User Manual - Transforming Images

Basic Information

Image Location:

To optimize the functionality of software, it is essential to organize all images to be processed within the designated 'Images' folder. Ensure that this folder is located alongside the program's code files.

Program Execution:
The software is run through a command prompt (CMD) in the Windows environment or a terminal in the Linux system. When accessing either of these environments, it is crucial to navigate to the folder containing the 'main.py' file to initiate the program execution.

REQUIREMENTS:

Installation of Necessary Libraries:
Before using software, it is mandatory to install the following libraries to ensure proper performance:

- Matplotlib
- Pillow

Installation Instructions:
To install each library, follow the steps below:

1. Open the command prompt (CMD) in the case of the Windows system or the terminal in Linux.
2. Navigate to the root folder of your project where the 'main.py' file is located.
3. Use the following command to install the desired library:

```
pip install <lib_name>
```

Make sure to replace `<lib_name>` with the specific name of the library being installed, such as 'matplotlib' or 'pillow'.

4. Press 'Enter' to initiate the installation process.

These steps will ensure that all necessary dependencies are correctly configured for the smooth operation of software. Be sure to follow these guidelines for a seamless user experience.

Running the Program

To execute software, there are two available approaches, both detailed below:

1. Execution via Terminal:
Open the terminal in the project's destination folder and enter the following command:

```
python main.py –image <name_image> --resize <width> <height>
--crop <x> <y> <width> <height> --rotate <angle>
```

Example of Parameter Input:

Let's assume we want to resize, crop, and rotate an image. The parameter input may look like the example below:

```
python main.py –image original_image.jpg --resize 800 600 --crop 100 100 600
400 --rotate 45
```

Make sure to replace `<name_image>`, `<width>`, `<height>`, `<x>`, `<y>`, and `<angle>` with specific values for your operation.

Explanation of Input Parameters:

When using the provided command in the above code, it is crucial to understand the meaning of each entered parameter:

- **image <name_image>:**
  This command loads the desired image to enable subsequent transformations of resizing, cropping, and rotation.
  `<name_image>` must be replaced with the exact name of the image located in the 'Images' folder, following the appropriate format, for example: Image01.jpeg.
- **resize <width> <height>:**
  The resizing parameter adjusts the image according to the specified width and height proportions by the user. Values can be larger, smaller, or equal to the dimensions of the original image. Replace `<width>` and `<height>` with the desired values, providing integers.
- **crop <x> <y> <width> <height>:**
  This parameter performs a crop on the image, starting at a point specified by the user `<x>` and `<y>` and extending to the desired width and height specified in the `<width>` and `<height>` parameters. All values must be provided as integers.
- **rotate <angle>:**
  By providing this parameter, the original image is rotated by the specified angle in `<angle>`. Values greater than zero will result in counterclockwise rotation, while values less than zero will cause clockwise rotation. Make sure to provide the angle in degrees, accepting floating-point values.

After entering all parameters and pressing 'Enter', the program will generate the resulting images for comparison, as illustrated in Figure 1. This process allows flexible manipulation of images according to the provided instructions.

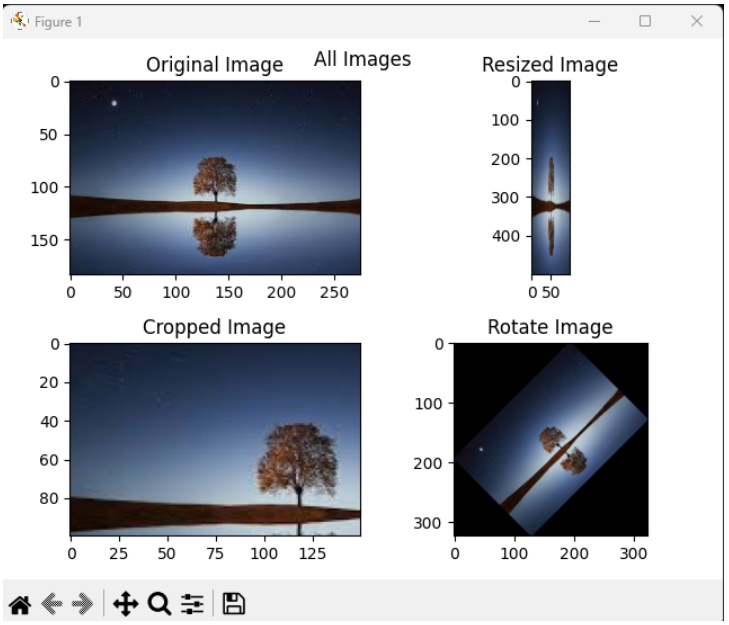

Figure 1

Error Handling:

It is important to note that if the user enters a parameter but does not provide a corresponding value, the program will identify this scenario as an error and will not function correctly. For example, entering a command like:

```
python main.py --resize
```

The program will report an error, indicating that values are required for the specified parameters.

However, if the user chooses not to provide the name of a parameter, the program will still run but enter a user-friendly interface mode. In this interface, the program will guide the user interactively, prompting the necessary parameters one by one and providing clear messages to guide the process.

This design aims to make the interaction with software more intuitive, even in situations where the user may not recall all required parameters or values. Follow the instructions provided in the interface for a smoother and more efficient user experience.

2. **User-Friendly Interface:**

   The User-Friendly Interface provides the user with greater control over the desired actions for each image, offering the flexibility to perform a single action or all actions simultaneously, similar to the previous example.

   To initiate the program through the User-Friendly Interface, the user simply needs to enter the following command:

   ```
   python main.py
   ```

   The first display that will appear is the W**elcome** screen, as illustrated in the Figure 2:

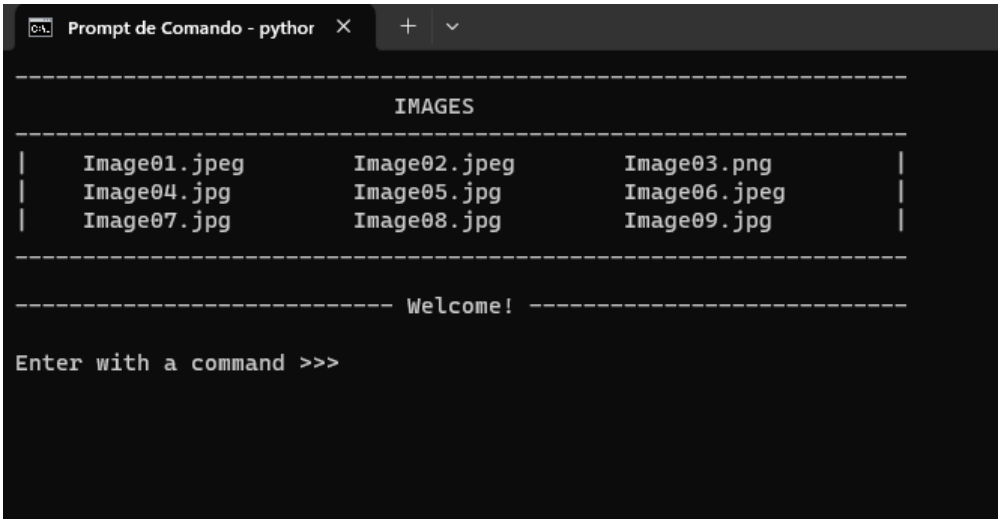

Figure 2

Upon observing the initial part of the interface, it is possible to identify the list of names of images located in the 'Images' folder. Subsequently, the program prompts the user to input a specific command. Below are the available commands:

- **HELP:** This command serves as an informative tool, presenting the user with a list of available commands, their descriptions, and the necessary parameters for their usage (when applicable). It is a useful tool for quickly recalling command names and obtaining detailed information about each one. The Figure 3 illustrates the appearance of the HELP command screen.

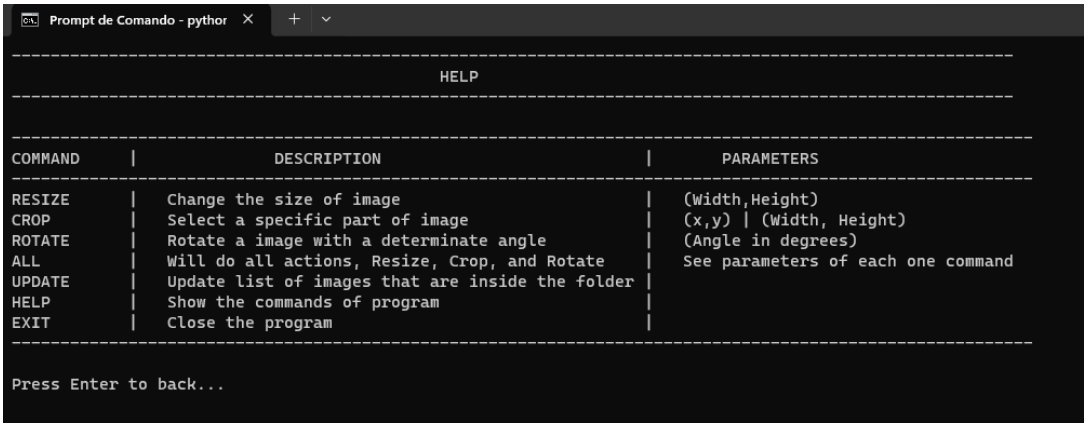

Figure 3

In addition to the HELP command, the user-friendly interface provides other specific commands to facilitate user interaction with the software:

- **UPDATE:** The UPDATE command allows the updating of image names in the 'Images' folder. If the user adds or removes images while the project is running, they can use this command to update the names without the need to close and restart the project.
- **EXIT:** This command terminates the program, providing a secure exit from the interface.

> 💡 After entering the RESIZE, CROP, ROTATE, or ALL commands, the user will be prompted to enter the name of the image they want to work with. Only after this input will the redirection to the specific page of the chosen command occur

- **RESIZE:** This command adjusts the proportions of the image according to the width and height values provided by the user. The values must be integers and separated by a comma. Example: `100, 300`.
- **CROP:** The CROP parameter crops the image, starting at a specific point (x, y) and extending to the desired width and height. Values for X, Y, width, and height must be integers and entered separately by commas.
- **ROTATE:** This command rotates the original image by the specified angle. Positive values indicate counterclockwise rotation, while negative values indicate clockwise rotation. The angle (<angle>) must be provided in degrees, accepting floating-point values.
- **ALL:** The ALL command executes the RESIZE, CROP, and ROTATE commands sequentially, presenting original and processed images. This provides a comprehensive view of the transformations performed.

> For additional questions, contact the product owner, who will be pleased to provide support and necessary clarifications.

Revision	Author	Version
A	*Anderson Cordeiro de Souza*	*v1.0.0*