Addis Ababa University

Master’s in Artificial Intelligence

Digital Image Processing (DIP) Laboratory Manual

Prepared by: Dr. Natnael Argaw Wondimu

Environment: GNU Octave

Preface

This manual is prepared in a workbook style for AI Master's students at Addis Ababa University. It provides a hands-on introduction to Digital Image Processing using GNU Octave, covering basic to advanced topics. Each lab includes objectives, theoretical background, procedures, Octave code, checkpoints, try-it-yourself prompts, and collaborative assignments. The manual is intended to build foundational knowledge in Digital Image Processing and serve as a base for advanced studies in Computer Vision.

Chapter 1: Image Sampling and Quantization (GNU Octave)

Objective

To introduce students to image sampling and quantization concepts using GNU Octave, with practical exercises and real standard images.

1. Setting Up the Environment

1.1 Installing GNU Octave

Description: GNU Octave is a high-level programming language compatible with MATLAB. It is widely used for numerical computations and image processing in open-source environments.

Steps: 1. Visit https://www.gnu.org/software/octave/download.html 2. Download the version suitable for your operating system (Windows, macOS, or Linux). 3. Follow the installation wizard instructions.

Verification: Open GNU Octave and type:

```
version
```

Output Description: Displays the version of Octave installed, confirming the setup.

1.2 Installing Image Package

Description: GNU Octave uses packages for extended functionalities. The image package allows reading, displaying, and processing images.

Code Snippet:

```
pkg install -forge image
pkg load image
```

Output Description: No error messages if successful. The image package is now active for the session.

2. Loading and Displaying Standard Test Images

2.1 Downloading Standard Images

Description: Digital Image Processing commonly uses standard test images like Lena, Cameraman, Peppers, etc. You can download them from public datasets.

Steps: - Download images from: https://sipi.usc.edu/database/ - Save them in your working directory.

2.2 Reading and Displaying an Image

Description: Load and visualize a grayscale image in Octave.

Code Snippet:

```
img = imread('cameraman.tif');
imshow(img);
title('Original Image');
```

Output Description: Displays the original Cameraman image in a separate window.

3. Image Sampling

3.1 What is Sampling?

Description: Sampling refers to the process of reducing the spatial resolution of an image (i.e., number of pixels).

3.2 Downsampling by a Factor of 2

Code Snippet:

```
sampled_img = img(1:2:end, 1:2:end);
imshow(sampled_img);
title('Sampled Image - Factor 2');
```

Output Description: Displays the image with half the original resolution in both dimensions.

3.3 Downsampling by a Factor of 4

Code Snippet:

```
sampled_img_4 = img(1:4:end, 1:4:end);
imshow(sampled_img_4);
title('Sampled Image - Factor 4');
```

Output Description: Heavily reduced image; details may be lost, edges become jagged.

4. Image Quantization

4.1 What is Quantization?

Description: Quantization involves reducing the number of gray levels used to represent an image.

4.2 Reducing to 4 Gray Levels (2-bit image)

Code Snippet:

```
quant_img = uint8(floor(double(img)/64) * 64);
imshow(quant_img);
title('Quantized Image - 4 Gray Levels');
```

Output Description: Image appears posterized, with distinct intensity bands.

4.3 Reducing to 2 Gray Levels (Binary Image)

Code Snippet:

```
binary_img = img > 128;
imshow(binary_img);
title('Binary Image (Threshold at 128)');
```

Output Description: Image shows only black and white areas. Good for edge outlines.

5. Cascaded Operations: Sampling + Quantization

5.1 Combined Process

Description: Downsample the image and then apply quantization for more compression.

Code Snippet:

```
cascaded_img = img(1:2:end, 1:2:end);
cascaded_img = uint8(floor(double(cascaded_img)/64) * 64);
imshow(cascaded_img);
title('Sampled and Quantized Image');
```

Output Description: Image is both low-resolution and posterized. Demonstrates impact of combined transformations.

6. Summary

- **Sampling** affects spatial resolution.
- **Quantization** affects intensity resolution.
- Both impact perceived quality, and are key to image compression techniques.

Suggested Exercises

1. Apply different downsampling factors (e.g., 3, 5) and compare results.
2. Experiment with quantizing to 8, 16, and 32 gray levels.
3. Implement interactive UI (sliders) using Octave’s GUI features for adjusting sampling/quantization dynamically.