

# Batch Image Resizer - CLI Tool

Firdaush Alam

July 2025

## 1 Introduction

This is a simple, practical CLI-based image resizing tool written in Python using the Pillow library. It begins with resizing for a single image, and then evolves to support batch resizing of images in a directory.

**Author:** Firdaush Alam

**Internship:** Python Developer Intern @ BroskiesHub

**Institution:** SECAB Institute of Engineering & Technology

## 2 Libraries Used

- `os` – for file and directory handling
- `argparse` – for CLI argument parsing
- `Pillow` (PIL) – for image operations

## 3 Code Structure

### 3.1 Function: `resize_single_image`

For understanding purpose, resizing of a single image is handled first.

```
def resize_single_image(source_path, destination_path, target_width, target_height):
    try:
        with Image.open(source_path) as original:
            resized = original.resize((target_width, target_height), Image.Resampling.LANCZOS)
            resized.save(destination_path)
            print(f"Saved resized image to: {destination_path}")
    except FileNotFoundError:
        print(f"Image is not available at the path: {source_path} Check & Verify")
    except Exception as err:
        print(f"There is some error in resizing the Image: {err}")
```

### 3.2 Function: `resize_images_in_directory`

All images are resized using the above function inside a loop. Supported formats include `.png`, `.jpg`, and `.jpeg`.

```
def resize_images_in_directory(source_dir, destination_dir, width, height):
    if not os.path.exists(destination_dir):
        os.makedirs(destination_dir)

    supported_extensions = (".png", ".jpg", ".jpeg")
    total = 0

    for item in os.listdir(source_dir):
```

```

        if item.lower().endswith(supported_extensions):
            original_path = os.path.join(source_dir, item)
            filename, _ = os.path.splitext(item)
            new_filename = f"{filename}_resized.jpg"
            save_path = os.path.join(destination_dir, new_filename)
            resize_single_image(original_path, save_path, width, height)
            total += 1

    print(f"\nThe resizing of {total} image(s) of your
    directory:{source_dir} is completed and stored at the location:{destination_dir}")

```

### 3.3 CLI Execution Block

```

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Resize
    images in bulk! Just provide source & target folder, plus width and height.")
    parser.add_argument("source_folder", type=str, help="Folder containing images you want to resize")
    parser.add_argument("destination_folder", type=str, help="Folder to save resized images into")
    parser.add_argument("width", type=int, help="Desired width in pixels")
    parser.add_argument("height", type=int, help="Desired height in pixels")

    args = parser.parse_args()

    resize_images_in_directory(
        source_dir=args.source_folder,
        destination_dir=args.destination_folder,
        width=args.width,
        height=args.height,
    )

```

## 4 Time Complexity

- `resize_single_image`:  $O(1)$
- `resize_images_in_directory`:  $O(n)$  for  $n$  image files

## 5 Limitations

- No aspect ratio preservation – fixed dimensions are used
- Only JPG, JPEG, PNG formats are accepted
- Converted output is always JPEG
- Unsupported image modes (e.g. RGBA) cause errors unless converted

## 6 Sample Execution

Here's a screenshot showing a sample run and its output.

```
C:\Windows\System32\cmd.exe x + v
C:\Users\Firdaush Alam\OneDrive\Desktop>python Image_Resizer.py A:\portfolio\images Resized_image_output 300 400
Saved resized image to: Resized_image_output\1424906_resized.jpg
Saved resized image to: Resized_image_output\4142747_resized.jpg
Saved resized image to: Resized_image_output\5277951_resized.jpg
Saved resized image to: Resized_image_output\8163-dark-code-wallpaper_resized.jpg
There is Some error in resizing the Image: cannot write mode P as JPEG
There is Some error in resizing the Image: cannot write mode RGBA as JPEG
Saved resized image to: Resized_image_output\afshin-t2y-3_PVkJCcXqg0-unsplash_resized.jpg
Saved resized image to: Resized_image_output\background_resized.jpg
There is Some error in resizing the Image: cannot write mode RGBA as JPEG
Saved resized image to: Resized_image_output\modern-empty-business-corporate-office-meeting-room-is-ready-businesspeople-late-night_r
esized.jpg
Saved resized image to: Resized_image_output\profile_resized.jpg

The resizing of 11 image(s) of your directory:A:\portfolio\images is Completed and stored at the location: Resized_image_output
C:\Users\Firdaush Alam\OneDrive\Desktop>
```

```
python Image_Resizer.py A:\portfolio\images Resized_image_output 300 400
Saved resized image to: Resized_image_output\1424906_resized.jpg
...
Saved resized image to: Resized_image_output\profile_resized.jpg
The resizing of 11 image(s) of your directory:
A:\portfolio\images is Completed and stored at the
location: Resized_image_output
```

## 7 Enhancement Ideas

- Use `convert("RGB")` to handle non-JPEG modes
- Add option to preserve aspect ratio
- Integrate a progress bar using `tqdm`
- Optionally retain original format (JPG/PNG)

The End