
No To Paid Software in the World of Photography

Eliezer Carvalho

eliezer.carvalho.11.8a@gmail.com

Abstract

Photography has become a constant presence in everyday life, accessible from mobile devices or dedicated cameras. With this evolution, the number of image editing and conversion tools available online has multiplied. Currently, the web offers a wide range of solutions, from converters for countless formats to software capable of performing advanced edits with just a few clicks. However, many of these tools are paid, especially when processing large quantities of images, making simple tasks unnecessarily expensive and tedious. The [Bulk Image Editor](#) and [Image Format Converter](#) project is a strong response to this problem, offering a foolproof solution that allows basic editing and conversion operations to be performed efficiently and free of charge.

1. Introduction

Driven by the evolution of mobile devices and the growing influence of social media, photography is an increasingly present phenomenon in our lives. Applications such as Instagram, TikTok, and other content-sharing platforms encourage intensive use of images, increasing the need to edit photographs and convert them between different formats quickly and easily. This reality affects both ordinary users and professionals, who are looking for simple solutions to optimise file quality and size.

In many scenarios, it is necessary to compress images to reduce their size without significantly compromising quality. On the other hand, there are also situations where conversion to heavier formats is necessary and crucial, increasing resolution or opening the

door to more advanced editing, which is common in the work of photographers and content creators. However, although these needs are common and recurring, most of the tools available online impose costs, usage limitations or watermarks, making simple tasks excessively restrictive and forcing users to resort to paid applications for simple image editing or conversion operations.

This project was created with the aim of addressing this problem. The main objective was to create two open source programmes designed to eliminate the need for paid solutions to perform basic editing and conversions between the most commonly used image formats. In addition, the programmes offer a ‘Bulk’ feature, which allows users to edit and convert multiple images simultaneously. This initiative aims to make image processing more accessible, practical and universal for all users.

2. Technical Definition

A digital image is the representation of a two-dimensional image using numerical values [1]. Each image is composed of millions of pixels, each containing colour information. In the most common colour systems, such as RGB (Red, Green, Blue), each colour channel is represented by a numerical value, encoded in binary, which when combined allows virtually any visible colour to be represented. Observing pixels at a close distance, such as when we approach a television screen or monitor, allows us to clearly understand this phenomenon.

Over the years, various digital image formats have appeared, each with its own characteristics. Among the most famous is TIFF [2], a format that allows images to be stored with or without compression, preserving high quality in exchange for greater memory usage. The JPG [3] file format (also known as JPEG) is one of the most widespread, particularly on mobile devices and the internet, due to its lossy compression, which reduces file size, even if there is some visual degradation. PNG [4], created in the 1990s, uses lossless compression, making it the perfect solution for images that require consistent quality.

Within the scope of the project in question, a fundamental format was also considered, although less widespread in the common context: the RAW format [5]. This designation represents a family of proprietary formats used by professional cameras in which the data captured by the sensor is stored in an almost raw form. This feature promotes the retention of detailed information about the photograph, including white balance, exposure and lens characteristics. This allows for a high level of post-editing. On the other hand, RAW files are considerably larger due to the wealth of data stored.

Image editing has clearly had a decisive influence on the evolution of digital photography, accompanying and driving its development over the years. In a context of the growing presence of digital photography in everyday life, editing has become a virtually indispensable process, used by ordinary users as well as professionals and content creators. Editing an image means giving the user a very high degree of creative control, allowing them to

transform ideas and aesthetic intentions into concrete photographic representations. The process can range from simple corrections, colour and light adjustments, to advanced manipulations that profoundly alter the visual content.

The software architecture of the project in question does not focus predominantly on providing advanced photo editing tools. However, it is important to highlight the recognition and preservation of the relevance of digital image editing and, more notably, video editing. Thus, the project offers users the opportunity to enjoy the potential of editing, combining it with practicality, speed, automation, and efficiency.

3. Project Architecture

This project originated from a particular circumstance: when attempting to perform basic photo editing, I encountered an excessive amount of paid software available on the internet, which made the process demotivating and inaccessible. Driven by this limitation, I decided to develop two applications with a graphical user interface (GUI), whose main objective would be simplicity, efficiency, and ease of use, eliminating the need for paid software once and for all.

With this purpose in mind, I began a research phase to identify the most suitable Python libraries for building the desired ecosystem. To organise this study and ensure a structured approach, the analysis was divided into three sections:

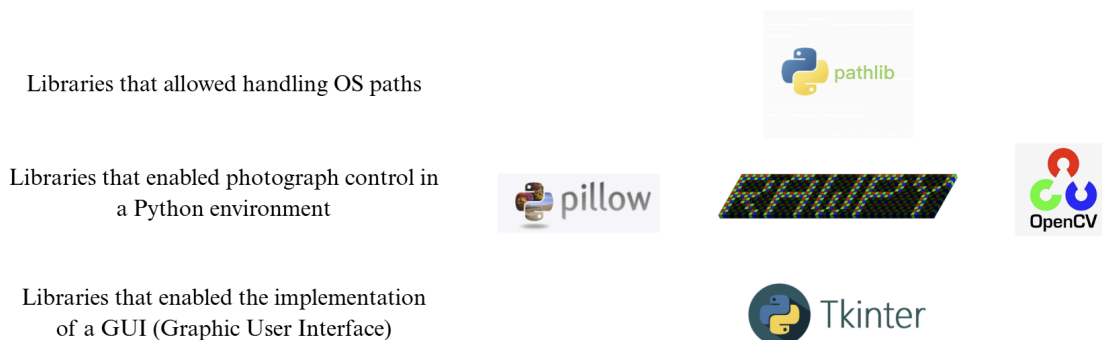


Figure 1

The use of the selected tools resulted in the creation of two files with the extension .py [6], forming a cohesive and functional ecosystem, contributing significantly to the success of the project. The separation of the source code by specific functions promotes good programming practices, improves the internal organisation of the project, and facilitates the reading, maintenance, and future evolution of the software. Additionally, the interface developed with Tkinter [7] is fully customisable by modifying the code, allowing for easy adjustment of the visual appearance, the behaviour of the graphic elements, and the overall user experience, making the project dynamic.

4. Modus Operandi

As mentioned earlier, the project consists of two files with the .py extension, carefully separated in order to optimise the understanding, maintenance and usability of the code. The first file, called BulkImageConverter.py, uses the standard Python library, pathlib [8], to allow the user to select a folder on their computer using Windows File Explorer. After selecting the folder, the user can select the desired format for converting the images using a combobox. In order to correctly identify and process existing files, several validation and security processes have been implemented. These mechanisms ensure that only supported formats are considered, avoiding surprises and code hallucinations. Of particular note is the use of the Rawpy library [9], which allows the handling and conversion of images from the RAW format family. This library was fundamental for the involvement of this family of such important formats. The conversion process is initiated with a single click. Easy. The software then automatically creates a new folder in the original path, where all photos converted to the selected format are saved in a customised folder.

The second file has an engineering structure similar to that of the first. In the second volume of the ecosystem, the BulkImageEditor.py application was designed for editing large quantities of images by selecting a folder from the device's local memory. The implementation of this software requires the use of three fundamental libraries. Firstly, the Pillow library [10] is used for manipulating photographs in various formats, namely reading, writing and basic image transformation. Secondly, OpenCV [11] was used, which, although geared towards computer vision tasks, was employed here as a mechanism for adjusting brightness and contrast. Finally, the Tkinter library was used to create the graphical components that allow direct interaction with the user. The modus operandi of both apps is quite similar, which contributes to intuitive and consistent use throughout the ecosystem. First, the user selects a reference image and sets the desired editing parameters, such as brightness and contrast. Then, the settings can be saved using a button provided by the Tkinter interface. The user then selects a folder in the system to which the previously stored settings will be applied. The software then edits all the photographs contained in the selected folder and saves the final set in a new folder, automatically generated by the programme.