**Image Processing & Analysis Toolkit – Report**

**Student Roll No:** 22671A7345  
**Course:** Image Processing & Analysis  
**Deadline:** 8th September 2025

**Table of Contents**

1. Introduction
2. Image Acquisition & Fundamentals
   * CMOS vs CCD
   * Sampling & Quantization
   * Point Spread Function (PSF)
3. Implemented Toolkit Modules
   * File Handling & Image Info
   * Color Conversions
   * Geometric Transformations
   * Filtering & Morphology
   * Enhancement Techniques
   * Edge Detection
   * Bitwise Operations
   * Compression & Saving
   * Histogram & Analysis
4. Results & Discussion
5. Applications
6. Conclusion

**1. Introduction**

Digital image processing is the foundation of modern computer vision, medical imaging, and multimedia applications. This project implements a **Streamlit-based GUI toolkit** using OpenCV, NumPy, Pillow, and Matplotlib. It provides an interactive environment to visualize and experiment with essential image operations such as filtering, transformation, edge detection, and enhancement.

The aim is to **bridge theoretical concepts with practical implementation**, making the learning process more hands-on and effective.

**2. Image Acquisition & Fundamentals**

**2.1 CMOS vs CCD**

* **CCD (Charge-Coupled Device):**
  + Light is converted into electric charge and transferred across the chip.
  + High-quality images with low noise.
  + Widely used in professional cameras and medical imaging.
  + Drawback: more power consumption and cost.
* **CMOS (Complementary Metal-Oxide-Semiconductor):**
  + Each pixel has its own amplifier and can be read individually.
  + Lower cost, faster readout, and low power usage.
  + Common in smartphones, webcams, and consumer electronics.
  + Drawback: slightly noisier compared to CCD.

➡️ **Relevance**: Most medical and scientific X-ray or MRI imaging devices used to rely on CCD, but modern portable and fast devices prefer CMOS.

**2.2 Sampling & Quantization**

* **Sampling:** Dividing a continuous image into discrete points (pixels).
  + Higher sampling = higher resolution.
  + Example: A 512×512 image has 262,144 pixels.
* **Quantization:** Mapping continuous intensity values into discrete gray levels.
  + Example: 8-bit quantization → 256 gray levels.
  + More quantization levels preserve finer details but increase storage.

➡️ **Relevance**: In X-ray images, proper sampling ensures bone structures are visible, while quantization controls brightness and contrast.

**2.3 Point Spread Function (PSF)**

* PSF describes how a single point source of light is recorded by an imaging system.
* It reflects blurring due to lens imperfections, motion, or diffraction.
* Mathematically, the **observed image** = (Original Image ⊗ PSF) + Noise.

➡️ **Relevance**: In medical imaging, PSF correction helps reduce blur in X-rays or CT scans.

**3. Implemented Toolkit Modules**

**3.1 File Handling & Image Info**

* Upload images in formats (PNG, JPG, BMP, TIFF).
* Extract metadata: dimensions, channels, file size, format.

**3.2 Color Conversions**

* BGR → RGB (display correction)
* BGR → Gray (simplification)
* BGR → HSV (color-based segmentation)
* BGR → YCrCb (used in compression algorithms like JPEG)

**3.3 Geometric Transformations**

* **Rotation** – arbitrary angles with or without resizing.
* **Scaling** – enlarging/reducing images.
* **Translation** – shifting objects in x/y directions.
* **Affine Transformation** – preserves parallelism but not lengths.
* **Perspective Transformation** – simulates 3D viewpoint.

**3.4 Filtering & Morphology**

* **Filtering:**
  + Mean, Gaussian, Median filters – reduce noise.
* **Morphological Operations:**
  + Erosion, Dilation, Opening, Closing – remove small noise, fill gaps.

**3.5 Enhancement Techniques**

* **Histogram Equalization** – improves global contrast.
* **Contrast Stretching** – spreads pixel intensity over full range.
* **Sharpening** – emphasizes edges using kernels.

**3.6 Edge Detection**

* **Sobel Filter** – detects vertical & horizontal gradients.
* **Laplacian** – detects intensity changes in all directions.
* **Canny** – multi-stage edge detection (noise reduction + gradient + threshold).

**3.7 Bitwise Operations**

* Logical operations between two images: AND, OR, XOR, NOT.
* Useful for masking and region-based operations.

**3.8 Compression & Saving**

* Save images in **PNG, JPEG, BMP** formats.
* Adjust JPEG quality (30–100%) to compare compression.

**3.9 Histogram & Analysis**

* Display histograms for grayscale or RGB channels.
* Helps analyze intensity distribution and contrast.

**4. Results & Discussion**

* The toolkit successfully applies multiple operations interactively.
* Side-by-side **comparison mode** helps visualize before/after transformations.
* Sample test images (gradient, coins, chessboard) verify correctness of algorithms.
* Noise reduction filters significantly improve quality but Gaussian blur may reduce sharpness.
* Histogram equalization enhances visibility in dark/bright images.

**5. Applications**

* **Medical Imaging:** noise removal in X-rays, MRI, CT scans.
* **Remote Sensing:** satellite image enhancement, vegetation segmentation.
* **Forensics:** restoration of blurred CCTV footage.
* **Photography:** image correction, sharpening, compression.

**6. Conclusion**

This project implemented a **comprehensive image processing toolkit** that integrates theory with practice. Students can experiment with filters, transformations, and enhancement techniques interactively, improving their understanding of fundamental image analysis concepts.

Future extensions can include:

* Machine learning–based segmentation.
* Real-time video processing.
* Integration with deep learning frameworks (TensorFlow, PyTorch).

