imerode(image, se1);
reconstructedResult2 = imreconstruct(erosionResult2, image);
topHatResult2 = image - reconstructedResult2;

figure;

subplot(2, 3, 1);
imshow(image);
title('Original Image');

subplot(2, 3, 2);
imshow(openingResult1);
title('Opening');
```

```
subplot(2, 3, 3);
imshow(topHatResult1);
title('Top-Hat');

subplot(2, 3, 4);
imshow(erosionResult2);
title('Eroded Image');

subplot(2, 3, 5);
imshow(reconstructedResult2);
title('Reconstructed Image');

subplot(2, 3, 6);
imshow(topHatResult2);
title('Top-Hat by Reconstruction');
```

Result of the code: