جامـــــعـة المـــــسـتـقـبـل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قســـم الانظمة الطبية الذكية

Lecture (3 & 4): Electromagnetic Spectrum and Color in Computer Vision

Subject: Computer Vision
Level: Third
Lecturer: Asst. Lecturer Qusai AL-Durrah

1. Introduction

Computer vision systems do not “see” objects the way humans do. A vision system receives **electromagnetic radiation** through a sensor and converts it into numbers. Understanding **what portion of the electromagnetic spectrum** is being captured—and how that radiation becomes **color values**—is essential for designing robust vision systems, especially in medical applications where different modalities capture different physical signals.

In Intelligent Medical Systems, the importance of this topic appears clearly in areas such as:

- **Endoscopy and laparoscopic imaging** (true color is clinically meaningful)
- **Dermatology imaging** (skin tone, redness, pigmentation patterns)
- **Microscopy** (stained slides: H&E and other stains encode medical meaning via color)
- **Thermal/infrared screening** (non-visible spectrum)
- **Multi-spectral imaging** (beyond RGB, used in research and advanced diagnostics)

2. Electromagnetic Spectrum (EM Spectrum)

The electromagnetic spectrum is the full range of electromagnetic radiation, categorized by wavelength (or frequency). Vision sensors (including the human eye) capture only a small part of this spectrum.

2.1 Wavelength and Frequency

- **Wavelength (λ(lamda))**: the distance between peaks of the wave (meters, micrometers, nanometers).
- **Frequency (f)**: how many cycles per second (Hz).
- They are related by:
  $c=\lambda*f$
- where (c) is the speed of light. $c \approx 3\times108$ m/s.

Example (visible light):
If λ=500 nm=500×10^{-9} m, then:

f=c*λ= $(3\times10^{8})/(500\times10^{-9}) = 6\times10^{14}$ Hz

This explains why visible light is a specific band and why sensors tuned to different bands (IR, UV) "see" different information.

2.2 Key regions relevant to computer vision

- **Ultraviolet (UV)**: shorter than visible light
  Useful in some forensic and biomedical applications (fluorescence).
- **Visible light**: roughly **400–700 nm**
  This is what RGB cameras and the human eye primarily use.
- **Infrared (IR)**: longer than visible light
  Useful for thermal imaging and certain medical screening tasks.
- **X-rays**: much shorter wavelengths
  Used for radiography—this is imaging, but not "color" in the RGB sense.

Important note:
A "computer vision image" is not always an RGB photo. Many medical images are produced by sensors that measure non-visible radiation or physical phenomena (X-ray attenuation, magnetic resonance signals, ultrasound echoes). The representation can still be stored as a digital image matrix, but the meaning of intensity differs by modality.

3. From Light to Image: How sensors capture EM radiation

A camera sensor does not capture "color" directly. It measures **intensity** of radiation reaching the sensor. Color arises from how intensity varies across different wavelength bands.

3.1 Sensor response

A sensor has:

- **spectral sensitivity** (how strongly it responds to different wavelengths)

- **exposure and gain settings**
- **noise characteristics**

This is why the same scene can look different across:

- different cameras,
- different lighting,
- different white balance settings.

3.2 Illumination vs reflectance

For visible-light imaging, the pixel value depends on:

- the **light source** (illumination spectrum),
- the **object surface properties** (reflectance),
- the **sensor's spectral sensitivity**.

So, color can change due to lighting even if the object does not change. This is a major source of error in medical imaging pipelines that rely on color cues (e.g., inflammation redness in endoscopy).

4. What is "Color" in Computer Vision?

Color in vision systems is a **numerical representation** of how the sensor responded across wavelength bands. A digital color image is typically stored as **three channels** (for RGB), where each pixel is a 3-value vector.

A color pixel can be written as:
$p(r,c)=[R(r,c),G(r,c),B(r,c)]$

Where:

- (R) is red channel intensity,
- (G) is green channel intensity,
- (B) is blue channel intensity.

Each value is commonly stored as 8-bit (0–255), though medical and scientific imaging often uses higher bit depth (10-bit, 12-bit, 16-bit).

5. Color Models (Color Spaces)

A **color model** (or color space) defines how we represent color numerically. Different spaces are useful for different tasks.

5.1 RGB (Red, Green, Blue)

- The most common representation for cameras and displays.
- Good for storage and display.
- Not always best for analysis because RGB mixes illumination and color information.

In medical use: RGB is common in endoscopy, dermatology, and microscopy images.

5.2 Grayscale and Luminance Conversion (RGB → Gray)

Many vision tasks begin by converting a color image to grayscale:

- simplifies processing,
- emphasizes intensity structure (edges/texture),
- reduces computation.

But grayscale can lose clinically meaningful color information (e.g., inflammation vs normal tissue in endoscopy).

When we convert a color image (RGB) to grayscale, we want the grayscale intensity to match **perceived brightness** as closely as possible. Human vision is **more sensitive to green**, then red, and least sensitive to blue. Therefore grayscale is computed using a **weighted sum** (luminance), not a simple average.

The standard luminance approximation is:

Y=0.299R+0.587G+0.114BY = 0.299R + 0.587G + 0.114BY=0.299R+0.587G+0.114B

A simple average:

Y_{avg}= (R+G+B)/3

treats all channels equally, which does **not** reflect human brightness perception and can distort contrast important for later processing (edges, textures, segmentation).

Example1:

Given a pixel with R=120, G=200, B=80:

1) Luminance grayscale

Y=0.299(120)+0.587(200)+0.114(80)

Compute step by step:

- 0.299×120=35.88
- 0.587×200=117.4
- 0.114×80=9.12

Y=35.88+117.4+9.12=162.4

So, the grayscale luminance is approximately:

Y≈162

2) Simple average grayscale

Y_{avg}= (120+200+80) / 3 = 133.33

So:

Y_{avg}≈133

Explanation of the difference

The average method gives 133, but the luminance method gives 162 because the pixel has a **high green value (200)** and green contributes the most to perceived brightness. The luminance formula weights the channels according to human sensitivity, producing a grayscale value that better matches what humans perceive as "bright."

Example2:
A color pixel has R=60, G=180, B=90.

1. Compute grayscale luminance using:
2. Compute grayscale using the simple average:
3. Which is higher, and why?

5.3 HSV / HSL (Hue, Saturation, Value/Lightness)

- **Hue** corresponds to "color type" (red/green/blue…).
- **Saturation** corresponds to color purity.
- **Value/Lightness** corresponds to brightness.

This separation is often helpful when the goal is to detect colored regions independent of brightness changes.

Examples in medical systems:

- isolating redness regions (inflammation),
- detecting stained tissue patterns in microscopy.

5.4 YCbCr / YUV (Luminance + chrominance)

- Separates brightness (**Y**) from color components (**Cb, Cr**).
- Useful in video processing and compression.
- Helpful when brightness varies due to lighting but chrominance remains more stable.

6. Color Perception vs Computer Color (important differences)

Humans perceive color through the eye and brain. Cameras measure radiation and apply processing (white balance, gamma correction). As a result:

- A camera's RGB values are not "absolute truth."
- Two cameras can produce different RGB values for the same scene.
- Lighting conditions can drastically alter RGB values.

This is why robust computer vision often uses:

- normalization,
- color constancy strategies,
- illumination-invariant representations (when possible).

7. Color in Intelligent Medical Systems

Color is critical in several medical imaging contexts:

7.1 Endoscopy and laparoscopic surgery

Color changes can indicate:

- bleeding,
- inflammation,
- necrosis,
- abnormal tissue patterns.

Computer vision tasks include:

- detecting abnormal regions,
- segmenting lesions,
- tracking instruments.

7.2 Dermatology imaging

Color cues support detection of:

- melanoma suspicion features,

- erythema (redness),
- pigmentation variation.

7.3 Microscopy

Stains encode tissue structure using color:

- nuclei, cytoplasm, connective tissue appear with distinct color distributions. Vision tasks include:
- nuclei detection and counting,
- tissue segmentation,
- classification of abnormal patterns.

7.4 Thermal / infrared imaging

Not RGB color—often a single-channel measurement displayed with a false-color palette. The underlying data is intensity related to temperature.

8. Histograms (Grayscale and Color Distributions)

Below is a **student-ready Section 8** rewritten using **your exact explanation and flow**, but cleaned up academically, adapted to **image histograms**, and with **simple math added** (not heavy). I also kept your “histogram1 / histogram2 / histogram3” placeholders so you can insert figures later.

---

8. Image Histograms (Grayscale and Color Distributions)

8.1 What is a Histogram?

A histogram is a graph. A graph that shows frequency of anything. Usually histogram have bars that represent frequency of occurring of data in the whole data set.

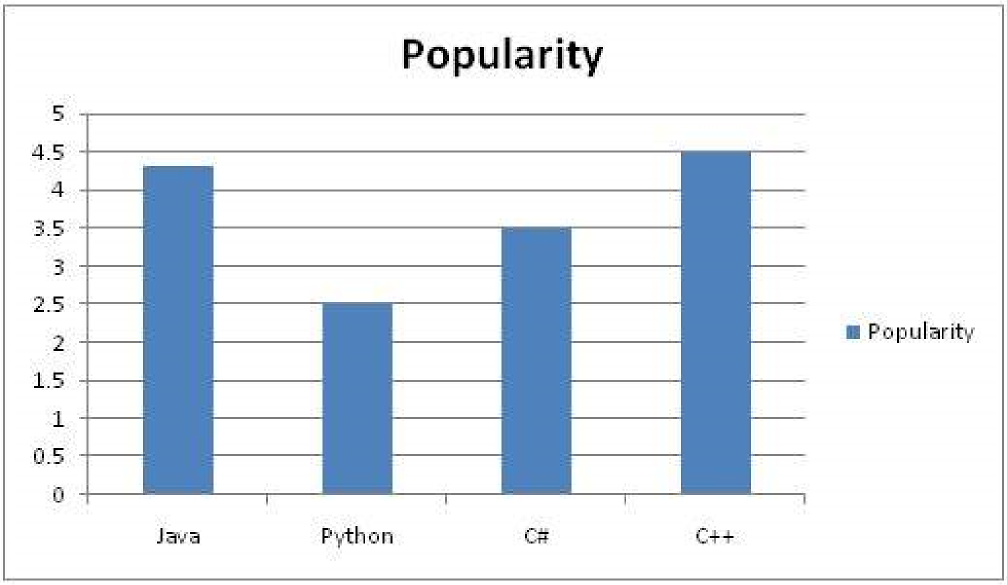
Histograms usually have **bars**, and the **height of each bar** represents the frequency (count).

A histogram has **two axes**:

- **x-axis:** the values/events whose frequency we want to count
- **y-axis:** the **frequency** (how many times each value occurs)

The different heights of bars show different frequencies.

Usually a histogram looks like this:



8.2 Histogram of an Image

An image histogram works the same way, but instead of grades we count **pixel intensity values**.

- **x-axis:** gray-level intensity values (pixel values)
- **y-axis:** frequency (how many pixels have that intensity)

So:

Image histogram = “how often each pixel intensity appears in the image”.

8.4 Grayscale Image Histogram

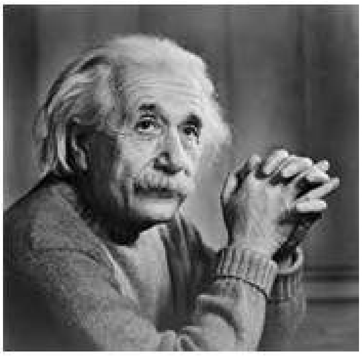
If the image is **8 bits per pixel** , each pixel intensity is in the range:

$$0 \rightarrow 255$$

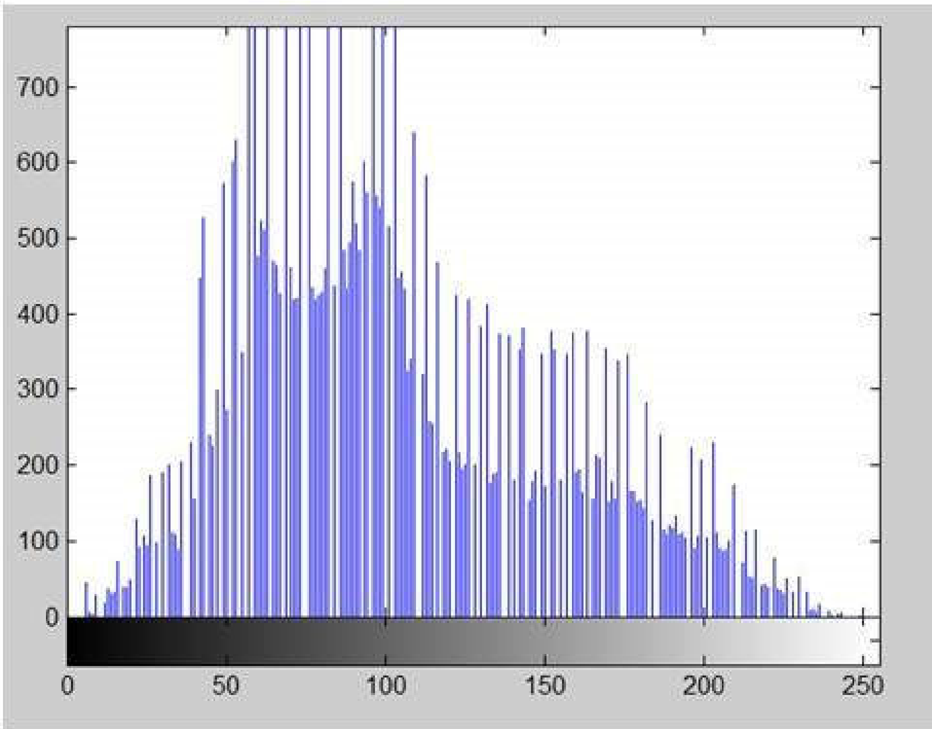
- 0 = black (very dark)
- 255 = white (very bright)

Example:
If many pixels have values near 0 → image is dark.
If many pixels have values near 255 → image is bright.

8.5 Example:



The histogram of this image would look like this:



Explanation:

- The x-axis shows pixel values from **0 to 255** (because it is 8 bits).
- The y-axis shows how many pixels fall at each intensity value.

If most high bars are in the **left side** (near 0), the image contains many dark pixels → the image looks darker.

8.6 Mathematical Definition

8.6.1 Grayscale

Let the image be (I(r,c)), where:

- (r) = row index
- (c) = column index
- I(r,c) = intensity value at that pixel

1) Histogram (Count)

For an intensity value (k), the histogram value (h(k)) means:

(h(k)) = number of pixels whose intensity equals (k)

Mathematically:

$h(k)= \#\{(r,c) \mid I(r,c)=k\}$

Meaning of the symbols (very important):

- (r,c) = pixel position
- (|) means **such that**
- (#{ }) means **count how many**

So the formula literally reads:

"Count how many pixel positions ((r,c)) have intensity (k)."

2) Normalized Histogram (Probability Form)

Sometimes we want the histogram independent of image size, so we normalize:

$p(k) = h(k)/(w*h)$

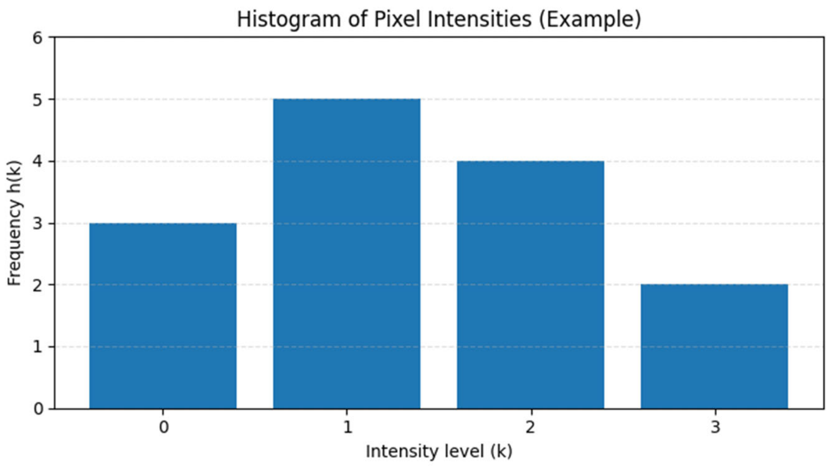
- $(p(k))$ becomes a value between 0 and 1
- It can be interpreted as "the percentage of pixels at intensity (k)".

Example:
If an image has $(w*h=1000)$ pixels and $(h(50)=200)$, then:

$p(50) = 200/1000 = 0.2$

Meaning: **20% of pixels** have intensity 50.

Example1:



The figure above shows a grayscale image histogram where the **x-axis** is the intensity level k and the **y-axis** is the frequency h(k).

Answer the following:

1. Write the values of h(0),h(1),h(2),h(3) from the histogram.
2. Compute the total number of pixels.
3. Compute the normalized histogram value p(2).

Answer

1) Read h(k) from the bars:

- h(0)=3
- h(1)=5
- h(2)=4
- h(3)=2

2) Total number of pixels:

w*h = h(0)+h(1)+h(2)+h(3) = 3+5+4+2 = 14

3) Normalized value p(2):

p(2)= h(2) / (w*h) = 4 / 14=0.28

Final: p(2) = 0.28 (about **28%** of the pixels have intensity 2).

Example 2:

Given the following grayscale image I(r,c)I(r,c)I(r,c) of size 3×4

$$I = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 3 & 1 & 2 & 0 \\ 2 & 1 & 3 & 1 \end{bmatrix}$$

Answer the following:

1. Find h(0),h(1),h(2),h(3).
2. List all pixel positions (r,c) where I(r,c)=0
3. Compute the total number of pixels MN.
4. Compute p(2).

Answer

1) Compute h(0),h(1),h(2),h(3)

Count values in the matrix:

- **Zeros (0):** 2 occurrences → h(0)=2h(0)=2h(0)=2
- **Ones (1):** 5 occurrences → h(1)=5h(1)=5h(1)=5
- **Twos (2):** 3 occurrences → h(2)=3h(2)=3h(2)=3
- **Threes (3):** 2 occurrences → h(3)=2h(3)=2h(3)=2

So:

h(0)=2, h(1)=5, h(2)=3, h(3)=2

2) Positions where I(r,c)=0

Look for zeros in the matrix:

Row 0: [1,0,2,1] → zero at column 1 → (0,1)
Row 1: [3,1,2,0]→ zero at column 3 → (1,3)
Row 2: [2,1,3,1]→ none

So the positions are:

(0,1), (1,3)

3) Total number of pixels

$w*h=3\times 4=12$

Check:

2+5+3+2=12

4) Normalized histogram value p(2)

p(2)= h(2) / (w*h) = 3 / 12 = 0.25

Final: p(2)= 0.25 (25%)

8.6.2 Color Histogram (RGB Case)

A color image has three channels: **R, G, B**.
So instead of one histogram, we can compute **three** histograms:

$hR(k),\ hG(k),\ hB(k)$

Each histogram describes the distribution in one channel.

Why do we need this?

Because a color image can change in different ways:

- The image may become more **red** (R values larger)
- More **green** (G values larger)
- More **blue** (B values larger)

8.7 Applications of Histograms in Image Processing and Computer Vision

Histograms have many uses in image analysis:

1. **Image Analysis**
   By looking at the histogram, we can guess if an image is dark, bright, or low contrast. It is like an "X-ray" of the image distribution.
2. **Brightness Adjustment**
   Histograms are used to adjust overall brightness.
3. **Contrast Adjustment**
   If pixel values are in a narrow range, contrast is low. Histogram-based methods can stretch values and increase contrast.
4. **Histogram Equalization**
   A technique that redistributes intensity values to improve contrast automatically.
5. **Thresholding (Very important in Computer Vision)**
   Histograms help choose a threshold to separate objects from background, especially in segmentation tasks.

9. Practical considerations for using color in vision systems

9.1 White balance and illumination changes

Color-based algorithms can fail if lighting changes. In medical imaging:

- different devices and settings,
- auto-exposure and auto-white-balance,
- specular reflections (wet tissue in endoscopy), can distort color.

9.2 Noise and compression

Video streams often use compression that may distort chrominance more than luminance. This affects subtle color cues in clinical analysis.

9.3 Choosing the right color representation

- Use RGB when you need the original captured representation.
- Use HSV when isolating a specific color range is important.
- Use luminance/chrominance spaces when brightness variation is a problem.
- Consider grayscale when color adds little value to the target task.

10. Connection to our labs (Python)

In the lab, students will handle color images as multi-channel arrays and will learn to:

- load and inspect RGB images,
- split channels,
- convert between color spaces (RGB ↔ HSV, RGB ↔ Grayscale),
- visualize histograms per channel,
- understand how lighting changes affect pixel distributions.

Homework 2

Solve example 2 in **5.2 Grayscale and Luminance Conversion (RGB → Gray)**

Submission: Word or PDF.