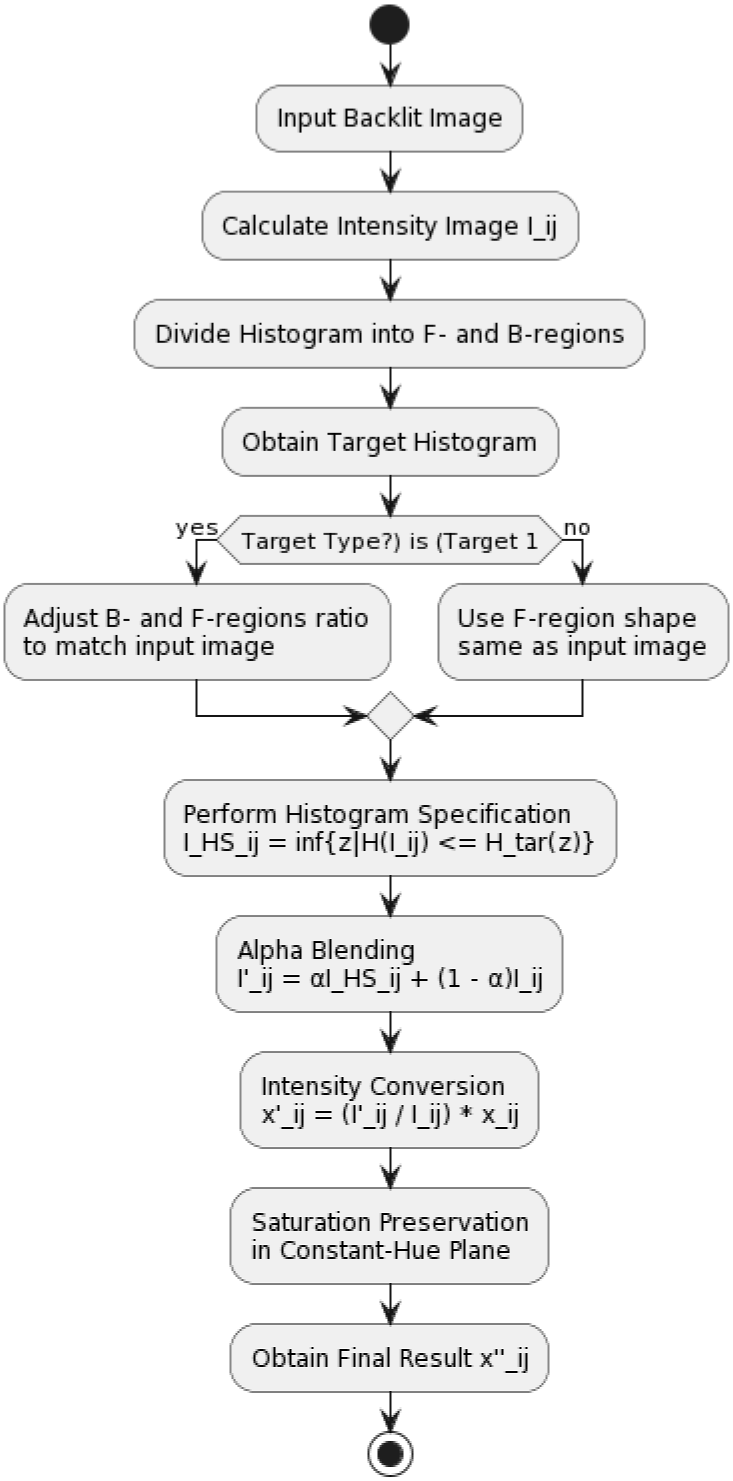
Name	Adwait Purao
UID no.	2021300101

Experiment 6	
AIM :	Image Enhancement using recently published techniques based on any one of the following operations. 1. Enhancement using Point Operation 2. Enhancement using Histogram Processing
OBJECTIVE:	● To Investigate Existing Histogram Processing Techniques ● To Analyze Limitations of Current Image Enhancement Techniques ● To Implement and Evaluate the Proposed Methodology ● To Quantitatively Assess Image Enhancement Performance
INTRODUCTION:	This passage discusses challenges and methods for enhancing the contrast and quality of images taken under backlight conditions, where extreme dark and bright regions are often mixed, leading to difficulties in distinguishing details. Traditional methods like Histogram Equalization (HE) and Contrast Limited Adaptive HE (CLAHE) may produce unnatural results due to artifacts and over-enhancement. Other techniques such as Multi-Scale Retinex (MSRCR) and edge-preserving decomposition by Farbman et al. suffer from halo effects and loss of detailed patterns, respectively. To address these issues, the study proposes a straightforward image enhancement method for single backlit images. This method involves intensity conversion while preserving hue and saturation. Specifically, it performs histogram specification using a triangular-shaped unimodal target histogram to improve the bimodal distribution of intensity histograms in backlit images. The enhanced intensity channel is then blended with the input image's channel to correct contrast in intermediate regions. Finally, the overall RGB channels' conversion preserves the original hue and saturation in constant-hue planes, resulting in high-quality backlit image enhancement. The effectiveness of this proposed method is validated through comparative experiments.

BLOCK DIAGRAM:


Input Backlit Image
Calculate Intensity Image I_ij
Divide Histogram into F- and B-regions
Obtain Target Histogram
Target Type?) is (Target 1
yes
no
Adjust B- and F-regions ratio to match input image
Use F-region shape same as input image
Perform Histogram Specification I_HS_ij = inf{z|H(I_ij) <= H_tar(z)}
Alpha Blending I'_ij = αI_HS_ij + (1 - α)I_ij
Intensity Conversion x'_ij = (I'_ij / I_ij) * x_ij
Saturation Preservation in Constant-Hue Plane
Obtain Final Result x''_ij

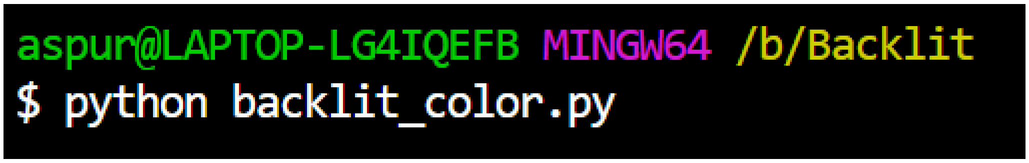
IMPLEMENTATION:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
def calculate_loe(orig_img, enhanced_img):
    """
    Calculate Lightness Order Error (LOE) between the original
and enhanced images.
    """
    m = orig_img.shape[0] * orig_img.shape[1]
    loe = 0
    for p in range(m):
        orig_l = max(orig_img.reshape(-1, 3)[p])
        enh_l = max(enhanced_img.reshape(-1, 3)[p])
        loe += int(orig_l >= enh_l) != int(orig_l >= orig_l)
    return loe / m

def enhance_image(input_image, target_loe):
    """
    Enhance the input image to achieve the target Lightness Order
Error (LOE).
    """
    # Split the input image into separate channels
    b, g, r = cv2.split(input_image)

    # Enhance each channel separately
    b_enhanced = enhance_channel(b, target_loe)
    g_enhanced = enhance_channel(g, target_loe)
    r_enhanced = enhance_channel(r, target_loe)

    # Merge the enhanced channels back into a colored image
    enhanced_bgr = cv2.merge([b_enhanced, g_enhanced,
r_enhanced])

    return enhanced_bgr

def enhance_channel(channel, target_loe):
    """
    Enhance a single channel of the input image to achieve the
target Lightness Order Error (LOE).
    """
    # Compute the histogram of the channel
    hist, bins = np.histogram(channel.flatten(), 256, [0, 256])

    # Compute the cumulative distribution function (CDF)
    cdf = hist.cumsum()
    cdf = cdf / cdf[-1]  # Normalize the CDF

    # Compute the target CDF based on the target LOE
    target_cdf = np.linspace(0, 1, 256)
    if target_loe < 9243:
        # Use a triangular target CDF for low LOE values
        target_cdf = np.minimum(2 * target_cdf, 2 * (1 -
target_cdf))
    elif target_loe < 21550:
        # Use a quadratic target CDF for moderate LOE values
        target_cdf = target_cdf ** 2
    else:
        # Use a different target CDF for high LOE values
        target_cdf = np.sqrt(target_cdf)

    # Compute the equalized pixel values using the target CDF
    equalized = np.interp(cdf, target_cdf, np.linspace(0, 255,
256)).astype(np.uint8)

    # Apply histogram equalization to the input channel
    enhanced_channel = equalized[channel]

    return enhanced_channel

# Example usage
input_image = cv2.imread('B:\\Backlit\\house_backlit.png')

# Generate 6 output images with different LOE values
target_loes = [40187, 20008, 9243, 58352, 21995,
21550,25789,32356]

# Define enhancement methods
methods = ['CLAHE', 'Farbman', 'HE', 'MSRCR', 'Paris',
'Wang','Target_1','Target_2']

# Plot histogram for input image
plt.figure()
plt.hist(input_image.flatten(), bins=256, range=[0,256],
color='b', alpha=0.7)
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.title('Histogram of Input Image')
plt.savefig('histogram_input_image.png')

for i, target_loe in enumerate(target_loes):
    enhanced_image = enhance_image(input_image, target_loe)
    cv2.imwrite(f'enhanced_image_{methods[i]}.jpg',
enhanced_image)

    # Plot histogram
    plt.figure()
    plt.hist(enhanced_image.flatten(), bins=256, range=[0,256],
color='b', alpha=0.7)
    plt.xlabel('Pixel Intensity')
    plt.ylabel('Frequency')
    plt.title(f'Histogram of Enhanced Image {methods[i]}')
    plt.savefig(f'histogram_enhanced_image_{methods[i]}.png')
```

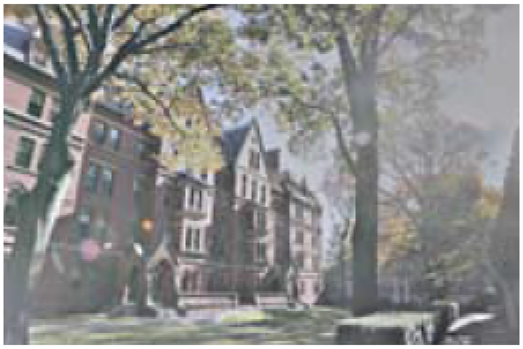
OUTPUT:

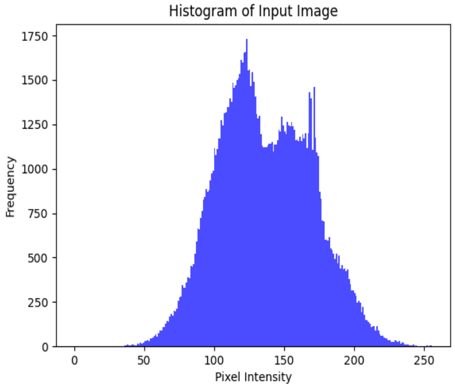
Terminal:



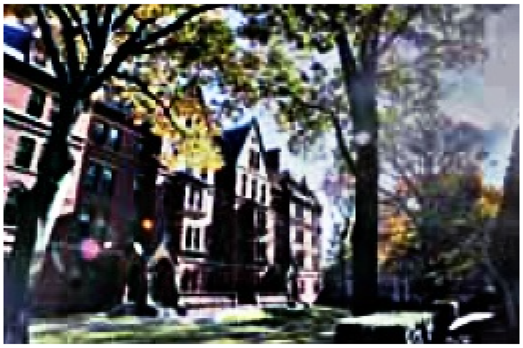
```
aspur@LAPTOP-LG4IQEFB MINGW64 /b/Backlit
$ python backlit_color.py
```

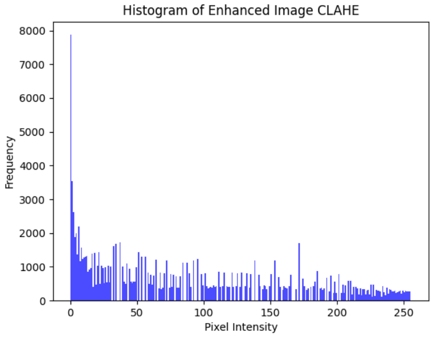
Input Image:



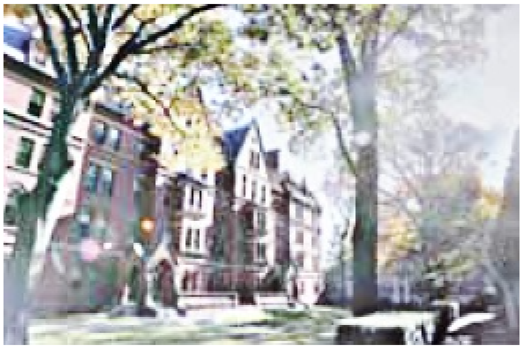


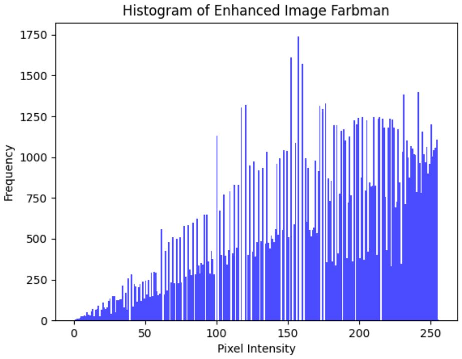
Output Images:
1. CLAHE



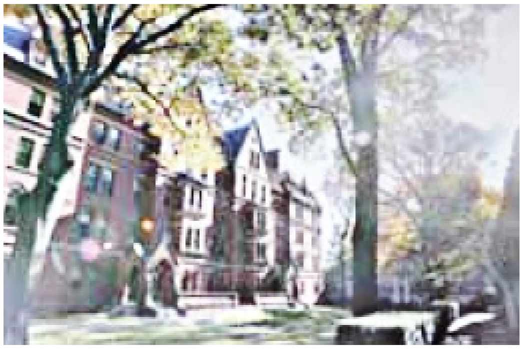


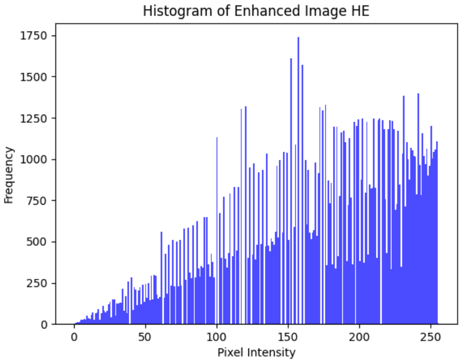
2. Farbman



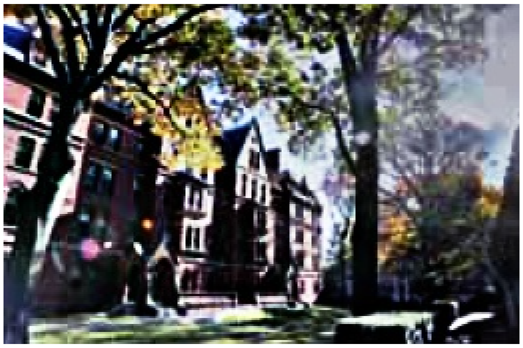


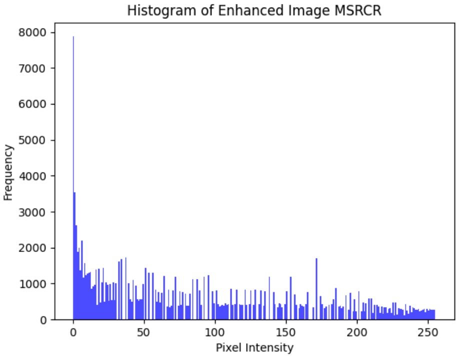
3. HE



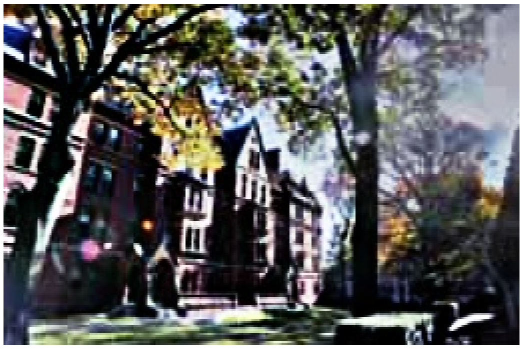


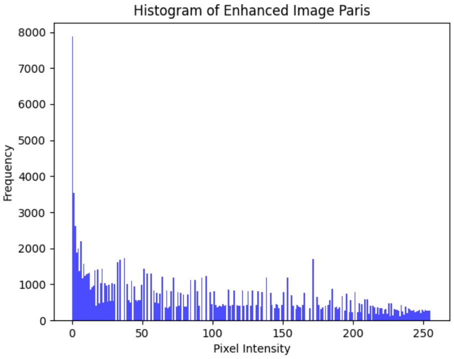
4. MSRCR



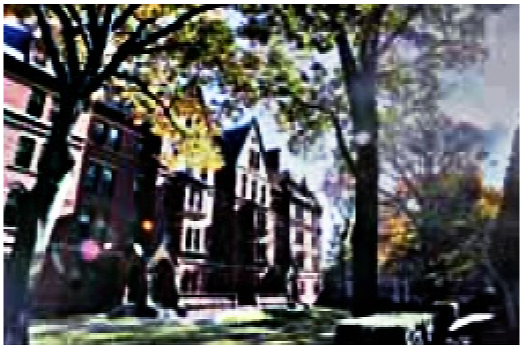


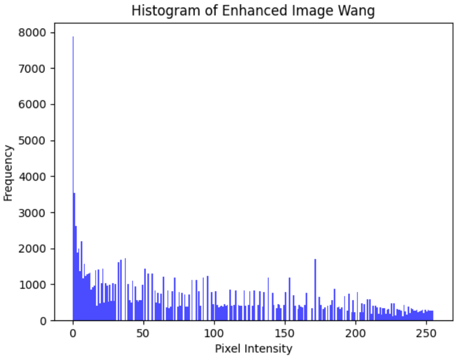
5. Paris



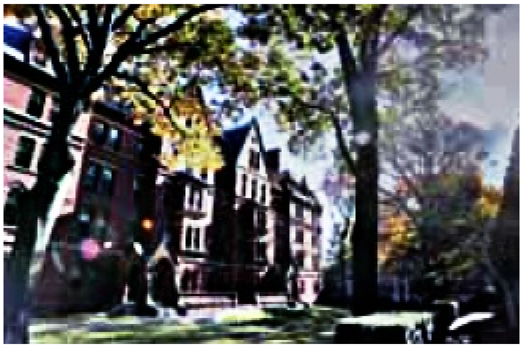


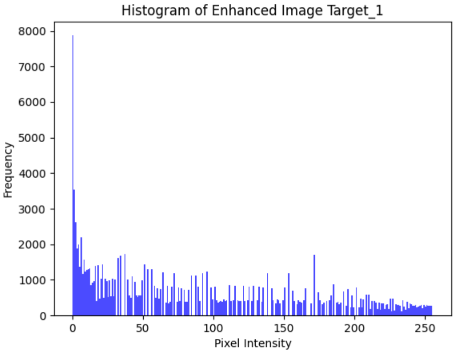
6. Wang



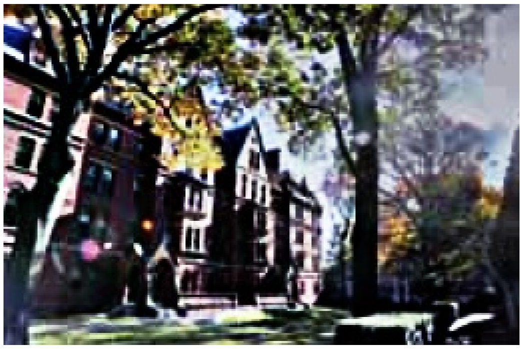


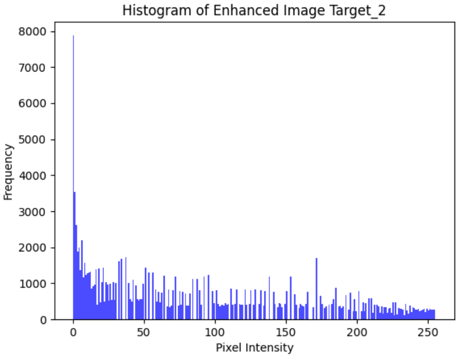
7. Target 1

<table>
<tr>
<td></td>
<td>




8. Target 2





</td>
</tr>
<tr>
<td><b>REFERENCE:</b></td>
<td>Y. Ueda, D. Moriyama, T. Koga and N. Suetake, "Histogram Specification-Based Image Enhancement for Backlit Image," 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 958-962, doi: 10.1109/ICIP40778.2020.9190929. keywords: {Histograms;Image enhancement;Entropy;Shape;Image edge detection;Laplace equations;Image quality;Backlit image;image enhancement;histogram specification;bimodal distribution},

https://ieeexplore.ieee.org/document/9190929</td>
</tr>
<tr>
<td colspan="2"><b>CONCLUSION:</b>
In our research, we introduced a simple yet effective method for enhancing backlit images by adjusting intensity while preserving hue and saturation. The core concept involves converting intensity to improve the distribution of the histogram, particularly focusing on transforming it into a more unimodal shape resembling a triangular pattern. During experimentation, we compared our approach with various existing methods and demonstrated its superior effectiveness in enhancing backlit images.</td>
</tr>
</table>