

# “IMAGE PREPROCESSING PIPELINE”

Theory:

An **image preprocessing pipeline** refers to a series of sequential steps or processes applied to an image to prepare it for further analysis or use. This pipeline is essential in various fields like computer vision, machine learning, and image analysis, where raw image data needs to be cleaned, enhanced, or transformed before being fed into algorithms or models.

## Components of an Image Preprocessing Pipeline

1. **Image Acquisition:**
2. **Image Enhancement:**
3. **Image Transformation:**
4. **Image Segmentation:**
5. **Image Conversion:**
6. **Feature Extraction:**
7. **Image Saving:**

## Benefits

- **Data Quality**
- **Algorithm Efficiency**
- **Feature Extraction**
- **Consistency.**

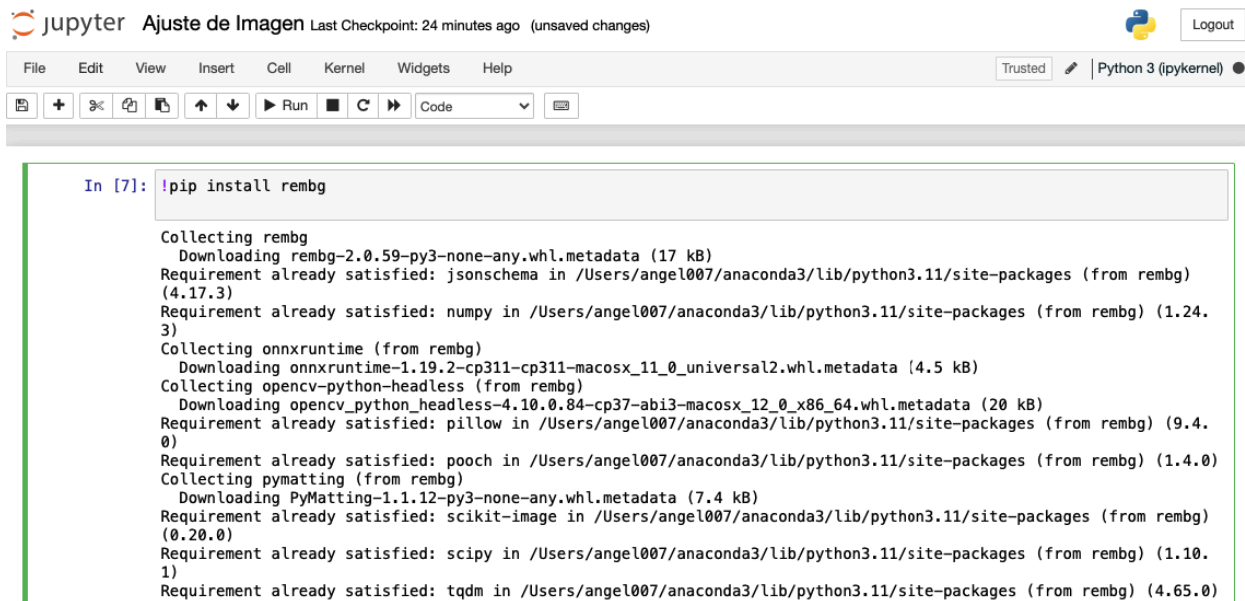
## Applications

- **Machine Learning:** Preprocessing is crucial before feeding images into machine learning models to improve performance and accuracy.
- **Medical Imaging:** Enhancing and preparing medical images for diagnosis or analysis.
- **Object Recognition:** Preparing images for algorithms that identify and classify objects.

# includes:

1. **Loading** → 2. **Background Removal** → 3. **Resizing** → 4. **Grayscale Conversion** → 5. **Saving.**

Each of these steps aligns with common techniques in image processing and computer vision, used to prepare and adjust images for various applications.



The image shows a Jupyter Notebook interface with the title 'Ajuste de Imagen'. The top bar indicates 'Last Checkpoint: 24 minutes ago' and 'unsaved changes'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The main area displays a code cell with the command 'pip install rembg' and its output. The output shows the collection and downloading of rembg and its dependencies, including onnxruntime, opencv-python-headless, pillow, pymatting, and scikit-image.

```
In [7]: !pip install rembg

Collecting rembg
  Downloading rembg-2.0.59-py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: jsonschema in /Users/angel007/anaconda3/lib/python3.11/site-packages (from rembg) (4.17.3)
Requirement already satisfied: numpy in /Users/angel007/anaconda3/lib/python3.11/site-packages (from rembg) (1.24.3)
Collecting onnxruntime (from rembg)
  Downloading onnxruntime-1.19.2-cp311-cp311-macosx_11_0_universal2.whl.metadata (4.5 kB)
Collecting opencv-python-headless (from rembg)
  Downloading opencv_python_headless-4.10.0.84-cp37-abi3-macosx_12_0_x86_64.whl.metadata (20 kB)
Requirement already satisfied: pillow in /Users/angel007/anaconda3/lib/python3.11/site-packages (from rembg) (9.4.0)
Requirement already satisfied: pooch in /Users/angel007/anaconda3/lib/python3.11/site-packages (from rembg) (1.4.0)
Collecting pymatting (from rembg)
  Downloading PyMatting-1.1.12-py3-none-any.whl.metadata (7.4 kB)
Requirement already satisfied: scikit-image in /Users/angel007/anaconda3/lib/python3.11/site-packages (from rembg) (0.20.0)
Requirement already satisfied: scipy in /Users/angel007/anaconda3/lib/python3.11/site-packages (from rembg) (1.10.1)
Requirement already satisfied: tqdm in /Users/angel007/anaconda3/lib/python3.11/site-packages (from rembg) (4.65.0)
```

## Use:

Rock-Paper\_Scissors Image classification

Source: Kaggle

Link: <https://www.kaggle.com/datasets/drgfreeman/rockpaperscissors>

**AUTHOR: Julien de la Bruère-Terreault (drgfreeman@tuta.io)**

LICENSE: CC-BY-SA 4.0

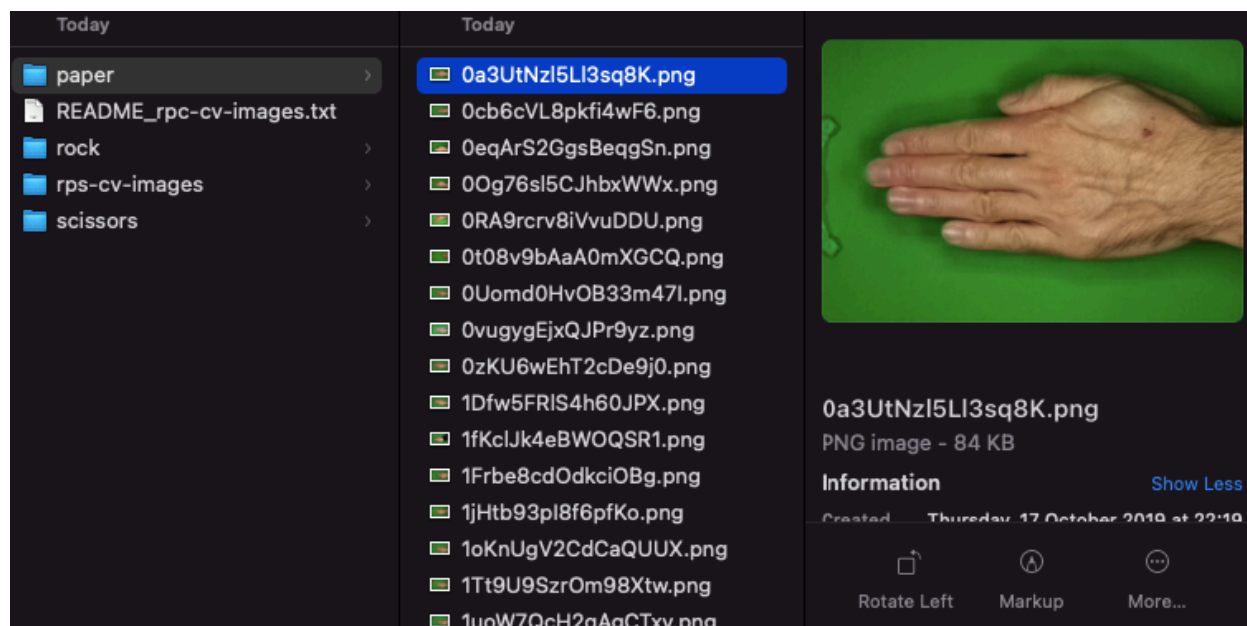
DESCRIPTION: This dataset contains images of hand gestures from the Rock-Paper-Scissors game. The images were captured as part of a hobby project where I developed a Rock-Paper-Scissors game using computer vision and machine learning on the Raspberry Pi (<https://github.com/DrGFreeman/rps-cv>)

CONTENTS: The dataset contains a total of 2188 images corresponding to the 'Rock' (726 images), 'Paper' (710 images) and 'Scissors' (752 images) hand gestures of the Rock-Paper-Scissors game. All image are taken on a green background with relatively consistent lighting and white balance.

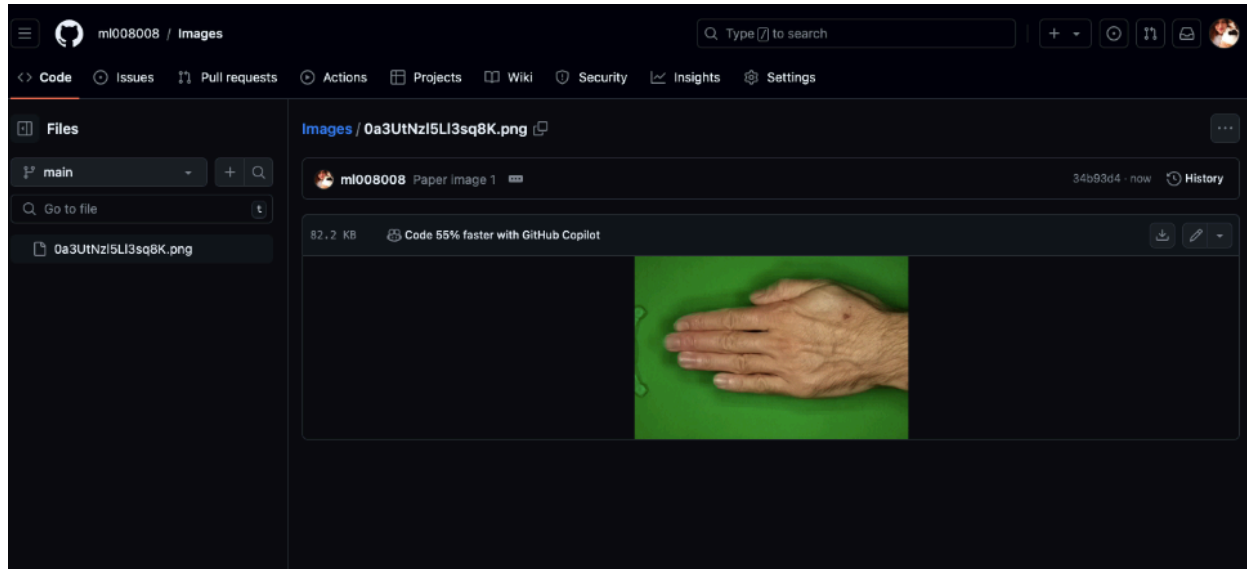
FORMAT: All images are RGB images of 300 pixels wide by 200 pixels high in .png format. The images are separated in three sub-folders named 'rock', 'paper' and 'scissors' according to their respective class.

Using:

Test\_Image: 0a3UtNzl5Ll3sq8K.png



GitHub images:



**Public Permalink:**

<https://github.com/ml00/Images/blob/34b93d40188f75544ed8c6548fb528f1eabc0750/0a3UtNzl5Ll3sq8KK.png>

archive.zip file

\*\*\*\*\*

LETS GO!