Assignment No.03

Aim :

Perform image sampling and quantization on a set of images and analyze the sample values in the sampling process and find different quantized levels in color models.

Objectives :

1) To understand the concepts of image sampling (spatial resolution reduction).
2) To implement image quantization (reducing intensity/color levels).
3) To study the effect of sampling and quantization on image quality.
4) To analyze quantized levels in different color models (RGB, HSV, YCbCr).
5) To visualize and compare original, sampled, and quantized images.

Expected Outcomes :

1) Students will be able to implement sampling and quantization using Python/OpenCV.
2) Ability to analyze the effect of reducing resolution and color levels.
3) Understanding of how quantization affects different color models.
4) Observation of trade-off between image quality and data storage.

Theory And Formulae :

1. Image Sampling

- Sampling is the process of selecting a subset of pixels to represent an image.
- Spatial resolution refers to the number of pixels used in representing an image.
- If an image of size M*N is sampled at a factor of k, the new resolution becomes:

$$M' = M/K \ , \ N' = N/K$$

- Higher sampling rate = better quality, but larger storage.
- Lower sampling rate = blockiness and loss of detail.

2. Image Quantization

- Quantization reduces the number of intensity or color levels in an image.
- For grayscale images, if the original bit depth = b, total levels = 2^b.
- Reducing to L levels:

$$q(x,y) = | \ f(x,y) \ / \ \Delta \ | * \Delta$$

where:

- $f(x,y)$= pixel intensity,
- Δ= 256/L = quantization step size.

- Example:
  - 8-bit image → 256 levels.
  - Quantization to 8 levels means each pixel can take only 8 discrete values.

3. Color Models in Quantization

- **RGB**: Quantization reduces each channel (R, G, B) separately.
- **HSV**: More perceptually uniform; quantization of Hue is critical.
- **YCbCr**: Separates luminance and chrominance, often better for compression.

Pseudocode :

A. Sampling Pseudocode

Input: Image, Sampling factor k

Output: Sampled Image

1. Read the input image

2. For every k-th row and column:

    Select the pixel and discard others

3. Store the new sampled image

4. Display the sampled image

B. Quantization Pseudocode
Input: Image, Number of levels L
Output: Quantized Image

1. Read the input image
2. Compute step size Δ = 256 / L
3. For each pixel f(x,y) in the image:
    q(x,y) = (f(x,y) // Δ) * Δ + (Δ/2)
4. Replace original pixel with q(x,y)
5. Display the quantized image

Formulae :

1. Sampling:

$$M' = \frac{M}{k}, \quad N' = \frac{N}{k}$$

2. Quantization Step Size:

$$\Delta = \frac{256}{L}$$

3. Quantized Value:

$$q(x,y) = \left\lfloor \frac{f(x,y)}{\Delta} \right\rfloor \cdot \Delta$$

4. Unique Colors after Quantization (RGB):
   Colors=L^3

Result :

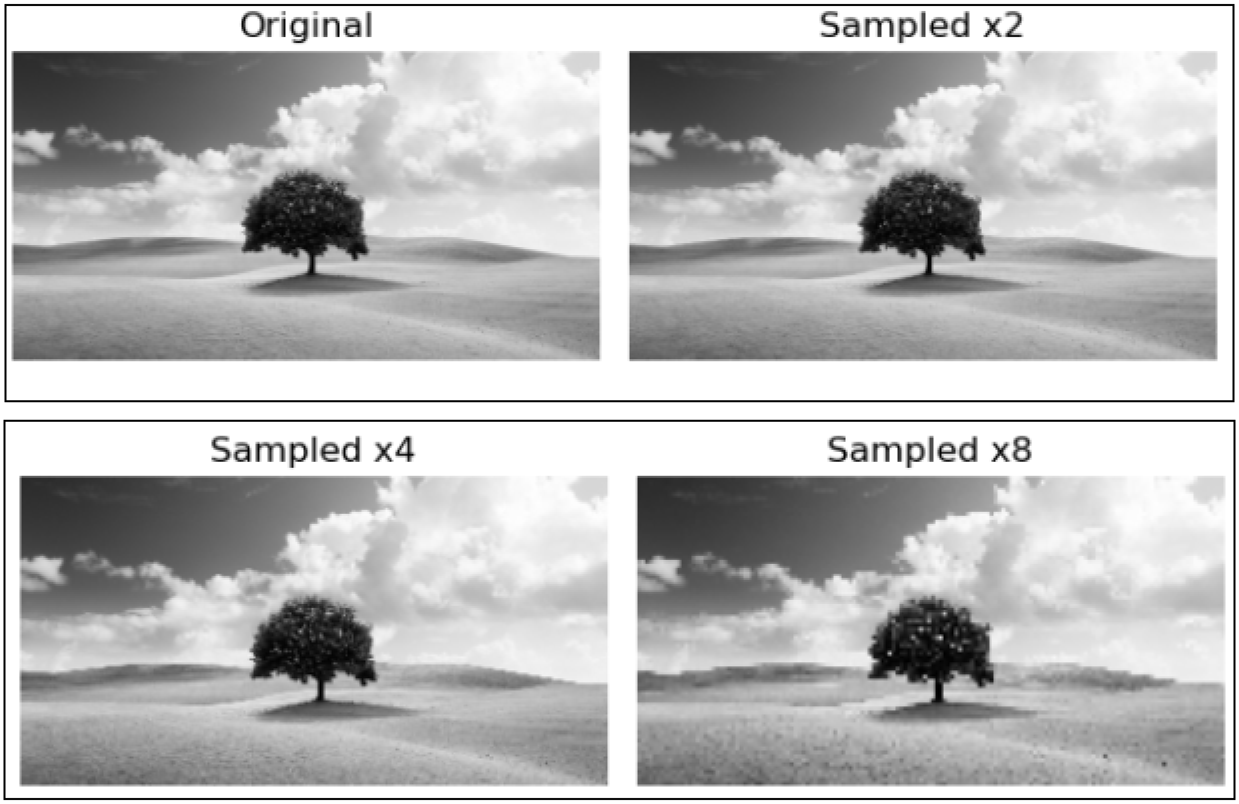

Fig Sampling of Gray Image

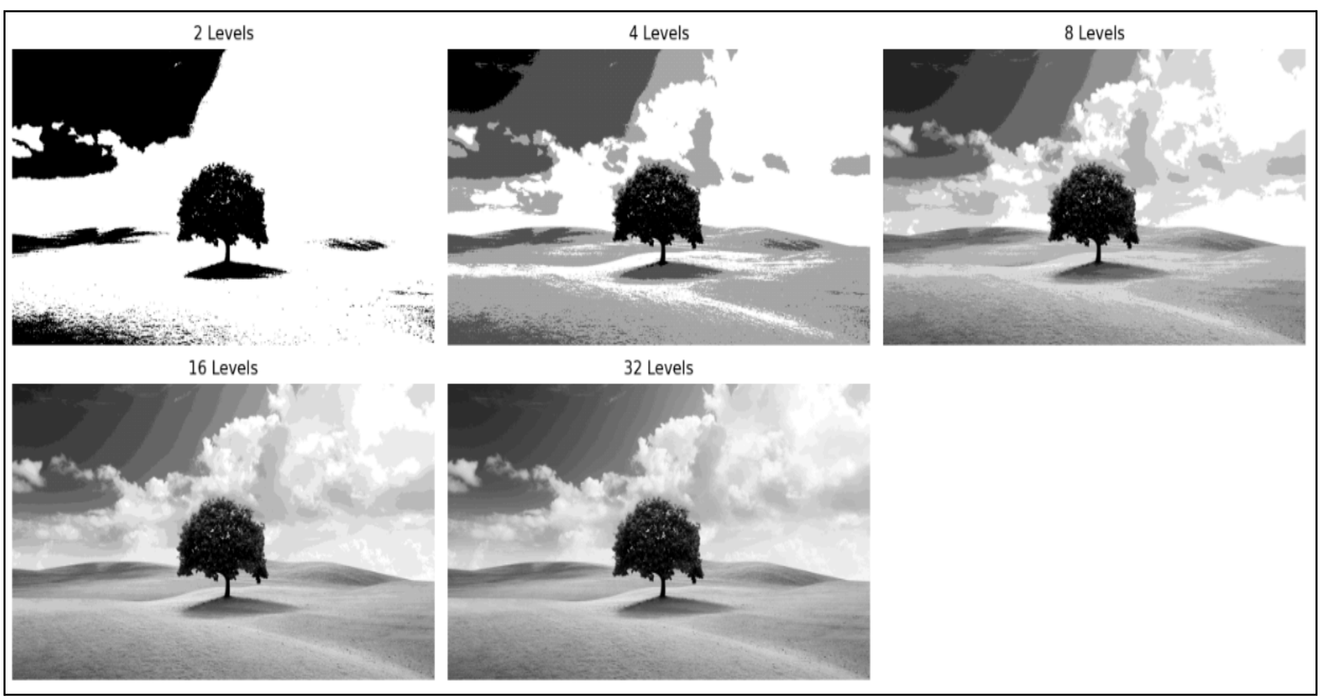

Fig. Quantization at different levels

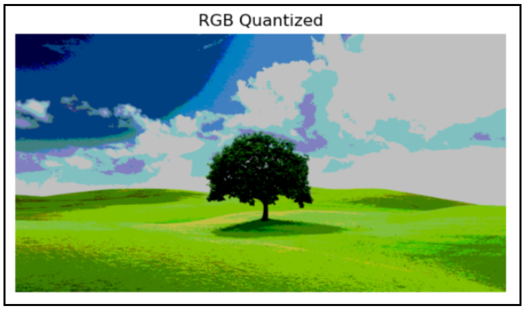

Fig. RGB Quantized Image

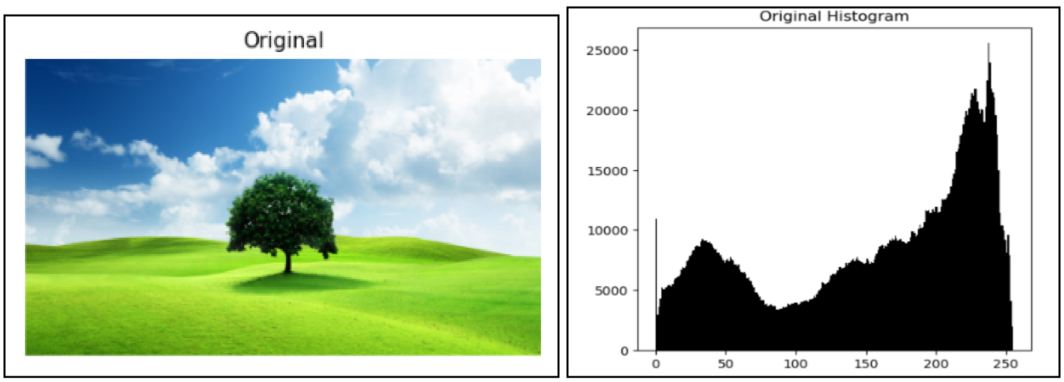

Fig. Original Image and Histogram

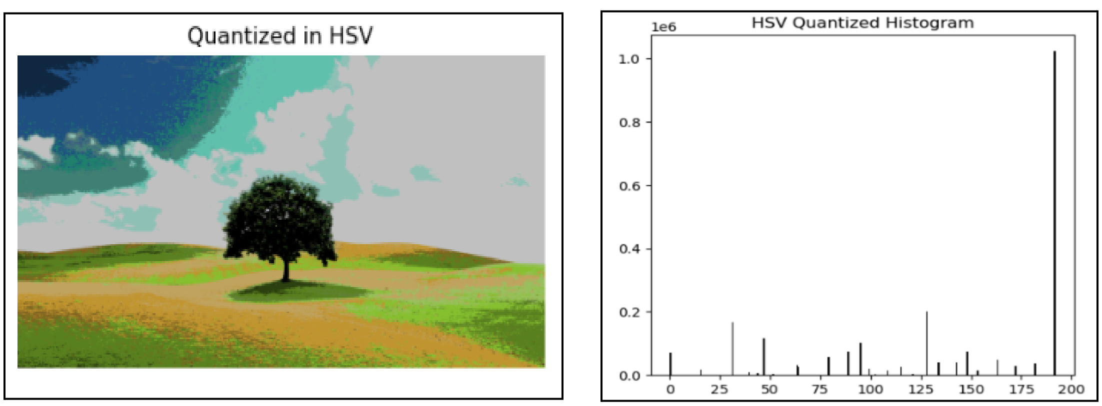

Fig. RGB Quantized Image and Histogram

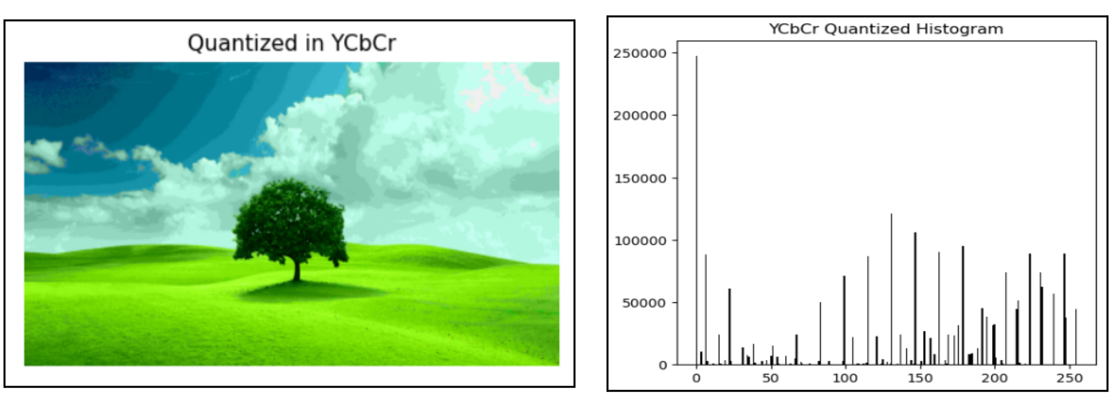

Fig. YCbCr Quantized Image and Histogram

Conclusion :

Image sampling and quantization are fundamental techniques in digital image processing. Sampling reduces spatial resolution by selecting fewer pixels, while quantization limits the number of intensity or color levels. The experiment shows that as resolution and levels decrease, image quality reduces with visible distortion, but storage requirements become smaller. Different color models respond differently, highlighting the balance between compression and perceptual quality.