**Project Report**

**Assignment 1**

**Group Members:**

1. Pratham Ramkripal Yadav (1017895120)
2. Jenil Sanjaykumar Pandya (1017921730)
3. Anant Chanchad (1017891080)
4. David Geis
5. Nikesh Chithambaram

<https://youtu.be/x-qr_keKlsU> Intro Video Link

<https://youtu.be/4GCZDqngQhU> Presentation Video Link

**Analysis of Problem 1:**

Create a short video (2 minutes) to introduce group members by covering the following points:

Names

Year (e.g., freshman, sophomore, etc.)

Individual strengths

Future aspirations

How the group works together as a team

The video will be uploaded to YouTube, and the link will be shared for submission.

Bonus points are awarded for creativity in the introduction.

**Analysis of Problem 2:**

1. **Reading the Image** : The `imread` function is used to read the image file 'falcon.jpg'.

2. **Applying Gaussian Filter:** The `fspecial` function is used to create a Gaussian kernel. The `imfilter` function applies this kernel to the grayscale image.

3. **Image Sharpening:** The sharpened image is obtained by doubling the grayscale image and subtracting the Gaussian blurred image.

4. **Displaying the Image:**The `imshow` function is used to display the final processed image.

**Issues During Implementation:**

Blurring the grayscale image using a Gaussian filter to remove noise.

Enhancing the edges of the image by subtracting the blurred image from the original.

The sharpened image was not displaying correctly due to incorrect data type conversion.

Ensuring the processed image data was in the correct format for display was necessary.

Visualizing the sharpened image.

**Solutions to Overcome the Issues:**

Finding the Perfect Image that if Applied the Sharpened Filter would show some effect

Verify the processed image data type before displaying. Ensuring the data remains in the correct format (e.g., uint8 for grayscale images) solved the display issue.

**Analysis for Problem 3:**

1. **Reading the Image:**

Problem: Importing the image file into the MATLAB environment.

Analysis: The `imread` function is used to read the image file 'singapore.jpg'. This function loads the image data into the MATLAB workspace.

2. **Displaying the Original Image:**

Problem: Visualizing the original image.

Analysis: The `imshow` function is used to display the original image in a figure window named 'Original Image'.

3. **Converting to Grayscale:**

Problem: Converting the color image to a grayscale image if it's not already in grayscale.

Analysis: The `rgb2gray` function is used to convert the image to grayscale if the image has more than one color channel (depth > 1). If the image is already in grayscale, it is used as is.

4. **Applying Min Filter:**

Problem: Applying a minimum filter to the grayscale image.

Analysis: A custom function `filterImage` is used with an arbitrary filter size (7 was used in this example) and a filter type of “min” to apply the minimum filter. The function iterates through the image, determining the minimum value in each pixel’s neighborhood based on the filter size.

5. **Applying Max Filter:**

Problem: Applying a maximum filter to the grayscale image.

Analysis: The `filterImage` function is again used, this time with a filter size of 5 and a filter type of “max”, to apply the maximum filter. This process is similar to the minimum filter but looks for the maximum value in the neighboring region.

6. **Applying Median Filter:**

Problem: Applying a median filter to the grayscale image.

Analysis: The `filterImage` function is used with a filter size of 3 and a filter type of “median” to apply the median filter. This function finds the median value in the neighboring region for each pixel.

**Issues:**

1. **Filter Size Error:**

Issue: Ensuring the filter size is an odd number.

Analysis: The function checks if the filter size is odd. If not, it displays an error message and aborts the filter application.

2. **Non-Grayscale Image Error:**

Issue: Ensuring the image is in grayscale before applying the filter.

Analysis: The function checks if the image is grayscale by verifying that the depth = 1. If the image is not grayscale, it aborts the filter application with an error message.

3. **Image Display:**

Issue: Properly displaying the filtered images.

Analysis: Ensuring the processed images are in the correct format (e.g., uint8) before displaying them using the `imshow` function.

4. **Filter Type:**

Issue: Specify a valid filter type to the filterImage function.

Analysis: Ensuring the filter type specified is “min”, “max”, or “median”. If the filter type is anything else it displays an error message and aborts the filter application.

**Solutions:**

1. **Filter Size Error:**

Solution: Implemented a check to ensure the filter size is an odd number before applying the filter. Displayed an appropriate error message if the filter size is even.

2. **Non-Grayscale Image Error:**

Solution: Added a check to confirm the image is grayscale before applying the filter. Displayed an appropriate error message if the image has more than one color channel.

3. **Image Display:**

Solution: Verified that the processed image data remains in the correct format for display. Ensured that the images are correctly visualized using the `imshow` function.

4. **Filter Type:**

Solution: Verified that the filter type passed to the function is valid. Displayed an appropriate error message if the filter type is not valid.

**Contribution of Each Individual Member:**

**Pratham Ramkripal Yadav**:

1. Implemented code for reading images, converting them to grayscale, and experimented with various Gaussian filter parameters.
2. Experimented with different kernel sizes and standard deviations for the Gaussian filter.
3. Drafted the report for Problem 2.

**Jenil Sanjaykumar Pandya**:

1. Assisted in debugging the file not found issue, provided the correct path to the image file, and ensured the processed image data was in the correct format for display.
2. Worked on applying the Gaussian filter and verified the final image display, ensuring the correct visualization of the sharpened image.
3. Edited the entire intro video and presentation video, as well as the report and presentation.

**David Geis**

1. Implemented the code for reading the image and converting it to grayscale, and wrote the filterImage function to apply min, max, and median filters.
2. Assisted in debugging the filter size and non-grayscale image errors, and verified the correctness of the implemented filters and their outputs.
3. Worked on code optimization and ensuring the efficiency of the filter application, and assisted in the final review of the report and presentation of the results.

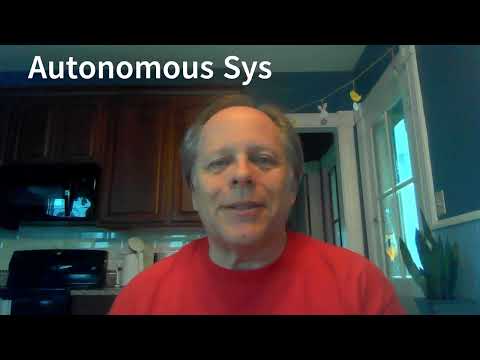
**Nikesh Chitambaram**

1. Assisted in debugging the file not found issue, provided the correct path to the image file, and ensured the processed image data was in the correct format for display.
2. Worked on applying the Gaussian filter and verified the final image display, ensuring the correct visualization of the sharpened image.
3. Assisted in documentation and writing the report, and edited the entire intro video and presentation video, as well as the report and presentation.

**Anant Chanchad**

1. Helped with the initial setup and importing the image, and implemented the code for reading the image and converting it to grayscale.
2. Assisted in testing the different filter types, wrote the filterImage function to apply min, max, and median filters, and verified their outputs.
3. Came up with the speech for the intro video, worked on code optimization, ensuring the efficiency of the filter application, and assisted in the final review of the report and presentation of the results.

INTRODUCTION VIDEO

[](https://www.youtube.com/embed/x-qr_keKlsU?feature=oembed)

PRESENTATION VIDEO

[](https://www.youtube.com/embed/4GCZDqngQhU?feature=oembed)