The University of Melbourne
School of Computing and Information Systems
COMP90086 Computer Vision, 2023 Semester 2

Assignment 3: Image inpainting

Due:	7pm, 13 Oct 2023
Submission:	Source code (in Jupyter Notebook) and written responses (as .pdf)
Marks:	The assignment will be marked out of 7 points, and will contribute 7% of your total mark.

In this assignment, you will design and improve algorithms for inpainting (filling the gaps in) an image.

1. Existing Code: reconstruction error analysis [2 pt]

Run the existing Jupyter Notebook and observe how well (or badly) it reconstructs the image. Try both images and varying locations for the mask. For each image choose 5 different mask locations and compute the rms (root mean square) error for the reconstruction at each mask location. Calculate the average of these rms errors for each image. In your written report, give the mask locations and rms errors you observe.

2. Context size and shape [2 pt]

The sample code uses a 5x5 window to calculate the value for the pixel in the bottom right corner. Context pixels marked O, target pixel marked X:

OOOOO
OOOOO
OOOOO
OOOOO
OOOOX

Try varying the shape and size of this context window to see how the quality of the reconstruction varies. Describe the reconstruction qualitatively as well as giving the quantitative reconstruction error. Time the algorithm to see how the context window affects the running time.

In your written report, describe at least 5 context window shape/sizes and describe their qualitative and quantitative performance on each of the two images.

3. Reconstruction design [3 pt]

Modify the design of the inpainting algorithm and try to improve the quality of the reconstructions both qualitatively and quantitatively. Some things to consider include:

- Try different context shapes (change the definition of make_context_locations)
- Context pixels don't have to be adjacent
- Maybe top down isn't the right order to fill things in?
- Maybe sum of squared differences isn't the right way to find the best pattern?
- Maybe you could randomly sample from the values weighted by pattern difference?

In your written report, describe your design choices and explain why you chose them and how you tested their effectiveness.

Submission

You should make two submissions on the LMS: your code and a short written report explaining your method and results. The response to each question should be no more than 400 words. Submission will be made via the Canvas LMS. Please submit your code and written report separately under the Assignment 2: Code and the Assignment 2: Report links on Canvas.

• Your code submission should include the Jupyter Notebook (please use the provided template) with your code and any additional files we will need to run your code, if any (do not include the Image files).
• Your written report should be a .pdf with your answers to each of the questions. The report should address the questions posed in this assignment and include any images, diagrams, or tables required by the question.

Evaluation

Your submission will be marked on the correctness of your code/method, including the quality and efficiency of your code and the appropriateness of design decisions. You should use built-in Python functions where appropriate and use descriptive variable names. Your written report should clearly explain your approach and any experimentation used to produce your results, and include all of the specific outputs required by the question (e.g., images, diagrams, tables, or responses to sub-questions).

Late submission

The submission mechanism will stay open for one week after the submission deadline. Late submissions will be penalised at 10% of the total possible mark per 24-hour period after the original deadline. Submissions will be closed 7 days (168 hours) after the published assignment deadline, and no further submissions will be accepted after this point.

Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on the LMS.