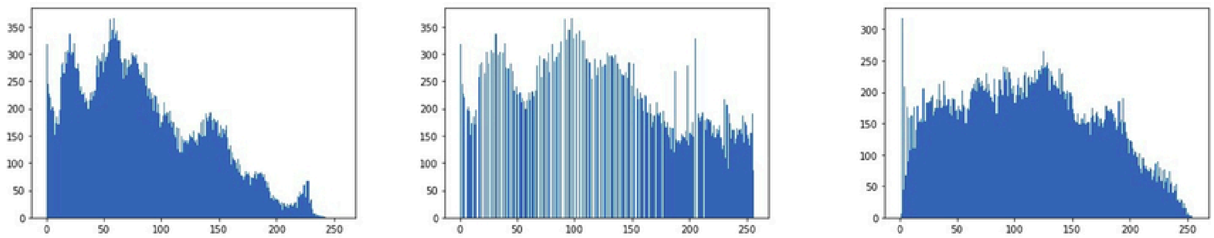
Table of contents

Lecture 9

(i) (ii) (iii)

(a) Image enhancement using Histogram Equalization and CLAHE



(i) (ii) (iii)

Topics:

1. White Balance
2. Histogram
3. Histogram Equalization
4. Morphological Transformation

White Balance

British Academy of Photography

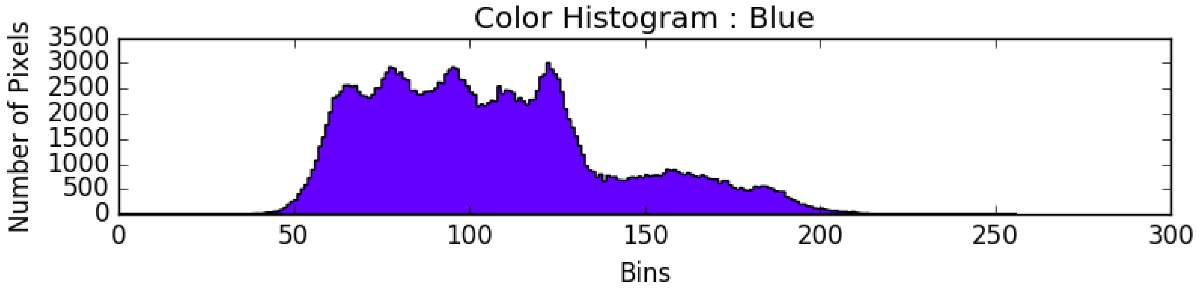
The goal of white balance is to ensure that the colors in an image appear natural and neutral, particularly under different lighting conditions. Ideally, a white object should look white in the image, regardless of the light source.

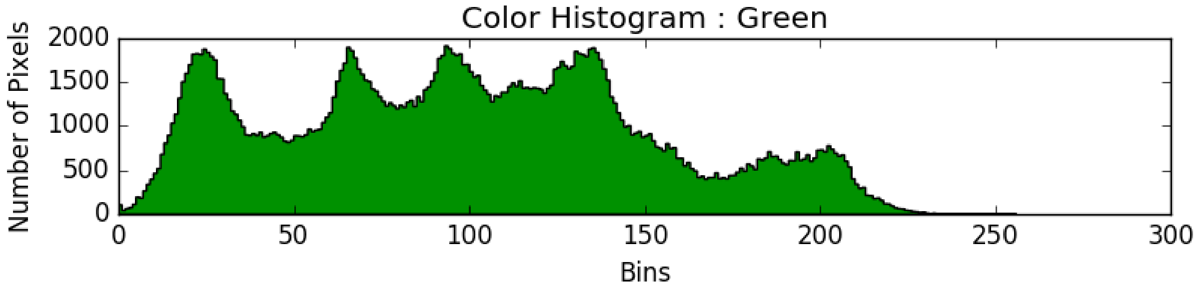
White balance control is vital in image processing for computer vision because it ensures color accuracy, consistency, and improved performance of algorithms. By correcting for color casts introduced by different lighting conditions, we can enhance the quality of images and facilitate more effective analysis and interpretation in various applications.

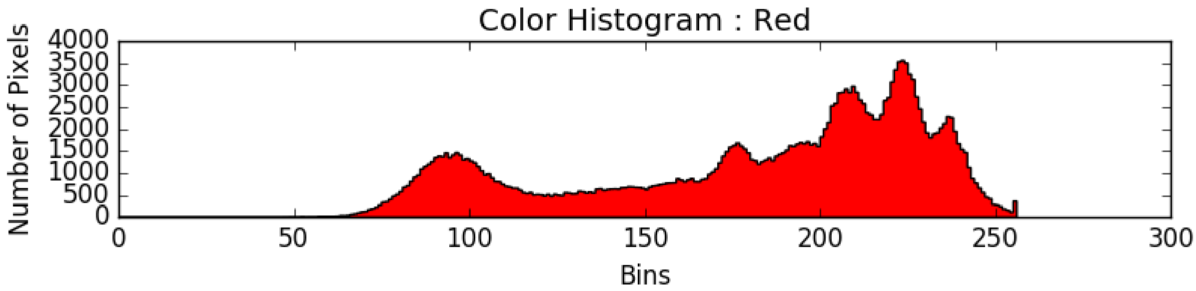
Steps:

1. **Set a scale factor**
2. **Get average for each channel**
3. **Divide the scale factor by the average value and get the scale value for each channel**
4. **Multiply each channel with the scale value**

What is Histogram?







Histogram: A visual representation of the distribution of quantitative data.

In image processing, **Histogram equalization** is a technique used to enhance the contrast of an image by effectively redistributing the intensity values (pixel values) across the entire range of possible values.

How Histogram Equalization Works?

1. **Histogram Calculation:** The histogram shows how many pixels fall into each intensity level (e.g., from 0 to 255 for an 8-bit image).
2. **Cumulative Distribution Function (CDF):** The CDF at a given intensity level indicates the cumulative number of pixels with intensity values less than or equal to that level. This helps in understanding how pixel intensities are distributed across the image.
3. **Normalization:** The CDF is then normalized to map the pixel values to the full range of intensity values (0 to 255). This normalization ensures that the output image will utilize the entire range of possible intensity values.
4. **Mapping Intensity Values:** Each pixel in the original image is then mapped to a new intensity value based on the normalized CDF. This mapping redistributes the pixel values, enhancing the contrast of the image.

Contrast is the difference in luminance or color that makes an object (or its representation in an image or display) visible against a background of different luminance or color.

Histogram Equalization Techniques:

The most commonly used histogram equalization techniques in image processing and computer vision are:

1. **Global Histogram Equalization (GHE):**
    - This is the basic form of histogram equalization, where the entire image histogram is transformed to achieve a uniform distribution of pixel values.
    - It is effective for improving the overall contrast of the image, but it may not work well for images with non-uniform illumination or local contrast issues.
2. **Adaptive Histogram Equalization (AHE):**
    - AHE divides the image into small, contextual regions and applies histogram equalization to each region independently.
    - This helps to enhance local contrast and preserve more details, especially in areas with non-uniform illumination.
    - However, AHE can also amplify noise in relatively homogeneous regions of the image.
3. **Contrast Limited Adaptive Histogram Equalization (CLAHE):**
    - CLAHE is an extension of AHE that includes a contrast limiting step to prevent amplification of noise.
    - It limits the amplification of contrast by clipping the histogram at a predefined value before applying the equalization.
    - CLAHE is widely used for improving the contrast and enhancing the details in medical images, satellite imagery, and other applications where local contrast **is important.**

Learn more: Bi-Histogram Equalization (BHE), Dualistic Sub-Image Histogram Equalization (DSIHE), Brightness Preserving Bi-Histogram Equalization (BBHE) etc.

These histogram equalization techniques offer different trade-offs between contrast enhancement, noise amplification, and brightness preservation, making them suitable for various image processing and computer vision applications.

Morphological Transformations

Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images. It needs two inputs, one is our original image, second one is called **structuring element** or **kernel** which decides the nature of operation.

1. Erosion

NRDC

The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object (Always try to keep foreground in white). So what it does? The kernel slides through the image (as in 2D convolution). A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded (made to zero).

So what happens?
All the pixels near boundary will be discarded depending upon the size of kernel. So the thickness or size of the foreground object decreases or simply white region decreases in the image. It is useful for removing small white noises (as we have seen in colorspace chapter), detach two connected objects etc.

cv.erode(img, kernel, iterations = 1)

2. Dilation

It is just opposite of erosion. Here, a pixel element is '1' if at least one pixel under the kernel is '1'. So it increases the white region in the image or size of foreground object increases. Normally, in cases like noise removal, erosion is followed by dilation. Because, erosion removes white noises, but it also shrinks our object. So we dilate it. Since noise is gone, they won't come back, but our object area increases. It is also useful in joining broken parts of an object.

cv.dilate(img, kernel, iterations = 1)

3. Opening

Opening is just another name of **erosion followed by dilation**. It is useful in removing noise, as we explained above.

cv.morphologyEx(img, cv.MORPH_OPEN, kernel)

4. Closing

Closing is reverse of Opening, **Dilation followed by Erosion**. It is useful in closing small holes inside the foreground objects, or small black points on the object.

cv.morphologyEx(img, cv.MORPH_CLOSE, kernel)

Learn more: Morphological Gradient, Top Hat, Black Hat …