# Image Processing

## Homework 1: Connected Component Analysis & Color Correction

Due Date: 2025/09/27 23:59

## Introduction

This homework focuses on implementing image analysis algorithms for segmenting different objects in images and color correction algorithms.

You will implement these algorithms from scratch using only Numpy and the Python standard libraries. The goal is to understand how these algorithms work and how they affect image quality.

## Implementation (70%)

### Task 1: Connected Component Analysis (40%)

In this task, you are asked to implement two connected component analysis (CCA) algorithms to identify and label the cells in a given image. Two types of approaches and connectivities should be implemented:

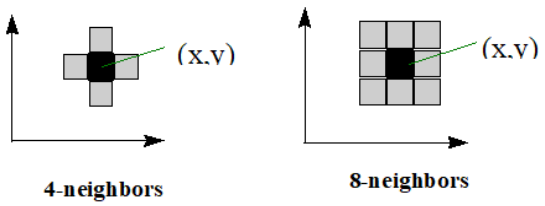
- Implement the **Two-pass Algorithm (20%)**
  - 4-connectivity (10%), 8-connectivity (10%)
- Implement the **Seed-filling Algorithm (20%)**
  - 4-connectivity (10%), 8-connectivity (10%)
- Please explain these two methods and compare the results in the report.

**Bonus (5%): Implement any other connected component analysis methods**

Please explain the method used and compare it to the two connected component analysis methods mentioned above in the report.

You can refer to p.96-98 of the lecture 2 slides on "digital image fundamental" or search online for more details. Below are some implementation hints:

- For **connectivity**, 4-connectivity considers a pixel's four direct neighbors: above, below, left, and right. 8-connectivity expands this by including the four diagonal neighbors, totaling eight adjacent pixels.

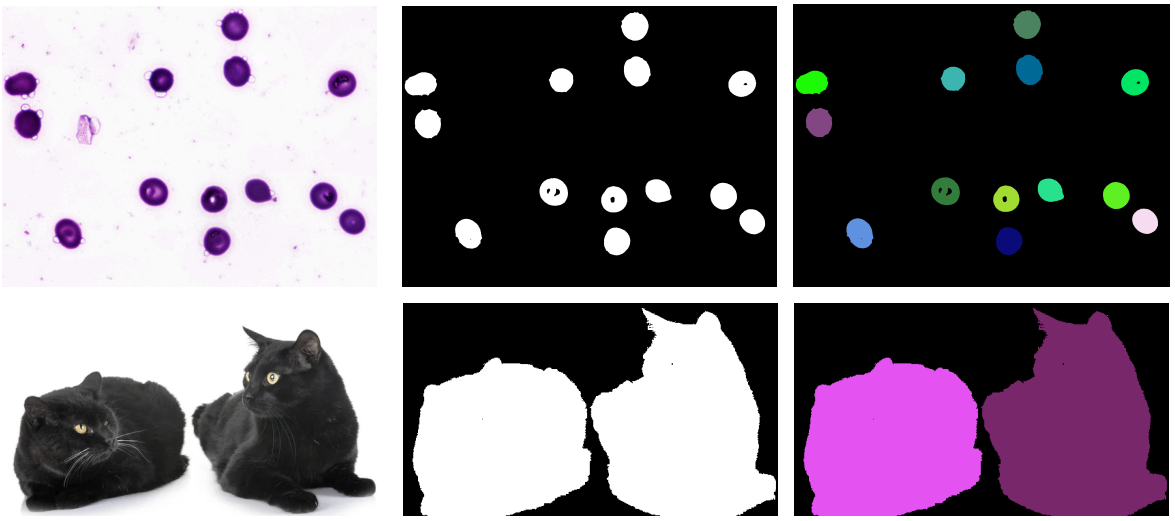



- For the **Two-pass Algorithm,** you can use a data structure like **union-find** to resolve equivalences between labels.
- For the **Seed-filling Algorithm,** you can use a data structure like **queue** or **stack** to add unvisited neighbors and assign the same label.

(These are suggestions for your reference; you are not required to follow them exactly.)

Moreover, to improve object segmentation and visualization, you may apply image processing techniques such as **removing/filling small regions** and **assigning colors to different labels**. Note that the colors do not need to match those in the examples, but the visualization should be clear and easy to interpret.

Examples:



origin binary labeling

## Task 2: Color Correction (30%)

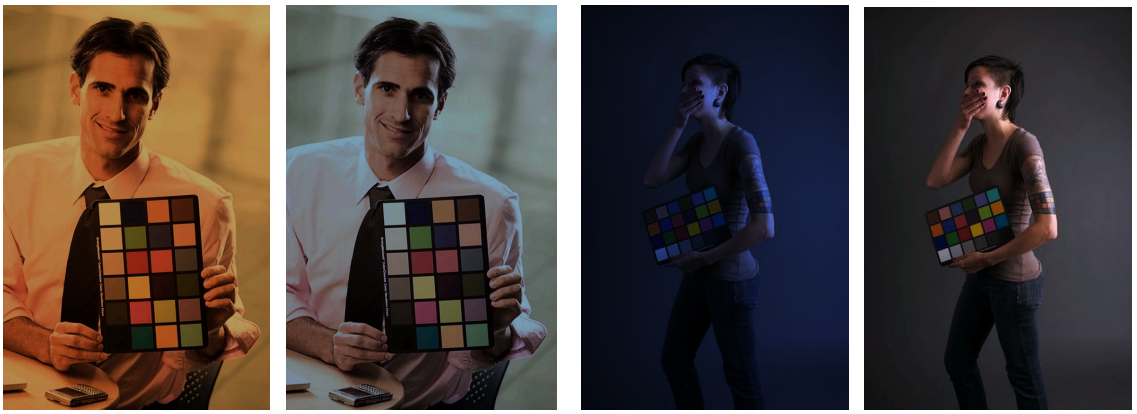
In this task, you should modify the color temperature of the input images to recover from the incorrect white balance of the given image. You can use the ColorChecker (the color palette) as a reference to determine if it's correct.

- Implement the **White Patch Algorithm (15%)**
- Implement the **Gray-world Algorithm (15%)**
- Please explain these two methods and compare the results in the report.

**Bonus (5%): Implement any other color correction methods**

Please explain the methods used and specify improvements in the image quality compared to the two white balance methods mentioned above in the report.

Examples:



origin result origin result

## Requirements

- You may only use OpenCV, Numpy, and other Python standard libraries for the implementation.
- Do not directly call packages to execute the entire algorithm; otherwise, you will receive no points.

# Report (30%)

- You should write your report following the report template.
- The report should be written in **English**.
- Please save the report as a **.pdf** file.

# QA Page

If you have any questions about this homework, please ask them on the following Notion page. We will answer them as soon as possible. Additionally, we encourage you to answer other students' questions if you can.

https://www.notion.so/tonychi123/HW1-QA-Sheet-2704c97a402280df9412c19ab61b164f?source=copy_link

# Submission

**Due Date: 2025/09/27 23:59**

Please compress all your code files and report (.pdf) into {student_id}_hw1.zip.

The file structure should look like:

```
📂{student_id}_hw1.zip
  ├ 📜 connected_component.py
  ├ 📜 color_correction.py
  ├ 📜 (other code files if you have)
  └ 📜 report.pdf
```

**Wrong submission format leads to -10 points.**

**20% off per late day.**