Solving Nested Rectangles Detection Problem

Introduction: This document provides an overview of the solution to the nested rectangles detection problem using both Python and C++ implementations.

Python Solution: sol_nested_rectangle.py

Description: This script detects nested rectangles in an image using OpenCV in Python.

Usage: Run the script in a conda environment with OpenCV installed. Use the command "python sol_nested_rectangle.py" to execute the script.

Output: The script identifies nested rectangles in the input image, highlights them, and labels them with their level of nesting. The processed image is saved in the "output_rectangles" folder.

C++ Solution: sol_nested_rectangle.cpp

Description: This C++ program detects nested rectangles in an image using OpenCV.

Compilation: Compile the program using the g++ compiler with OpenCV libraries linked. Use the following command:

```
bash$ g++ -fdiagnostics-color=always -g sol_nested_rectangle.cpp -o sol_nested_rectangle -I/usr/local/include/opencv4 -L/usr/local/lib -lopencv_core -lopencv_imgproc -lopencv_highgui -lopencv_imgcodecs -pthread
```

Execution: Run the compiled executable using the command "./sol_nested_rectangle".

Output: Similar to the Python solution, the C++ program identifies nested rectangles, highlights them, and labels them with their level of nesting. The processed image is saved in the "output_rectangles" folder.

Conclusion: Both the Python and C++ solutions successfully detect nested rectangles in images and provide the level of nesting for each rectangle.