COMP/ENGN 4528 Computer Vision
Mini-Project Assignment S1-2025

1 Overview

The aim of the mini-project is to develop independent research skills in finding good research problems, designing effective solutions, and implementing them using modern techniques. The project involves planning, developing and implementing computer vision and machine learning algorithms, conducting experiments, and evaluating results, as well as communicating the research in the form of a report.

Given an open-ended prompt and a single image, you are required to propose a task and solve it using techniques from deep learning and computer vision. In a two page written report, excluding references, you will outline your proposed task, method, and results, and provide a reflection on the potential societal impacts of your work.

This assignment is to be completed individually, is graded out of **100** marks, and is worth **15%** of your final mark for this course.

1.1 Objectives

The objectives of this assignment are to:

1. identify, formulate and solve a problem in computer vision;
2. design and develop a practical and innovative computer vision system;
3. communicate effectively in a written format; and
4. demonstrate professional and responsible practice.

1.2 Guidelines

1. **Individual work.** This is an individual assignment. All students must work individually when coding and writing the report.
2. **Independent research.** Projects are *unsupervised*. The tutors are not required to monitor the progress of your project, however, course staff will provide generic support and guidance.
3. **Late submission.** The report is to be uploaded to the Wattle site before the due time. This course does not allow late submissions. That is, any submission after the deadline will receive a **mark of zero**. Therefore, it is important to factor upload time into your submission schedule and *submit early*; you can always resubmit an updated version later.

1.3 Academic Integrity

You are expected to comply with the university policy on academic integrity and plagiarism. Please ensure you cite appropriately any resources that you use (lecture notes, papers, online documents, code), and complete all tasks independently. All academic integrity violations will be reported and can result in significant penalties. More importantly, working through the assignment yourself will help you learn the material and will set you up for success in the final exam.

We will be using Turntin for the submission of written work, and we reserve the right to ask you to orally explain your work. You must *not* use the project for any other ANU course or vice versa—this constitutes academic misconduct (see ANU policy on plagiarism `https://www.anu.edu.au/students/academic-skills/academic-integrity`).

Figure 1: The image to be used in this mini-project (see file on Wattle), showing the John Curtin School of Medical Research at ANU. Metadata: 5616×3744 RGB image; 17 mm focal length; $1/400$ exposure time. Copyright: All rights reserved, The Australian National University. Used with permission. Do not distribute externally.

2 The Task

You are provided with a ***single*** colour image (see Figure 1) and are given the open-ended prompt: *what can be (deep) learned from just this one image?*

2.1 Requirements

1. You must use deep learning (i.e., training, optimising or fine-tuning a neural network is a mandatory component of your approach).
2. You must use the provided image in some form (downsampling is allowed).
3. You may not capture or acquire additional images for training. The aim is to see what can be learned from a single image, so please do not collect or use a dataset of different images.

To reiterate, **this one image is your entire dataset** for training, optimising or fine-tuning your network. You may not use other images.[a] This is intentionally a very *strange* requirement that will require some creativity from you to come up with an interesting deep learning task.

[a]What constitutes another image? This is somewhat ambiguous, but see Section 2.3 for some allowed differences or perturbations.

2.2 Considerations

You will be proposing your own research project, which must have a significant computer vision and deep learning component, contain some level of technical challenge, and be of an appropriate level of complexity for a 5-week project. It is your responsibility to ensure that the project can be completed in the given time frame. You should expect to commit about 5 hours per week on average.

Three approaches—not specific to this assignment—are typical for developing a project idea:

1. **An extension of prior work:** for example, if a work assumes a static camera, can you extend it to handle a moving camera?
2. **A new application of prior work:** for example, could a new loss function for recovering thin structures in MRI data be used to recover overhead power cables in an outdoor image segmentation task?
3. **A new solution to an existing problem:** for example, can you develop a new approach to sign language recognition?

A good place to look for ideas is the proceedings of a recent (2020–25) top-tier computer vision conference (e.g., CVPR, ICCV, ECCV, NeurIPS and SIGGRAPH) to get a sense for the sort of tasks addressed in computer vision research. Many proceedings can be found for free online, such as at `http://openaccess.thecvf.com`. Choose a paper that interests you and consider whether you could extend or improve the work, apply one of the ideas to a different problem, or use it as inspiration. It can be helpful to look at the code (if available) to get a concrete view of their method. You will need to highlight in your report what improvements or extensions you made, or where you obtained your ideas or inspiration.

To get you started, here are some ideas for neural network-based tasks involving a single image:

1. image compression by fitting coordinate network parameters (see, e.g., NeRF [2] and NeRFCodec [1]) [easy];
2. image super-resolution with coordinate networks [easy];
3. learning scene-specific features with a self-supervised loss on image crops [moderate];
4. learning a generative image model from a single image (e.g., extend SinGAN [3] to diffusion models) [hard];
5. fine-tuning a pre-trained generative image model with a single image to increase the likelihood of similar samples [moderate];
6. fine-tuning a pre-trained generative image model with a single image to inpaint missing regions from the image [moderate];
7. image manipulation with single-image generative models [4] [moderate];
8. detect object and scene symmetries by fitting a symmetry-aware network [hard];
9. segmenting the image into meaningful regions by training a CNN with self-supervised losses [moderate];
10. predicting relative depth from a single image by training a CNN with self-supervised losses [moderate];
11. predicting the location of Harris corners with a CNN [easy];
12. predicting Canny edge locations with a CNN [easy]; and
13. something else entirely—be creative!

To understand some of these ideas, you may need to do a literature search or read the relevant section of the textbook.

2.3 FAQs

1. Can I downsample the provided image? Yes.
2. Can I use augmentations or crops of the provided image? Yes.
3. Can I make use of a pre-trained large language model (LLM)? Yes.
4. Can I make use of a pre-trained vision–language model (VLM)? Yes, so long as you train or fine-tune a neural network to address your problem.
5. Can I make use of a pre-trained generative model like Stable Diffusion? Yes, so long as you train or fine-tune a neural network to address your problem.
6. Can I re-use code from existing libraries and papers? Yes, with appropriate citations. You will be marked on *your* ideas and approach, but putting together something new from existing components can be a good strategy for research.

3 The Report

- **Length:** 2 pages (additional pages containing only figures with captions and a reference list are allowed).
- **Format:** PDF, using the LaTeX, .docx or .odt template on Wattle.
- **Code:** single ZIP file containing all required source code and a readme file explaining how to test the code.
- **Contents:** [length guidance in brackets]
  1. **Title:** Project title.
  2. **Problem Definition:** Define the task that you have identified, in words and mathematically. What is the input to your method and what is the output? Why is the problem important and/or interesting? What would a success look like and how would you measure it? **[~1 paragraph]**
  3. **Method:** What did you do? Outline your proposed method using a flowchart, equations, and algorithms where relevant. **[~4 paragraphs]**
  4. **Results:** How did you conduct your experiments, how did you obtain a test set, and what are your results? Did your approach succeed and how do you know? If relevant, how does it compare to state-of-the-art approaches for this problem? You may want to include an ablation study that shows how important different aspects of your design are (e.g., removing your new loss function leads to a 3% performance decrease). Provide both quantitative results (tables of numbers) and qualitative results (e.g., images) where appropriate. **[~2–3 paragraphs]**
  5. **Reflection:** What is the broader impact of your work on people other than researchers, including ethical aspects and future societal consequences? Other things to consider include: who may benefit from this research, who may be put at disadvantage from this research, what are the consequences of failure of the system, and does the task/method leverage biases in the data? **[~1 paragraph]**
  6. **Conclusion:** What did you find out? What would you do next, if you had more time? **[~3 sentences]**
  7. **References:** Starting on a new page (`\newpage` in LaTeX), include a list of references. Use the bracket citation style like in this document (e.g., [3]).

Two pages is intentionally very short: it will be a challenge to say everything that you want to say. This is for two reasons: (1) to encourage concision, prioritisation, and use of figures rather than words; and (2) to encourage you to do a realistic amount of work for this mini-project, and no more—it is about exploration, having fun and learning, rather than writing a lengthy report.[a]

[a] Let us know if this was or was not the case in the course surveys!

4 Coding Languages and Frameworks

Python and PyTorch are strongly recommended due to their ease of use and widespread take-up in the computer vision and machine learning communities, however you are free to choose other programming languages and frameworks, such as Matlab, Java, C/C++, TensorFlow, etc. You may also use external libraries, such as OpenCV, OpenGV, VXL, etc.

5 Accessibility

Please be mindful to communicate your research in a way that is accessible to as many people as possible. Keep your sentences short and clear unless there is a pressing need. Ensure your figures are large enough to be legible and easily interpreted. Importantly, **do not rely on colour** to, for example, distinguish between lines in a plot. I am colour blind and will not be able to interpret your results. Instead, use symbols, line styles or labels to further disambiguate your plots. (Sometimes this is not possible. For example, the results of an image colourisation algorithm cannot be made more accessible to those with colour blindness.)

6 Research Advice

(Adapted from some good advice I was given by Steve Gould.)

6.1 Progress

Research is difficult. The many unknowns and the long-term focus often means that more concrete and shorter-term activities (e.g., an assignment for a course) take precedence. Don't let this happen. Set realistic milestones and stick to them.

I expect you to work consistently throughout the duration of project and not just around deadlines. You are the owner of the project and responsible for it.

If your project involves software development, consider setting up a GitHub repository where you can also manage tasks and milestones. This also makes it easy to deploy your software after the project.

6.2 Experimentation

Not all your experiments will be reported in your final report. Indeed, many experiments are done to convince yourself that your code and model are behaving as expected (i.e., for debugging). Think of these as software unit tests.

Trends are more interesting than single results. If your model contains a parameter that is designed to induce some effect, then make sure you vary the parameter and observe the effect. If a baseline model falls within your family of models, then make sure you can reproduce the baseline results by appropriately setting the parameters of your model. If your model is producing counter-intuitive results, then try injecting ground-truth signals into various places in your experimental pipeline to debug where the problem may be. Don't forget to remove the ground-truth when running your final experiments!

When planning milestones, give yourself enough time to re-run experiments. Things often go wrong the first time! In the same spirit, test your code on a small dataset first to make sure it runs to completion before setting off a multi-day experimental run. During the multi-day run, log as much data as you can and monitor progress.

Always report results on a hold-out test set that was not used to tune model parameters. However, it is also useful to collect and observe results on the training set. A big discrepancy between test and train may indicate that your model is over-fitting to the training data. Poor performance on your training data may indicate a problem with your model or training procedure.

7 Submission Requirements

7.1 Files

Upload a **PDF file of your report** and a **ZIP file of your code** by the due date. You must use the following file names: **uXXXXXXX.pdf** and **uXXXXXXX.zip**, replacing uXXXXXXX with your university ID. Your ZIP file must contain all your *.py and/or *.ipynb files. There are **separate** upload boxes on Wattle for the report and the code.

7.2 Report Formatting Instructions

Use:

1. Headings, as given in this document;
2. Times New Roman font or similar;
3. single-spaced font;
4. ~12pt font for the body text;
5. one-inch margins (top, bottom, left, right); and
6. images where appropriate to explain your answers.

Figures and tables are critical for communicating your results clearly. Always refer to these from the body text of the report (e.g., "As shown in Figure 2, the results indicate that. . . "). Use:

1. descriptive captions;
2. axis labels;
3. a legend;
4. appropriate axis scaling (e.g., perhaps a log scale is more informative?);
5. appropriate annotations;
6. ~10pt Times New Roman font for captions;
7. a sufficient size for visibility; and
8. your own material, or public-domain and cited images only.

8 Marking Criteria

Category	Indicative criteria	Mark
Presentation	• Consistent and appropriate formatting • Correct spelling and grammar • Clear and concise explanation and interpretation • Adheres to page limit (2 pages + figures/references) • Uses LaTeX, .docx or .odt template • Symbols and acronyms defined • Appropriate referencing and bibliography • Figures, tables and captions are clear and properly integrated into the text • Logical structure • Written for a technically knowledgeable, non-expert audience • Visual communication aids are used (e.g., a flowchart to outline the method)	/20
Problem definition and background	• Problem definition is clearly and precisely defined • The input(s) and expected output(s) are reported • An outcome that constitutes a success is outlined • The importance of the problem is motivated	/20
Technical expertise and scientific quality	• The chosen techniques and approaches are appropriate for the project • The student has demonstrated skill and understanding of the techniques used • The project is well scoped and contains an appropriate level of technical material • Limitations in the methods used are discussed • The project contains some novelty and creativity • Overall quality of the system design and implementation	/20
Experimental method, results and analysis	• Experiments validate any claims • Comparison to previous work using standard metrics, if included • Necessary ablation studies included • Clear documentation of the implementation details and datasets • Analysis of the results is appropriate and well communicated • An insightful discussion and interpretation of the results • Plausible explanations are provided for any inconsistent results	/20
Reflection	• Thoughtful reflection on the ethical and societal concerns • Considers at least two salient concerns • Considerations are relevant to the project	/15
Conclusions and future work	• Clear expression of the project conclusions • Conclusions drawn are supported by the results • Suggestions for further work are appropriate and insightful	/5
Total		/100

References

[1] Sicheng Li, Hao Li, Yiyi Liao, and Lu Yu. NeRFCodec: neural feature compression meets neural radiance fields for memory-efficient scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21274–21283, 2024.

[2] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020.

[3] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[4] Yael Vinker, Eliahu Horwitz, Nir Zabari, and Yedid Hoshen. Image shape manipulation from a single augmented training sample. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13769–13778, 2021.