



Machine Learning, Course Project 2018

Important – Read before starting

- The deadline for completing and submitting your assignment is strictly Wednesday 23rd January 2019 at 18:00.
- VLE will be set up to not accept late submissions meaning that you will get zero marks if your submission is late. Please plan ahead (it is recommended that you upload and verify your work a day before).
- You must complete the project completion form (shown later) and include it in your report. Submissions without the statement of completion will not be considered.
- You must complete a plagiarism declaration form and include it in your report. Submissions without the form will not be considered.
- Projects must be submitted using VLE only. Physical copies or projects (including parts of) sent by email will not be considered.
- For your convenience, a draft and final submission area will be set up in VLE. Only projects submitted in the *final* submission area will be graded. Projects submitted to the draft area are not considered.
- It is suggested that after submitting your project, you redownload it and check it again. It is your responsibility to ensure that your upload is complete, valid, and not corrupted. You can reupload the assignment as many times as you wish within the deadline.
- Your project must be submitted in ZIP format without passwords or encryption. Project submitted in any other archiving format will not be considered.
- The total size of your ZIP file should not exceed 38 megabytes.
- Your submission should include your report in PDF format, your source code, and executable file(s).
- It is expected that you submit a quality report with a proper introduction, discussion, evaluation of your work, and conclusions. Also, make sure you properly cite other people's work that you include in yours (e.g. diagrams, algorithms, etc...).
- In general, I am not concerned with which programming language you use to implement this project. However, unless you develop your artifact in BASIC, C, C++, Objective C, Swift, Go, Pascal, Java, C#, Matlab, or Python, please consult with me to make sure that I can correct it properly.
- This is not a group project.
- Plagiarism will not be tolerated.

Fast Frontal Face and Eye Detection using Viola-Jones Object Detection

- As part of your report, write a technical section about how Viola-Jones Object Detection/Haar Cascades work. Discuss why they are especially useful for real-time applications such as face detection in digital cameras. Three to five pages worth of good material should suffice. Don't rip off Wikipedia.
- Note regarding any artifacts you develop: do not implement the training or classification components yourself. Use OpenCV (<https://opencv.org>).
- You are required to implement the following two artifacts:
 1. Obtain a dataset containing faces and non-faces (you'll find plenty of these datasets on the internet) and use OpenCV to train a Cascade classifier. OpenCV have a user guide to help you get started. Present and discuss your results in the evaluation section of your report.
 2. Obtain frontal face and eye cascades from here: <https://github.com/opencv/opencv/tree/master/data/haarcascades>. Write a program which uses these two cascades to detect faces and eyes in images you input to the program. Faces should be outlined with a green rectangle, and eyes should be outlined in red. If your computer has a webcam, it is easy (and very cool) to do real-time video tracking using OpenCV.
- In your report, briefly discuss any alternative learning algorithm which is suitable for face/eye detection in images (you choose). Compare it to Viola-Jones.
- Make sure that your report has a good evaluation section for any artifacts you develop.

Statement of completion – MUST be included in your report

Item	Completed (Yes/No/Partial)
Viola-Jones technical discussion	
Artifact 1	
Artifact 2	
Comparison to alternative algorithm	
Experiments and their evaluation	
Overall conclusions	