UNIVERSITY OF LONDON

Intelligent Signal Processing

Module: Intelligent Signal Processing

Coursework: end-of-term

Submission Deadline: Please refer to the Coursera.

- Please Note: You are permitted to upload your Coursework in the final submission area as many times as you like before the deadline. You will receive a similarity/originality score which represents what the Turnitin system identifies as work similar to another source. The originality score can take over 24 hours to generate, especially at busy times e.g. submission deadline.

- If you upload the wrong version of your Coursework, you are able to upload the correct version of your Coursework via the same submission area. You simply need to click on the 'submit paper' button again and submit your new version before the deadline.

  In doing so, this will delete the previous version which you submitted and your new updated version will replace it. Therefore your Turnitin similarity score should not be affected. If there is a change in your Turnitin similarity score, it will be due to any changes you may have made to your Coursework.

- Please note, when the due date is reached, the version you have submitted last, will be considered as your final submission and it will be the version that is marked.

- **Once the due date has passed, it will not be possible for you to upload a different version of your assessment. Therefore, you must ensure you have submitted the correct version of your assessment which you wish to be marked, by the due date.**

Your overall total word count should not exceed 2500 words (Weighted at 50% of final mark for the module)

Coursework Description

The End of term assignment for Intelligent Signal Processing consists of a series of three individual exercises. These exercises cover the last five topics of the course:

- Capturing, representing and processing camera input
- Computer vision: movement detection
- Computer vision: face detection
- Audio file formats. Compressing audio signals
- Video file formats. Compressing video signals

The exercises are strongly based on the subjects covered during the course, but also invite the student for further investigation.

It is recommended that the students carefully read all the sections of this document, both to ensure a good understanding of the coursework exercises, in addition to knowing what to submit.

There are three exercises, and you are required to submit the following files:

A. One PDF file containing a report for all exercises.
B. One PDF file containing code you developed with annotations for all exercises.
C. One ZIP file (containing sub-folders that include files for each exercise, i.e., ex1, ex2, …, ex <*N*>.)
D. One Video demonstration for all exercises in an MP4 format (maximum six minutes duration)

Please note you MUST submit the supporting
(B) Code PDF file & (D) Video demonstration
of all exercises otherwise
NO MARKS will be awarded for
(C) *code implementation tasks*.

Exercise 1

The city of Laramie (Wyoming, US) will launch a series of initiatives in order to improve the city planning, traffic flow, public transportation and parking efficiency.

One of these initiatives consists of developing an automated traffic monitoring system using computer vision, and the major of the city has contacted you to develop this system.

First, to evaluate your skills, you are asked to perform two tasks, which consist of developing two motion detection applications.

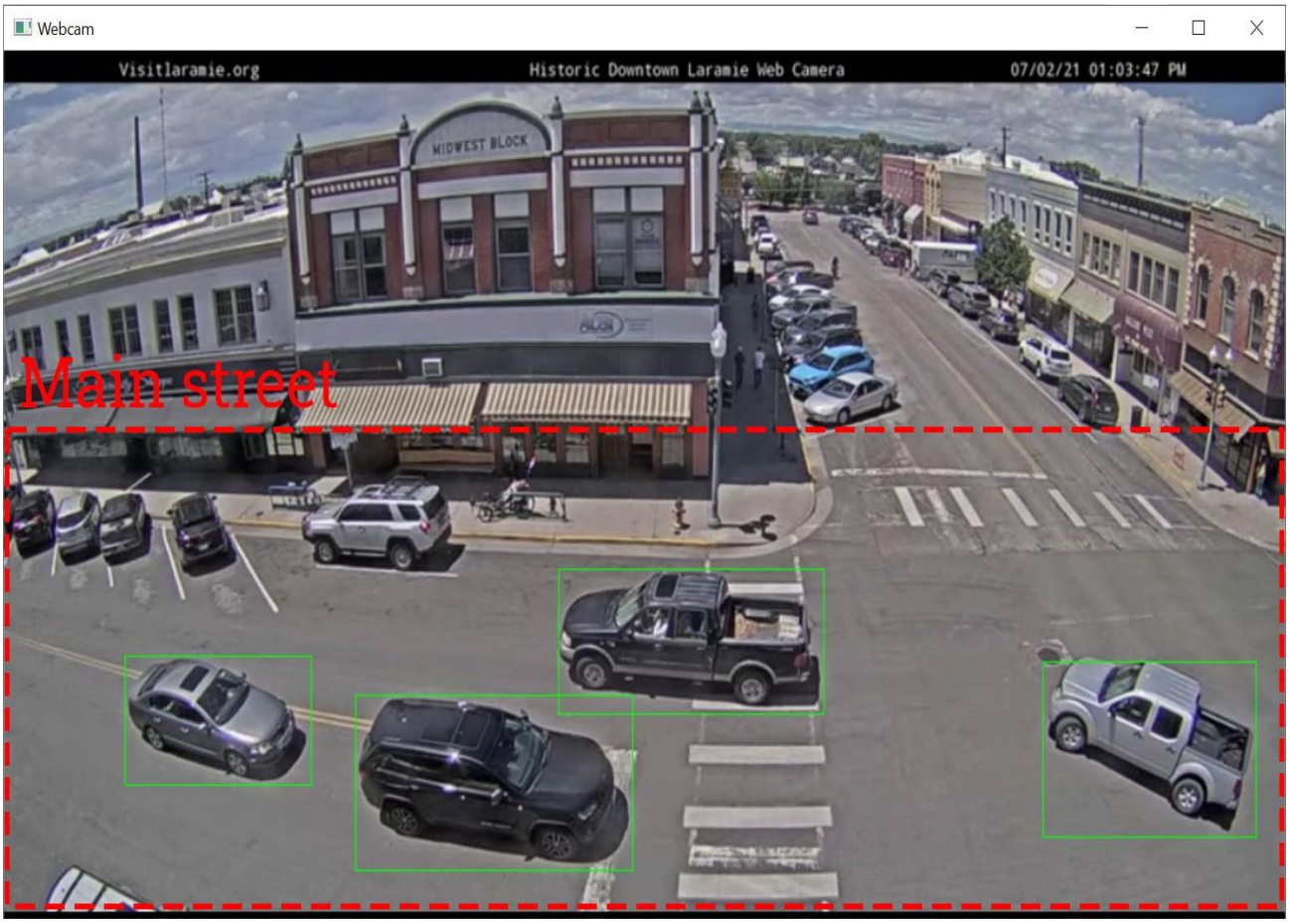
Task 1

The first task consists of developing an application (*exercise 2.1*) for detecting and tracking moving cars from a camera recording.

The application must run on a Jupyter notebook written in Python and use the OpenCV library.

Technically, the algorithm of the application must be based on the frame differencing and background subtraction techniques.

To verify that the application is operating as expected, your client has sent you a recording from one of the city traffic cameras (Traffic_Laramie_1.mp4), and the following image showing the expected results. (Note: your application can focus only on detecting cars that are in the 'main street').



Task 2

In order to prevent traffic jams, your client is also very interested in the number of cars that go from the city's downtown to the city centre in peak hours.

Therefore, Task 2 consists of developing a computer vision application (*exercise 1.2*) for counting the number of cars that go from the city's downtown to the city centre for a specific time interval.

As in Task 1, the application must run on a Jupyter notebook written in Python, use the OpenCV library, and be based on the frame differencing and background subtraction techniques.

To verify that the application works correctly, you should use the recordings Traffic_Laramie_1.mp4 and Traffic_Laramie_2.mp4 provided by the client and fill in the following table with the data generated by the application.

	Total number of cars	**Cars per minute**
Traffic_Laramie_1.mp4		
Traffic_Laramie_2.mp4		

List of deliverables

For Exercise 1, you should submit in a ZIP file:

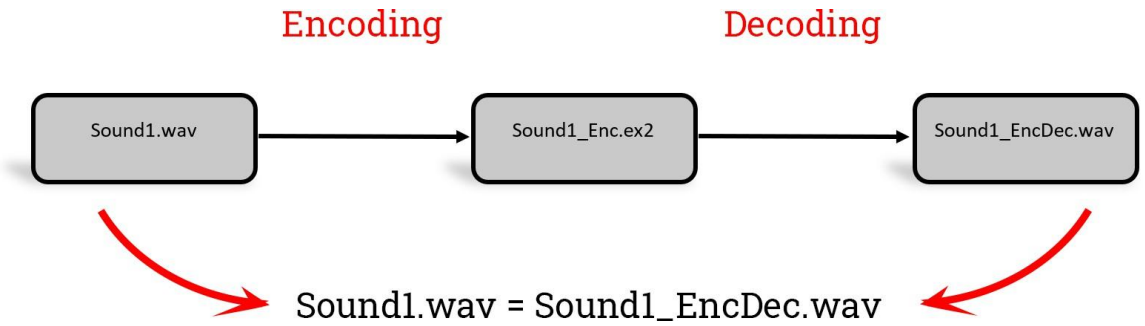
- The Jupyter notebooks of the applications exercise 1.1 and exercise 1.2.
- A written report in PDF format, approximately 500 words. This report must include:
  - The table of Task 2.
  - A brief description of the frame differencing and background subtraction techniques.
  - A brief analysis of the application.

Exercise 2

As we already know, data compression can be lossless or lossy. Lossless compression techniques involve no loss of information, while lossy compression techniques reduce file size by eliminating data in the file.

In this exercise, you are going to create your own application (*exercise 2*) for encoding and decoding a series of uncompressed audio files (WAV files). The application must run on a Jupyter notebook written in Python and be based on the lossless data compression method 'Rice coding'.

Therefore, if Sound1.wav is an uncompressed audio file, the application should be able to encode this file and save it in your drive with the name Sound1_Enc.ex2 (we can use, for example, the extension 'ex2' to identify encoded files). Then the application should be able to decode the file Sound1_Enc.ex2 and create a new WAV file, with the name Sound1_Enc_Dec.wav. As the application is based on a lossless data compression method, the files Sound1.wav and Sound1_Enc_Dec.wav should be exactly the same.



To analyse the effectiveness of the Rice coding algorithm with bit length K = 2 bits and K = 4 bits to compress the audio files Sound1.wav and Sound2.wav, fill in the following table.

	Original size	**Rice (K = 4 bits)**	**Rice (K = 2 bits)**	**% Compression (K = 4 bits)**	**% Compression (K = 2 bits)**
Sound1.wav					
Sound2.wav					

Ideas for further development:

1. Find ways to improve the compression rates obtained in the previous table. You can try to 'improve' the Rice coding algorithm you have used in your application or implement another lossless data compression method.

List of deliverables

For Exercise 2, you should submit in a ZIP file:

- The Jupyter notebook of the application *exercise 2*.
- A link to the application running in a Jupyter notebook link in Coursera.
- A written report in PDF format, approximately 500 words. This report must include:
    - A brief analysis of the application.
    - The requested table and an analysis of the results. In particular, justify why the compression rates of the files are so different.
    - A brief description of the further development implemented.

Exercise 3

The Narbonne Online Film Festival receives every year more than one hundred films in digital format.

The problem is that some of these films are not submitted in the format specified by the festival organisation, and they must be converted.

The organisation wants to automate this process by developing an application (*exercise 3*) for examining the format of the films and, if necessary, convert them.

You have been selected by the festival to develop such application. The application should meet the following requirements:

1. The application must run on a Jupyter notebook written in Python, use ffprobe to examine the files, and ffmpeg to convert the films.
2. From a series of video files, the application should generate a brief report in TXT indicating which films do not respect the digital format specified by the festival and what are the 'problematic' fields.
3. The application should automatically convert the submitted films with an incorrect format. The program will create a new copy of the films by adding '_formatOK' at the end of the name.
4. The format of the films specified by the festival organisation is:
   - Video format (container): mp4
   - Video codec: h.264
   - Audio codec: aac
   - Frame rate: 25 FPS
   - Aspect ratio: 16:9
   - Resolution: 640 x 360
   - Video bit rate: 2 – 5 Mb/s
   - Audio bit rate: up to 256 kb/s
   - Audio channels: stereo

To verify that the application works correctly, the film festival organisation has sent you a series of film excerpts in a ZIP file (see attached file Exercise3_Films.zip).

List of deliverables

For Exercise 3, you should submit in a ZIP file:

- The Jupyter notebook of the application *exercise 3*.
- A link to the application running in a Jupyter notebook link in Coursera.
- The report in TXT that the application examining the films submitted by the organisation.
- A written report in PDF format, approximately 500 words. This report must include:
  - A brief description of how ffprobe and ffmpeg has been installed and configured in your machine.
  - A brief analysis of the application.
  - A brief description of the terms video format (container), video codec, audio codec, frame rate, aspect ratio, resolution, video bit rate, audio bit rate and audio channels.

Exercise 4

In this exercise, you are required to investigate and document emerging computer vision and audio processing applications used to address real-world problems using suitable academic resources (*i.e., books, conference/journal papers, and some credible websites*).

1. Review and critically discuss at least three emerging computer vision applications that you find most interesting with justifications.
2. Provide technical discussions on two popular or state-of-the-art computer vision techniques adapted with suitable diagrams, where relevant.
3. Discuss two potential creative and novel uses of these computer vision and audio feature processing suitable for addressing real-life problems.

List of deliverables

For Exercise 4, you should write a report of approximately 1000 words to concisely capture the above areas of discussion.

Deliverables

A. Exercises Report

Submit a single report in a PDF format containing discussions on tasks carried out for all exercises. Generate separate links and include them in the report as discussed in the exercises.

B. Code report

Copy/paste the textual code you have developed for all four exercises with appropriate annotations in a single PDF file.

C. Code files

The source code of the application for all four exercises is in separate folders and compressed in one ZIP file.

D. Video demonstration

A screencast recording demonstrating that the application meets all the exercise requirements in an MP4 format (maximum length of SIX minutes).

- Please ensure the screencast recording contains your voice-over as a minimum and no background music that is irrelevant to the exercises.
- For screen recording you can use applications such as Google Chrome Extensions such as Screen recorder, and Screencastify - Screen Video Recorder. For video editing, you can use open source tools: Shotcut, kdenlive or other tools such as DaVinci, VideoPad, Movie Maker, VSDC, Openshot, and iMovie.

Marking Criteria:

Description	Marks
Exercise 1	
The Jupyter notebook for *exercise 1.1* correctly detects and tracks the cars on the recording Traffic_Laramie_1.mp4.	2
The Jupyter notebook for *exercise 1.1* installs the necessary libraries for running the application (using the package-management *pip*) and includes markdown cells describing the code in detail.	1
The Jupyter notebook for *exercise 1.2* correctly counts the number of cars that go from the city's downtown to the city centre.	3
The Jupyter notebook for *exercise 1.2* installs the necessary libraries for running the application (using the package-management pip) and includes markdown cells describing the code in detail.	1
The written report includes a brief description of the frame differencing and background subtraction techniques.	2
The written report includes the table of Task 1 and a brief analysis of the application.	1
Exercise 2	
The Jupyter notebook for *exercise 2* correctly encodes an uncompressed WAV audio file using the lossless data compression method 'Rice coding'.	2
The Jupyter notebook for exercise 2 correctly decodes a compressed 'ex2' audio file.	2
The submission includes a link to the application running in a Jupyter notebook link in Coursera. The notebook for *exercise 3* includes markdown cells describing the code in detail.	1
The written report includes the requested table and an analysis of the results (the effectiveness of the Rice coding algorithm).	2
The written report includes a brief analysis of the application.	1
The application includes further development.	2
Exercise 3	
The application *exercise 3* correctly examines the series of film excerpts provided and generates a brief report in TXT indicating which films do not respect the digital format specified by the festival and what are the 'problematic' fields.	3
The application *exercise 3* automatically converts to the correct format the submitted films with an incorrect format. The application creates a new copy of the films by adding '_formatOK' at the end of the name.	3
The submission includes a link to the application running in a Jupyter notebook link in Coursera. The notebook for exercise 3 includes markdown cells describing the code in detail.	1
The written report includes a brief description about how ffprobe and ffmpeg has been installed and configured in your machine.	1
The written report includes a brief analysis of the application.	1

The written report includes a brief description of the requested terms.	1
Exercise 4	
Review of at least three emerging computer vision applications.	3
Technical discussions on two popular or state-of-the-art computer vision techniques	3
Discussion on two potential creative and novel uses of these computer vision and audio feature processing suitable for addressing real-life problems.	3
Appropriate academic structure, writing style, with references, and captions to figures/tables.	1
Code file PDF	
No submission, unable to open, or irrelevant	*0*
Partial code files in the PDF provided with limited or no annotations	*0.5*
Includes all developed code for all exercises with appropriate annotations	1
Video demonstration	
No submission, unable to open, irrelevant or unclear what is demonstrated.	0
Poor video demonstration showing limited or no completion of any of the exercises with some or no voice narration.	0.5
Adequate video demonstration file was submitted but shows some exercises partially completed with attempted voice narration.	1
Reasonable video demonstration file showing completion of most of the exercises with attempted voice narration.	2
Good video demonstration file was submitted which shows most completions of the exercises with clear voice narration and concise core code explanations.	3
Excellent video demonstration of all exercises with appropriate voice narration and core code walkthrough.	4
Total	**45**

Penalties for exceeding the word count:

Please refer to the latest programme regulations documents made available to you within the virtual learning environment platform