# COMP9517: Computer Vision

# 2025 Term 3

# Group Project Specification

# Maximum Marks Achievable: 40

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

Project work is in Weeks 6-10 with deliverables due in Week 10.

**Deadline for submission is Friday 21 November 2025 18:00:00 AET.**

Instructions for online submission will be posted closer to the deadline.

Refer to the separate marking criteria for more 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 solutions and results in both oral and written form.

Group members can meet with their tutors once per week in Weeks 6-10 during the usual consultation session hour to discuss progress and get feedback.

---

### Description

Healthy and productive agriculture is essential to human life and of great importance to economic development of nations. Insect pests, however, cause extensive damage to plants, resulting in crop yield losses and reduced food availability. To monitor insect pests during the cropping season, and to develop and implement effective pest management strategies, advanced tools are needed. Computer vision methods can help to quickly and reliably detect and identify insects from image and video recordings.

#### Task Statement

The goal of this group project is to develop and compare different computer vision methods for detecting and classifying insects in images from natural agricultural environments.

#### Public Dataset

The public dataset to be used in this group project is AgroPest-12 and is available from Kaggle (see reference and link at the end of this document). It contains 11,502 training images as well as 1,095 validation images and 546 test images, showing insects from 12 classes, along with the ground-truth class labels and bounding boxes.

## Method Development

Many traditional, machine learning, and deep learning-based computer vision methods could 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. More specifically, as the task is to both detect and classify objects, one complete method is a combination of a detector and a classifier, and two such methods must be developed.

We do not expect you to develop everything from scratch, but we do encourage you to try different methods, or modifications of existing ones, or use more state-of-the-art methods that have not been presented in the lectures. As a general guideline, the more you code things yourself rather than copy straight from elsewhere, the better. In any case, always cite your sources, including all papers, tools and repositories you have used.

### *Comprehensive Method Development*

If you are aiming for higher marks (33-36), we expect to see sufficient variety in the methods you develop. For example, as a group, you could develop the following:

- Machine learning-based methods, comparing different handcrafted feature descriptors and classification methods.
- Deep learning-based methods, comparing different deep neural network architectures and data augmentation methods.

### *Advanced Method Development*

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

- Resample the dataset to make it an imbalanced classification problem and show how your methods perform in that case. Also, various techniques could be explored to try and correct for this imbalance, including reweighting, resampling, and augmentation.
- Evaluate how your methods are affected by different levels of distortions that may happen in practice, such as noise, blurring, low brightness/contrast, or occlusion in the test images. This would be similar to studies about adversarial attacks.
- Extend the methods to provide explanation of the predictions so that users get an idea of what the methods have actually learned. You could experiment with different explainable AI methods such as attention maps and conduct some analysis based on them.

These are just some examples. Do discuss with your group and see if you can come up with other original ideas that add value to your methods and results. You can further discuss your ideas with the tutors during the weekly consultation hours.

### Training and Testing

The given dataset has already been split into training, validation, and test subsets. It is very important to never mix them up or cause data leakage. Whether your methods require training (supervised methods) or not (unsupervised methods), they will probably have (hyper)parameters that you need to fine-tune to get optimal performance. Use the training and validation sets for this and use the test set only for final performance evaluation. Using (partly) the same data for both training/fine-tuning and testing leads to biased results that are not representative of real-world performance.

To assess the performance of your methods, compare the predicted bounding boxes quantitatively with the corresponding annotated boxes using appropriate metrics, and likewise for the predicted classes. Specifically, compute at least the mean average precision (mAP) to evaluate detection performance, and compute the precision, recall, F1 score, accuracy, and the area under the curve (AUC) to evaluate the classification performance. Furthermore, compare the training and testing times of the different methods.

### Results and Discussion

Show the quantitative results in your video presentation and written report (see deliverables below). Also show representative examples of successful results as well as examples where your methods failed (no method generally yields perfect results). Give some explanation why you believe your methods failed in these cases. Also, if one of your methods clearly works better than the other(s), discuss possible reasons why. Finally, discuss some potential directions for future research to further improve performance.

### Practicalities

The dataset is less than 600 MB in total, so method training and testing should be feasible on a modern desktop or laptop computer. If more computing resources are needed, you could consider using the free version of Google Colab. Otherwise, you are free to use only a subset of the data, for example 75% or even just 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 at most 10 minutes (any content beyond that will be ignored). The presentation must start with an introduction of the problem and then explain the used methods, show the obtained results, and discuss these results 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 (points will be deducted if this is not the case). It is up to you to decide who presents what (introduction, methods, results, discussion, demonstration). 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 done using either the video recording functionality of PowerPoint itself (see for example this YouTube tutorial) or using other recording software such as OBS Studio, DaVinci Resolve, Adobe Premiere, and many others. It is up to you (depending on your preference and experience) which software to use, as long as the final video satisfies the requirements mentioned above and below.

The entire presentation must be in the form of a video (720p or 1080p MP4 format). Note that video files can become quite large (depending on the level of compression used). To avoid storage problems for this course, the video upload limit is 100 MB per group, which should be more than enough for this type of presentation. If your video file is initially larger, use tools like HandBrake to re-encode with higher compression. For example, in Handbrake, choose the ”Very Fast 720p30” Preset and then go to the Video tab to lower the Quality as needed. That way, even a 1-hour video presentation can be compressed to under 100 MB without significant loss of quality. Minor compression artifacts will not affect your mark.

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 (in 2-column IEEE format; maximum 10 pages of main text including all diagrams, figures, tables, but excluding references; you may add any number of additional pages for references only).

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

1. Introduction: Present a brief introduction including the relevance and your understanding of the task and the 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 theories where necessary.

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 experimental results and findings, comparing the different methods you developed.
6. Conclusion: Summarise what worked and what did not work, any limitations of the developed 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.

The upload limit for the report is 10 MB per group. If your report is initially larger than that, you may need to resize images or increase the compression.

Code

The source code of the developed software must be submitted as a ZIP file. The file should mainly contain your own code. Libraries or code obtained from other sources should be clearly described in a README file. Include proper documentation about how to run the code, use inline comments, and make the code well structured.

The upload limit for the source code is 25 MB per group. To stay within this limit, 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 or report without also 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.

---

**Dataset**

The dataset to be used in the group project is the following:

Rupankar Majumdar. AgroPest-12: Image Dataset for Crop Pest Detection. Kaggle, September 2025. https://www.kaggle.com/datasets/rupankarmajumdar/crop-pests-dataset

**References**

The following papers may be helpful in getting a deeper understanding of the problem and exploring potential methods for insect identification:

Sourav Chakrabarty et al. Application of artificial intelligence in insect pest identification – a review. Artificial Intelligence in Agriculture 16(1):44-61, March 2026.
https://doi.org/10.1016/j.aiia.2025.06.005

Kaili Wang et al. AP162: A large-scale dataset for agricultural pest recognition. Computers and Electronics in Agriculture 237B:110520, October 2025.
https://doi.org/10.1016/j.compag.2025.110520

Xiaoping Wu et al. IP102: A large-scale benchmark dataset for insect pest recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
https://doi.org/10.1109/CVPR.2019.00899

---

**Released:** 17 October 2025