

**Coursework Administrative Details**

Module/Lecture Course:	Advanced Computer Vision
Deadline for submission:	25/4/2023 14:00
Deadline for marks and feedback to be returned to students:	After the Board of Examiners
Submission instructions:	Please submit all files on Blackboard.
Submission file type(s) required:	A PDF file for the report. A ZIP file containing everything else, including the Jupyter notebook, multimedia files, optionally additional .py and .sh files.
Format:	Report as a PDF document. Accompanying implementation and multimedia files in a ZIP file
Contribution:	The report contributes 100% to the final mark for the module.

In accordance with University procedures, **submissions that are up to 5 working days late will be subject to a cap of the module pass mark, and later submissions will receive a mark of zero.**

Content and skills covered by the assignment:

- Have a strong understanding of how to work with images and video, and to usefully manipulate visual data
- Have a good understanding of advanced deep learning models for classifying and understanding images.
- Train and test generative deep learning models using real-world data.
- Effective written communication.
- Planning, organising and time management.
- Problem solving and analysis.

General Requirements

Students are expected to work on the coursework individually.



Students are expected to:

- 1- Implement human feature extraction and style translation networks
- 2- Understand the limitations of existing techniques, especially with unconventional or limited datasets, and the importance of appropriate image augmentation and algorithm selection
- 3- Understand how deep learning models might fit back into a real-world pipeline

Submission

Please follow the instructions very carefully to avoid any score penalty.

You should submit your report in PDF, along with a ZIP file containing the Jupyter notebook that can replicate (within reason), the multimedia files as requested in the questions, and optionally additional .py and .sh files.

To maintain a small file size, please apply proper image and video compression to the multimedia files. For the same reason, please do not re-submit the original mp4 data files. If you use other datasets, models, etc, please include lines in the code that download them automatically (e.g. “!git clone xxx” or “!wget xxx”).

You may find that you have too much code to be included in a single Jupyter notebook, and that you wish to move bits of code into separate, importable python files. This is fine; in this case, please include a Jupyter notebook in the root directory, which imports functions locally.

Assignment Questions

The techniques of computer graphics used in games have improved significantly in the past few years. Thanks to the highly realistic rendering in games, it has been proposed to utilise games video, i.e., synthetic images, to improve the quality of old real-life movies. In particular, movies produced in the 70s usually suffer from a range of visual artifacts, such as washed-out colour and limited contrast, due to the hardware limitations at the time.

In this assignment, you will implement a deep learning solution to enhance the visual quality of human beings in 70s movies using game videos. While it is possible to translate image styles using the whole image, the results would not be good as foreground image styles (i.e. human beings) are mixed up with background styles. As the focus is to enhance the visual quality of human beings, it is preferable to extract the patches of pixels that are relevant to humans using different human features.

A dataset is available on Blackboard for this assignment, which consists of .mp4 video files from movies in the 70s styles and from a video game.



1. Human Feature Analysis

1.1. Human Patch Extraction: For both the game and movie videos under the “Train” folder, adapt a deep learning method to detect individual humans and extract the human image patches. Store each patch as one image file, and gather 1,000 (or more) image files of human patches. Randomly sample and submit 50 image files. Explain, justify and evaluate the method you adapted. Max words: 100. **(10%)**

1.2. Classification: Using the images created in 1.1, propose and develop a method that classifies each image into one of the five classes: full-body standing, full-body sitting, half-body, head-only, and others (you may need to define what each class includes in your system). Randomly sample and submit 10 image files per class. Explain, justify and evaluate your method. Max words: 100. **(10%)**

1.3. Training Data Selection: From the images created in 1.1 and considering the findings in 1.2 as well as your own insights, propose and develop a method that selects the most useful images for human style transfer training (in the next stage). Randomly sample and submit 50 image files. Explain and justify your method. Max words: 100. **(10%)**

1.4. Performance: Identify possible causes of inaccuracy in Question 1.1, and discuss possible solutions to improve accuracy, with consideration of computational resources. Max words: 100. **(10%)**

2. Real-world application

2.1. Image Model Deployment: *Implement any unpaired image-to-image network (e.g. Cyclegan) for converting images between the game and movie domains. Download a pretrained model or train from the beginning, apply it to your dataset of frames, and analyse its performance in both directions using appropriate metrics. Compare successful and failure cases using at least 10 images for each and display them in the report.* Max words: 250. **(20%)**

2.2. Local (temporal) Enhancement: Using your 2.1 model, transfer the style of humans in the *video* under the “Test” folder and submit the *video*. Critically *evaluate* the result and suggest possible *causes* of failure. *Design* an improved model using local methods in 1.1 to 1.3 or other *advanced* temporal approaches. Create and submit a *new video* and *display* at least 10 images of key frames to *compare* and show the *improvement*. Max words: 250. **(30%)**

3. Overall: Marks for good scientific report writing (clarity, brevity, precision, good use of diagrams/tables/visualisations, etc.): **(10%)**

Reports and Guidelines



The report format should follow the instructions given in the questions. In all your answers, we are expecting to see enough detail (e.g. some maths, analysis of results, visualisations, references to papers or other materials) to demonstrate your understanding of the subject. There should be diagrams, figures, and tables to demonstrate the results and analysis. There should also be references where suitable to support and justify your solution.

The coursework aims to evaluate knowledge and understanding of both the fundamentals and the recent advances in computer vision, and not the student programming skill. Therefore, we ask you to implement the solutions using any Python libraries you are most comfortable with. You are free to re-use any pre-trained model, pre-written library/implementation or extra datasets as you see fit, as long as attribution is clearly given in both the code and the report. However, simply lifting existing codebases—without adapting them to the problem at hand, or otherwise demonstrating an understanding of how they work—will not result in high marks.

You can complete this assignment on Google Colab, or on the NCC (you will need to register for an account). Please note that debugging on the NCC is much less practical, and the resources offered by both services are similar.

Word Limit policy

The word count for each question will:

- *Include* all the text, including in-text citations, quotations, footnotes and any other item not specifically excluded below.
- *Exclude* diagrams, tables (including tables/lists of contents and figures), equations, executive summary/abstract, acknowledgements, declaration, bibliography/list of references and appendices. However, it is not appropriate to use diagrams or tables merely as a way of circumventing the word limit. If a student uses a table or figure as a means of presenting his/her own words, then this is included in the word count.

Examiners will stop reading once the word limit has been reached, and work beyond this point will not be assessed. Checks of word counts may be carried out on submitted work. Checks may take place manually and/or with the aid of the word count provided via electronic submission.

Plagiarism and collusion

Your assignment will be put through the plagiarism detection service.

Students suspected of plagiarism, either of published work or work from unpublished sources, including the work of other students, or of collusion will be dealt with according to Computer Science Department and University guidelines.



FAQ

Please read the FAQ section before asking a question.

Q. *“Can you tell me exactly what method I should implement in doing Question X?”*

A. Unfortunately no. One of the core parts of this assignment is for you to propose/design your solution based on the knowledge you obtained within and outside this module. This is very similar to a real-world working environment, where you need to suggest and propose solutions to your supervisor when working on a project. Telling you what methods to implement would deflect this purpose and this part of the evaluation.

Q. *“If I do X, what grades will I get?”*

A. We cannot “pre-grade” your assignment since this will be unfair to other students who do not share this piece of information. However, we do have a formative feedback session where we can tell you how you can improve your solutions; please make sure you attend that. Also, please read this assignment brief very carefully to understand what the assignment requires.

Q. *“I’ve spent a million hours training these models and I still need more time. This assignment is impossible!”*

A. This is often a problem with deep learning. Some models may take a long time to train; so you need to choose your experiments carefully. These problems are intentionally difficult; you are not expected to produce near-perfect outputs. You’ll never have enough computing power to try everything; well-reasoned, well-explained models with poor results can still get good marks.

Q. *“This would be far easier if I had an expensive GPU machine!”*

You are not being examined on how much GPU power you’ve bought. Please be explicit in your reports about the hardware you used to train your model, and the time it took. Better performing models will not score higher marks where this increase in performance is judged to have come solely from better hardware (e.g. because they have been trained for many more epochs, or because you were able to test many different hyperparameter combinations). Equally, everyone has access to the same basic resources (e.g. Google Colab): so waiting until just before the submission deadline to start training your models is a very bad idea!

Q. *“I found code online which looks similar to what I need. Can I use it?”*

A. Yes, but you must cite the code in both the written report and in the comments at the top of the code. I will then carefully cross-reference this code with your implementation to see how well you have adapted their code. If you have simply copied and pasted without evidence of experimentation or tailoring, do not expect to get a



good grade even if your results are good. Whereas if I see evidence of original interpretation, novel comprehension and application of the theory in the lectures, even if the experimental results are not as strong, you can get very high marks. However, if you pass off other people's code as your own, and "forget" to cite their code, or work together with other students, you will very likely get caught (see the submission Plagiarism and Collusion section on Ultra to read about the tools used to detect this). This incurs a very severe departmental penalty.

Q. "I just read this interesting new paper which is very different to techniques from the course. Can I use it?"

A. Yes! You can use any papers, and even any (open-source) implementations, you like. Make sure you cite it properly in the code and reports: see the previous question.

Q. "Can I use external datasets or even pretrained models?"

A. Yes! But as with external bits of code, make sure you give proper attribution in the report and your code: and make sure you have adapted it to suit your purposes. Avoid obscenely large additional datasets, that require much greater computing power:

Q. "Is downloading a pre-trained model an essential step to be assessed?"

A: No. A pre-trained model can be from irrelevant tasks, e.g. Zebra-Horse or other game-video transfer models. Training from scratch with your own design of model architecture is also fine.

Q. "What will be the difference between using a pre-trained model or not?"

A: You are welcome to build up your own models. For pre-trained models, there are pre-trained movie-game models but not necessarily to be CycleGAN.

If you choose to design your own model, you are more flexible to focus on discussing the impacts of architecture, e.g. the number of layers, adding/removing res-blocks.

If you choose to apply a pre-trained model, you can discuss the domain-shifting problem, e.g. the difference between source data distribution and the target movie-game domain distribution.

Q. "What are the relationship between 2.1 and 2.2 and do we implement one or multiple models?"

A: 2.2 aims to improve 2.1 and there are two ways. (1) Using methods in 1.1 to 1.3, you can improve the same model in 2.1 by using local information. (2) Developing an advanced model that can consider the temporal information in videos and get extra credit.

Q. "Many questions in section 2 is associated with that in section 1. Do we train an end-to-end model for both questions, or do we train separate models for individual questions?"



A: Both are fine, you have the flexibility. Apparently, build separate models for each single task is easier. Combining multiple tasks into one end-to-end model can be difficult and gives very interesting results. If you have got one model that provided reasonable results, you can demonstrate a second sophisticated solution (multi-task losses of detection, classification, temporal and style transfer) to get extra credit even though the results may be not perfect.