COMP9517: Computer Vision

2022 Term 3

Group Project Specification

Maximum Marks Achievable: 40

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

> Project work is in Weeks 6-10 with a demo and report due in Week 10.
> Refer to the separate marking criteria for detailed information on marking.
> Submission instructions and a demo schedule will be released later.

Introduction

The goal of the group project is to work together with peers in a team of 4-5 students to solve a computer vision problem and present the solution in oral and written form.

Each group can meet with their assigned tutors once per week in Weeks 6-9 during the usual consultation session on Fridays 11am-12pm 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 tasks, 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.

Note that we give high marks only to groups who developed something new or tried more state-of-the-art models not used before for the given task. The more you use or build on existing solutions, the lower the mark.

Description

An important and challenging computer vision task is object detection in real-world images or videos. Example applications include surveillance, traffic monitoring, robotics, medical diagnostics, and biology. In many applications, the large volume and complexity of such data make it impossible for humans to perform accurate, complete, efficient, and reproducible recognition and analysis of the relevant image information.

The goal of this group project is to develop and evaluate a method for detection of starfish in underwater videos of coral reefs. Australia's Great Barrier Reef, the world's largest coral reef, is under threat in part because of overpopulation of the coral-eating crown-of-thorns starfish (COTS). Monitoring COTS outbreaks requires large-scale video surveillance using

underwater cameras and automated COTS detection methods. The challenge is to develop methods that can analyse the video images accurately and efficiently.

Tasks

Dataset

The dataset to be used in the group project is from the Kaggle COTS detection competition [1] and is described in full detail in the associated paper [2]. In summary, the training set consists of three videos containing in total tens of thousands of images with corresponding manual annotations (bounding boxes around COTS objects). The official annotated test set is not publicly available (though the images can be accessed via an API). In this project we will not use this test set, but only the available training set, which hereafter we will simply refer to as "the dataset", for our own training and testing, described below.

Methods

Many traditional and/or machine or deep learning-based computer vision methods could be used to detect COTS in the video images. In this project you are challenged to use concepts taught in the course and other methods from literature to develop your own COTS detection methods and evaluate their performances.

More specifically, each group is expected to develop two different methods and compare their performances to see which one works better. Some methods of the COTS competition participants are publicly available [1]. You can study them to get inspiration, but you should not use them (we will check for this, see the plagiarism notice below). Develop your own methods using other state-of-the-art techniques or models.

Training

If your methods require training (that is, if you use supervised rather than unsupervised detection approaches), you can use part of the dataset for this purpose, as described in the testing step (see next). Even if your methods do not require training, they probably do have hyperparameters that you may want to fine-tune to get optimal performance. In that case, too, you must use a part of the dataset that will not be used for testing, because using the same data for both training/fine-tuning and testing leads to biased results that are not representative of actual performance. So, either way, do split the dataset (see next).

Testing

For the testing of your methods, you must use data that have not been used in the training or fine-tuning stage. Thus, the dataset must be randomly split into 80% for training and the remaining 20% for testing. Performance evaluation must be done using the F2 metric (see the paper [2] for details on how to compute this metric). Show the F2 scores in your demo and written report (see deliverables below), and if one method works clearly better than the other, try to explain what the reasons for this could be. Also compare the performances of your methods to those of the top-performing methods on the COTS competition leaderboard and try to explain the differences.

<u>Visualisation</u>

In addition to quantitative testing (described above) your program must also show the detections. That is, for each image, it should not only detect each COTS, but also draw its corresponding bounding box. Use a unique colour per COTS to draw the box. Furthermore, the count (number of detected COTS in the image) should be reported either by printing to the terminal or (better) directly on the image (in one of the corners of the window).

Deliverables

The deliverables of the group project are 1) a group video demo and 2) a group report. Both are due in Week 10. More detailed information on the two deliverables:

Video Demo: Each group will 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. This part of the presentation should be in the form of a short PowerPoint slideshow. Following this part, the presentation should 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 cut off). 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 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 be easily 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 such as HandBrake to reencode with higher compression.

During the scheduled lecture/consultation hours in Week 10, that is Wednesday 16 November 2022 4-6pm and/or Friday 18 November 2022 10am-12pm, the video demos will be shown to the tutors and lecturers, who will mark them and will ask questions about them to the group members. Other students may tune in and ask questions as well. Therefore, all

members of each group must be present when their video is shown. A roster will be made and released closer to Week 10, showing when each group is scheduled to present.

Report & Code: Each group will also submit a report (max. 10 pages, 2-column IEEE format) along with the source code, before 18 November 2022 18:00:00 AEST.

The report must be submitted as a PDF file and include:
1. <u>Introduction</u>: Discuss your understanding of the task specification and dataset.
2. <u>Literature Review</u>: Review relevant techniques in literature, along with any necessary background to understand the methods you selected.
3. <u>Methods</u>: Justify and explain the selection of the methods you implemented, using relevant references and theories where necessary.
4. <u>Experimental Results</u>: Explain the experimental setup you used to evaluate the performance of the developed methods and the results you obtained.
5. <u>Discussion</u>: Provide a discussion of the results and method performance, in particular reasons for any failures of the method (if applicable).
6. <u>Conclusion</u>: Summarise what worked / did not work and recommend future work.
7. <u>References</u>: List the literature references and other resources used in your work. All external sources (including websites) used in the project must be referenced.

The complete source code of the developed software must be submitted as a ZIP file and, together with the 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 (each is 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 similar assignments in previous years (if applicable) and publicly available materials. See the Course Outline for the UNSW Plagiarism Policy.

As a group, you are free in how you divide the work among the group members, but all group members are supposed to contribute approximately equally to the project in terms of workload. An online survey will be held at the end of term allowing students to anonymously evaluate their group members' relative contributions 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] Kaggle. TensorFlow – Help Protect the Great Barrier Reef: Detect crown-of-thorns

starfish in underwater image data. 2021-2022.
https://www.kaggle.com/competitions/tensorflow-great-barrier-reef/

[2] J. Liu et al. The CSIRO Crown-of-Thorn Starfish Detection Dataset. arXiv:2111.14311, November 2021. https://arxiv.org/abs/2111.14311

Released: 12 October 2022

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder