

HW5B

CS103 Fall 2025 Unan

due: 11.23.2025 Sunday 11:59pm

*** Individual Work Only

*** AI Tools Allowed (with Documentation and Screenshots)

There are two different versions of Homework 5. **HW5A** focuses on *team collaboration and traditional problem solving*, while **HW5B** focuses on *individual work with responsible AI assistance*. You may complete either **HW5A** or **HW5B**, but not both. Please make sure that you **do not use any AI tools if you choose HW5A**, as it is designed to evaluate your teamwork and independent coding skills. Conversely, if you choose **HW5B**, you must **work individually** and follow the AI usage and documentation rules described in that assignment.

Objectives

In this assignment, you will:

- Learn to process and analyze images using OpenCV in Python.
- Use AI tools (such as ChatGPT, Copilot, Gemini, or others) responsibly to assist you in developing small parts of the program.
- Practice step-by-step problem solving, testing, and reflection on AI-assisted programming.

IMPORTANT: AI can help you with specific coding steps or debugging, but if you try to make AI “write the entire homework,” it will not meet all requirements and may fail to integrate properly. You are expected to understand and verify every part of your code.

Setup Instructions

- Create a directory named **HW5_ALT** under your **cs103fa25** folder.
- Inside this directory, create a Python file named **hw5b.py**.
- Add the following mandatory identification functions:

```
def myName():
    return "James Bond"
def myBlazerID():
    return "jbon12"
# Call these functions
print("My Name is =", myName(), " and my BlazerId is =",myBlazerID())
```

- **Function Definition:**
 - For each question, define the appropriate function and include calls to test your functions with the provided test cases.
 - Include the mandatory functions `myName()` and `myBlazerID()` that return your name and Blazer ID (as in previous assignments).
- **Docstrings:**
 - Create docstrings for each function. Missing docstrings will incur a penalty of 2 points each.
- **Submission:**
 - Submit the following three files on Canvas:
 - `hw5b.py` - Your fully working Python script.
 - `ai_usage.pdf` - A short report (1-2 pages) containing:
 - A brief explanation of how you used AI tools in this assignment.
 - Which prompts you used and for which parts.
 - Screenshots of at least three AI prompts and their responses.
 - A short reflection (4-5 sentences) describing:
 - What kind of help AI provided.
 - What needed debugging or corrections.
 - What you learned from using AI.
 - Screenshots should clearly show both your prompt and the AI's answer.
 - Submissions without screenshots will lose up to 10 points.
 - [Independent Completion Form \(ICF\)](#)

Part 1: Image Statistics (30 points)

Write a function `summarize_image(image_path)` that:

- Accepts an image file path (e.g., "cat.jpg").
- Returns a dictionary containing:
 1. "width" – image width
 2. "height" – image height
 3. "channels" – number of color channels
 4. "mean_color" – average RGB color as a tuple of three integers
 5. "brightness" – average brightness (mean of all pixel intensities)

Example Call

```
result = summarize_image("flowers.jpg")
print(result)
```

Expected Output (approximate):

```
{'width': 800, 'height': 600, 'channels': 3, 'mean_color': (120,
100, 90), 'brightness': 103.5}
```

Part 2: Apply Filters (30 points)

Write a function `apply_filters(image_path)` that:

- Loads the image.
- Converts it to grayscale.
- Applies a **blur filter** and an **edge detection filter (Canny)**.
- Displays all three images (`original`, `grayscale`, and `edge-detected`) in separate windows.
- Saves the grayscale and edge-detected images to disk (e.g., `gray.jpg` and `edges.jpg`).

Part 3: Object Counting (30 points)

Write a function `count_objects(image_path)` that:

- Converts the image to grayscale.
- Applies thresholding to separate objects from the background.
- Uses `cv2.findContours()` to detect objects. (*feel free to explore alternative ways*)
- Returns the number of detected objects.

Bonus [+10 Points]:

Draw a bounding box around each detected object using `cv2.rectangle()` and display the result.

Example Output

Detected objects: 7

AI Reflection (in `ai_usage.pdf`) (10 points)

In your PDF, include a short reflection answering the following:

- Which AI tools did you use, and for what parts?
 - Here is the UAB approved AI tools: <https://www.uab.edu/ai/tools>
- What was the most useful AI suggestion?
- What didn't work as expected?
- How did you debug or improve the AI's output?
- What did you learn about programming with AI assistance?

Include screenshots of your prompts and AI responses.

Grading Rubric

| Component | Points |
|---------------------------------|-----------------------------|
| <code>summarize_image()</code> | 30 |
| <code>apply_filters()</code> | 30 |
| <code>count_objects()</code> | 30 |
| Bonus: Bounding boxes | +10 |
| AI usage PDF (with screenshots) | 10 |
| Total | 110 (100 + 10 bonus) |

Academic Integrity Policy

This is an AI-assisted assignment, not an AI-generated one.

You must:

- Understand every part of your submission.
- Be able to explain your code if asked.
- Provide genuine AI usage evidence with screenshots.

If your submission appears fully auto-generated, incomplete, or inconsistent with your own coding style, it may result in a penalty or loss of credit.

Suggested Step-by-Step AI Workflow

1. Ask AI how to load and display an image using OpenCV.
2. Ask AI how to get image dimensions and compute averages using NumPy.
3. Ask AI how to apply filters like grayscale, blur, and edge detection.
4. Ask AI how to count objects using contours.
5. Combine the code parts, test them, and write the report.