Manipulating Images

In Python

Submitted By: Jaskirat Singh

Enrolment no.: 12813202722

Branch: B.Tech CSE-2

Guru Tegh Bahadur Institute of Technology

Table of Contents

Acknowledgement………………………………………………………………4

1. Introduction………………………………………………………………….5
   1. What is image processing?
   2. Advantages of using code
   3. Everyday uses
2. Background………………………………………………………………….9
   1. Basics of digital image representation
   2. Different image processing techniques
3. Tools and Technologies……………………………………………………..11
   1. Python as our choice of language
   2. Discussing relevant libraries
   3. Technologies of higher scope
4. Image handling in Python…………………………………………………..16
   1. Opening an image
   2. Basic editing operations
5. Image Enhancement
   1. Different types of blurs
   2. Upscaling and enhancing images
6. Color Manipulation
   1. Converting between different color spaces
   2. Adjusting color balance and other metrics
   3. How to create basic Instagram filters
7. Image Transformation
   1. Geometric transformations
   2. Perspective shifts and how they’re useful
   3. Domain transformations
8. Feature Extraction
   1. Difference of Gaussians
   2. Edge detection
   3. Corner detection
9. Advanced Techniques

9.1. Object recognition

9.2. Image segmentation and it’s uses

1. Application and Case Studies

10.1. ASCII art generator

10.2. Facial Recognition

10.3. Real-life use cases and tasks

1. Conclusion

11.1. Summary of the project

11.2. Future prospects in the field

11.3. Where to find the code in this project

References

Acknowledgment

I, Jaskirat Singh, would like to express my deepest gratitude to Professors and the entire faculty of the C.S.E. department at Guru Tegh Bahadur Institute of Technology for their unwavering support and guidance throughout this summer training report. Their invaluable insights and encouragement have been instrumental in shaping my understanding and approach to image processing in Python.

I am also profoundly thankful to Guru Tegh Bahadur Institute of Technology for providing the necessary resources and a conducive environment for research and learning. The opportunity to work in such a stimulating academic setting has greatly enriched my knowledge and skills.

Additionally, I would like to extend my appreciation to my peers and colleagues who offered their support and shared their expertise, which was vital in completing this project.

Thank you all for your contributions and encouragement.

Signatures:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Jaskirat Singh) (Institute Representative)

Introduction

This section will be covering the basics of what we are trying to achieve with this project, as well as discussing some real-life uses of everything to follow so as to provide context for all these technologies and techniques.

1.1 - What is image processing?

Image processing is a technique used to perform operations on an image to enhance it or extract useful information. It is a type of signal processing where the input is an image, such as a photograph or video frame, and the output can be either an improved image or a set of characteristics or parameters related to the image. Image processing typically involves treating a two-dimensional image as a matrix of pixel values and applying various algorithms to manipulate these values to achieve the desired effect.

One of the primary objectives of image processing is to improve the visual appearance of images to make them more suitable for display or further analysis. This involves tasks like noise reduction, contrast enhancement, and sharpening. For instance, noise reduction techniques are employed to remove unwanted random variations in pixel intensity, making the image clearer. Contrast enhancement methods adjust the dynamic range of pixel intensities, making features within the image more discernible.

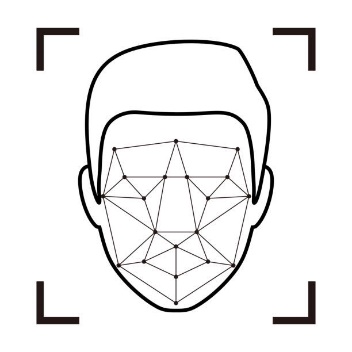
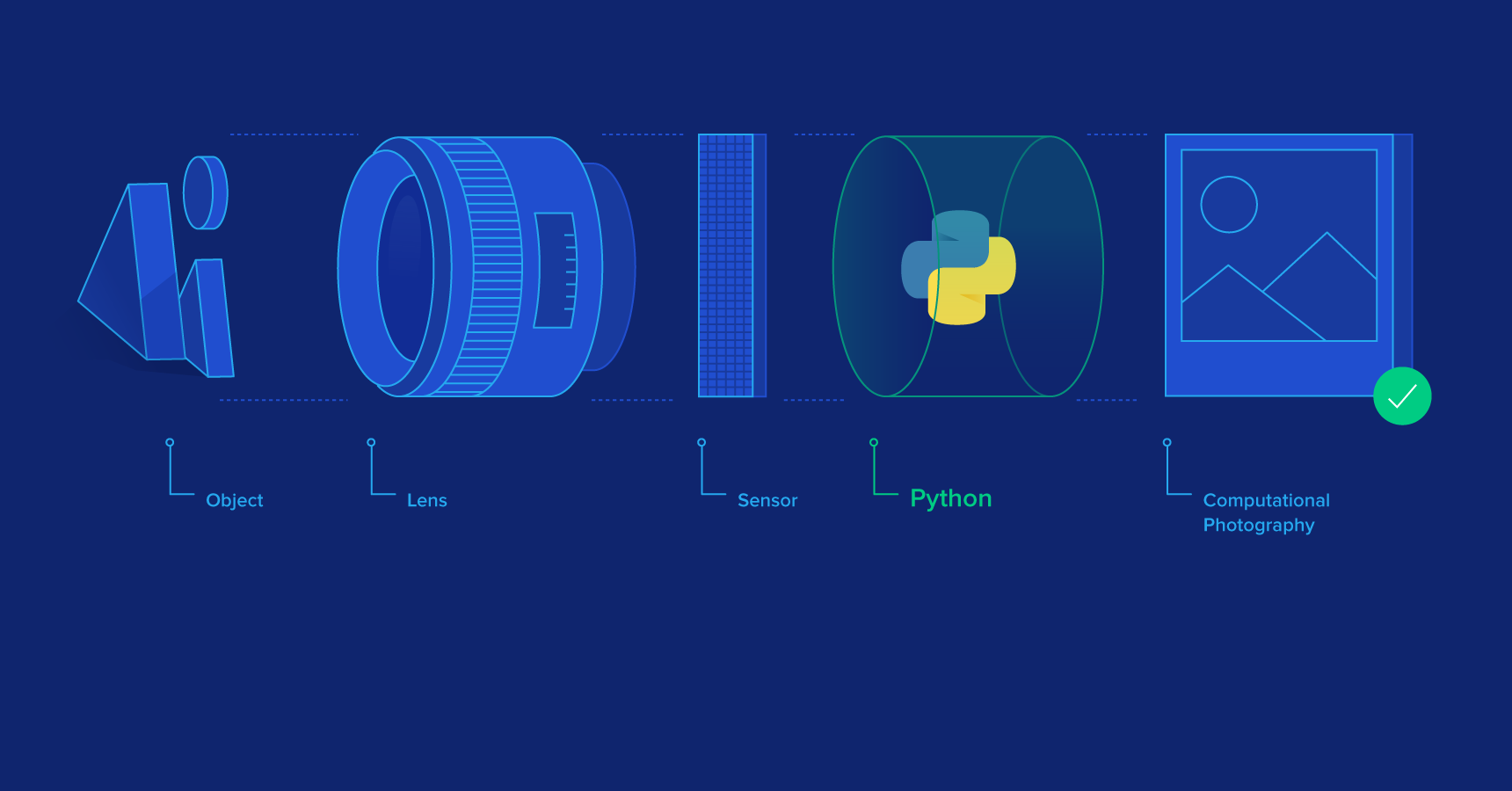
Another critical aspect of image processing is feature extraction, where specific elements or attributes of an image are identified and analysed. Techniques such as edge detection, corner detection, and blob analysis are used to extract meaningful features from images. These features can then be used in various applications like object recognition, image segmentation, and computer vision tasks. For example, edge detection algorithms highlight significant transitions in intensity, which correspond to object boundaries within the image, facilitating object detection and recognition.

Image processing is widely used in numerous fields, including medical imaging, remote sensing, surveillance, and multimedia. In medical imaging, it helps in enhancing images obtained from X-rays, MRI scans, and CT scans for better diagnosis and treatment planning. In remote sensing, satellite images are processed to monitor and analyze environmental changes. Surveillance systems use image processing for real-time monitoring and detection of suspicious activities. In multimedia, it enhances the quality of images and videos for better user experience. The versatility and efficacy of image processing make it an indispensable tool in modern technology.

1.2 - Advantages of using code

The advent of digital technology has revolutionized the field of image processing, making code-based approaches far superior to traditional manual methods. One of the primary reasons for using code in image processing is the efficiency and accuracy it offers. Manual techniques, such as hand-drawing masks for segmentation or adjusting image properties through physical means, are time-consuming and prone to human error. In contrast, code-based methods can process large volumes of images quickly and consistently, ensuring high precision and repeatability.

Automating image processing through coding allows for the implementation of complex algorithms that would be impossible or impractical to perform manually. Algorithms like Fourier transforms, wavelet transforms, and machine learning-based approaches can analyze and manipulate image data in ways that exceed human capabilities. These advanced techniques can extract intricate patterns, enhance minute details, and perform sophisticated analyses that traditional methods cannot achieve. For example, convolutional neural networks (CNNs) can identify and classify objects within images with a level of accuracy and speed that manual inspection cannot match.



Another significant advantage of using code for image processing is the flexibility and scalability it offers. With programming languages such as Python and powerful libraries like OpenCV, Pillow, and scikit-image, developers can easily customize and extend image processing pipelines to suit specific needs. Whether it's developing a new filter, integrating with other software, or processing images in real-time, the versatility of code-based approaches allows for a wide range of applications. This adaptability is particularly beneficial in dynamic fields like medical imaging and surveillance, where requirements and standards continuously evolve.

Moreover, coding enables the integration of image processing with other digital systems, enhancing the overall functionality and interconnectivity of technological solutions. For instance, image processing algorithms can be embedded in machine learning workflows, augmented reality applications, and automated quality control systems in manufacturing. This seamless integration allows for the creation of comprehensive, intelligent systems that leverage image data to make informed decisions, improve user experiences, and streamline operations.

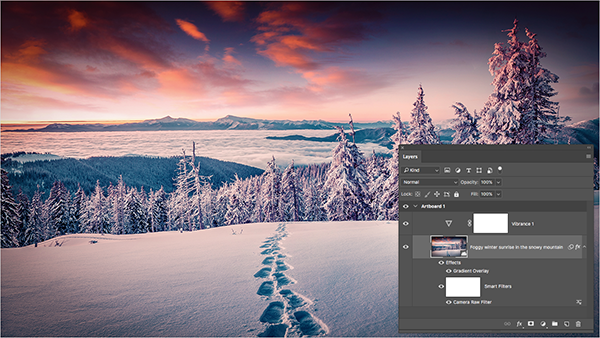
1.3 - Everyday uses

This section will discuss, in slight depth, the various uses of image processing that we see in our day-to-day life. A lot of these concepts will be important further in the project when we implement them using simpler techniques.

1. Photo Editing –

Enhancement and Retouching: Image processing techniques are widely used in photo editing software like Adobe Photoshop and GIMP. Users can enhance image quality by adjusting brightness, contrast, and sharpness, and retouch photos by removing blemishes or unwanted objects.

Filters and Effects: Various filters and effects, such as sepia, black and white, and artistic filters, can be applied to images to create different visual styles.



2. Facial Recognition –



Security and Authentication: Facial recognition technology, powered by image processing algorithms, is used for security purposes in smartphones, computers, and surveillance systems. It enables unlocking devices and verifying identities by analysing facial features.

Social Media Tagging: Social media platforms like Facebook and Instagram use facial recognition to automatically tag people in photos, making it easier to organize and share images.

3. Optical Character Recognition (OCR) –



Document Digitization: OCR technology converts printed or handwritten text into digital format. It is used in scanning documents, processing forms, and converting books into editable text files, making data more accessible and searchable.

License Plate Recognition: Image processing in OCR is used by law enforcement and toll systems to read vehicle license plates for identification and tracking purposes.

4. Content Moderation –

Social media and Online Platforms: Image processing algorithms are used to detect and filter inappropriate content, such as violence, nudity, or hate speech, ensuring a safer online environment.

Spam Detection: Platforms use image analysis to identify and block spam or misleading content, maintaining the integrity of the information shared.

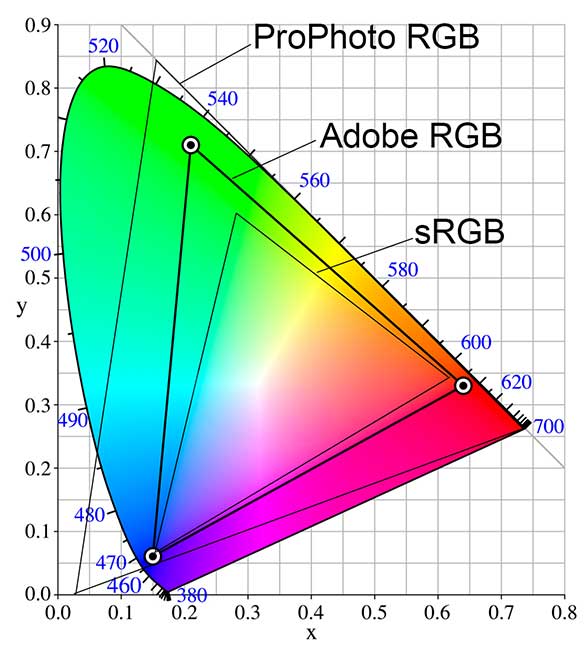
Background

Before we delve deeper, this section is all about introducing the reader to various concepts and terminology used in digital image processing. Reading this will be essential to understanding the rest of the project.

2.1 - Basics of digital image representation

Digital images are representations of visual information in a format that computers can process. At the most fundamental level, a digital image is composed of a grid of tiny elements known as pixels. Each pixel represents a single point in the image and contains information about its color and intensity. The arrangement of these pixels in rows and columns forms the complete image. The resolution of a digital image, which determines its clarity and detail, is defined by the number of pixels along its width and height.

Pixels are often represented by values that indicate their color. In grayscale images, each pixel holds a single value representing the intensity of light, typically ranging from 0 (black) to 255 (white) in an 8-bit image. This range provides 256 possible shades of gray. In color images, each pixel contains multiple values corresponding to different color channels. The most common color model used is the RGB (Red, Green, Blue) model, where each pixel has three values, one for each primary color. These values usually range from 0 to 255, allowing for over 16 million possible color combinations (256^3).



*Different and commonly used color spaces*

Another important aspect of digital image representation is the concept of color depth, which refers to the number of bits used to represent the color of a single pixel. Higher color depth allows for more colors and finer gradations. For example, an 8-bit image has a color depth of 8 bits per channel, resulting in 24 bits per pixel in an RGB image. This provides a good balance between image quality and file size. For applications requiring extremely high image fidelity, 16-bit or even 32-bit color depth may be used.

Digital images can also be represented in various formats, each with its own advantages and use cases. Common formats include JPEG, PNG, and GIF. JPEG is widely used for photographs due to its efficient compression, which reduces file size at the cost of some image quality. PNG supports lossless compression, making it suitable for images requiring high quality and transparency support. GIF, on the other hand, is limited to 256 colours and is often used for simple graphics and animations.

Understanding the basics of digital image representation is crucial for image processing tasks. It provides the foundation for manipulating images, whether it's enhancing them, extracting features, or analysing their content. By knowing how images are structured and represented digitally, one can effectively apply various image processing techniques to achieve desired outcomes.

2.2 – Different image processing techniques

* Image Filtering: Enhances or suppresses features using filters like blurring, sharpening, and edge detection.
* Geometric Transformations: Modifies spatial properties of images through translation, rotation, scaling, and perspective transformation.
* Color Manipulation: Adjusts color properties, converts color spaces, and enhances contrast and color balance.
* Image Segmentation: Partitions an image into meaningful regions based on pixel characteristics like color and intensity.
* Feature Extraction: Identifies and quantifies key attributes such as edges, corners, and blobs for image matching and recognition.
* Morphological Operations: Processes binary images to manipulate object shapes using dilation, erosion, opening, and closing.

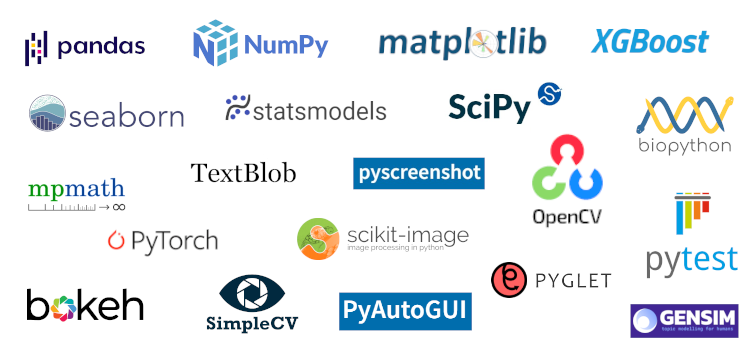
Tools and Technologies

This section will be making a case for why we’re using python to begin with, as well as the different relevant tools it offers for this project in particular.

3.1 - Python as our choice of language

Python is an excellent choice for image processing projects due to its simplicity, versatility, and extensive library support. As a high-level programming language, Python's easy-to-read syntax and dynamic typing make it accessible for beginners and powerful for experienced developers. This ease of use accelerates development and allows for quick prototyping, which is particularly beneficial in a research and experimentation-heavy field like image processing. Python's robust standard library and its integration with other technologies and tools further enhance its utility.

One of the main reasons Python is favored for image processing is its rich ecosystem of specialized libraries. Libraries such as OpenCV, Pillow, and scikit-image provide comprehensive tools for a wide range of image manipulation tasks, from basic operations like reading and writing images to advanced techniques like feature extraction and machine learning integration. OpenCV, for instance, offers an extensive set of functions for image processing and computer vision, enabling efficient implementation of complex algorithms. Pillow, a friendly fork of the Python Imaging Library (PIL), simplifies tasks like image creation, transformation, and filtering. Scikit-image leverages NumPy arrays, facilitating seamless integration with other scientific computing libraries.



*Different Python libraries*

Furthermore, Python's strong community support and wealth of documentation make it an ideal environment for both learning and development. Extensive online resources, tutorials, and forums provide valuable assistance, ensuring that solutions and examples are readily available for a wide array of image processing challenges. Additionally, Python's compatibility with other data analysis and visualization libraries, such as NumPy, SciPy, and Matplotlib, allows for comprehensive data manipulation and insightful result presentation. This integration capability is crucial for developing and evaluating image processing algorithms, making Python the go-to language for this project.

3.2 - Discussing relevant libraries

As discussed in the previous section, there are numerous useful libraries we will be using throughout our project. Here are a few of them along with install commands in pip for each one:

1. OpenCV (Open-Source Computer Vision Library):

Overview: OpenCV is a powerful library designed for real-time computer vision and image processing. It offers a wide range of functionalities, including image filtering, geometric transformations, and advanced algorithms for feature detection and object recognition.

pip install opencv-python

Key Features:

* Image Manipulation: Functions for resizing, cropping, and rotating images.
* Filtering: Includes a variety of filters like Gaussian blur and edge detectors.
* Feature Detection: Tools for detecting edges, corners, and objects, such as the Canny edge detector and Haar cascades.
* Integration: Supports integration with machine learning libraries for tasks like image classification and object detection.

2. Pillow (Python Imaging Library):

Overview: Pillow is a user-friendly fork of the original Python Imaging Library (PIL). It simplifies image processing tasks with a straightforward API, making it suitable for a variety of image manipulation needs.

pip install pillow

Key Features:

* Basic Operations: Functions for opening, saving, and displaying images in various formats.
* Image Enhancement: Tools for adjusting brightness, contrast, and color balance.
* Transformation: Capabilities for rotating, resizing, and cropping images.
* Image Creation: Allows for the creation of new images and drawing shapes and text onto them.

3. scikit-image:

Overview: Scikit-image is a part of the SciPy ecosystem, designed for image processing tasks in Python. It provides a collection of algorithms and utilities for image analysis, built on top of NumPy arrays.

pip install scikit-image

Key Features:

* Segmentation: Algorithms for image segmentation, such as thresholding and region growing.
* Feature Extraction: Tools for extracting features like edges and textures.
* Morphological Operations: Functions for operations like dilation and erosion, useful in binary image processing.
* Integration: Seamlessly integrates with other scientific libraries like NumPy and Matplotlib, enabling sophisticated data analysis and visualization.

4. NumPy:

Overview: NumPy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a wide array of mathematical functions to operate on these arrays.

pip install numpy

Key Features:

* Array Operations: Efficient manipulation of image data represented as arrays, enabling complex mathematical operations.
* Performance: Fast computation through vectorized operations, crucial for processing large images or datasets.
* Integration: Works well with other scientific computing libraries, providing a foundation for image processing and data analysis tasks.

5. Matplotlib:

Overview: Matplotlib is a comprehensive plotting library used for creating static, animated, and interactive visualizations in Python. It is particularly useful for displaying and analysing image data.

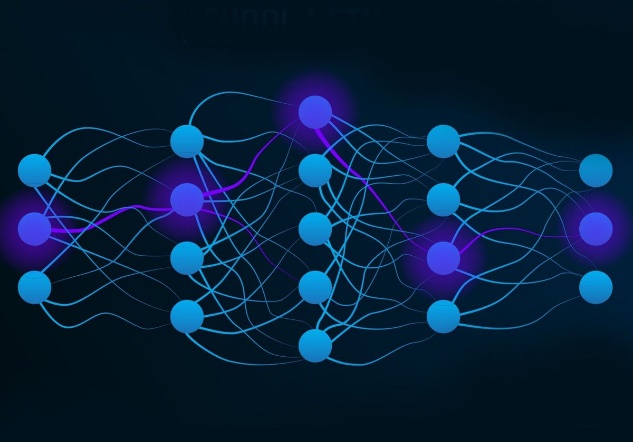
pip install matplotlib

Key Features:

* Image Display: Functions for displaying images and visualizing image data in various formats.
* Visualization: Tools for creating plots and graphs to analyze and interpret image processing results.
* Customization: Allows for extensive customization of plots and visualizations, enhancing the presentation of results.

3.3 - Technologies of higher scope

At the cutting edge of image processing, several advanced technologies are pushing the boundaries of what can be achieved with digital images. One such technology is **Deep Learning-based Image Analysis**, which leverages neural networks to perform tasks that were once considered too complex for traditional algorithms. Convolutional Neural Networks (CNNs) are particularly noteworthy in this field, as they excel at tasks such as image classification, object detection, and segmentation with remarkable accuracy. Techniques like transfer learning and data augmentation are used to train these networks on large datasets, enabling them to generalize well to new, unseen images. Although these methods offer unprecedented performance, they require significant computational resources and are beyond the scope of basic image processing projects.

Another frontier in image processing technology is **Generative Adversarial Networks (GANs)**. GANs consist of two neural networks, a generator and a discriminator, that are trained together in a competitive setting. This approach is used to generate realistic synthetic images, enhance image resolution, and perform style transfer. GANs have demonstrated impressive results in creating high-quality images from noisy inputs and generating realistic images from text descriptions. While these techniques represent the forefront of image generation and enhancement, they involve complex model architectures and extensive training processes, which are typically beyond the scope of standard image processing projects.

Additionally, **3D Image Processing and Computer Vision** technologies are advancing rapidly, driven by applications in fields like robotics, virtual reality, and medical imaging. Techniques for 3D reconstruction and analysis allow for the creation of detailed three-dimensional models from 2D images or sensor data. This includes advancements in stereo vision, depth sensing, and multi-view reconstruction. These technologies enable applications such as augmented reality and precise surgical planning but require sophisticated hardware and advanced algorithms that are not typically covered in basic image processing projects.

These cutting-edge technologies illustrate the rapid progress in image processing and its expanding applications. While they offer exciting possibilities for future work, they require specialized knowledge and resources that extend beyond the typical scope of introductory image processing projects.

Image handling in Python

Now that we’ve established the basics of image processing and all the techniques we will and will not be using in this project, we will now be discussing the main program created for this project, all its features and how they came to be. As such, going forward, the project will be a lot more oriented towards the code end of things, with screenshots and basic info provided for everything. If you’d like to research any of these topics in further detail, it can be done so at the references provided at the end of the document.

4.1 - Opening an image