Bonus Homework 01: Document Scanner

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-10-13

Due: 2025-10-22, 11:59pm, on Canvas

General Instructions:

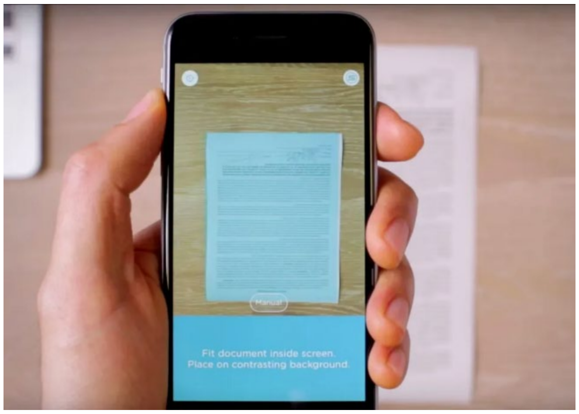
- You should solve this homework and submit your report **individually**. Identical submissions will receive a grade of zero.
- For bonus assignments, getting help from other students is okay and encouraged, but the **TAs** are **not** required to help you.
- Ask any questions on **Ed Discussion** (instead of emailing).
- The deliverables are outlined for each problem, and you should carefully **follow the instructions**. Failing to follow instructions will result in **points being subtracted**.
- Follow the instructions specified by each problem to submit the homework. Upload your **code** or other deliverables as instructed.
- **Late submission:** Late or missed submission will not be accepted. Any excused absence must be documented and disclosed to the instructor (extensions will be granted on a case-by-case basis).

Bonus Homework 01: Document Scanner

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-10-13

Due: 2025-10-22, 11:59pm, on Canvas



Bonus Problem 1 (1.5pts) – In this problem you will build the first stage of a ***document scanner*** that can automatically detect the boundaries of a document in an image. Write a Python program that

1. Takes as input an image called input.jpg containing a document (e.g., a piece of paper on a table).
2. Automatically detects the four corners of the document even if
   - The image is rotated or captured from an angle (i.e., perspective distortion)
   - The document is not centered in the frame
3. Draws lines around the detected document boundary for visualization.
4. Returns the ***(x, y)*** pixel coordinates of the four document corners, in the following order:
   - Top-left
   - Top-right
   - Bottom-right
   - Bottom-left

Deliverables:

- A Python script (e.g., document_scanner.py) that performs the above steps on an input image, called input.jpg, that is located in the same directory as the Python script.
- The script should
  - ***Display*** the input image with the detected boundary drawn on it.
  - ***Print*** out the 4 corner coordinates in the required order.

Notes:

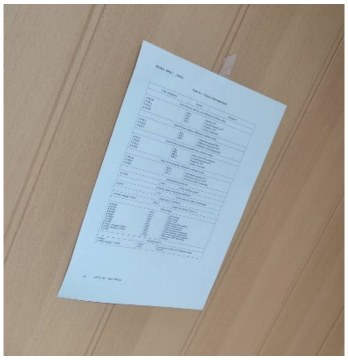
- You are encouraged to research online for solution ideas, algorithmic approaches, and for example codes that can help you design your document boundary detection system.
- You are free to decide how to implement your algorithm (there is no single required method). For example, you may use techniques such as edge detection, Hough line detection, or corner detection.
- However, you ***must not*** use any pre-built or high-level document scanner or document boundary detection package (e.g., OpenCV's cv2.scan or similar third-party libraries). The goal is to implement the boundary detection logic yourself using basic computer vision tools.
- Your code ***must run in our course's*** cv ***Conda environment***—we will test it and grade you based on that. A code that errors out will not receive a grade. So, it is crucial that that your code can correctly load the image called "***input.jpg***" in the same directory.
- You are expected to debug and solve any implementation issues that arise during development. This is part of the learning process for this problem.
- The code should be able to handle a wide range of input images with different orientations and lighting conditions, e.g., and image like

Bonus Homework 01: Document Scanner

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-10-13

Due: 2025-10-22, 11:59pm, on Canvas



- We will test the robustness of your algorithm by feeding it challenging images such as the one above. Your code should be robust in order to receive full score.
- You do ***not*** need to perform perspective correction (this will be done in a later problem). Only output what is asked of you in the deliverables.

Running the program on your phone:

If you wish to test your program on a smartphone, you can use Apps such as

- **Android:** Pydroid 3
- **iPhone:** Pythonista or Pyto

You can then use a photo you take with your phone as the input image.