CS 5330: Pattern Recognition and Computer Vision

Northeastern University

OpenCV Workshop
Lab 5: Histogram

** Contributed by Fall 2024 TAs: Byunghyun Ko, Yihan Wang and Taiwei Cui*

Histogram

1. Introduction to Histograms
2. Types of Histograms
3. Grayscale Histogram Implementation
4. Color Histogram Implementation

Introduction to Histograms

- Histogram: graphical representation that shows the frequency of pixel intensity values in an image
    - Pixel Intensity: ranges from 0 (black) to 255 (white) for grayscale
    - Usage of Histogram: can helps understand how pixels are distributed across an image – this is particularly helpful with image thresholding, contrast adjustment, and object detection

Types of Histograms

- Grayscale Histogram: plots the intensity values of pixels in grayscale image
  - The x-axis represents pixel intensity (0-255) and the y-axis represents the number of pixels with each intensity
  - Usages: image brightness and contrast analysis, thresholding, image equalization, image matching and comparison, object detection, etc.
- Color Histogram: created separately for each of the three color channels – tells you how intensities are distributed for each color
  - Usages: image classification and retrieval, color enhancement and correction, color segmentation dominant color extraction, object recognition

Grayscale Histogram Implementation

- Load an image and convert it to grayscale

Code

```
import cv2
Import numpy as np
# Load the image in grayscale mode
img = cv2.imread('image.jpg', 0)
```

- “0” in *cv2.imread()* tells OpenCV to load the image in grayscale mode

Grayscale Histogram Implementation

- Calculate the Histogram:

```
hist = cv2.calcHist([img], [0], None, [256], [0, 256])
```

- cv2.calcHist(): This OpenCV function computes the histogram of an image.
  - **[img]**: The input image, which is passed as a list because OpenCV expects the image data in this format.
  - **[0]**: The index of the channel for which the histogram is calculated. Since the image is grayscale, there's only one channel (0).
  - **None**: This is where a mask would go if you wanted to calculate the histogram for only a specific region of the image. Here, we use the entire image, so no mask is needed.
  - **[256]**: This represents the number of bins. Since the intensity values of the grayscale image range from 0 to 255, we use 256 bins to represent all possible values.
  - **[0, 256]**: The range of pixel values we want to consider (0 to 255).

Grayscale Histogram Implementation

- Plot the histogram using Matplotlib

Code

```
import matplotlib.pyplot as plt
# Plot the grayscale histogram
plt.plot(hist)
plt.title('Grayscale Histogram')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.show()
```

- plt.plot(hist): This function plots the computed histogram, showing how often each intensity value occurs in the image.The **x-axis** represents the pixel intensity values (from 0 to 255).
- The **y-axis** represents the frequency, i.e., how many times each intensity value appears in the image.

Color Histogram Implementation

- Load an image in color mode

Code

```
import cv2
Import numpy as np
# Load the image in color mode
img = cv2.imread('image.jpg’)
```

- Note the absence of “0” in parameters -> not loading it in grayscale mode

Color Histogram Implementation

Code

```
# Initialize colors for BGR channels
colors = ('b', 'g', 'r')

# Loop through each color channel (Blue, Green, Red)
for i, col in enumerate(colors):
    # Calculate the histogram for each channel
    hist = cv2.calcHist([img], [i], None, [256], [0, 256])

    # Plot the histogram for the current channel
    plt.plot(hist, color=col)

plt.title('Color Histogram')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.show()
```

Color Channels (BGR): OpenCV uses the BGR (Blue, Green, Red) color model by default. We calculate a separate histogram for each channel.
•colors = ('b', 'g', 'r'): This tuple represents the three color channels—Blue, Green, and Red.
•for i, col in enumerate(colors): This loop iterates over each color channel. i is the channel index, and col is the color name ('b', 'g', or 'r').

•cv2.calcHist([img], [i], None, [256], [0, 256]): This computes the histogram for the current channel (i).**[img]**: The image from which the histogram is being computed.
•**[i]**: The index of the color channel (0 for Blue, 1 for Green, 2 for Red).
•**None**: We're not using a mask, so all pixels in the channel are considered.
•**[256]**: The number of bins to represent the pixel intensities (0-255).
•**[0, 256]**: The range of pixel intensities to consider for the histogram (0-255).

•plt.plot(hist, color=col): This command plots the histogram for the current channel. The **color** argument ensures that the histogram is plotted in the respective channel's color.For the Blue channel, it plots the histogram in blue.
•For the Green channel, it plots the histogram in green.
•For the Red channel, it plots the histogram in red.

Summary

- Grayscale Histogram:
  - Only one channel, representing the intensity of light from 0 (black) to 255 (white).
  - Used to understand image brightness, contrast, and for basic operations like thresholding.
- Color Histogram:
  - Three separate histograms, one for each color channel (Blue, Green, Red).
  - Helps analyze the color distribution in an image, used for tasks like object recognition, color enhancement, and image classification.