| **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:** |  |
| **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:**    **Input Image:**    **Output Images:**  **1. CLAHE**    **2. Farbman**    **3. HE**    **4. MSRCR**    **5. Paris**    **6. Wang**    **7. Target 1**    **8. Target 2** |
| **REFERENCE:** | 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> |
| **CONCLUSION:**  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. | |