# COMP9517: Computer Vision

# 2024 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 15 November 2024 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 Weeks 6-10 during the usual consultation session hours to discuss progress and get feedback.

The group project is to be completed by each group separately. Do not copy ideas or any materials from other groups. If you use publicly available methods or software for some of the steps, these must be properly attributed/referenced. Failing to do so is plagiarism and will be penalised according to UNSW rules described in the Course Outline.

You are expected to show creativity and build on ideas taught in the course or from computer vision literature. High marks will be given only to groups that developed methods not used before for the project task. We do not expect you to develop everything from scratch, but the more you use existing code (which will be checked), the lower the mark.

---

**Description**

Recognizing individual animals from photographs is important for many wildlife studies, including population monitoring, behaviour analysis, and management. This is often done by expert manual analysis of the photographs, in particular segmentation of the relevant body parts of the animals, which is very labour-intensive. The continuing collection and expansion of image datasets spanning multiple years of observation is causing a growing need for computer vision methods to automate the task.

## Task

The goal of this group project is to develop and compare different computer vision methods for segmenting sea turtles from photographs. More specifically, the task is to segment the head, flippers, and the carapace of each turtle.

## Dataset

The dataset to be used in this group project is called SeaTurtleID2022 and is available from Kaggle (see references with links at the end of this document). It contains 8,729 photographs of 438 unique sea turtles collected over 13 years in 1,221 encounters, making it the longest-spanned dataset for animal reidentification.

Each photograph comes with annotations such as identities, encounter timestamps, and segmentation masks of the body parts. Further details are provided in the paper associated with the dataset (see references). On WebCMS3 we provide a Jupyter notebook showing how to load the turtle photographs and corresponding annotations.

## Methods

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. For example, you could compare one machine learning-based method and one more traditional method, or two deep learning-based methods using different neural network architectures.

Although we do not expect you to develop everything from scratch, we do expect to see some new combination of techniques, or modifications of existing ones, or the use of more state-of-the-art methods that have not been tried before for sea turtle segmentation.

As there are virtually infinitely many possibilities here, it is impossible to give detailed criteria, but as a general guideline, the more you develop things yourself rather than copy straight from elsewhere, the better. In any case, always do cite your sources.

## Training

If your methods require training (that is, if you use supervised rather than unsupervised segmentation approaches), you must ensure that the training images are not also used for testing. Even if your methods do not require training, they may have hyperparameters that you need to fine-tune to get optimal performance. In that case, too, you must use the training set, not the test set. Using (partly) the same data for both training/fine-tuning and testing leads to biased results that are not representative of real-world performance.

Specifically, you must use open-set splitting of the dataset into training, validation, and test subsets, as defined by the creators in the metadata of the SeaTurtleID2022 dataset. In their paper (see references below) the creators explain that open-set splitting gives a much more realistic performance evaluation than closed-set or random splitting.

### Testing

To assess the performance of your methods, compare the segmented regions quantitatively with the corresponding manually annotated (labelled) masks by calculating the intersection over union (IoU), also known as the Jaccard similarity coefficient (JSC).

That is, for each photograph in the test set, compute the IoU of the predicted head segment with the corresponding head segment in the annotation, and similarly for the flippers and the carapace ("turtle"). Then compute the mean IoU (mIoU) over the whole test set for each of the three categories separately (head mIoU, flippers mIoU, and carapace mIoU).

Show these quantitative results in your video presentation and written report (see deliverables below). Also show representative examples of successful segmentations 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.

Furthermore, if one of your methods clearly works better than the other, discuss possible reasons why. And, finally, discuss some potential directions for future research to further improve the segmentation performance for this dataset.

### Practicalities

The SeaTurtleID2022 dataset is less than 2 GB 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 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). More detailed information on the deliverables:

### Video

Each group must prepare a video presentation of at most 10 minutes showing their work. 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.

The entire presentation must be in the form of a video (720p or 1080p MP4 format) of at most 10 minutes (anything beyond that will be ignored). All group members must present (points may be deducted if this is not the case), but it is up to you to decide who presents which part (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 easily 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, Camtasia, 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.

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 concerns or questions about the presented work, they may contact the group members by email for clarification.

## Report

Each group must also submit a written report (in 2-column IEEE format, max. 10 pages of main text, and in addition any number 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 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 results and method performance, in particular reasons for any failures of the method (if applicable).
6. Conclusion: Summarise what worked / did not work and recommend future work.
7. References: List the literature references and other resources used in your work. All external sources (including websites) used in the project must be referenced. The

references section does not count toward the 10-page limit.

Code

The complete source code of the developed software must be submitted as a ZIP file and, together with the video and report, will be assessed by the markers. Therefore, the submission must include all necessary modules/information to easily run the code. Software that is hard to run or does not produce the demonstrated results will result in deduction of points. The upload limit for the source code (ZIP) plus report (PDF) together will be 100 MB. Note that this upload limit is separate from the video upload limit (also 100 MB).

Plagiarism detection software will be used to screen all submitted materials (reports and source codes). Comparisons will be made not only pairwise between submissions, but also with related assignments in previous years (if applicable) and publicly available materials. See the Course Outline for the UNSW Plagiarism Policy.

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**

**Dataset:** SeaTurtleID2022: A dataset with 438 different sea turtle individuals. Can you recognize them? Kaggle, 2024.

**Paper:** L. Adam, V. Čermák, K. Papafitsoros, L. Picek. SeaTurtleID2022: A long-span dataset for reliable sea turtle re-identification. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2024, pp. 7146-7156.

---

**Released:** 11 October 2024