# Stitch: Python Program for Image Stitching

Stitch is a Python program for image stitching. The program uses default values for most options; the following options are configurable:

- `confidence_threshold`: Confidence level for selecting image matches. Default: `0.5`.
- `try_use_gpu`: Boolean to indicate if GPU should be used. Default: `True`.
- `final_megapix`: Resolution for final stitched image (megapixels). Default: `5`.
- `crop`: Boolean to indicate if cropping should be performed. Default: `False` (not changeable).
- `warper_type`: Type of image warper. Default: `plane` (not changeable).
- `detector`: Feature detector type. Default: `sift` (not changeable).

To change any of these options, add them to the INI file `stitch.ini`:

```
[OPTIONS]
img_dir = ./img
out_dir = ./result
final_megapix = 5
try_use_gpu = True
confidence_threshold = 0.5
output = output
```

# Precondition for Each Program Run

## Directory Structure

1. **Input Directory (`img_dir`, change default in `stitch.ini`)**
    - Path: `./img`
    - Content: JPEG files (`.jpg`) to be stitched into a panorama.
2. **Output Directory (`out_dir` and `output`, change default in `stitch.ini`)**
    - Path: `./result`
    - Filenames produced: `output.jpg` (stitched image before postprocessing) and `outputfixed.jpg` (stitched image after border-fix postprocessing).

Ensure the directories exist and contain the required files before running the program.

# Functionality of the Program

1. **Logging Configuration**
    - Logs to console and to `stitch.log`.
    - Removes the log file if it exceeds 10 MB.
2. **Configuration Reading**
    - Reads options from `stitch.ini` and applies defaults when needed.

3. **Directory and File Handling**
    - Converts relative input/output paths to absolute paths and logs them.
    - Lists input files and logs their full paths.
4. **Image Stitching**
    - Initializes a `Stitcher` object with configured parameters.
    - Uses the stitcher to create a panorama and saves the result.
5. **Post-Processing for Border Removal**
    - Replaces black border pixels with a mean color sampled from non-black border pixels.
    - Saves the post-processed image as `outputfixed.jpg`.

# Libraries Used

- `os` for file and directory operations.
- `numpy` for numerical operations.
- `cv2` (OpenCV) for image processing.
- `logging` for logging messages.
- `configparser` for reading configuration files.

# Installation

The ZIP archive `stitch.zip` contains a working directory and sample data.

- `stitch.exe`: Standalone executable generated with PyInstaller:
    - Build command example: `pyinstaller --onefile stitch.py`
- `readme.md`: This file.
- `stitch.py`: Python source code (just for documentation).
- `lfstitcher.py`: Python source code (just for documentation).
- `stitch.ini`: Configuration file (example).

# Usage

## Capture Input Images

Capture overlapping images covering the subject. Keep the following as consistent as possible:

- Same focus plane (small deviations tolerated)
- Same exposure (automatic correction applied to some degree)
- Same color balance

## Create Stitched Image

1. Remove any existing images in `img` and `refimg` if present.

2. Copy captured images into `img`.
3. Run `stitch.exe`.

Example: Metzgeria furcata (L.) Dumort.