



DEVELOPMENT OF IMAGE EDITOR USING DATA SCIENCE TECHNIQUES FOR THE ENHANCEMENT OF IMAGE QUALITY

Ritesh Kumar¹, Anuradha Misra^{2*}, Praveen Kumar Misra³

Article History: Received: 14.01.2023

Revised: 31.03.2023

Accepted: 08.05.2023

Abstract

This project is an image editor using Python programming language. The main objective of this project is to provide a user-friendly interface that allows users to edit their images easily. The editor provides features such as cropping, resizing, rotating, adjusting brightness and contrast, and applying filters. The application is built using Python libraries such as Streamlit, Pillow, and OpenCV. The user can upload an image, and the editor displays the image with a variety of editing tools. The user can then apply the desired changes and save the edited image to their computer. Overall, this project provides a simple yet powerful tool for users to edit their images using Python.

Keywords: Image Editing, Opencv, Streamlit,

¹UG student, Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University, Lucknow Campus, India,

²Assistant Professor, Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University, Lucknow Campus, India,
ORCID: 0000-0002-3790-8798

³Assistant Professor & Coordinator, Department of Mathematics and Statistics, Dr.Shakuntala Misra National Rehabilitation University, Lucknow, India
ORCID: 000-0002-3539-2994

Email: ¹ritesh.kumar16@student.amity.edu, ³praveenkumarmisra@gmail.com

Corresponding Author: Anuradha Misra^{2*}

Email: ^{2*}amisra@lko.amity.edu

DOI: 10.31838/ecb/2023.12.s3.270

1. Introduction

The demand for image editing tools has grown exponentially in recent years, thanks in large part to the rise of social media and the increasing importance of visual content in online communication. With the growth of this demand, there has been a corresponding rise in the need for accessible and easy-to-use image editing tools. Python has emerged as a popular choice for digital image processing and editing due to its versatile and powerful libraries.

This project aims to create an image editor using Python that provides users with a powerful yet intuitive tool for editing their images. The image editor is built using several popular Python libraries, including Pillow and OpenCV. Pillow provides the image editing and processing functions, while OpenCV provides advanced image processing capabilities, such as facial recognition and object detection.

One of the main challenges of creating an image editor is creating a user-friendly interface that makes it easy for users to perform complex tasks without requiring them to have advanced technical knowledge. This project uses Streamlit, a standard Python library for creating GUIs, to create an intuitive and easy-to-use interface that allows users to perform common image editing tasks such as cropping, resizing, and adjusting brightness and contrast.

In addition to basic image editing functions, the image editor also provides users with a variety of filters that they can apply to their images. These filters range from simple color adjustments to more complex effects, such as blur and edge detection. Users can preview these filters before applying them to their images, allowing them to experiment with different settings and achieve the desired results.

Overall, this project demonstrates how Python can be used to create a powerful and accessible image editor that provides users with a range of useful tools and features. The combination of Python libraries such as Pillow, OpenCV, and Streamlit makes it possible to create an image editor that is both powerful and easy

to use, opening up a range of possibilities for digital image processing and editing.

Application of Image Editor

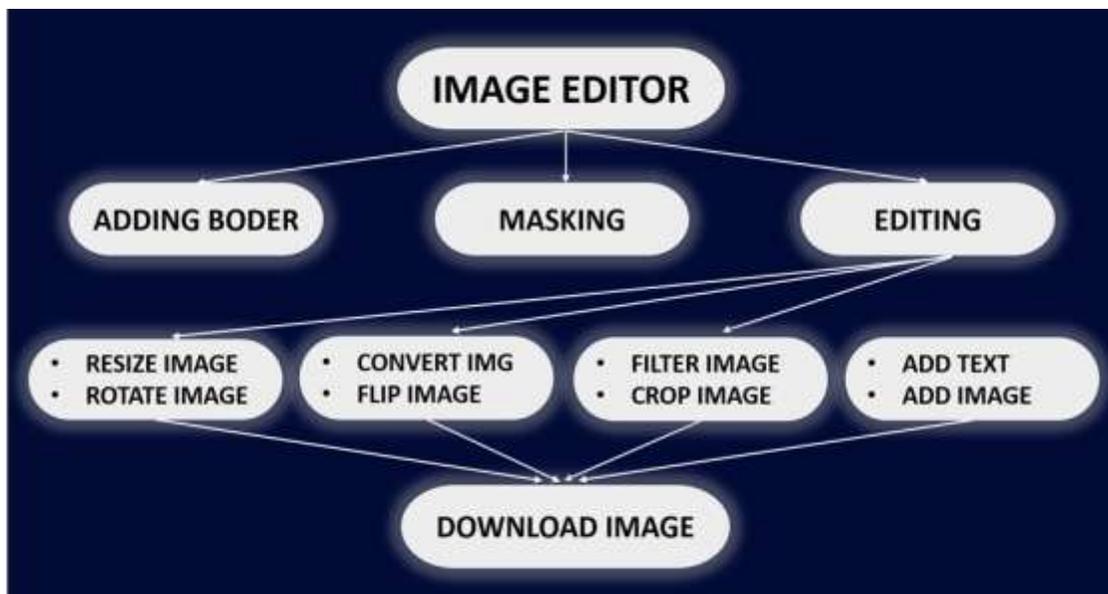
Picture Editor is an item that licenses pictures to be adjusted and besides changed over to different plans. Picture editors conventionally oversee just bitmapped pictures like GIFs, JPEGs and BMPs; regardless, a couple editors support the two bitmaps and depictions (see vector plans). Ordinary limits are Manually cropping and resizing the image

- 1) Add filters to adjust brightness, contrast and colours.
- 2) Add borders to image
- 3) Adding text over image
- 4) Adding image over image
- 5) Adding a layer of different image over the real image
- 6) Rotate the image

2. Related Work

1. Adobe Photoshop: A widely used professional image editing software that provides a range of functionalities for photo manipulation, including image retouching, cropping, resizing, and applying various filters and effects.
2. GIMP (GNU Image Manipulation Program): An open-source image editor that provides advanced photo editing features similar to Adobe Photoshop, including layer management, colour correction, and image enhancement.
3. Paint.NET: A free and user-friendly image editor that offers basic image editing functionalities, such as resizing, cropping, and colour adjustment.
4. Pixel Editor: A free and cloud-based image editor that offers a range of features for image editing, including layers, colour correction, and filters.
5. Fotor: An online photo editor that provides basic image editing functionalities, such as cropping, resizing, and adding filters.

Proposed work

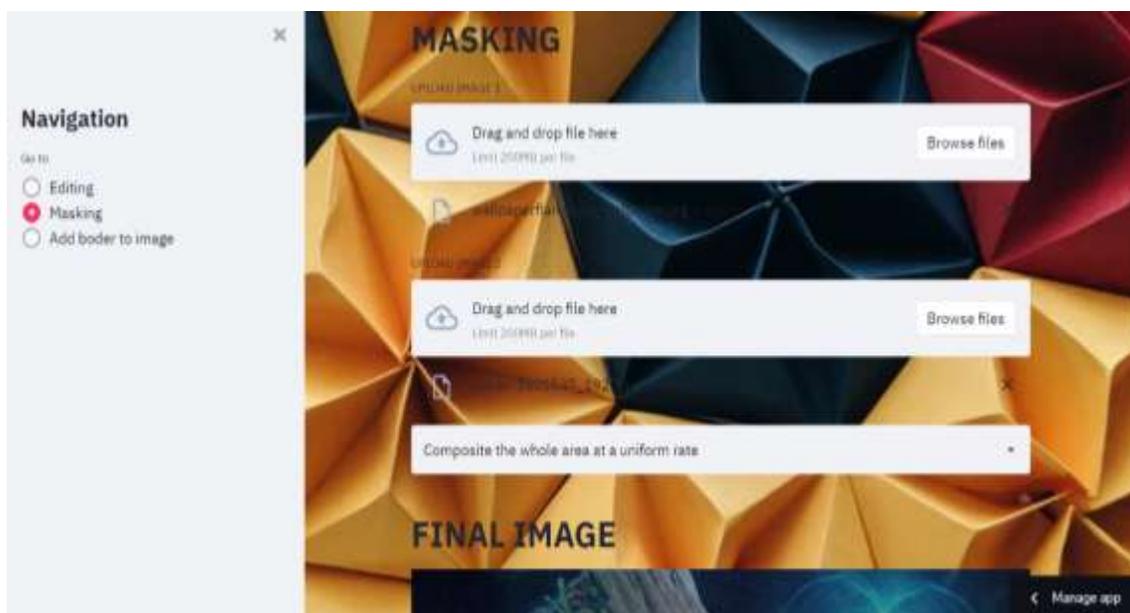


Masking :
Original Images :-





Procedure:-





The proposed work for the image editor using Python includes the following steps:

1. **Research and analysis:** The first step in creating an image editor using Python is to conduct research and analysis on the existing image editing tools and the functionalities they offer. This will help in identifying the essential features that the editor should have to make it competitive.
2. **Requirements gathering:** Based on the research and analysis, the next step is to gather the requirements for the image editor. This includes identifying the essential features such as cropping, resizing, adjusting brightness and contrast, applying filters, and others.
3. **Design and architecture:** Once the requirements are gathered, the next step is to design the architecture of the image editor. This includes defining the different modules, the classes and functions required, and the flow of data between them.
4. **Development:** The actual development of the image editor starts with coding the different modules, classes, and functions identified in the architecture design. The image editor can be built using Python libraries such as Streamlit, Pillow, and OpenCV.
5. **Testing:** Once the development is complete, testing the image editor is critical to ensure that all the functionalities work correctly, and there are no bugs. This step also includes identifying and fixing any issues or errors that may arise during the testing phase.
6. **Deployment:** After testing, the image editor is ready for deployment. This involves creating an installer for the application and making it available for download on various platforms such as Windows, Mac, and Linux.
7. **Maintenance and support:** After deployment, the image editor will require maintenance and support to ensure that it continues to function correctly and remains competitive. This includes updating the application with new features and bug fixes, providing technical support to users, and ensuring that the editor remains compatible with the latest operating systems and hardware.

In summary, the proposed work for creating an image editor using Python involves conducting research and analysis, gathering requirements, designing the architecture, developing the application, testing, deployment, and providing maintenance and support.

3. Conclusion

The image editor created using Python is a powerful and accessible tool that provides users with a range of useful functionalities for editing their images. The application is built using several Python libraries, including Pillow, OpenCV, and Streamlit, making it a versatile and robust editor.

The research and analysis conducted during the development of the image editor helped identify the essential features that the application should have, making it competitive with other image editing tools in the market. The requirements gathering and architecture design ensured that the editor was designed to meet the needs of its users, making it intuitive and easy to use.

The development and testing of the image editor ensured that it functions correctly and is free of bugs, while the deployment and maintenance ensured that the application was available to users on different platforms and remained up-to-date with the latest operating systems and hardware.

Overall, the image editor using Python provides an excellent example of how Python can be used to create a powerful and accessible tool for digital image processing and editing. The project demonstrates the potential of Python libraries in digital image processing, and highlights the importance of research, design, and testing in creating high-quality software applications.

Future Scope

The image editor created using data science techniques can further be utilized in the following areas, but not limited to-

Image recognition and classification: Data science techniques such as machine learning and deep learning can be used to train an image editor to recognize and classify objects in images. This can help in automating the image editing process by identifying the objects and making appropriate adjustments to them.

Image enhancement: Data science techniques can be used to enhance the quality of images, such as improving their resolution, sharpness, and color quality. This can help in making images more visually appealing and useful in various applications.

Image restoration: Data science techniques can be used to restore damaged or degraded images by removing noise, scratches, and other imperfections. This can be useful in preserving historical images and documents.

Image synthesis: Data science techniques such as generative adversarial networks (GANs) can be used to create new images by synthesizing them from existing ones. This can be useful in creating new content for digital media and in various applications such as virtual reality.

Image compression: Data science techniques can be used to compress large image files without losing significant amounts of image quality. This can be useful in reducing storage requirements for images and in making them easier to share and transmit over the internet.

4. References

- Pillow library documentation: <https://pillow.readthedocs.io/en/stable/>
- Open CV documentation: <https://docs.opencv.org/4.5.2/>
- Tkinter documentation: <https://docs.python.org/3/library/tkinter.html>
- Python Imaging Library Handbook: <https://effbot.org/imagingbook/>
- Joseph Howse , "Python Image Processing with OpenCV" by: <https://www.packtpub.com/product/python-image-processing-with-opencv/9781785282690>