SE3071 – Digital Image Processing

Year 3, Semester 1

Group Assignment 1

Table of content

1. Introduction

The Image Processing Tool is a versatile and user-friendly software application designed to facilitate various aspects of image manipulation, enhancement, and transformation. Developed using Python and leveraging libraries such as Tkinter, OpenCV, and NumPy, this tool offers a rich graphical user interface (GUI) that empowers users to interact with digital images effortlessly. Whether you're a novice looking to enhance your photos or a student exploring the world of image processing, this tool simplifies the image editing process.

2. Key Features

Open and Display Images: Users can open digital images from their local file system, and the tool displays both the original and modified images side by side in the GUI.

Basic Image Operations:

- **Color:** Users can perform operations to retain the color of the original image.
- **Black & White (B&W):** Conversion to black and white
- **Gray-scale:** Another option for converting the image to gray-scale.

Image Transformations:

- **Rotate:** Allows users to specify an angle and rotate the image accordingly.
- **Crop:** Enables users to crop the image to a defined region.

Flip: Flips the image horizontally, effectively mirroring it.

Image Filters:

Sharpen: Applies a sharpening filter to enhance details and edges in the image.
Smooth: Uses a Gaussian blur filter to reduce noise and create a smoother appearance.
Edge Detection: Applies the Canny algorithm to detect and highlight edges in the image.
Emboss: Applies an embossing filter to give the image a three-dimensional appearance.

Image Segmentation:

Region-Based Segmentation: Allows users to select and extract regions with a specific color from the image.

Image Adjustments:

Tonal Transformations: Adjusts contrast and brightness for improved visual quality.
Color Balancing: Equalizes the histogram for each color channel to enhance overall image
Image Enhancement: Utilizes the SRCNN (Super-Resolution Convolutional Neural Network) model to enhance image quality, particularly useful for upscaling images while preserving detail.

3. Related Technologies

Python

Python is a programming language that may be used to create desktop gui apps, websites, web applications, mathematics, system scripting and so many. It has an autonomous memory management system and a dynamic type of system. It contains a large comprehensive library and supports.

TensorFlow

TensorFlow is an open-source machine learning software that focuses on deep neural networks from start to finish. TensorFlow is a collection of libraries, tools, and community resources that are diverse and comprehensive. It enables programmers to construct and deploy cutting-edge machine learning based applications.

Keras

Google developed Keras, a high-level deep learning API for building neural networks. It is written in python and helps with neural network development. It is modular, quick, and simple to use. It was created by Google developer Francois Chollet. Low-level computation is not handled by Keras. Instead, it makes use of a library known as the “Backend”. Keras provides the ability to swap between different back ends.

OpenCV-python

OpenCV is a large open-source library for image processing, machine learning, and computer vision. python, C++, java, and other programming languages are among the languages supported by OpenCV. It can recognize objects, faces, and even human writing by analyzing efficient library for numerical operations.

NumPy

NumPy is the most important python package for scientific computing. It is a library that includes a multidimensional array object, derived objects (such as masked arrays and matrices), and a variety of routines for performing fast array operations, such as mathematical, logical, shape manipulation, sorting, selecting, basic linear algebra, basic statistical operations, random simulation, and more.

4. Algorithms used

Convolutional Neural Networks (CNNs):

The Image Processing Tool employs Convolutional Neural Networks (CNNs) in various operations. CNNs are a class of deep learning models designed for processing structured grid data, such as images. These networks are known for their ability to automatically learn features from raw pixel data, making them incredibly powerful for image-related tasks.

SRCNN (Super-Resolution Convolutional Neural Network):

The SRCNN (Super-Resolution Convolutional Neural Network) is a specific CNN architecture used for image super-resolution. Super-resolution refers to the process of enhancing the resolution and quality of an image, allowing it to be displayed or printed at a larger size without significant loss of detail. SRCNN is designed to upscale low-resolution images while preserving important details.

Key Points about SRCNN:

Layers: SRCNN typically consists of three main layers: the input layer, multiple convolutional layers, and the output layer.

Training: To train the SRCNN model, a dataset of low-resolution and high-resolution image pairs is used. The model learns to predict high-resolution details from low-resolution inputs.

Activation Functions: Activation functions like Rectified Linear Unit (ReLU) are commonly used in SRCNN. ReLU introduces non-linearity to the model and helps in learning complex patterns.

Convolutional Layers: The convolutional layers perform feature extraction and mapping from low-resolution to high-resolution, using learned filters (kernels).

Output Layer: The output layer produces an image that is super-resolved, having more details and higher quality than the input low-resolution image.

Usage: SRCNN is frequently employed in applications like upscaling old, low-quality images, enhancing video frames, and improving the resolution of medical images for diagnosis.

Gaussian Blur:

Gaussian blur is a fundamental image processing operation that smoothens or blurs an image by applying a Gaussian function to each pixel. The intensity of each pixel is replaced by a weighted average of its neighbors. This technique is particularly effective in reducing noise and removing unwanted fine details.

Canny Edge Detection:

The Canny edge detection algorithm is used to identify edges in images. It works by detecting areas with rapid changes in intensity, which typically correspond to object boundaries. Canny edge detection is a multi-stage process involving gradient calculation, non-maximum suppression, edge tracking, and hysteresis.

Embossing Filter:

Embossing is an image filtering technique used to create a three-dimensional or raised effect. It involves applying an embossing kernel to the image, which shifts pixel values based on local image gradients. This process enhances the perception of depth and detail.

Color-Based Segmentation:

Color-based segmentation is a method for dividing an image into regions based on color similarity. The Image Processing Tool uses a color picker dialog to select a color, and then all areas with a color close to the chosen one are extracted. This approach is particularly useful for isolating objects or regions of interest based on color.

Histogram Equalization:

Histogram equalization is an image enhancement technique that adjusts the intensity levels in an image to enhance its contrast and improve its visual quality. This operation can be applied to each color channel (e.g. red, green, and blue) separately to balance color distribution.

5. Model Architectures

In the Image Processing Tool, the primary model architecture used is the SRCNN (Super-Resolution Convolutional Neural Network) for image super-resolution.

SRCNN Model Overview:

The SRCNN is a deep convolutional neural network designed for single-image super-resolution. Super-resolution is the process of enhancing the resolution or quality of an image. The SRCNN is particularly useful when dealing with low-resolution images and is designed to predict high-resolution details.

Model Architecture:

Input Layer:

The model expects a low-resolution image as input. In the Image Processing Tool, the input is typically a grayscale image.

Convolutional Layers:

Convolutional Layer 1:

Input Shape: Variable, as the SRCNN can handle images of different sizes.

Number of Filters: 64

Filter Size: 9x9

Activation Function: Rectified Linear Unit (ReLU)

Padding: 'Same' to keep the output size the same as the input.

Role: The first convolutional layer extracts features from the low-resolution input image.

Convolutional Layer 2:

Number of Filters: 32

Filter Size: 1x1

Activation Function: ReLU

Padding: 'Same'

Role: This layer refines the features extracted by the previous layer, helping to learn more complex representations.

Convolutional Layer 3:

Number of Filters: 1

Filter Size: 5x5

Activation Function: Linear (Identity)

Padding: 'Same'

Role: The final convolutional layer produces the high-resolution output image.

Output Layer:

The output layer's purpose is to generate the high-resolution image, retaining the spatial dimensions of the input.

Key Features and Concepts:

Training Data: The SRCNN model is trained on a dataset containing pairs of low-resolution images and their corresponding high-resolution counterparts. The model learns to map low-resolution inputs to high-resolution outputs.

Activation Functions:

ReLU (Rectified Linear Unit) is used as the activation function in the first two convolutional layers. ReLU introduces non-linearity and helps the model capture complex patterns in the data.
The last layer uses a linear (identity) activation function, which allows it to predict pixel values directly.

Filter Sizes: The SRCNN employs filters of different sizes in its convolutional layers. These different filter sizes enable the model to capture both local and global features in the input image, helping it recover high-frequency details.

Output Resolution: The SRCNN's output resolution matches that of the input low-resolution image, but it aims to predict high-quality pixel values. It essentially enhances the clarity and quality of the image without changing its size

Customizable Input Size: The SRCNN can handle images of varying dimensions. This flexibility is essential for handling images of different resolutions.