**Digital Image Processing**

**Project: Image Filter Utility Suite**

****

**Submitted to**

**Muhammad Siddique**

**Submitted by**

**Shams ul Islam (01-134212-166)**

**Muneeb Hussain Anjam (01-134221-060)**

**Abdul Jabar (01-134221-002)**

**Department of Computer Science,**

**Bahria University, Islamabad.**

**02 June 2025**

1. **Abstract**

This project presents a desktop application developed using Python that provides a suite of image processing filters such as Sobel edge detection, Gaussian blur, histogram equalization, face detection, pencil sketch, and cartoon effects. The utility was built using the OpenCV, NumPy, and Tkinter libraries and offers an intuitive GUI to load, preview, filter, and save images. The project demonstrates key concepts in digital image processing while offering practical application usability.

1. **Introduction**

Digital image processing plays a vital role in today’s software and AI-driven world, with applications ranging from photography and media to surveillance, healthcare, and self-driving cars. The goal of this project is to design a compact and user-friendly desktop utility that allows users to apply a variety of image filters for enhancement, analysis, or stylistic transformation of images.

1. **Literature Review**

Several open-source tools and commercial applications exist for image filtering and editing, such as Adobe Photoshop, GIMP, and Snapseed. Python’s OpenCV library offers a rich collection of image processing tools used widely in academia and industry. This project bridges that gap by bundling various effects like edge detection, face detection, blurring, and stylization into one interface.

1. **Methodology**

The project uses:

* OpenCV for core image processing (filters, face detection).
* NumPy for matrix and pixel-level operations.
* PIL (Python Imaging Library) to interface OpenCV images with Tkinter.
* Tkinter for building the GUI and event-driven interface.

Each filter is defined as a function, and all filters are stored in a dictionary. When a user selects a filter from the dropdown and clicks “Apply,” the respective function is called.

Implemented filters:

* Sobel Edge Detection
* Canny Edge Detection
* Gaussian Blur
* Median Blur
* Histogram Equalization
* Image Sharpening
* Face Detection (Haar cascade classifier)
* Pencil Sketch Effect
* Cartoon Effect

1. **Implementation**

The application is implemented in Python using a modular structure (main.py, filters.py, gui.py). Simplified view of the process:

1. Load image using file dialog and OpenCV.
2. Display original image in the GUI.
3. Select a filter from dropdown and apply transformation.
4. Display the processed image.
5. Option to save the new image.

Tools used:

* Python 3
* OpenCV
* NumPy
* Tkinter
* PIL (Pillow)

Challenges:

* Face detection may fail on side profiles or poor lighting.
* Sketch effect required careful tuning to avoid whiteouts.
* Limited true resolution upscaling without deep learning.

1. **Results**

The application was tested on various JPEG and PNG files, including portraits, landscapes, and cartoon images. Filters performed as expected, offering both analytical (edges, histograms) and creative (cartoon, sketch) transformations.

Results were validated through:

* Visual confirmation of edge and blur filters.
* Proper face box placement.
* Visual appeal and accuracy of pencil sketch and cartoon filters.

Sample outputs (you can insert screenshots here).

1. **Discussion**

This project demonstrates practical use of digital image processing concepts in a real-world application. While simple in structure, the application effectively demonstrates filtering, detection, and image manipulation. Its limitations lie in the absence of GPU acceleration or advanced AI upscaling. Future versions could incorporate:

* Deep learning-based super-resolution (e.g., ESRGAN).
* Real-time video filtering.
* Batch processing support.
* Custom kernel input by the user.

1. **Conclusion**

The Image Filter Utility is a compact, intuitive tool that leverages Python’s powerful libraries to showcase image processing techniques. It provides a modular base for further experimentation and can be extended with AI, deep learning, or web integration. This project helped solidify our understanding of image filtering, GUI development, and OpenCV pipelines.