

**COMP501 – Computing Technology in Society**

Semester 1, 2021

Assignment 1: ICT Fundamentals

**Total Marks 100 (Contribution to the final mark: 20%)**

**Due: 23:59 pm, Sunday, AUT calendar week 5.**

Late penalty: Late submissions will be penalised at 5% of the actual assignment mark for every calendar day. E.g. assignments submitted 5 minutes late are penalised as one day late. Late work of 5 or more days will not be marked.

**ASSIGNMENT AIM**

The assignment has 4 parts:

1. It aims to give students an understanding of how computer systems represent real-life data such as positive, negative numbers, floating point numbers, text, etc., at the lowest level seen by the programmer, namely the **binary numbers**.
2. It prepares students with the ability to install multi-operating systems using VirtualBox (<https://www.virtualbox.org/>) and comparatively evaluate the **Linux/UNIX** Operating Systems (OS). This assignment is also prepared for students to understand the basic concepts covering Linux/UNIX file systems, commands, and working environments.
3. It helps students understand the basic idea of contemporary Machine Learning and **Deep Learning** using PyTorch ([https://www.pytorch.org/](https://www.tensorflow.org/)) and Google Colab (<https://colab.research.google.com/>).
4. It provides students some foundations of **Hypertext Markup Language (HTML)**; you will have a chance to learn about HTML elements and structure. Students will build some simple blocks of a website.

**Instruction for Submission**

The assignment must be submitted on AUT ONLINE / Blackboard in soft copy.

File(s) to be submitted:

1. A zip file contains all your assignment files and folders.
2. The submitted folder might contain PDF files, docx files, or a website

Convert or scan your working papers to make a single PDF file, name it as “**LAST NAME\_Student ID.zip”**, compress all files into **zip** including documents, images and video files. Submit it on Blackboard at:

**Assignment 1: ICT Fundametals**.

**Miscellaneous requirements:**

* The assignment will not be marked if:
  + It contains any form of malware (e.g. computer virus)
  + Not submitted in correct file format (see section: REPORT SUBMISSION INSTRUCTION)
* Keep a backup copy of your assignments to be:
  + uploaded to *“Turnitin”* anti-plagiarism service – if requested.
  + submitted as a hard copy – if requested.

***Note: Show all your workings if possible***

**Part One (35 marks):**

**Question 1:** Converting Between Number Bases (6 marks)

Consider A is the last 3 digits of your **AUT ID number**, so if your ID is 123456 then A = 456.

*Please fill the below boxes with your own ID number. It’s NOT 123456*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 0 | 1 | 2 | 8 | 9 | 3 | 3 |

ID =

Perform the following conversions between different number based systems (3 marks for each):

Assume the number A is a decimal number (base 10): A is 933

1. Convert A10 (from Decimal) to Binary

933/2=466 R1

466/2=233 R0

233/2=116 R1

116/2=58 R0

58/2=29 R0

29/2=14 R1

14/2=7 R0

7/2=3 R1

3/2=1 R1

1/2=0 R1

MSD to LSD

* 1110100101

1. Convert A10 (from Decimal) to Hexadecimal

1110100101

0011 1010 0101

* 3A5

**Question 2:** Unsigned Arithmetic Operations (6 marks)

Carry out the operations, assume that the numbers are unsigned and unlimited bits to represent:

1. Base 2: 01101111 + 11001101

1 1111

01101111

+11001101

100111100

1. Base 16: DBEAF1CA + F1CAF1CA

111 11

DBEAF1CA

+F1CAF1CA

1CDB5E394

10+10=20, 16/20, 1-4

1+12+12=25, 16/25, 1-9

1+1+1=3

15+15=30, 16/30, 1-14-E

1+10+10=21, 16/21, 1-5

1+14+12=27, 16/27, 1-11-B

1+11+1=13

13+15=28, 16/28, 1-12-C

**Question 3:** Bitwise Logical Operations (12 marks)

Assume that numbers are represented as signed, 8-bit 2’s complement representation.   
Carry out the following operations (2 marks each):

1. 0011 1110 | 0111 0001 (note: OR operation)

2 x 0 makes 0, everything else = 1

0111 1111

1. 1010 1011 & 0100 1101 (note: AND operation)

2 x 1 makes 1, everything else = 0

0000 1001

1. 1010 1010 ^ 1101 1100 (note: Exclusive OR operation)

2 x 1 makes 0, 2 x 0 makes 0, everything else = 1

0111 0110

1. 0011 0101 << 4 (note: Shift left arithmetic operation)

0101 0000

**Question 4:** 2’s Complement Conversion (5 marks)

Assume that numbers are represented as signed, 8-bit 2’s complement representation.

If the last two digits of your AUT ID is **B**, if your ID is 123456 then **B =** 56.  
Work out the following question (replace B with your last 2 digits of your ID):

Convert –**B**10 to 8-bit 2’s complement Binary; give the answer in 8 bits binary number.

B = 33

33/2=16 R1

16/2=8 R0

8/2=4 R0

4/2=2 R0

2/2=1 R0

1/2=0 R1

100001

8 bit binary

0010 0001

To convert to negative Flip 2 C plus 1

1101 1110 + 1

1101 1111

**Question 5:** ASCII Characters (6 marks)

The Appendix gives a table for 7-bit ASCII. Using this table, we can work out the hexadecimal value corresponding to the encoding of this ASCII string “ABBA” (assume each 7-bit code occupies the space of an 8-bit byte with the MSB=0) as **4142424116**

Find the hexadecimal and Binary values corresponding to **your full name** (note that there are spaces in the string); answer the following:

1. Your full name in Hexadecimal (base 16)

Ye Na Chang

59 65 20 4e 61 20 43 68 61 6e 67 00

1. How many bits are used (do not count the end of string byte)

11 x 7 = 77

**Part Two (30 marks):**

To answer the assignment questions, you use CLI (command-line interface) and provide:

* 1. **The text or screen-shots of command(s)** that you type to perform a task.
  2. **The text or screen-shots of any console output** from those commands (the output from a directory listing, for example).
  3. Make sure that you include ALL commands you used to do a task. This includes any commands you have to type to move to a specific directory. Make sure **your pasted texts** or **screen-shots** are clear enough to show where you are or have moved to.

**Question 1:** Setting up three different Linux operating system (6 marks)

Obtain your own **copy of mandatory Linux** distribution. You can download any ISO or VDI images from the Internet (at - <http://linuxlookup.com/linux_iso> or <http://virtualboxes.org/images>). Once you get a copy of an image file, you can install it on Virtual Box. Tutorial on how to use VirtualBox could be found at <https://www.youtube.com/watch?v=sB_5fqiysi4>.

Choose and install any **three** of the Linux operating systems, listed in <https://www.techradar.com/nz/best/best-linux-distros>, and **display some screen-shots** of your installation (max 3 screen-shots for each installation).

**Question 2:** Manipulate directory structures in Unix/Linux (24 marks – 2 marks each)

Use the OS that you set up from Question 1. Assume you are started at your **home** directory.   
Let your first name be **XXX**, so **XXX = “Mark”**, if your name is Mark. **XXX** from now on refers to your first name (e.g. Mark).

Perform a command that displays the absolute path of your **home** **directory** (your current location).

1. Create a new directory inside your home directory and name it “**XXX”**
2. Now navigate to the **XXX** directory and create directory **XXX-HOMEWORK** and change your current working directory to **XXX-HOMEWORK**.
3. Create three new subdirectories called **Part1, Part2,** and **Part3** in **XXX-HOMEWORK** directory.
4. Create a new file called “**MyFave.txt”** using the **touch** command and **insert** three lines into the file (you may use **echo** command and **>>** command).
   1. The first line should contain your **name** and **ID** number.
   2. The second line should be the first **sentence** of your **favourite song**.
   3. The third line should be the **name** of your **favourite movie**.

And display the contents of the file “**MyFave.txt”** to the standard output scraceen (you may use **cat** command).

1. Display the **number** of **words** in the file “**MyFave.txt”** (you may use **wc** command).
2. Copy the file “**MyFave.txt”** to directory “**Part1”** and rename it to “**MyFaveCopy.txt”**.

Make another copy of “**MyFaveCopy.txt” just made in** directory “**Part1”** and name it “**MyFaveCopyCopy.txt”** (also store in the same directory “**Part1”**).

Then, display the contents of the directory “**Part1”** using the long format.

1. Copy all the contents of directory “**Part1**” to directory “**Part2”** and display the contents of the directory “**Part2”**.
2. Assume that you are now at the current working directory **XXX-HOMEWORK;** create 15 new files (in directory **XXX-HOMEWORK**) named as follows:
   1. *FICT.bak*
   2. *unix.txt*
   3. *thistest.bak*
   4. *wumbo.file*
   5. *wumbology.txt*
   6. *moreThings.woot*
   7. *FundaIT.txt*
   8. *doodad.text*
   9. *coursetxt*
   10. *Test-1-2.xtxt*
   11. *Test-2-2.xtxt*
   12. *Test-1-1.bak*
   13. *Test-1.txt*
   14. *File-1.bat*
   15. *Assignment1.file*

And display a listing of all the files and directories in long format in the current working directory **XXX-HOMEWORK**.

1. Display a listing of all the files in the current working directory ending with letter ‘**t’** using one command, e.g. unix.tx**t** or doodad.tex**t**.
2. Display a listing of all the files in the current working directory starting with ‘**T**’ and ending with **‘t’** using one command, e.g. **T**est-1.tx**t**.
3. **Move** everything (files/directories) containing letter ‘**t’** to the directory **Part3** using one command.
4. Display a listing of the contents of the current directory XXX-HOMEWORK. All files that contain letter ‘t’ should now be gone.

**Part Three (25 marks):**

This part of the assignment will help you to understand the basic idea of Machine Learning and Deep Learning using PyTorch ([https://www.pytorch.org/](https://www.tensorflow.org/)) and Google Colab (<https://colab.research.google.com/>).

Assume that you are employed by New Zealand Autodriving Vehicle research group, they are creating an automatic detection system that allows the car to recognize a few common New Zealand road signs. Your job in the team is to qualify how well the system performs under different environments. The full list of most 50 common **New Zealand road signs** is presented in the Appendix of this assignment.

To do this, you will need to explore if there is an available Artificial Intelligence tool capable to quickly and accurately detect these road sign types on a public road. Thus, you try to test a fast version of an available object detection system - YOLO (<https://pjreddie.com/darknet/yolo/>). The detection accuracies of these objects will be evaluated on photos and a single video **taken by yourself**, and you will analyze the results to conclude if that tool is sufficient enough for your task.

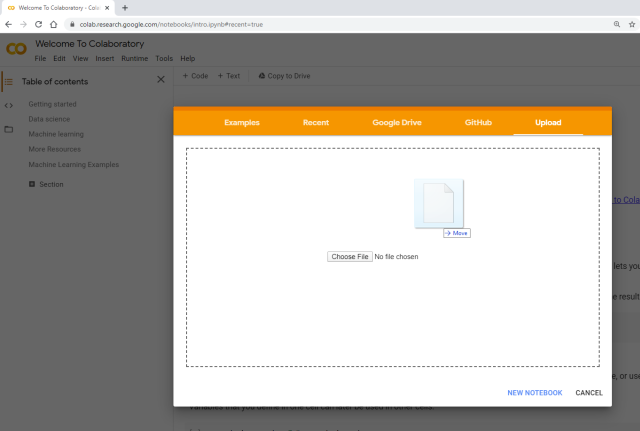
The research team has provided you 2 different **Yolo pre-trained weights** that you can use to predict the signs in photos or videos. The model weight details are presented in the below table.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy (%)*\** | Frame per second*\** |
| yolov3\_416 | 57.52 | 35 |
| yolov3\_tiny | 55.69 | 220 |

*\*The performance was collected by running on high-end GPU devices, the real results might be different on low-end devices. These weights are able to detect the NZ road signs, but not other country signs. Other country road signs might give a lower accuracy rate.*

**Instruction to Start with Google Colab**

1. Download the attached file: **Assignment\_1\_Traffic\_yolo.ipynb** from the Assessment page.
2. Upload it to Google Colab: <https://colab.research.google.com/>



1. Click File >> Upload Note Book to upload the **.ipynb** file
2. Click Runtime >> Change runtime type >> GPU if you would like to use GPU to accelerate the prediction process
3. There are 7 different cells (click Run cell to run each cell) which are ordered by following order (*you actually can just need to run the first cell once, and the last cell many times, to finish the assignment; other cells are used for extra functionalities*):

* 1st cell: where you can get the Yolo repo, pre-trained weights and set up the python environment (You only need to run this cell once)
* 2nd cell: where you can upload all of the necessary files from your local drive, including images (.jpg), weights (.pt) or video (.mp4)
* 3rd cell: where you can mount your google drive to the working directory (You only need to run this cell once). This is a very helpful tool where you don’t need to re-clone the files and setup the environment again.
* 4th cell: where you can copy the repo folder to your google drive or copy the saved folder from your google drive back to the current working directory.
* 5th cell: check whether the system is running on GPU or not
* 6th cell: run the prediction using command line
* 7th cell: run the prediction using file upload

**NOTE:** *you can switch between different weight files by modifying the path of –weights in either 6th cell or 7th cell depends on which cell you are running the prediction on.*

Cell 7

Cell 6

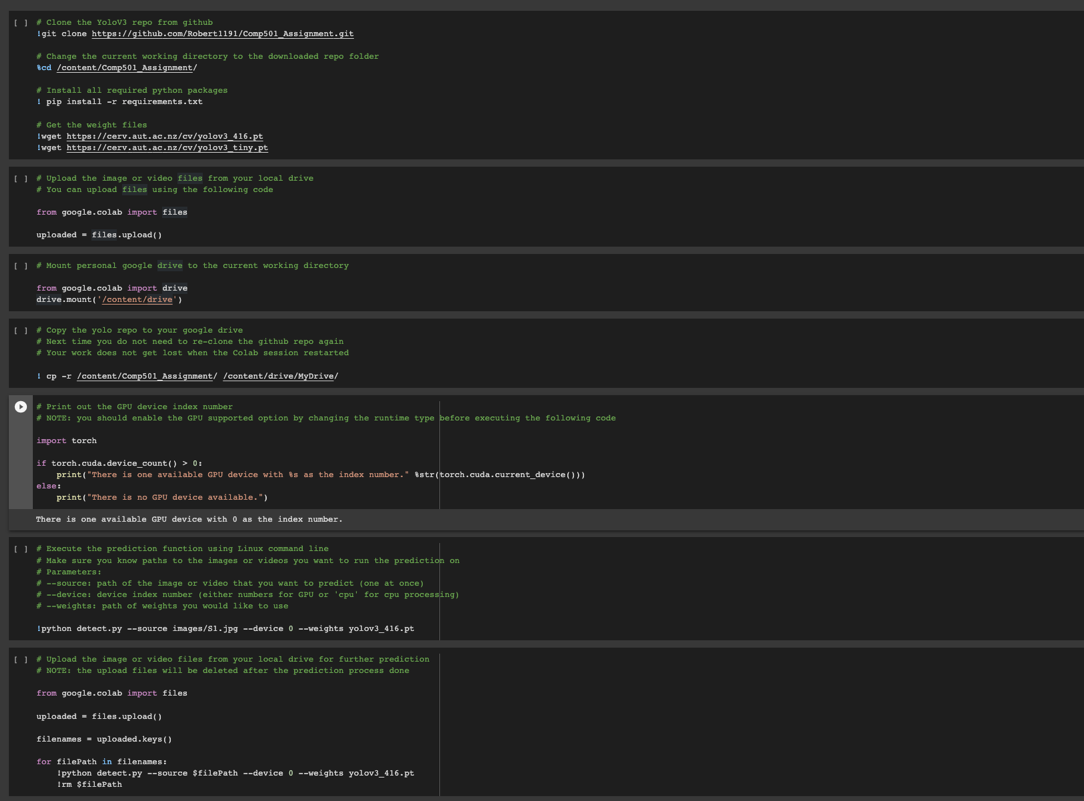
Cell 5

Cell 4

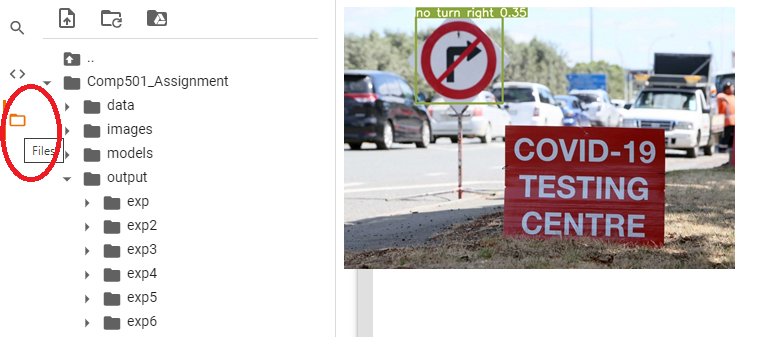
Cell 3

Cell 2

Cell 1



1. The output image could be something like this (outputs are stored in **output** folder):



One sign is successfully detected in the image (1/1 = 100%).

****

Or, two signs are detected in the above image. However, only one of them gives the correct identified name (1/2 = 50%).

**Question 1:** Collect dataset / photos (5 marks)

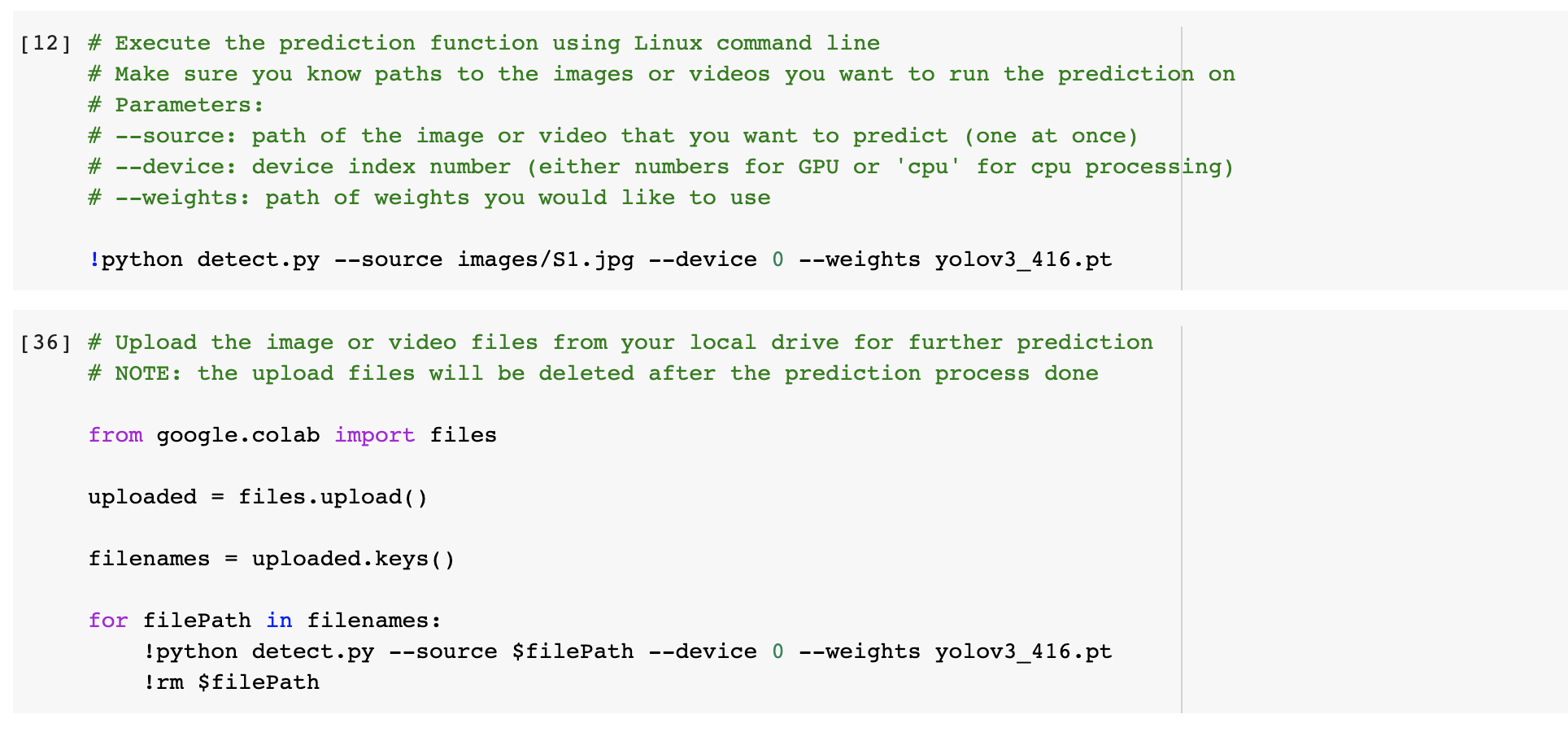
* **Take at least 10 photos** that include any items in the list of classes which is shown at the Appendix part of this document. Pictures can be **taken by yourself via cell phone or found on the Internet**.
* **Resize** the images so that each is less than **500 KB** in size. Save all the pictures to folder **Images**, and all are stored as **jpg** type.
* List your collected images in the table below (**0.5 marks for one image)**.

|  |  |
| --- | --- |
| Replace by yours | Replace by yours |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* *The best way to do it is to open the image using IrfanView (*[*https://www.irfanview.com/*](https://www.irfanview.com/)*), press Ctrl + A, then Ctrl + C, and paste your photos on the table below using Ctrl + V, 1 mark for each correct photo. Or you can use the free online photo edit tool (*[*https://www.online-image-editor.com/*](https://www.online-image-editor.com/)*).*
* *In case you are using iPhone to take the photos, you can simply send it to your own email, it will get converted to a .jpg image when you download. Or you can use the open-source HEIC to JPG convert tool (*[*https://www.heictojpg.com/*](https://www.online-image-editor.com/)*) to convert your image to .jpg format.*

**Question 2:** Test the prediction performance of YOLO object detection system using images as the inputs (10 marks)

* Run the last two cells to do the prediction on a particular image or video.



* Record your finding on the below table (**Remember to replace by yours)**. Repeat the same process with two different pre-trained weights (**yolov3\_416** and **yolov3\_tiny**). At the end, you should have two different tables, one for each pre-trained weight.

|  |  |
| --- | --- |
| Images | Performance (%) |
|  | 100 |
|  | 0 |
| Average | 50% |

**Question 3:** Test the prediction performance of YOLO object detection system using video as the input (10 marks)

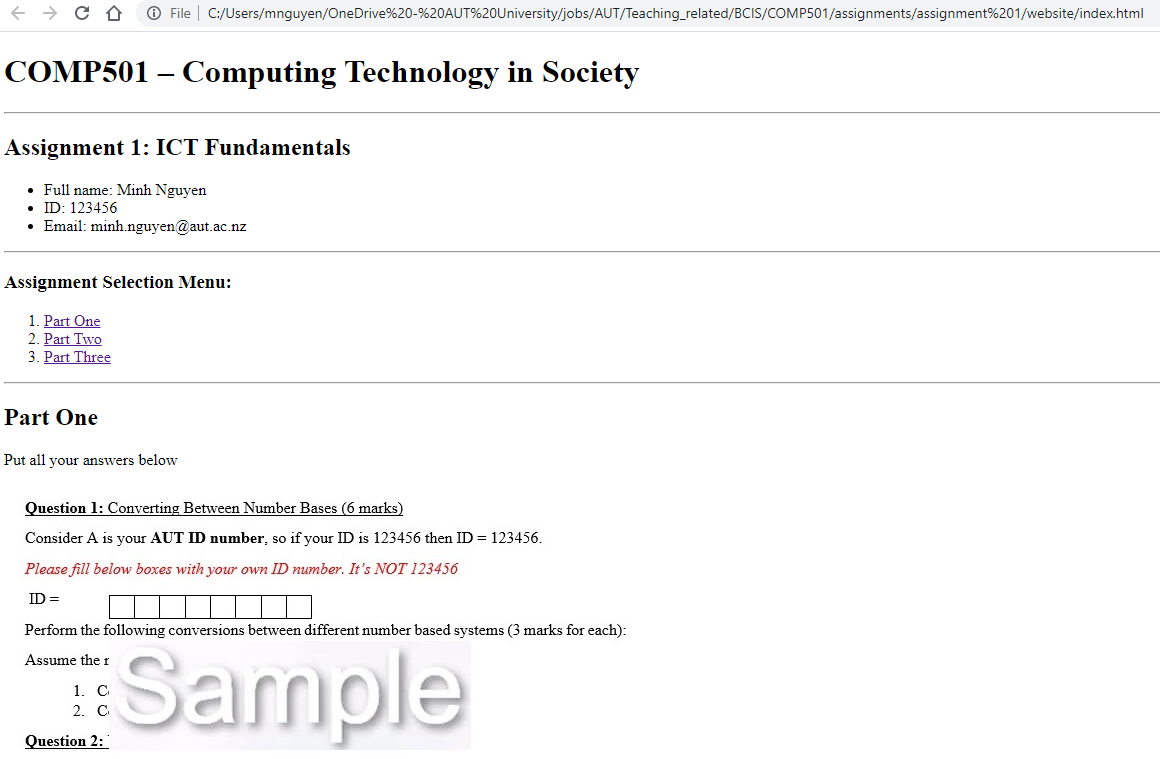
* Record a short video*\** of different New Zealand road signs with **.mp4** format (minimum 10s and 30s as maximum). This video could be taken by using your phone device or professional camera devices. The recommended resolutions are 720p or 1080p. You could use 2k or 4k video as well, but it would take a bit longer time to process. (**5 marks)**
* Run the video through the prediction process. The final video should be able to show at least **one corrected class name for a minimum of 2s**. (**5 mark)**

[](https://www.youtube.com/watch?v=_tOpAhm65WU)[](https://www.youtube.com/watch?v=dEUDZBEwIXo)*\*NOTE: you can take many videos as you wish and pick the one that has the best performance. Please take a look at the below videos for your reference* ***(Right-click on the picture, open Link).***

**Part Four (10 marks):**

This part of the assignment requires you to use some basic HTML tags to present the entire assignment nicely on a static website. Note that, if you do not do this part, you can submit the assignment as a word or PDF document, with other files (in a zip file):

* Start by downloading the template **website.zip** file, extract it and edit the contents using any IDE or Notepad.
* Use Chrome/Firefox/IE/Edge to open the **index.html** file to see the output (as below).
* Press Windows + left arrow combination, to make the website allocate half of the screen.
* You can go back to the file, and right click >> open with Notepad; and press Windows + right arrow combination, to make the website allocate the other half of the screen.



**Question 1:** Successfully present the entire assignment to be displayed correctly on a website using HTML language. (5 marks)

**Question 2:** Use your own imagination and artistic skills, turn the page into something nice, clear, and attractive. (5 marks)

**APPENDIX - The ASCII character set**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 00 | NUL | 01 | SOH | 02 | STX | 03 | ETX |
| 04 | EOT | 05 | ENQ | 06 | ACK | 07 | BEL (\a) |
| 08 | BS (\b) | 09 | HT (\t) | 0A | LF (\n) | 0B | VT (\v) |
| 0C | FF (\f) | 0D | CR (\r) | 0E | SO | 0F | SI |
| 10 | DLE | 11 | DC1 | 12 | DC2 | 13 | DC3 |
| 14 | DC4 | 15 | NAK | 16 | SYN | 17 | ETB |
| 18 | CAN | 19 | EM | 1A | SUB | 1B | ESC |
| 1C | FS | 1D | GS | 1E | RS | 1F | US |
| 20 | SP | 21 | ! | 22 | " | 23 | # |
| 24 | $ | 25 | % | 26 | & | 27 | ' |
| 28 | ( | 29 | ) | 2A | \* | 2B | + |
| 2C | , | 2D | - | 2E | . | 2F | / |
| 30 | 0 | 31 | 1 | 32 | 2 | 33 | 3 |
| 34 | 4 | 35 | 5 | 36 | 6 | 37 | 7 |
| 38 | 8 | 39 | 9 | 3A | : | 3B | ; |
| 3C | < | 3D | = | 3E | > | 3F | ? |
| 40 | @ | 41 | A | 42 | B | 43 | C |
| 44 | D | 45 | E | 46 | F | 47 | G |
| 48 | H | 49 | I | 4A | J | 4B | K |
| 4C | L | 4D | M | 4E | N | 4F | O |
| 50 | P | 51 | Q | 52 | R | 53 | S |
| 54 | T | 55 | U | 56 | V | 57 | W |
| 58 | X | 59 | Y | 5A | Z | 5B | [ |
| 5C | \ | 5D | ] | 5E | ^ | 5F | \_ |
| 60 | ` | 61 | a | 62 | b | 63 | c |
| 64 | d | 65 | e | 66 | f | 67 | g |
| 68 | h | 69 | i | 6A | j | 6B | k |
| 6C | l | 6D | m | 6E | n | 6F | o |
| 70 | p | 71 | q | 72 | r | 73 | s |
| 74 | t | 75 | u | 76 | v | 77 | w |
| 78 | x | 79 | y | 7A | z | 7B | { |
| 7C | | | 7D | } | 7E | ~ | 7F | DEL |

**APPENDIX – The top 50 common NZ road signs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Image** | **Class name** | **Image** | **Class name** |
| A picture containing text, sign, vector graphics  Description automatically generated | 10k speed limited |  | 90k speed limited |
|  | 20k speed limited |  | 70k speed limited |
|  | 30k speed limited |  | 80k speed limited |
|  | 40k speed limited |  | 90k speed limited |
|  | 50k speed limited |  | 100k speed limited |
| **Image** | **Class name** | **Image** | **Class name** |
|  | 110k speed limited |  | bus stop |
|  | attention |  | disable parking |
|  | bus lane |  | except bus |
|  | bus only |  | give way to oncoming vehicles |
|  | bus parking |  | give way |

|  |  |  |  |
| --- | --- | --- | --- |
| **Image** | **Class name** | **Image** | **Class name** |
|  | go straight |  | no entry |
|  | motorcycle parking |  | no exit |
|  | must go straight |  | no parking |
|  | must turn left |  | no speed limited |
|  | must turn right |  | no stopping |

|  |  |  |  |
| --- | --- | --- | --- |
| **Image** | **Class name** | **Image** | **Class name** |
|  | no turn left |  | priority over oncoming vehicles |
|  | no turn right |  | school |
|  | no u turn |  | road closed |
|  | one way left |  | road works |
|  | one way right |  | roundabout giveway |

|  |  |  |  |
| --- | --- | --- | --- |
| **Image** | **Class name** | **Image** | **Class name** |
|  | stop |  | traffic lights ahead |
|  | straight or left |  | turn left and right |
|  | straight or right |  | turn left |
|  | taxi stand |  | turn right |
|  | temporary |  | wrong way |