جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
College of Computer Science and Information Technology
تامولعملا ةينقتو بساحلا موليك
ةيلك

DEPARTMENT OF COMPUTER ENGINEERING

ARTI404 – Image Processing

STUDENT PROCEDURAL MANUAL
Lab 2

2025 - 2026

IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

College	Department	Course
College of Computer Science and Information Technology	Department of Computer Engineering	ARTI404 – Image Processing

Practical Session Plan

Session Topic/Title	Session No.	Session Duration (Minutes)
Digital Image Fundamentals	2	100

1- Session Outcomes

Outcome#1: Explain how digital images are represented and manipulated in a computer.

2- Tool(s)/Software

- Python 3
- Anaconda
- IDE for Python: Jupyter, Spyder.
- Libraries: Pillow, OpenCV

3- Procedural steps (Tasks)

Image Sampling and Quantization

Step#1: Import the libraries

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Step#2: Define Functions

```
def sample_image(image, factor):
    """
    Downsamples the image by the given factor.
    Args:
        image (numpy array): Original image.
        factor (int): Factor by which to downsample.
    Returns:
        numpy array: Downsampled image.
    """
    height, width = image.shape[:2]
    sampled_image = cv2.resize(image, (width // factor, height
// factor), interpolation=cv2.INTER_NEAREST)
    return sampled_image

def quantize_image(image, levels):
    """
```

```
    Reduces the number of grayscale levels in the image.
    Args:
        image (numpy array): Original image.
        levels (int): Number of grayscale levels.
    Returns:
        numpy array: Quantized image.
    """
    quantized_image = np.floor(image / (256 // levels)) * (256
// levels)
    quantized_image = quantized_image.astype(np.uint8)
    return quantized_image

def plot_images(original, sampled, quantized):
    """
    Plots the original, sampled, and quantized images side by
side.
    Args:
        original (numpy array): Original image.
        sampled (numpy array): Sampled image.
        quantized (numpy array): Quantized image.
    """
    plt.figure(figsize=(12, 4))
    plt.subplot(1, 3, 1)
    plt.imshow(original, cmap='gray')
    plt.title('Original Image')
    plt.axis('off')

    plt.subplot(1, 3, 2)
    plt.imshow(sampled, cmap='gray')
    plt.title('Sampled Image')
    plt.axis('off')

    plt.subplot(1, 3, 3)
    plt.imshow(quantized, cmap='gray')
    plt.title('Quantized Image')
    plt.axis('off')

    plt.show()
```

Step#3: Import the image and set the parameters

```
image_path='../images/lena_gray_256.tif'
sampling_factor=14
quantization_levels=9

# Load image in grayscale
original_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
if original_image is None:
    print(f"Error: Unable to load image at {image_path}")

# Sample and quantize
sampled_image = sample_image(original_image, sampling_factor)
```

```
quantized_image = quantize_image(original_image,
quantization_levels)

# Plot results
plot_images(original_image, sampled_image, quantized_image)
```

Arithmetic Operations

```
from PIL import Image
img1=Image.open('../images/lena_gray_256.tif')
img2=Image.open ('../images/cameraman.tif')

resize=(400,400)
img1=img1.resize(resize,Image.Resampling.LANCZOS)
img2=img2.resize(resize,Image.Resampling.LANCZOS)

im1arr=np.asarray(img1)
im2arr=np.asarray(img2)

addition=im1arr+im2arr
resultImage=Image.fromarray(addition)
resultImage.show()
```

Sets and Logical Operations

```
img3=Image.open('../images/A.png')
img3.show()
img4=Image.open('../images/B.png')
img4.show()

resize=(400,400)
img3=img3.resize(resize,Image.Resampling.LANCZOS)
img4=img4.resize(resize,Image.Resampling.LANCZOS)

im3arr=np.asarray(img3)
im4arr=np.asarray(img4)

union=im4arr|im3arr
resultImage2=Image.fromarray(union)
resultImage2.show()
```

Practical Session Plan

Task#1: Change the sampling and quantization parameters and observe the effects.

Task#2: Read two images, convert them into an array, and perform the following operations on them.

1. Subtract two images and display the result.
2. Add one image with a constant value of 175 and display it.
3. Apply the set difference operation on two Gray-Scale images.
4. Apply the symmetric difference operation on two Gray-Scale images.
5. Apply Intersection operations on two Gray-Scale images.

4- Assessment

The student will be asked to accomplish the tasks within the lab time.

5- Resources

OpenCV documentation