COMP3007/COMP4106
Computer Vision Coursework Description

2024-25

1 Introduction

We have looked in class at different kinds of convolutional neural networks (CNNs) and the basics of how they work. We have also seen some simple examples in MATLAB for both classification and segmentation. In this coursework, you will have a chance to research this topic in more detail, and build your own networks in MATLAB to work on an existing dataset.

2 Key dates

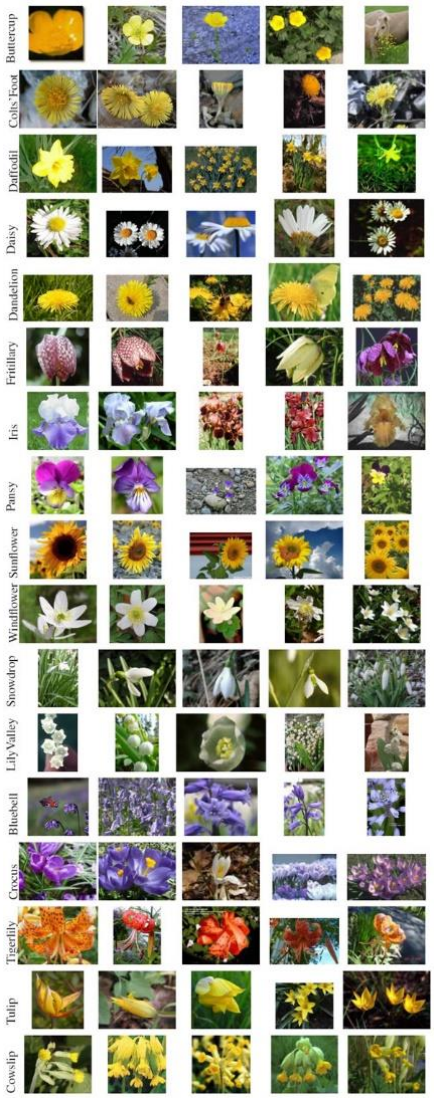
Submission deadline of MATLAB code and report: 6th May 2025 1pm.

More details given in the Module Assessment Sheet in Moodle. Note that COMP3007 and COMP4106 have different assessment sheets (COMP4106 has an additional presentation component)

3 Detailed requirements

Dataset:

We will work with the Oxford Flower Dataset, 17 class version (see right). This dataset consists of photos of flowers, grouped into categories depending on flower type. It also comes with some segmentation maps of the flower regions.



Your datasets for our classification and segmentation tasks will be provided locally on Moodle.

More details on the background of the dataset is available here

https://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html

Related papers are also provided at that link.

4 Method

You have two main tasks.

First, you must research and write MATLAB **code** to (a) classify and (b) segment images from the dataset using CNNs.

Second, you must write up your methods as a **research paper**, according to a fixed template and set of subheadings.

Both tasks are explained below.

Coding

You need to develop one CNN to ***classify*** the images, and one CNN to ***segment*** the images.

The classification network **must be written by you from scratch, in MATLAB, *building up CNN layers in code*.**

The other segmentation network can, if you like, *use an existing pretrained model*, but you would need to work out how to reuse/retrain this for the given task. You can write *both* models yourself from scratch in code if you prefer.

Classification

Use the 17-class dataset ("17Flowers" on Moodle) provided to train a CNN classification model to classify a new image into one of the 17 flower classes.

You will need to consider:

- How to organise the data files
- How to load and preprocess the data, and arrange into classes
- How to build and train a CNN model from scratch
- How to evaluate how well it is working
- How to improve efficiency in your code

You MUST

1. Implement a classification network which takes as input colour images of size 256x256x3 (ie. RGB). Train this network in your code.
2. Save your trained network model after training, and submit this as part of your submission.

Other considerations:

- **Classification categories**. The images (1,360 images across 17 categories) are all in one folder and sequentially numbered. You will need to put them into an arrangement suitable for loading into MATLAB (e.g. in folders). The paper says there are 80 examples of each category.

Segmentation

There is a 1-class dataset on Moodle ("DaffodilSeg", the daffodil flower class), comprising flower photos AND segmentation groundtruth maps. You only need to segment this *one* category (not all 17 flower types).

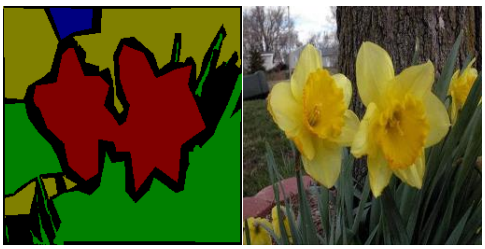
You will need to consider:

- How to organise the data files
- How to load and preprocess the data
- How to load and use the labelled data
- How to build and train a CNN model, or use/retrain an existing model
- How to evaluate how well it is working
- How to improve efficiency in your code

You MUST

1. Implement and train a segmentation network which takes in colour images of size 256x256x3 (ie. RGB). You can build this from scratch or re-use a pretrained model in MATLAB.
2. Save the trained network model file and submit as part of your submission.

- The **segmentation groundtruth** exists as colour maps. If you inspect the images you will see that value 0 is null/boundaries, 1 is flower, 3 is background. Some images also have class 2 as leaves and 4 as sky. These last two classes are more unreliable, and for this coursework we can ignore them or count them both as background as appropriate, **making only 2 classes we care about: flower and background**. An example image with all 5 classes looks like:



- Note there are only 71 labelled images in this class.

What to submit

Classification:

One MATLAB file called **classification.m**; One trained network model file called **classnet**

Segmentation:

One MATLAB file called **segmentation.m**; One trained network model file called **segmentnet**

Report: One PDF matching the template and page requirements.

You will be shown how to save a network model file. If you use an existing network for the segmentation task, you must submit your retrained/adapted version of the model

Create a zip file where you will include the MATLAB code files, saved model files, and report with the following name: **YourStudentID_ModuleCode.zip.** For example, if your ID is 012345678 and you are an undergraduade student, please name the zip file as 012345678_COMP3007.zip. Change the module code where appropriate (3007/4106).

There is **no need** to include the datasets in the submitted zip file **IF you haven't changed the structure** of them – if you have (e.g. deleted/renamed files) please include your new data as a ZIP called **data.zip** (it should only be about 50MB). Explain the organisation of your new data if you have changed it.

Your .m code must train when run (ie. it must load the datasets provided from subfolders within the working folder: daffodilSeg\ImagesRsz256, daffodilSeg\LabelsRsz256, \17flowers). This is in case we need to re-run it. Each task must save a trained network file in the working folder (names as above), and these models must also be submitted as noted earlier.

If you split your code in multiple files for reusability, please be advised that the main **classification.m** and **segmentation.m** files will be the only ones that will be run during the assessment. Despite that, all the submitted files will be assessed by inspection.

Don't know where to start?

1. Try the HoG/SVM classification lab and the deep learning labs again
2. Work through the CNN examples in the lab. As a first step, try to build a classification network based on examples from the labs.
3. **If you can't complete both tasks** (segmentation and classification), just write up the one network you have managed in the paper, for partial code marks. Still make the paper as thoroughly researched and well written as you can – the full range of marks for the paper will still be available, even if code is incomplete.
4. Use the provided lab to ask questions about these earlier lab sessions if you have any.
5. There will be a chance to ask questions again after the Easter break.

Report:

You need to submit a report that describes your work, written in the style of a scientific paper. A template in Word and LaTeX are provided in Moodle, which is in an IEEE conference paper format. You need to **follow the template** format in terms of font size and layout (double column).

In the report, you must include the following sections:

Abstract

Introduction

Method

Evaluation (including a discussion of results)

Conclusion

References

The length of the report needs to be **minimum of 3 pages but no more than 4 pages (References can be in the 5th page)**. Scientific writing tips were introduced in the tutorials.

The report needs to be submitted in .pdf format on Moodle.

5 Marking Criteria

Matlab code 40%	
Classification approach 20%	Mark will consider implemented approach, quality of code, effectiveness of model on unseen images
Segmentation approach 20%	Mark will consider implemented approach, quality of code, effectiveness of model on unseen images
Report 60%	
20%	Description of **methodology**. Explain the networks/data used.
15%	**Explanation and presentation** of the results obtained.
15%	**Discussion** of the strengths and weaknesses of the chosen approach and methods; conclusions; future work
10%	**Scientific writing** and clarity

A note on accuracy. We will take into account network performance on unseen test images. However, ***most* performance marks come from the implementation and how you explain and evaluate the code in the report**. We realise networks can take a while to train, and so we will not expect them to be perfect. They need to work ok, but you shouldn't need to train for a very long time – that is not the aim here.

The code must train when run (ie. it must load the datasets provided from subfolders within the working folder, e.g. \17flowers, \daffodilSeg\ImagesRsz256, \daffodilSeg\LabelsRsz256.) Explain how you have organised data within those folders.

Each task must save a network model file in the working folder, and these must also be submitted as noted earlier.

NOTE: COMP4106 students will also need to do a presentation component – details to follow.

6 Plagiarism

Copying code or report from other students, from previous students, from any other source, or soliciting code or report from online sources and submitting it as your own is plagiarism and will be penalized as such. FAILING TO ATTRIBUTE a source will result in a mark of zero – and can potentially result in failure of coursework, module or degree. All submissions are checked using both plagiarism detection software and manually for signs of cheating. If you have any doubts, then please ask.

You must NOT use code lifted from GenerativeAI systems such as ChatGPT. We expect approaches to be sufficiently explained in the report. Likewise, the report MUST be your own work. If you use code or piece of text taken from external sources, they must be cited appropriately.