Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CM2203
Module Title: Informatics
Lecturer: Dr Sylwia Polberg
Assessment Title: Informatics Portfolio 2
Assessment Number: 2 out of 3
Date Set: 23rd of February 2022
Submission Date and Time: 25th of March at 9:30am
Return Date: 6th of May 2022

This assignment is worth **30%** of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1   If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;

2   If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:

https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

Description	Type		Name
Cover sheet	**Compulsory**	One PDF (.pdf) file	[student number].pdf
Source code	**Compulsory**	One .zip folder containing the code	[student number].zip
README	**Compulsory**	One .txt file	[student number]_README.txt

The files need to be submitted to appropriate parts in the Assessment→ Portfolio 2 area on Learning Central. Any code submitted will be run on a system equivalent to those available in the Windows laboratory and must be submitted as stipulated in the instructions above.

Any deviation from the submission instructions above (including the number and types of files submitted) will result in a mark of zero for the assessment or question part OR a reduction in marks for that assessment or question part.

In all coding tasks, please use one of the following: Java (JDK 1.14 or JRE 1.8) or Python (version 3.8). **If you require a different version, please contact the module leader.** In the README file, state which IDE you have used. **Use of Maven, Google Colab and Jupyter Notebook is not permitted.**

Staff reserve the right to invite students to a meeting to discuss coursework submissions

Automatic anti-plagiarism and similarity checking tools can be used to process the submissions.

You can submit multiple times on Learning Central. ONLY files contained in the last attempt will be marked, so make sure that you upload all files in the last attempt.

Assignment

In this assignment, we are going to use excerpts from the following datasets:
- AI Generated Faces from Generated.Photos
  https://generated.photos
- Turath-150K Image Database of Arab Heritage
  https://danikiyasseh.github.io/Turath/
- ANIMAL-10N Dataset
  https://dm.kaist.ac.kr/datasets/animal-10n/
  Song, H., Kim, M., and Lee, J., "SELFIE: Refurbishing Unclean Samples for Robust Deep Learning" In Proc. 36th Int'l Conf. on Machine Learning (ICML), Long Beach, California, June 2019

The assignment is worth 30 points in total and is compromised of the following tasks. The classification scheme as well as the data can be found in the .zip file accompanying this portfolio. Please also use the attached template; if the template uses a different programming language than you want to use, please contact the module leader.

You can only use standard libraries that come with the language you have picked unless stated otherwise. For example, calling a kNN classifier from a scikit-learn package instead of implementing your own from scratch will yield 0 points.

Additional clarifications concerning this portfolio may be posted on the discussion board on Learning Central, so please remember to check it.

Task 1 [10] My first not-so-pretty image classifier

By using the kNN approach and three similarity measures, build image classifiers. **You need to implement the kNN approach yourself**, however, you can use libraries for any similarity measures (remember that some measures can make assumptions on the sizes of images etc.) You can assume that k=100 (if the code takes too long to run, feel free to decrease it to as low as k=10). You are allowed to use libraries to read and write to files, and to perform image transformations if necessary.

Task 2 [2] Basic evaluation

Evaluate your classifiers. **On your own,** implement methods that will output precision, recall, F-measure, and accuracy of your classifiers.

Task 3 [6] Cross validation

Evaluate your classifiers using the k-fold cross-validation technique covered in the lectures (use the training data only). Assume the number of folds is 100 (if the code takes too long to run, feel free to decrease it to as low as 10 folds). Output their average precisions, recalls, F-measures and accuracies. **You need to implement the validation yourself.**

Task 4 [3] The curse of k

Independent inquiry time! Picking the right number of neighbours k in the kNN approach is tricky. Find a way you could approach this more rigorously. In comments, state the approach you could use, and provide a reference to it. The reference needs to be to a handbook or peer-reviewed publication; a link to an online tutorial will not be accepted.

Task 5 [6] Similarities

Independent inquiry time! In Task 1, you were allowed to use libraries for image similarity measures. Pick two of the three measures you have used and implement them yourself!

Task 6 [3] I can do better!

Independent inquiry time! There are much better approaches out there for image classification. Your task is to find one, and using the comment section of your project, do the following:
- State the name of the approach, and a link to a resource in the Cardiff University library that describes it
- Briefly explain how the approach you found is better than kNN in image classification (2-3 sentences is enough).

Learning Outcomes Assessed

1. Execute and evaluate various techniques in knowledge discovery and data mining.

Criteria for assessment

Credit will be awarded against the following criteria. The tasks are connected and increase in difficulty.

Task	Fail (0-39%)	3rd (40-49%)	2.2 (50-59%)	2.1 (60-69%)	1st (70-100%)
1	There are severe errors in the code Solution is missing or is completed in a way that breaks assessment requirements	The core kNN algorithm is present The code is generally bug-free and correct At least one appropriate similarity is used The code is reasonably documented	The core kNN algorithm is present The code is bug-free and correct At least one appropriate similarity is used The code handles all the inputs and outputs properly as given in the template The code is reasonably documented	The core kNN algorithm is present The code is bug-free and correct Most of the requested similarities are present, and are appropriate for the task The code handles all the inputs and outputs properly as given in the template The code is well documented	The core kNN algorithm is present The code is bug-free and correct The code handles all the inputs and outputs properly as given in the template All three similarities are present and appropriate The code is very well documented
2	0.5 points is awarded per correct implementation of each of the requested measures [2 points in total].				
3	The implementation is missing or has major problems	The validation is implemented The code has moderate issues	The validation is implemented The code is mostly bug-free and correct	The validation is implemented The code is bug-free and correct	The validation is implemented The code is bug-free and correct The average measures are calculated and outputted The code is very well documented

4	Unsuitable approach is proposed Reference is missing	A good approach is proposed The reference could use improvements	A good approach is proposed The reference material is good but not cited all that well	A good approach is proposed The reference material is good, but citation could use improvement	A good approach is proposed The reference material is good, and the citation follows the approach approved by the university
5	Nothing is done Code is seriously flawed Documentation is missing	One similarity is correctly implemented and fairly documented	One similarity is correctly implemented and well documented A start has been made on the second similarity	One similarity is correctly implemented and well documented Progress has been made on the second similarity	Each similarity is correctly implemented Each similarity is well documented
6	No approach is stated Explanation is missing	A good approach is stated The explanation is correct, but not entirely clear	A good approach is stated The explanation is correct and sufficiently clear	A good approach is stated and well referenced The explanation is correct and sufficiently clear	A good approach is stated and well referenced The explanation is clear, correct, and succinct

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on the 6th of May via Learning Central and/or email.

Feedback from this assignment will be useful for any other modules requiring programming or machine learning.