

**COMP501 – Computing Technology in Society**

Semester 1, 2023

Assignment 1: ICT Fundamentals

**Total Marks: 100 (Contribution to the final mark: 40%)**

**Due: 23:59 pm, Sunday, 2 April 2023**

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 [CANVAS](https://canvas.aut.ac.nz/courses/1249) 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

Compress all files to “**LAST NAME\_Student ID.zip”**, the file includes all documents, images and video files. Submit it on [CANVAS](https://canvas.aut.ac.nz/courses/1249/assignments).

**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 above: Instruction for Submission)
* 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 where possible***

**Part One (35 marks):**

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

Consider A is the two **NON ZERO** digits in **positions 5 and 6** of your **AUT ID number**, so

Examples: If your ID is 36125612 then **A** = 56.

If your ID is 36125061 then **A** = 56

If your ID is 36120561 then **A** = 56.

*Please fill the below boxes with your own ID number.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 2 | 1 | 7 | 0 | 2 | 5 | 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):

1. Convert **A**10 (from Decimal) to Binary = call the answer **M**2

A= 25

25/2=12, r= 1

12/2=6, r= 0

6/2=3, r= 0

3/2=1, r= 1

½=0, r=1

Binary number (M2)= 011001

1. Convert **A**10 (from Decimal) to Hexadecimal = call the answer **N**16

Decimal to hexadecimal🡺25 to hexadecimal

Step 1: convert decimal to binary

25/2=12, r= 1

12/2=6, r= 0

6/2=3, r= 0

3/2=1, r= 1

½=0, r=1

Binary number (M2)= 011001

Step 2: convert binary to hexadecimal

N16= 19

**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: **M**2 + 110011012

0110012 + 110011012

= 111001102

1. Base 16: **N**16 + 9916

1916 + 9916

= B216

**Question 3:** 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 56 then call **B**10 **= -**56.  
Work out the following question (replace with your last 2 digits of your ID):

Last two digits of ID= 53

Therefore B10= -53

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

*(\*) note that the number is negative, so the answer will start with ‘1’.*

Step 1: convert 53 to binary

53/2=26, r=1

26/2=13, r=0

13/2= 6, r=1

6/2=3, r=0

3/2=1, r=1

½=0, r=1

0/2=0,r=0 binary number(in 8 bits)= 0011 0101

Step2:convert positive binary to negative binary

p

=1100 1011

**Question 4:** Bitwise Logical Operations (12 marks)

Assume that you have the number **C**2 found in Question 3, represented as signed, 8-bit 2’s complement representation. Carry out the following operations (3 marks each):

C2= 1100 1011

1. **C**2 | 0111 0001 (note: OR operation)

1100 1011 | 0111 0001

= 1111 1011

1. **C**2 & 0100 1101 (note: AND operation)

1100 1011 & 0100 1101

= 0100 1001

1. **C**2 ^ 1101 1100 (note: Exclusive OR operation)

1100 1011^ 1101 1100

= 0001 0111

1. **C**2 << 4 (note: Shift left arithmetic operation)

**=**1100 1011 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**

Use Ascii table; 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) (4 marks)

Full name: Emily Lau

In hexadecimal: 454D494C794C417516

1. How many bits (not bytes) are used (do not count the end of string byte) (2 marks)

8\*8 =64

**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 one Linux operating system (5 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. A tutorial on how to use VirtualBox can be found at <https://www.youtube.com/watch?v=sB_5fqiysi4>.

Choose and install any Linux operating systems, listed in <https://www.techradar.com/nz/best/best-linux-distros>, and **display some screen-shots** of your.

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

Use any OS that you set up from Question 1 (Puppy is allowed). 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”**

Now navigate to the **XXX** directory and create directory **XXX-COMP501WORK** and change your current working directory to **XXX-COMP501WORK**.

Shape

Description automatically generated

1. Create three new subdirectories called **Sec1, Sec2,** and **Sec3** in **XXX-COMP501WORK** directory.

Text, letter

Description automatically generated

1. 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 screen (you may use **cat** command).

Text, letter

Description automatically generated

1. Display the **number** of **words** in the file “**MyFave.txt”** (you may use **wc** command).

A picture containing background pattern

Description automatically generated

1. Copy the file “**MyFave.txt”** to directory “**Sec1”** and rename it to “**MyFaveCopy.txt”**.

Make another copy of “**MyFaveCopy.txt” just made in** directory “**Sec1”** and name it “**MyFaveCopyCopy.txt”** (also store in the same directory “**Sec1”**).

Then, display the contents of the directory “**Sec1”** using the long format.

Text, letter

Description automatically generated

1. Copy all the contents of directory “**Sec1**” to directory “**Sec2”** and display the contents of the directory “**Sec2”**.

Text, letter

Description automatically generated

1. Assume that you are now at the current working directory **XXX-COMP501WORK;** create 15 new files (in directory **XXX-COMP501WORK**) named as follows:
   1. *FICTION.bak*
   2. *myunix.txt*
   3. *thistle.bak*
   4. *mumbo.file*
   5. *mumble.txt*
   6. *moremumbo.woot*
   7. *FundaOfIT.txt*
   8. *dooda.text*
   9. *coursetext*
   10. *Test-1.xtxt*
   11. *Test-2.xtxt*
   12. *Test-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-COMP501WORK**.

Table

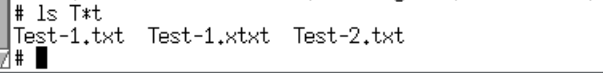
Description automatically generated

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**.

Text

Description automatically generated with low confidence

1. 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**.



1. **Move** everything (files/directories) containing letter ‘**t’** to the directory **Sec3** using one command.



1. Display a listing of the contents of the current directory **XXX-COMP501WORK**. All files that contain letter ‘t’ should now be gone.

A picture containing text

Description automatically generated

**Part Three (25 marks):**

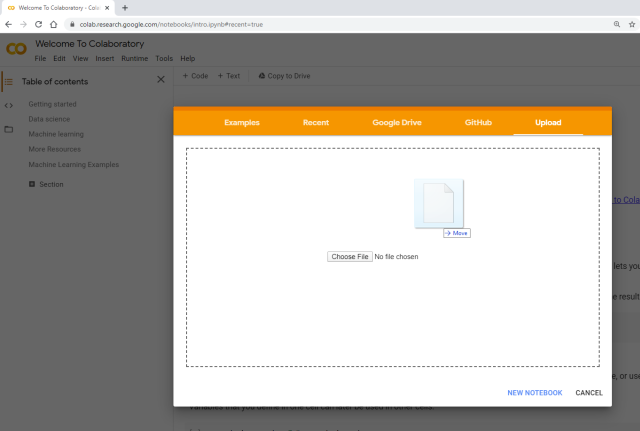
This part of the assignment will help you to understand the basic idea of Machine Learning and Deep Learning using PyTorch ([https://www.pytorch.org/](https://www.tensorflow.org/)) and Google Colab (<https://colab.research.google.com/>).

Assume that you are employed by Auckland Transport (AT), and they ask you to write a program to detect and count how many cars and trucks are flowing through a particular motorway section during different hours of the day. You will use the images optioned live at <https://www.journeys.nzta.govt.nz/traffic-and-travel-information/traffic-cameras/auckland/>. The decision is used to adjust the ramp signals (also known as ramp metering) to manage the rate at which vehicles move down the ramp and onto the motorway.

To do this, you will need to explore if there is an available Artificial Intelligence tool capable of quickly and accurately detecting these transportation 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 are evaluated on photos taken by yourself, and you will analyse the results to conclude if that tool is sufficient enough for your task (and the report could be sent to AT for consideration).

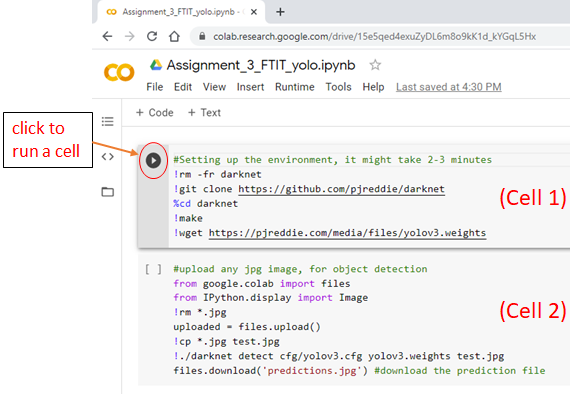
**Instruction to Start with Google Colab**

1. Download the attached file: **Assignment\_1.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. There are only two cells (click Run cell to run each cell) (*you can just need to run the first cell once, and the last cell many times, to finish the assignment*):

* 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 the necessary files from your local drive, including images (.jpg) to achieve the prediction with YOLO



The output image could be something like this:

test.jpg: Predicted in 19.392162 seconds.

truck: 80%

truck: 78%

truck: 55%

car: 97%

car: 90%

car: 68%

car: 64%

Prediction is done in 26.339985132217407 seconds

  
  
So there are 7 cars/trucks are successfully detected in the image.

test.jpg: Predicted in 18.271086 seconds.

truck: 92%

truck: 75%

truck: 68%

car: 97%

car: 96%

car: 87%

car: 78%

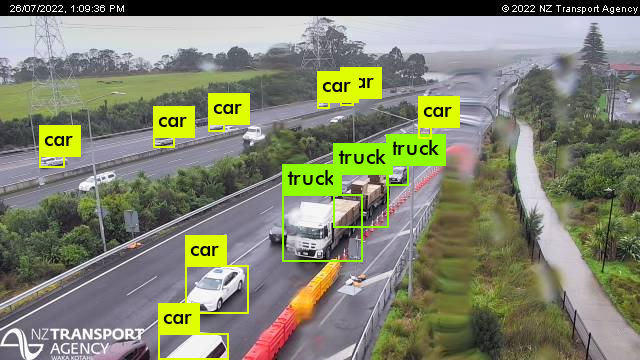
car: 65%

car: 59%

car: 59%

car: 50%

Prediction is done in 25.13233256340027 seconds

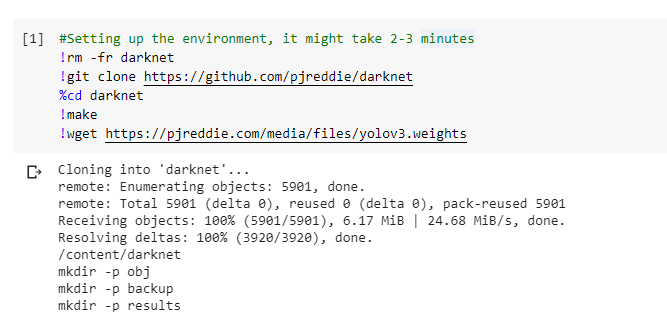
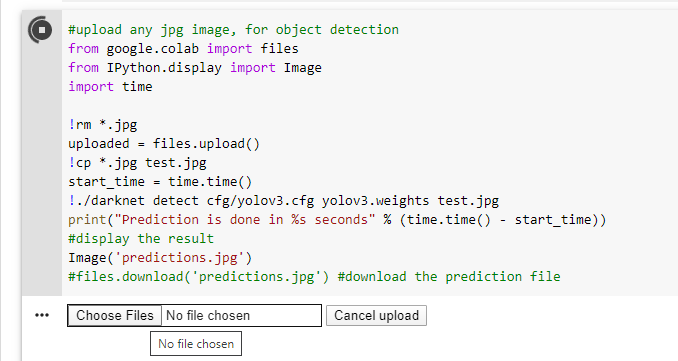


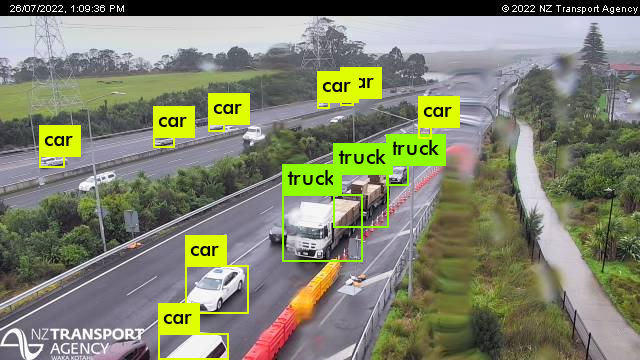
And here, there are 11 cars/trucks are successfully detected in the image.

**Question 1:** Collect dataset / photos (10 marks)

* Go to: <https://www.journeys.nzta.govt.nz/traffic-and-travel-information/traffic-cameras/auckland/>
* Choose the motorway section closest to your home, for example: <https://www.journeys.nzta.govt.nz/traffic-and-travel-information/traffic-cameras/auckland/723>.
* Normally, photos are refreshed at least every minute or so, right click and save at least 30 different photos within one hour, e.g. 10am-11am.
* Save all the pictures to folder **Images**, and name them XXX\_01.jpg, XXX\_02.jpg, …. XXX\_30.jpg. Let your first name be **XXX**, so **XXX = “Mark”**, if your name is Mark.

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

* Start by running the first cell to set up the environment (you only need to run it once).  
  
* Run the second cell and click “Choose Files” to upload a **jpg** image.   
  
* After that, you should first see the weights being downloaded; and then the object detection system will search for all objects in the image and return output with bounding boxes and annotations of detected objects like this below:

* You will need to collect three things: (1) how many cars/trucks you can see on the original image (left), please just ignore very tiny ones, (2) how many cars/trucks that the computer can count on the right, (3) the accuracy in percentage between the two (e.g. 0% - if none of the cars/trucks is identified, 50% - if there are 20 cars/trucks, but only 10 bounding boxes of the cars/trucks are identified, 100% - if all cars/trucks are identified.
* Repeat running the second cell to upload all your photos and save all the prediction results.
* **Copy and paste** the predicted photos on the table below, replace the images with your images, also collect the processing time, and the accuracy, e.g. 2/2 = 100%. (1 mark for each correct photo).

note: all small or not seen cars are not counted

Processing time: 23.4 seconds

Number of cars detected: 2🡪 accuracy = 2/2 = 100%

Number of trucks detected: 2 🡪 accuracy = 2/4 = 50%

|  |  |
| --- | --- |
| Original images (Replace by yours) | AI detected images (Replace by yours) |
| A picture containing text, way, road, scene  Description automatically generated  **Image 1**  Numbers of cars in original: 11  Number of trucks in original: 2 | Processing time: 23.51 seconds  Number of cars detected: 11 🡪 accuracy = 11/11 = 100%  Number of trucks detected: 3🡪 accuracy = 3/2 = 100% |
| A picture containing text, way, scene, road  Description automatically generated  **Image 2**  Numbers of cars in original: 7  Number of trucks in original: 2 | Processing time: 24.14 seconds  Number of cars detected: 7🡪 accuracy = 7/7 = 100%  Number of trucks detected: 4 🡪 accuracy = 4/2 = 100% |
| A picture containing text, way, road, scene  Description automatically generated  **Image 3**  Numbers of cars in original: 2  Number of trucks in original: 4 |  |
| A highway with cars on it  Description automatically generated with low confidence  **Image 4**  Numbers of cars in original: 6  Number of trucks in original: 3 | Processing time: 24.50 seconds  Number of Cars detected: 6 🡪 accuracy = 6/6 =100%  Number of trucks detected: 1 🡪accuracy = 1/3 = 33% |
| A picture containing text, way, road, scene  Description automatically generated  **Image 5**  Numbers of cars in original: 7  Number of trucks in original: 2 | Processing time: 22.50 seconds  Number of cars detected: 2 🡪 accuracy = 2/7 = 29%  Number of trucks detected: 0 🡪 accuracy = 0/2 = 0% |
| A picture containing text, way, road, scene  Description automatically generated  **Image 6**  Numbers of cars in original: 10  Number of trucks in original: 1 | Processing time: 23.5 seconds  Number of cars detected: 8 🡪 accuracy = 8/10 = 80%  Number of trucks detected: 1 🡪 accuracy = 1/1 = 100% |
| A picture containing text, way, road, scene  Description automatically generated  **Image 7**  Numbers of cars in original: 10  Number of trucks in original: 1 | Processing time: 22.9 seconds  Number of cars detected: 8 🡪 accuracy = 8/10 = 80%  Number of trucks detected: 1 🡪 accuracy = 1/1 = 100% |
| A high angle view of cars on a highway  Description automatically generated with medium confidence  **Image 8**  Numbers of cars in original: 5  Number of trucks in original: 0 | Processing time: 22.5 seconds  Number of cars detected: 5 🡪 accuracy = 5/5 = 100%  Number of trucks detected: 1 🡪 accuracy = 1/0 = 100% |
| A highway with many cars on it  Description automatically generated with low confidence  **Image 9**  Numbers of cars in original: 8  Number of trucks in original: 2 | Processing time: 22.8 seconds  Number of cars detected: 3 🡪 accuracy = 3/8 = 38%  Number of trucks detected: 1 🡪 accuracy = 1/2 = 50% |
| A picture containing text, way, road, scene  Description automatically generated  **Image 10**  Numbers of cars in original: 3  Number of trucks in original: 2 | Processing time: 23.2 seconds  Number of cars detected: 2🡪 accuracy = 2/3 = 67%  Number of trucks detected: 0 🡪 accuracy = 0/2 = 0% |
| A picture containing text, way, scene, road  Description automatically generated  **Image 11**  Numbers of cars in original: 9  Number of trucks in original: 2 | Processing time: 22.9 seconds  Number of cars detected: 8🡪 accuracy = 8/9 = 89%  Number of trucks detected: 2 🡪 accuracy = 2/2 = 100% |
| A picture containing text, way, road, scene  Description automatically generated  **Image 12**  Numbers of cars in original: 11  Number of trucks in original: 2 | Processing time: 23.1 seconds  Number of cars detected: 11🡪 accuracy = 11/11 = 100%  Number of trucks detected: 1 🡪 accuracy = 1/2 = 50% |
| A picture containing text, way, scene, road  Description automatically generated  **Image 13**  Numbers of cars in original: 11  Number of trucks in original: 3 | Processing time: 22.9 seconds  Number of cars detected: 11🡪 accuracy = 11/11 = 100%  Number of trucks detected: 3 🡪 accuracy = 3/3 = 100%  11 |
| **Image 14**  Numbers of cars in original: 10  Number of trucks in original: 2 | Processing time: 23.3 seconds  Number of cars detected: 10🡪 accuracy = 10/10 = 100%  Number of trucks detected: 0 🡪 accuracy = 0/2 = 0%  11 |
| A picture containing text, way, road, scene  Description automatically generated  **Image 15**  Numbers of cars in original: 8  Number of trucks in original: 2 | Processing time: 21.8 seconds  Number of cars detected: 7🡪 accuracy = 7/8 = 88%  Number of trucks detected: 1 🡪 accuracy = 1/2 = 50%  1 |
| A picture containing text, way, scene, road  Description automatically generated  **Image 16**  Numbers of cars in original: 12  Number of trucks in original: 2 | Processing time: 22.6 seconds  Number of cars detected: 12🡪 accuracy = 12/12 = 100%  Number of trucks detected: 1 🡪 accuracy = 1/2 = 50%  1 |
| A picture containing text, way, scene, road  Description automatically generated  **Image 17**  Numbers of cars in original: 9  Number of trucks in original: 5 | Processing time: 22.9 seconds  Number of cars detected: 9🡪 accuracy = 9/9 = 100%  Number of trucks detected: 2🡪 accuracy = 2/5 = 40%  1 |
| A picture containing text, way, road, scene  Description automatically generated  **Image 18**  Numbers of cars in original: 12  Number of trucks in original: 1 | Processing time: 22.1 seconds  Number of cars detected: 11🡪 accuracy = 11/12 = 92%  Number of trucks detected: 0🡪 accuracy = 0/1 = 0%  1 |
| A picture containing text, way, road, scene  Description automatically generated  **Image 19**  Numbers of cars in original: 10  Number of trucks in original: 4 | Processing time: 24.3 seconds  Number of cars detected: 10🡪 accuracy = 10/10 = 100%  Number of trucks detected: 4🡪 accuracy = 4/4 = 100%  1 |
| **Image 20**  Numbers of cars in original: 9  Number of trucks in original: 2 | Processing time: 23.1 seconds  Number of cars detected: 9 🡪 accuracy = 9/9 = 100%  Number of trucks detected: 2🡪 accuracy = 2/2 = 100%  1 |
| **Image 21**  Numbers of cars in original: 12  Number of trucks in original: 4 | Processing time: 23.5 seconds  Number of cars detected: 12 🡪 accuracy = 12/12 = 100%  Number of trucks detected: 5🡪 accuracy = 5/4 = 100%  1 |
| **Image 22**  Numbers of cars in original: 12  Number of trucks in original: 1 | Processing time: 22.9 seconds  Number of cars detected: 10 🡪 accuracy = 10/12 = 83%  Number of trucks detected: 2🡪 accuracy = 2/1 = 100%  1 |
| **Image 23**  Numbers of cars in original: 8  Number of trucks in original: 2 | Processing time: 23.0 seconds  Number of cars detected: 3 🡪 accuracy = 2/8 = 25%  Number of trucks detected: 2🡪 accuracy = 2/2 = 100%  1 |
| A picture containing text, way, scene, road  Description automatically generated  **Image 24**  Numbers of cars in original: 10  Number of trucks in original: 2 | Processing time: 23.4 seconds  Number of cars detected: 7 🡪 accuracy = 7/10 = 70%  Number of trucks detected: 1🡪 accuracy = 1/2 = 50%  1 |
| **Image 25**  Numbers of cars in original: 12  Number of trucks in original: 0 | Processing time: 23.4 seconds  Number of cars detected: 12 🡪 accuracy = 12/12 = 100%  Number of trucks detected: 0🡪 accuracy = 0/0 = 100%  1 |
| **Image 26**  Numbers of cars in original: 8  Number of trucks in original: 1 | Processing time: 22.8 seconds  Number of cars detected: 6 🡪 accuracy = 6/8 = 75%  Number of trucks detected: 2🡪 accuracy = 2/1 = 100%  1 |
| **Image 27**  Numbers of cars in original: 7  Number of trucks in original: 2 | Processing time: 23.5 seconds  Number of cars detected: 6 🡪 accuracy = 6/7 = 86%  Number of trucks detected: 1🡪 accuracy = 1/2 = 100%  1 |
| A picture containing text, way, scene, road  Description automatically generated  **Image 28**  Numbers of cars in original: 10  Number of trucks in original: 1 | Processing time: 23.3 seconds  Number of cars detected: 10🡪 accuracy = 10/10 = 100%  Number of trucks detected: 0🡪 accuracy = 0/1 = 0%  1 |
| A picture containing text, way, scene, road  Description automatically generated  **Image 29**  Numbers of cars in original: 12  Number of trucks in original: 0 | Processing time: 23.2 seconds  Number of cars detected: 10🡪 accuracy = 10/12 = 83%  Number of trucks detected: 0🡪 accuracy = 0/0 = 100%  1 |
| **Image 30**  Numbers of cars in original: 11  Number of trucks in original: 2 | Processing time: 23.0 seconds  Number of cars detected: 11🡪 accuracy = 11/11 = 100%  Number of trucks detected: 0🡪 accuracy = 0/2 = 0%  1 |

**Question 3:** Evaluation (5 marks)

Analyse the prediction results and write your evaluation report. Please summarise the prediction accuracies by filling up the below table (10 marks).

|  |  |  |  |
| --- | --- | --- | --- |
| Transportation types | Total number appeared in all images | Total number accurately detected in all images | Average accuracy  (%) |
| Cars | 272 | 239 | 81.2% |
| Trucks | 59 | 43 | 65.8% |
| Buses | 1 | 0 | 0 |
| Bikes | 0 | 0 | 0 |
| Total | 0 | 0 | 0 |

**Evaluation report:** Overall the accuracies detected by AI are not all correct. Some are unidentified or identified as something else or it identidied a vehicle as two different types of transportation which affected the overall results. An example was shown in image 9 where it detected atruck as a train and a truck or in image 3 where it detected a caravan as a car, truck and a van. It is indeed very hard to determine different types of of vehicles like caravans, utes and trailers to count as either a car, truck or either disclude them as well as that, a single truck carrying two trailers were counted as a truck twice. Some are incorrectly identified as truck or a car which effected the results because they were counted towards the type of transportation where as the vehicle that was suppose to be correctly identified is not counted towards the total number. This was shown image 4 where a car is identified as a truck where as the actual had 3 trucks appeared but it was NOT counted towards the total which affected the accuracy resulting in a 33% accuracy where as it should’ve been a 0% because they were not correctly identified.

**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 asignment 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 in Questions 1& 2 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)

Note: attached zip file with assignment under the file name: LauEmily\_22170253.zip

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

Note: attached zip file with assignment under the file name: LauEmily.22170253.zip

**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 |