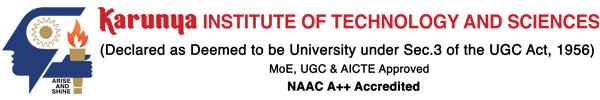
****

**Division of Electronics and Communication Engineering**

**2023-2024 (EVEN SEM)**

**III IA EVALUATION REPORT**

***for***

**DIGITAL SIGNAL PROCESSING-PROJECT BASED COURSE**

***Title of the project: Implementation Of Image Compressor Using Python***

***A report submitted by***

|  |  |
| --- | --- |
| ***Name of the Student*** | ***V. Franclin*** |
| ***Register Number*** | ***URK22EC2001*** |
| ***Subject Name*** | ***Digital Signal Processing*** |
| ***Subject Code*** | ***18EC2015*** |
| ***Date of Report submission*** |  |

**Project Rubrics for Evaluation**

**First Review:** Project title selection - PPT should have four slides (Title page, Introduction, Circuit/Block Diagram, and Description of Project).

**Second Review:**  PPT should have three slides (Description of Concept, implementation, outputs, results and discussion)

Rubrics for project (III IA - 40 Marks):

Content - 4 marks (based on Project)

Clarity - 3 marks (based on viva during presentation)

Feasibility - 3 marks (based on project)

Presentation - 10 marks

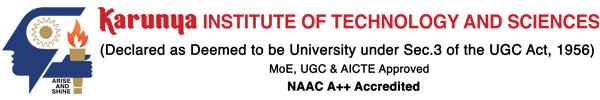
Project Report - 10 marks

On-time submission - 5 marks (before the due date)

Online submission-GCR - 5 marks

**Total marks: \_\_\_\_\_/ 40 Marks**

**Signature of Faculty with date:**

****

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO.** |
|  | **INTRODUCTION** | **3** |
|  | **DESCRIPTION OF THE PROJECT** | **3** |
|  | **CONCEPT INVOLVED** | **4** |
|  | **TOOLS** | **4** |
|  | **IMPLEMENTATION** | **5** |
|  | **RESULTS WITH GRAPH/SIMULATION** | **5** |
|  | **INFERENCES** | **6** |
|  | **CONCLUSION** | **6** |

**CHAPTER 1**

**INTRODUCTION**

In today's digital age, efficient management of image data is essential for various applications. Python provides robust tools for image compression, a technique vital for reducing image file sizes while maintaining acceptable visual quality. In this tutorial, we'll delve into building an image compressor using Python, utilizing well-known libraries such as Pillow or OpenCV. By undertaking this project, you'll gain a solid understanding of image processing fundamentals and compression methodologies, equipping you to handle images effectively in your Python endeavors.

**CHAPTER 2**

**DESCRIPTION OF THE PROJECT**

1. **Get the image:** Start by loading the image you want to compress.
2. **Reduce the file size:** Use Python to shrink the image's file size while keeping its quality as good as possible.
3. **Save the compressed image:** Once compressed, save the smaller image file.
4. **Enjoy the smaller file:** Now you have an image that takes up less space but still looks good.

**CHAPTER 3**

**CONCEPT INVOLVED**

1. **Using Tools:** Use tools like PIL or OpenCV to handle images in Python.
2. **Making Data Manageable:** Convert image data into a form Python can work with, often using arrays.
3. **Squeezing Data:** Use techniques to shrink image files, like reducing detail or applying specific compression methods.
4. **Squishing and Un-squishing:** Turn compressed data into a smaller file and then back into an image.
5. **Making Things Faster:** Speed up the process, especially for big batches of images.
6. **Handling Mistakes:** Deal with problems, like when an image doesn't work or compression fails.
7. **Checking and Tweaking:** Test the compressor with different images to make sure it works well without losing too much quality.

**CHAPTER 4**

**TOOLS**

1. **PIL (Python Imaging Library):** It helps open, edit, and save different image formats.
2. **OpenCV (Open Source Computer Vision Library):** It's great for working with images and videos, including compression tasks.
3. **NumPy:** This helps handle image data efficiently by treating it like numbers.
4. **Scikit-image**: It offers many tools for working with images, like making them smaller.
5. **imageio:** This library reads and writes various image formats, useful for handling different kinds of images**.**

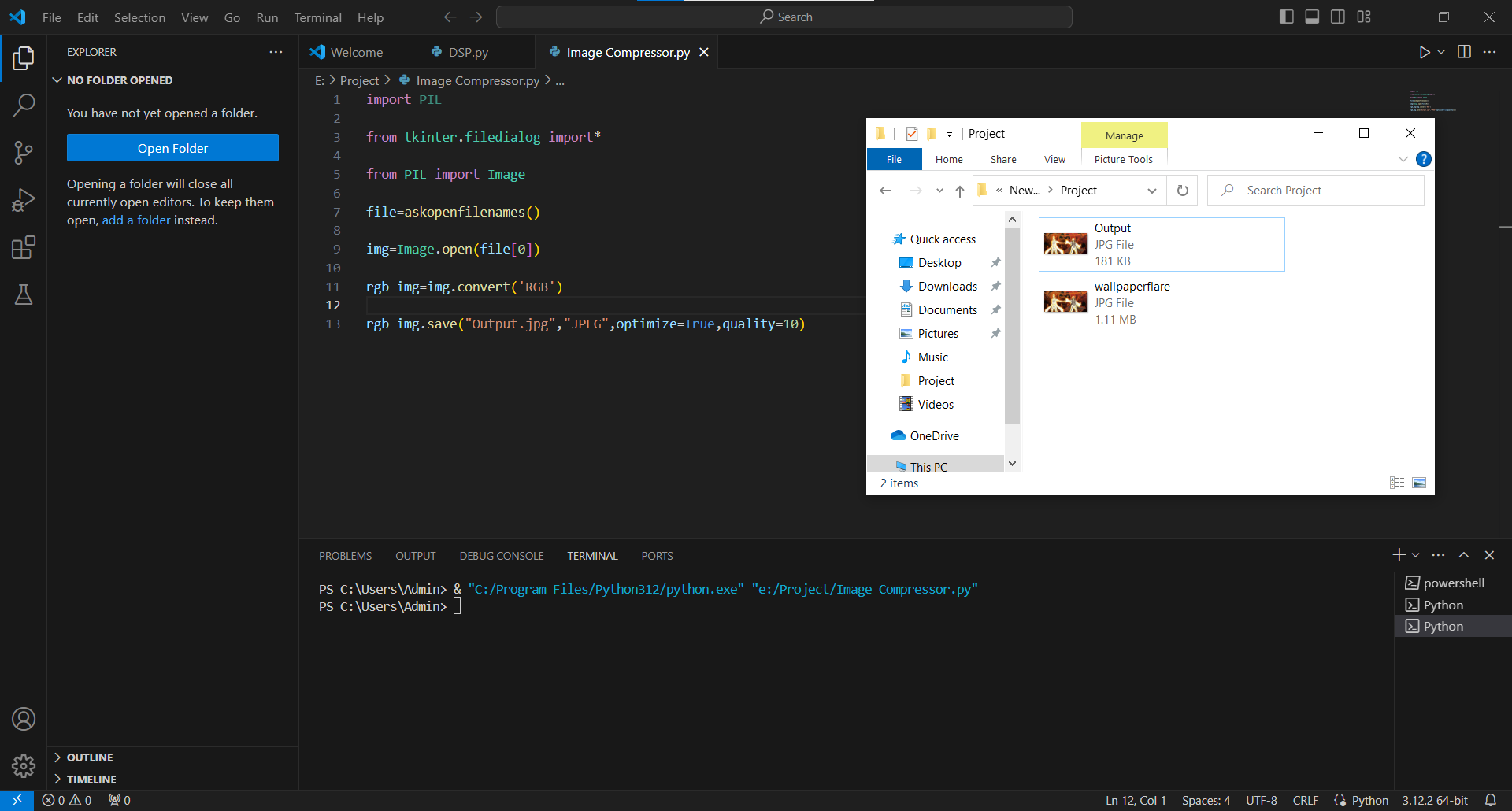
**CHAPTER 5**

**IMPLEMENTATION**

* **Step 1:** Install PIL. The first step is to install the PIL library.
* **Step 2:** Import the Required Libraries.
* **Step 3:** Load the Image.
* **Step 4:** Resize the Image.
* **Step 5:** Save the Compressed Image.
* **Step 6:** Verify the Compressed Image.

**CHAPTER 6**

**RESULTS WITH GRAPH/SIMULATION**



**CHAPTER 7**

**INFERENCE**

* <https://www.youtube.com/watch?v=GTZBAiIS9UE&list=WL&index=2&t=202s>
* <https://www.tutorialspoint.com/how-to-compress-images-using-python-and-pil>

**CHAPTER 8**

**CONCLUSION**

In conclusion, the Python-based image compressor efficiently reduces image size while preserving quality. It offers benefits like storage conservation and faster loading times for web pages. With ongoing refinement, it holds promise for diverse applications.