COMP9517: Computer Vision

2025 Term 1

Group Project Specification

Maximum Marks Achievable: 40

The group project is **worth 40% of the total course mark**.

Project deliverables due in Week 10.
Deadline for submission is Friday 25 April 2025 18:00:00 AET.
Instructions for online submission will be posted closer to the deadline.
Refer to the separate marking criteria for detailed information on marking.

Introduction

The goal of the group project is to work together with peers in a team of 5 students to solve a computer vision problem and present the solution in both oral and written form.

Group members can meet with their assigned tutors once per week in during the usual consultation session hours to discuss progress and get feedback. Consultation with lecturers is also available *in-person* in the lecture theatre (MathewsThA), 3-4pm Wednesday in Weeks 7 and 9, and 2-4pm Wednesday in Week 10.

Description

Task

The goal of this group project is to develop and compare different computer vision methods for classifying aerial scenes from remote sensing images. Such a task has many real-world applications in urban planning, environmental monitoring and disaster response by enabling automated analysis of large-scale satellite and drone imagery.

Dataset

The dataset to be used in this group project is available from Kaggle (https://www.kaggle.com/datasets/ankit1743/skyview-an-aerial-landscape-dataset).

It contains 15 landscape categories, with each category containing 800 high-quality images. It is thus a well-balanced dataset and the task is to develop and evaluate computer vision methods for classifying the images into 15 landscape categories.

Methods

Many traditional machine learning and deep learning-based computer vision methods can be used for this task. You are challenged to use concepts taught in the course and other techniques from literature to develop your own methods and test their performance.

At least two different methods must be developed and tested. For example, you could compare one feature extraction and machine learning-based method and one deep learning method, or two deep learning-based methods using different neural network architectures.

In general, we do not expect you to develop everything from scratch. We do encourage you to try different techniques, or modifications of existing ones, or the use of more state-of-the-art methods that have not been presented in the lectures. In any case, always do cite your sources, especially for papers, tools and GitHub repositories that you have referenced or used.

Comprehensive Method Development

If you are aiming for high marks (33-36 marks), we expect to see sufficient varieties in the methods that you develop. For example, as a group, you might develop all the following:

- Machine learning-based methods, comparing different feature descriptors (such as LBP and SIFT), and classifiers (such as kNN and SVM).
- Deep learning methods, comparing different deep learning architectures (such as ResNet, EfficientNet and SENet), and data augmentation methods.

Advanced Method Development

If you are aiming for even higher marks (37 and above), we expect to see some interesting ideas and research skills. For example, you might consider adding some of the following:

- What if the dataset is resampled to make it an imbalanced classification problem? For example, the number of images might follow a long-tail distribution, e.g., 800 images in category 1, 700 images in category 2, and gradually decreasing to 50 images in category 15. For such imbalanced classification problems, there are various techniques that you can try, such as varying the training process (e.g., re-weighting and re-sampling) and applying data augmentation methods specifically targeting the minority classes [1].
- What if users of your system want to understand if the system has learned the right information? There are many Explainable AI methods that can be applied to image classification models, such as the Grad-CAM method [2] that produces attention maps. You can experiment with such methods and conduct some analysis based on the visual outputs. And if you add some noise, blurring, or occlusion in the test image, does the model still give you the correct classification and right attention maps? This would be similar to studies about adversarial attacks.
- Any other interesting ideas that you want to try. You can discuss your ideas with the tutors or lecturers during the consultation hours if you are unsure about the validity or soundness of the idea.

Training and Testing

We expect that this task will require training and testing. For this task, in a standard setting, you should do an 80-20 split for the dataset. This means that, for each category, randomly choose 20% of the 800 images as the test set, and use the remaining 80% from all categories as the training set. To evaluate the performance of your methods, use at least three evaluation metrics for each method, such as precision, recall and F1-score.

You can vary the setting if you are developing something different, like those mentioned under the Advanced Methods. Then you can also include other evaluation metrics, suitable for your specific experimental settings, to have more comprehensive analyses. You can also include 5-fold cross validation for evaluating the classification performance.

Result Discussion

Show these quantitative results in your video presentation and written report (see deliverables below). Also show representative examples of successful cases as well as examples where your methods failed (no method generally yields perfect results). Give some explanation why your methods succeed or fail in these cases. In addition, compare the results obtained from the various methods and discuss your findings from such comparisons (e.g., why certain methods perform better than the others), and you can do ablation studies (e.g., compare how your method works with different parameter settings).

Practicalities

We have tested that training and testing of ResNet-18 models can be completed within 10 minutes on a desktop computer with a Nvidia RTX 3090 GPU. If you need a powerful computer, you can consider using the free version of Google Colab. Otherwise, you are free to use only a subset of the data, for example 75% or 50%. Of course, you can expect the performance of your methods to go down accordingly, but as long as you clearly report your approach, this will not negatively impact your project mark.

Deliverables

The deliverables of the group project are 1) a video presentation, 2) a written report, and 3) the code. The deliverables are to be submitted by only one member of the group, on behalf of the whole group (we do not accept submissions from multiple group members).

Video

Each group must prepare a video presentation of about **10 minutes** showing their work (any content beyond that will be ignored). The presentation must start with an introduction of the problem and then explain the developed methods, show the obtained results, and discuss the findings as well as ideas for future improvements. For this part of the presentation, use PowerPoint slides to support the narrative. Following this part, the presentation must include a demonstration of the methods/software in action. Of course, some methods may take a long time to compute, so you may record a live demo and then edit it to stay within time.

All group members must present, but it is up to you to decide who presents which part. In order for us to verify that all group members are indeed presenting, each student presenting their part must be visible in a corner of the presentation (live recording, not a static head shot), and when they start presenting, they must mention their name. Overlaying a webcam recording can be easily done using either the video recording functionality of PowerPoint itself (see for example this YouTube tutorial) or using other recording software such as Zoom.

Also note that video files can easily become quite large (depending on the level of compression used). To avoid storage problems for this course, **the video upload limit will be 100 MB per group**, which should be more than enough for this type of presentation. If your video file is larger, use tools like HandBrake to re-encode with higher compression.

The video presentations will be marked offline (there will be no live presentations). If the markers have any questions, they may contact the group members by email for clarification.

Report

Each group must also submit a written report (**PDF** file, in 2-column IEEE format, using Word, Latex or Overleaf, **maximum 10 pages** of main text, with additional pages of references).

The report must be submitted as a PDF file and include:

1. Introduction: Discuss your understanding of the task specification and dataset.
2. Literature Review: Review relevant techniques in literature, along with any necessary background to understand the methods you selected.
3. Methods: Motivate and explain the selection of the methods you implemented, using relevant references and describing important method details.
4. Experimental Results: Explain the experimental setup you used to test the performance of the developed methods and the results you obtained.
5. Discussion: Provide a discussion of the results and method performance, comparing the various methods that you developed.
6. Conclusion: Summarise what worked / did not work, limitations of the implemented methods and recommend future work.
7. References: List the literature references and other resources used in your work. The references section does not count toward the 10-page limit.

Code

Code submission should mainly contain your own programs. Libraries or code obtained from other sources should be clearly described in the **README**. Include proper documentation about how to run the code, inline comments and make the code well structured. A good example can be found at: https://github.com/xuehaouwa/Fashion-MNIST (Note: we recommend .md file for the README, but you can use other format as well).

The upload limit for the source code plus report (all in a ZIP file) will be 25 MB. Note you may need to downsize the images (using Paint or other tools, not simply resizing within the Word or Latex file) before inserting them into the report to keep the file size within the limit. For the code, *do not* include trained models, input images or result images.

Student Contributions

As a group, you are free in how you divide the work among the group members, but all group members must contribute roughly equally to the method development, coding, making the video, and writing the report. For example, it is unacceptable if some group members only prepare the video and report without contributing to the methods and code.

An online survey will be held at the end of the term allowing students to anonymously evaluate the relative contributions of their group members to the project. The results will be reported only to the LIC and the Course Administrators, who at their discretion may moderate the final project mark for individual students if there is sufficient evidence that they contributed substantially less than the other group members.

References

[1] Y. Zhang et al., “Deep long-tailed learning: A survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(9): 10795-10816, 2023. (https://arxiv.org/abs/2110.04596)

[2] R. Selvaraju et al., “Grad-CAM: Visual explanations from deep networks via gradient-based localization”, IEEE International Conference on Computer Vision, pp. 618-626, 2017. (https://arxiv.org/abs/1610.02391)

Released: 17 March 2025