COM2028 *Introduction to AI*
Assessment Overview and Coursework Guideline

February 13, 2019

Overall Coursework Project Description

This coursework concerns the automated recognition of letters and words with the help of AI techniques. Letters and words can be given in various formats. It will be up to you to choose a suitable machine learning technique to achieve the project objectives.

The overall assessment pattern for COM2028 is shown in the following table, which shows the coursework project counts 30% of the module.

Unit(s) of Assessment	*Weighting Towards Module Mark (%)*
Coursework (individual)	30%
Exam (Two Hour Unseen Examination)	50%
Lab exercises and questions	20%
Qualifying condition(s) A weighted aggregate mark of 40% is required to pass the module	

Schedule

Week 1-8: Between week 1 and 8, there are lab exercises each week and questions that students should complete and submit through `SurreyLearn`. Those exercises and questions are mostly self-explanatory. On completion of each lab work, preliminary scores will be obtained through answering questions on `SurreyLearn`. Key solutions to the lab exercises will be available one week after each lab session. A discussion with the lecturer will confirm

the scores with further feedback. This should take place no later than week 8.

Regular engagement with the lab tasks will enable good digestion of the module content and prepare students for the coursework project. Coursework guidelines will be provided before the end of week 5. As the course is progressing, we will start to prepare for the coursework, getting familiar with the related concepts and techniques.

Week 8-10: You will be focusing more on the coursework project. You may find that the foundation built upon all lab activities is crucial for a successful completion of the coursework. Most importantly, with such foundation, working through the challenging tasks in the coursework becomes more enjoyable and satisfying.

Week 10, Tuesday, 7 May 2019 at 4pm: Coursework due. A poster, the developed programs and screenshots of the results should be submitted to `Surreylearn`.

Week 11 and 12 A viva with demonstration of the developed system will take place in week 11 and 12. Feedback on the coursework will be given within 15 working days or before the exam whichever sooner.

Coursework Assessment

Coursework Objectives

- To understand the workflow of developing a useful computer vision system
- To develop machine learning techniques and evaluate their performance

Coursework poster

It is important to demonstrate your understanding of the subject area. There are many useful guidelines on how to design an academic poster, for examples: http://www.surrey.ac.uk/library/documents/learning/SPLASH%205_Poster%20Design.pdf, or http://guides.nyu.edu/posters

Source code and screenshots

Any developed source code should be submitted together with screenshots that demonstrate the results of the developed system.

Viva

Students will present their posters and the developed systems during the viva which involves questions and discussions.

Marking criteria for the poster and viva

- The introduction to the project, providing appropriate background information, and detailing the objectives of the project. Literature review should be carried out on relevant topics and how they may relate to your work. You are expected to find and use other sources of information, but must show their origins by referencing all sources used. If the reference list becomes long and cannot fit within the poster, put a few key reference in the main poster and leave the rest in the appendix. (10/100)
- Analysis of the problems, research and technical issues, challenges, and description of the approach you have developed with justification. Effective graphic illustration will be appreciated. (30/100)
- Implementation and evaluation of the methods. You need to discuss the system results, present the data you used, how many of them for training and testing and what is the recognition accuracy etc. Figures and tables can be useful to present such information, especially when multiple approaches are developed and comparison of various approaches/choices are conducted. (30/100)
- Discussions and findings of your investigation/study . (10/100)
- Clarity in presentation and ability to deal with questions in the viva. (20/100)

You should try to answer the following questions in the poster,

- What are the advantages and weaknesses of these techniques?
- If the system fails to achieve the desired project goals, can you explain why? What might be the issues the processing stages (e.g. data preparation, feature extraction, classification) that give rise to a poorer performance? Is the training appropriate and sufficient, or whether the

training is overfit/underfit? is your evaluation of the developed system thorough enough?

- What is your recommendation?

Contact

Dr Lilian Tang h.tang@surrey.ac.uk

Coursework Project Description

Recognising printed pages

The first part (and actually the easiest) is to extract letters and words from an image of a printed page. So the system will take an image of a printed text, and output a text of the words in that image. There are so many factors that make this task much easier than recognising the words from a handwritten note. The first two steps for this task is to *build a dataset* that is suited for training, and then *train a letter recogniser* based on such dataset. For the training part, you are free to use any of the methods that we have already used in the lab (SVM, K-NN, neural network, and CNN etc) or a combination of them.

Handwritten notes

The second task is to create a similar engine, but this time the engine should work on handwritten notes. Again, your task is to create the training dataset, using your own handwritten notes or those you can find from open source. Note the data should be pages of handwritten notes, then train a system to recognise each letter in the notes. Experiment with different classifiers, and report your findings, comparing with the recogniser that works on printed pages.

Signature recognition

Using similar techniques, build a system that accepts as an input an image of a signature, and returns the name of the person that this signature belongs to. Of course, you will have to train the system, using different people's signature data. You can start with only two people, but it would be worth to add more signature data for each person, and see how well your system can deal with that. The more training data you have, the better the system should learn. Of course, you need to make sure the testing data are not included in the training. Again, you can choose the type of classifier as well as the kind of features to use, you can experiment in order to find out the best choice (and report on why you have selected this specific configuration).

NOTE: Please make sure you have asked for permission if you are collecting signatures from other people so they are happy for you to use them in your study. You may find some public data from open source.

Building a writer identification system

Team up with someone, and exchange handwritten training data. Try to experiment with building an engine, that is able to understand *who* is the author of these writings. Try different configurations, and different ideas. Since this is a purely research oriented question, you need to be able to identify possible paths to the success. For example, what kind of features should you use? Raveled images, or HoG features? or automated learned feature through CNN? If you use HoG features, how large should the letters be in order to deal with the problem successfully?

Testing - Evaluation

For the testing stage, a simple comparison on the percentage of correctly classified letters would be enough. You can even go to a higher level, and incorporate a word analysis facility into the system. However, by doing that, it may mean that you also have to add text related capabilities to your system (for example a simple word or grammar checker etc).

For the writer identification task, the evaluation can be in a form of running the test on $2 \times m$ images, with m images from each person, and reporting the percentages of correct vs wrong classifications. The similar applies to the signature recognition system.

Tips

- Note that in order for the labeling and the letter segmentation to work well, your handwriting should be non-continuous! Do not stick the letters and the words together, it will be much more difficult to segment a word that has no spaces between the letters. For those interested, it is still possible to segment such words and understand the letters, but it involves sliding windows and averaging responses, something that is out of the scope of this coursework.
- Obviously for both cases, the easiest way to create the training data is to fill out pages with single letters. This will make it easy to program the whole system. For example if you want to train the system to recognise the letter *a* just fill out a page with a number of *a*'s, lets say 100, and then pass these 100 objects to your recogniser together with the label that represents the letter *a*.
- Multiple fonts. You can make your system more robust, by using multiple different fonts in the training process (of course for the task on recognising printed page).
- Some training and testing samples for print text recognition are provided in `WritingRecognition.zip`. You can use these data as a starting point. You can extend the training and testing sets as you wish.
- The program `extract_letters.py` is given, and it has a basic method of segmenting a page and returning the recognised letters.
- Plot the results using `matplotlib`. It is always better to visualise the results, and you can use `matplotlib` to do that. You can plot graphs on the success rates, and the recognition rates etc.

Letter size normalisation

If you take a look at the code in the `extract_letters.py` program, you will see that what this program does is to create bounding boxes for all the letters in a scanned image of a document.

By using the variable `rect`, you are able to get the bounding box for each of the letters. As we have seen before, you can use `numpy`'s slicing abilities, in order to get a part of a bigger matrix. For example if the variable `image` is a matrix `[100,100]` and you only need a small part that is `[10,10]` in the top left part of the image, you can write

```
part = image[0:9,0:9]
```

The syntax `list[x:y]` means returning the part of the list that is between `x` and `y`.

So you can use exactly the same logic, in order to extract each letter from the original image using the bounding box.

scipy.imresize

Letters have different shapes and sizes. So it it better if you normalise everything for easier classification process. A logical size should be around `[20,20]`.

```
1 letter_raw = image[bounding_box[0],etc....]
2 letter_norm = imresize(letter_raw,(20,20))
```

Letter extraction and size normalisation

Now you can use `letter_norm` both for training purposes and testing purposes. You *have* to keep everything constant, during the training and the testing process. For example, don't train your system with `[20,20]` patches and then try to run it on `[20,15]` patches.

Example of getting the letters from an image

Space exploration has long been about reaching far off destinations but now there's a race to exploit new frontiers by mining their minerals.

When Neil Armstrong first stepped on the Moon in 1969, it was part of a "flags and footprints" strategy to beat the Soviets, a triumph of imagination and innovation, not an attempt to extract precious metals.

No-one knew there was water on that dusty, celestial body. What a difference a generation makes.

Mysterious and beautiful, the Moon has been a source of awe and inspiration to mankind for millennia. Now it is the centre of a space race to mine rare minerals to fuel our future - smart phones, space-age solar panels and possibly even a future colony of Earthlings.

"We know that there's water on the Moon, which is a game-changer for the solar system. Water is rocket fuel. It also can support life and agriculture. So exploring the Moon commercially is a first step towards making the Moon part of our world, what humanity considers our world," says Bob Richards, CEO of Silicon Valley-based Moon Express, one of 25 companies racing to win the $30m in Google Lunar X Prizes.

It is considered to be among the top-three teams in the running for the prize. The other two are Pittsburgh-based Astrobiotic and Barcelona Moon Team.

Figure 1: The original image that contains a chunk of text

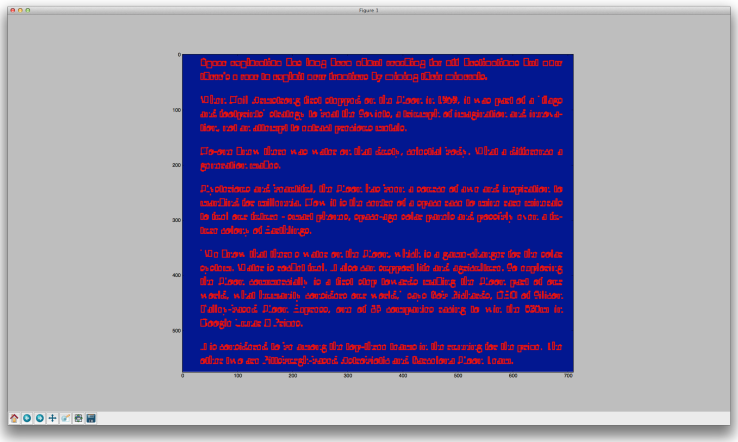

Figure 2: Extracting all the elements from the image. Similar to what we did in Week 2 with the coins, but for much more elements. *(zoom in to see clearer details)*

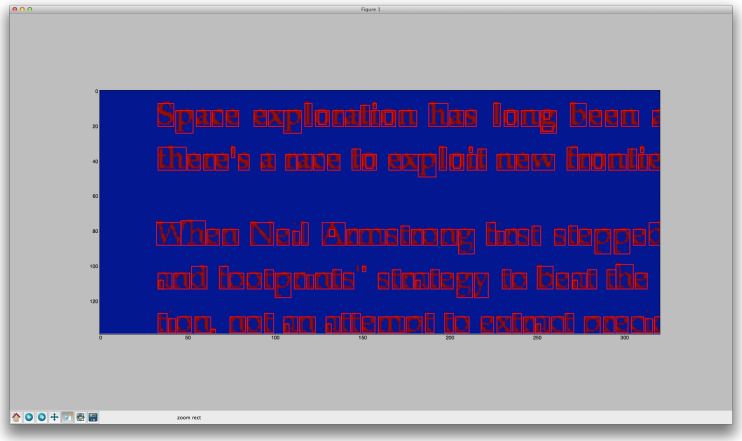

Figure 3: The first part from the image above *(zoom in to see clearer details)*