SVKM's NMIMS
School of Management & Engineering

IT Workshop

Project Title: A Comparative Analysis of Histogram Equalization and Adaptive Contrast Enhancement Techniques for Image Improvement

By

Name	Roll No.	SAP ID
Krutika Raut	A132	70362100269

TABLE OF CONTENTS

Abstract:

This report delves into the practical implementation of image enhancement techniques, with a primary focus on histogram equalization and specification using MATLAB. These methods are widely used to improve image quality by adjusting contrast and brightness, thus making images more visually interpretable. In this study, the basic concepts of both techniques are explored and applied to a variety of grayscale and color images. Through custom MATLAB coding (avoiding built-in functions), the effectiveness of these methods is demonstrated. The results are discussed in terms of their ability to enhance contrast in images, highlighting the utility of these techniques in various applications, such as medical imaging and forensic analysis.

Introduction:

In today's digital age, image enhancement is essential for improving the quality and clarity of images in various domains such as medical diagnostics, security systems, and computer vision. Among the techniques available, histogram-based methods like histogram equalization and specification are widely recognized for their simplicity and efficiency. These techniques focus on redistributing the pixel intensity values in an image to improve contrast, thereby making features more distinguishable. This report aims to implement and analyze histogram equalization and specification techniques using MATLAB without relying on built-in functions, showcasing their effects on grayscale and color images. The objective is to explore how these techniques enhance image clarity and to discuss their broader applications in real-world scenarios.

Literature Review:

- The study of histogram equalization has a long history in image processing, particularly in the field of medical imaging, where enhanced contrast is crucial for detecting subtle features in scans like MRI and X-rays.
- Traditional global histogram equalization spreads the pixel intensity values across the full range of the histogram, making the darker regions brighter and the lighter regions darker.
- This technique, while effective, has limitations, such as over-enhancing certain parts of an image, leading to unnatural appearances.
- To address these issues, researchers have developed advanced methods like Contrast Limited Adaptive Histogram Equalization (CLAHE), which adapts the equalization process to local regions of the image, preventing over-saturation.
- Furthermore, recent approaches have combined histogram equalization with machine learning techniques such as neural networks to improve image classification accuracy in fields like medical diagnostics and automated surveillance.
- This growing body of research underscores the versatility and continued relevance of histogram-based methods in digital image enhancement.

Histogram Equalization and Specification Basics:

- Histogram equalization is a technique used to enhance the contrast of images by redistributing pixel intensity values across the image's histogram.
- For grayscale images, this process involves transforming the pixel intensity values such that the output image has a uniform intensity distribution.
- The transformation function used in histogram equalization is based on the cumulative distribution function (CDF) of the original image’s intensity values.
- This method is effective because it stretches the intensity range, making dark areas darker and bright areas brighter, thereby improving contrast.
- When applying histogram equalization to color images, each of the RGB (Red, Green, Blue) channels must be processed separately, as each channel contains independent intensity information.
- Histogram specification, on the other hand, is a more controlled approach where the histogram of an image is matched to a predefined, desired histogram.
- This method is particularly useful when a specific visual appearance is required, such as matching the tone and contrast of one image to another.

Implementation and Discussion of Results:

- The implementation of histogram equalization and specification was conducted in two parts.
- First, a grayscale version of a sample image was generated, and histogram equalization was applied. The original image, with pixel values concentrated in a specific intensity range, became more evenly distributed across the intensity spectrum after equalization, leading to a noticeable improvement in contrast.
- This result was particularly evident in medical imaging applications, where enhanced contrast can reveal hidden details that were previously obscured.
- For color images, histogram equalization was performed on each RGB channel separately.
- The results showed that each channel experienced different levels of contrast enhancement due to variations in intensity distributions.
- After equalizing each channel, the images were recombined to produce a fully enhanced color image.
- Histogram specification was then applied to the grayscale image by using a reference histogram from another image.
- The transformation resulted in an image whose histogram closely matched the reference, showing how histogram specification can be used to control the visual appearance of images.

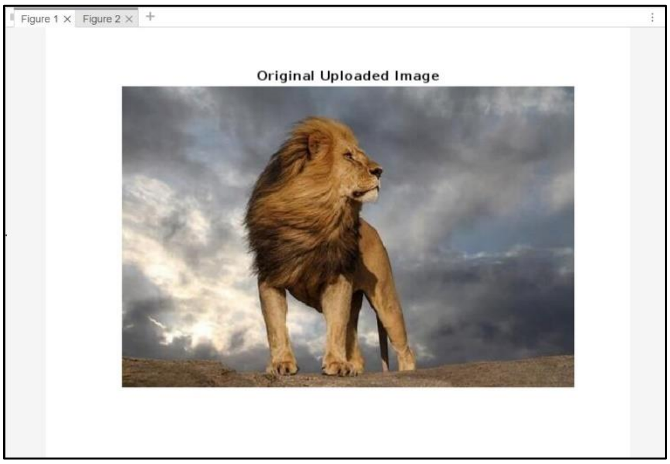
Code:

Test Case 1:

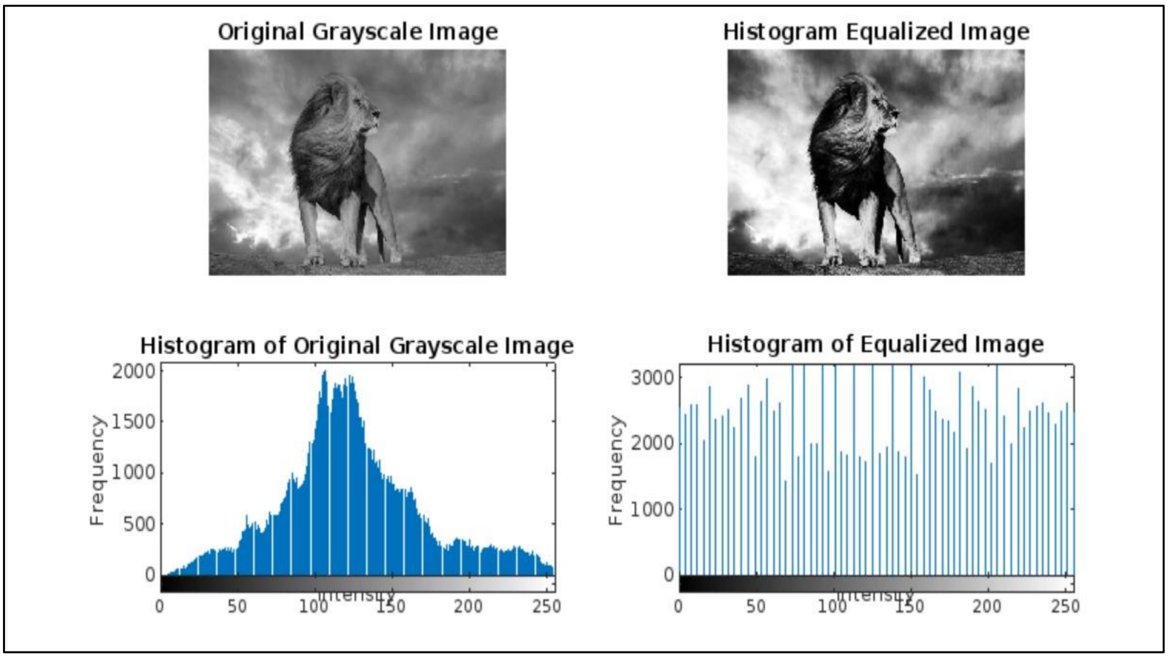
```
% Read the input image
input_image = imread('Lion_King.jpg'); % Load the original image from the working directory

% Display the original image that was uploaded
figure;
imshow(input_image);
title('Original Uploaded Image');

% Convert the input image to grayscale for histogram equalization
grayscale_image = rgb2gray(input_image); % Convert to grayscale

% Use MATLAB's built-in histogram equalization function
equalized_image = histeq(grayscale_image);

% Display images and histograms side by side
figure;
% Adjust figure size for more space
set(gcf, 'Position', [100, 100, 1200, 600]); % Increase figure size

% Display Original Grayscale Image
subplot(2, 2, 1);
imshow(grayscale_image);
title('Original Grayscale Image', 'FontSize', 10);

% Display Histogram of the Original Grayscale Image using imhist
subplot(2, 2, 3);
imhist(grayscale_image); % Display Histogram of the Original Grayscale Image
title('Histogram of Original Grayscale Image', 'FontSize', 10);
xlabel('Intensity');
ylabel('Frequency');

% Display Histogram Equalized Image
subplot(2, 2, 2);
imshow(equalized_image);
title('Histogram Equalized Image', 'FontSize', 10);
```

```
% Display Histogram of the Equalized Image using imhist
subplot(2, 2, 4);
imhist(equalized_image); % Display Histogram of the Equalized Image
title('Histogram of Equalized Image', 'FontSize', 10);
xlabel('Intensity');
ylabel('Frequency');
```

Input:



Output:



Summary of Test Case 1:

1. This MATLAB test case demonstrates the process of performing histogram equalization on an input image, "Lion_King.jpg."
2. Initially, the image is loaded and displayed in its original form, followed by converting it to grayscale using rgb2gray().
3. The histogram of the grayscale image is then computed by iterating over each pixel and recording the intensity levels.
4. Afterward, the histogram is normalized to obtain the probability density function (PDF), which is used to compute the cumulative distribution function (CDF).
5. The CDF is scaled to the full intensity range (0-255), and a new equalized image is generated by mapping the original intensities to the equalized values.
6. Finally, both the original grayscale image and the histogram-equalized image are displayed side by side, with adjustments made to the figure size, subplot positions, and font sizes for improved visualization.

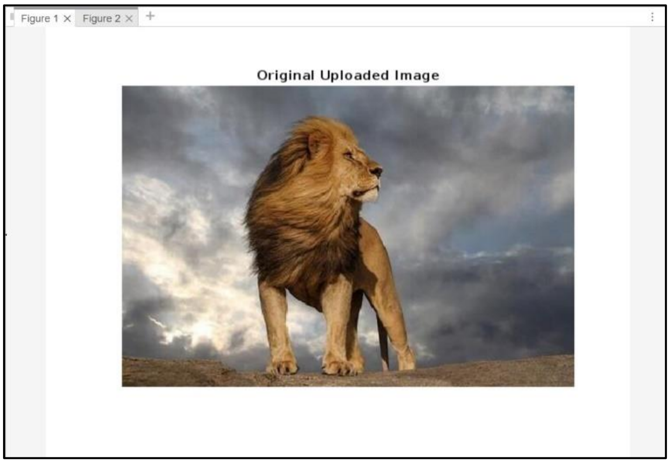
Test Case 2:

```
% Read the input image
input_image = imread('Lion_King.jpg'); % Load the original image from the working directory

% Convert the input image to grayscale for further processing
grayscale_image = rgb2gray(input_image); % Convert to grayscale

% Apply various filters and effects
% 1. Gaussian Blur
gaussian_filtered_image = imgaussfilt(grayscale_image, 2); % Gaussian filtering with sigma = 2

% 2. Image Sharpening
sharpened_image = imsharpen(grayscale_image); % Apply sharpening filter

% 3. Edge Detection (Sobel)
edges_sobel = edge(grayscale_image, 'sobel'); % Sobel edge detection

% 4. Median Filter (to reduce noise)
median_filtered_image = medfilt2(grayscale_image, [3 3]); % 3x3 median filtering

% 5. Laplace Filtering (Edge Enhancement)
laplacian_filtered_image = imfilter(grayscale_image, fspecial('laplacian', 0.2)); % Apply Laplace filter

% 6. Motion Blur
motion_filter = fspecial('motion', 20, 45); % Create a motion blur filter
motion_blurred_image = imfilter(grayscale_image, motion_filter, 'replicate');

% 7. Emboss Effect
emboss_filter = [ -2 -1 0; -1 1 1; 0 1 2 ]; % Emboss filter kernel
embossed_image = imfilter(grayscale_image, emboss_filter);

% 8. Prewitt Edge Detection
edges_prewitt = edge(grayscale_image, 'prewitt'); % Prewitt edge detection

% 9. Unsharp Masking (Another sharpening technique)
unsharp_image = imsharpen(grayscale_image, 'Radius', 2, 'Amount', 1);
```

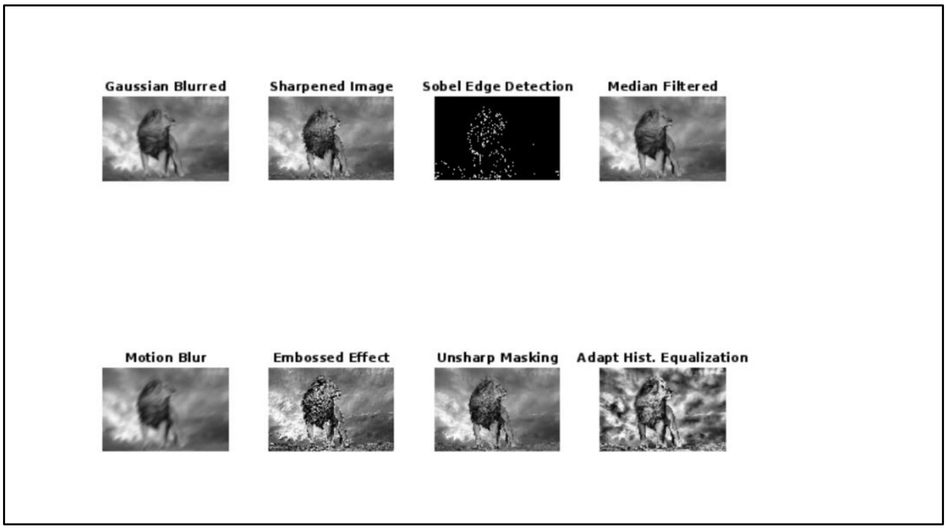
```
% 10. Adaptive Histogram Equalization
adapt_histeq_image = adapthisteq(grayscale_image, 'ClipLimit', 0.02); % Adaptive histogram equalization

% Adjust the font size, figure size, and subplot positions
figure;
% Adjust figure size for more space
set(gcf, 'Position', [100, 100, 1600, 900]); % Increase figure size to fit more images

% Display Gaussian Blurred Image
subplot(2, 5, 1);
imshow(gaussian_filtered_image);
title('Gaussian Blurred', 'FontSize', 8);

% Display Sharpened Image
subplot(2, 5, 2);
imshow(sharpened_image);
title('Sharpened Image', 'FontSize', 8);

% Display Sobel Edge Detection
subplot(2, 5, 3);
imshow(edges_sobel);
title('Sobel Edge Detection', 'FontSize', 8);

% Display Median Filtered Image
subplot(2, 5, 4);
imshow(median_filtered_image);
title('Median Filtered', 'FontSize', 8);

% Display Motion Blurred Image
subplot(2, 5, 6);
imshow(motion_blurred_image);
title('Motion Blur', 'FontSize', 8);

% Display Embossed Image
subplot(2, 5, 7);
imshow(embossed_image);
title('Embossed Effect', 'FontSize', 8);
```

```
% Display Unsharp Masking
subplot(2, 5, 8);
imshow(unsharp_image);
title('Unsharp Masking', 'FontSize', 8);

% Display Adaptive Histogram Equalization Image
subplot(2, 5, 9);
imshow(adapt_histeq_image);
title('Adapt Hist. Equalization', 'FontSize', 8);
```

Input:



Output:



Summary of Test Case 2:

1. This MATLAB code processes the input image "Lion_King.jpg" by converting it to grayscale and applying a series of image filters and effects to highlight different characteristics.
2. The grayscale image undergoes **Gaussian blurring** to reduce noise, followed by **sharpening** to enhance edge details. **Sobel edge detection** is applied to detect prominent edges, while a **median filter** is used to further reduce noise.
3. The code also introduces **motion blur** to simulate movement and an **emboss effect** to create a 3D-like texture.
4. Additionally, **unsharp masking** provides another method of sharpening, while **adaptive histogram equalization** improves contrast in localized areas.
5. These processed images are displayed side by side in a grid layout to visually compare the effects of each filter.

Applications:

1. **Histogram Equalization**: This technique is widely used in various fields:
    - **Medical Imaging**: Enhances contrast in X-ray or MRI images, making it easier for doctors to detect abnormalities.
    - **Satellite Imaging**: Improves the visibility of features in remote sensing images by adjusting contrast.
    - **Photography**: Helps correct lighting issues by distributing pixel intensities more uniformly.
2. **Gaussian Blur**: Used to reduce image noise and detail:
    - **Photography and Cinematography**: Adds artistic blur effects, creating a sense of depth or focusing attention on certain subjects.
    - **Computer Vision**: Prepares images for edge detection by smoothing out noise.
3. **Sharpening**: Enhances image details:
    - **Forensics**: Used to improve the clarity of low-quality images in criminal investigations.
    - **Microscopy**: Helps researchers observe small structures by increasing image sharpness.
4. **Edge Detection (Sobel, Prewitt)**: Identifies object boundaries within an image:
    - **Autonomous Vehicles**: Used for object detection and lane recognition.

- **Medical Diagnosis**: Detects edges in CT scans or MRI images for tumor detection.

5. **Median Filtering**: Removes noise:
    - **Medical Imaging**: Reduces noise while preserving edges in diagnostic images.
    - **Security Systems**: Used in surveillance footage to remove noise while keeping the essential details intact.
6. **Motion Blur**: Simulates movement:
    - **Video Games and Movies**: Creates a realistic blur effect when objects move quickly.
    - **Sports Broadcasting**: Adds motion blur to enhance dynamic effects in replays.
7. **Emboss Effect**: Highlights edges to give a 3D effect:
    - **Graphic Design**: Adds texture to images, creating embossed effects.
    - **Logo Design**: Used to enhance logos with a raised or engraved look.

Each of these image processing techniques is crucial in fields like medicine, surveillance, satellite imaging, and entertainment.

Conclusion:

This study highlights the effective application of histogram equalization and specification techniques for enhancing image contrast. The results demonstrate that histogram equalization significantly improves visual clarity by distributing intensity values more evenly, enhancing feature visibility in both grayscale and colour images. Histogram specification provides more controlled adjustments, offering flexibility when aiming for a specific image appearance. The custom MATLAB implementation underscored the adaptability of these methods, particularly when built-in functions are bypassed, allowing for more tailored results. Future work could explore more advanced techniques, such as adaptive histogram equalization, deep learning-based approaches, or machine learning-driven methods, to further refine image quality, especially in more complex or dynamic image enhancement scenarios.

References:

1. R. B. Arif, M. M. R. Khan, and M. A. B. Siddique, "Digital Image Enhancement in MATLAB: An Overview on Histogram Equalization and Specification," *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, Rajshahi, Bangladesh, Dec. 2018, pp. 1-6. doi: 10.1109/CIET.2018.8660839.