Possible Project1: autonomous drone (obstacle avoidance)

1. Dataset:

   Deep reinforcement learning gym for autonomous aerial robot visual navigation
   Paper: https://link.springer.com/article/10.1007/s10994-021-06006-6
   Or
   A 3D simulation Platform like Unreal Engine: https://github.com/aqeelanwar/DRLwithTL.git
   Paper: https://towardsdatascience.com/deep-reinforcement-learning-for-drones-in-3d-realistic-environments-36821b6ee077

2.

   a. It will be trained in a completely virtual game. The information provided that is useful would be the actual obstacles in the game. There are source code for this in the paper. https://github.com/harvard-edge/AirLearning

   b. I want the drone can learn how to maximize the efficiency of obstacle avoidance and navigation. Possible model: Deep Reinforcement Learning with Transfer Learning – Simulated Drone and Environment: https://github.com/aqeelanwar/DRLwithTL
   So far, I haven't really know the pros and cons since there are not many TA learning RL.

   c. (based on the second paper listed above) Generated by Tensorboard from Tensorflow
      i. MeanQ
      ii. success rate
      iii. Loss
      iv. Ret-VanLeer

3.

   It would really hard to set up the drone in real life but it is possible to generate a new virtual environment in the simulator to test the model.

Project 1 alternative: (may personal favourite solution since it is what Tesla is doing)

What this one is doing: Building a occupancy model that is able to map the surrounding into a 3D vector space based on cameras' input.
NOTE: All model, data for this Occupancy Model is open sourced. You can find tutorial and code on github.

1. Dataset: there are pre-processed data downloadable. Inside the README.md: https://github.com/autonomousvision/occupancy_networks
2. There is a really comprehensive tutorial and code in this link: https://github.com/autonomousvision/occupancy_networks

   a. no need

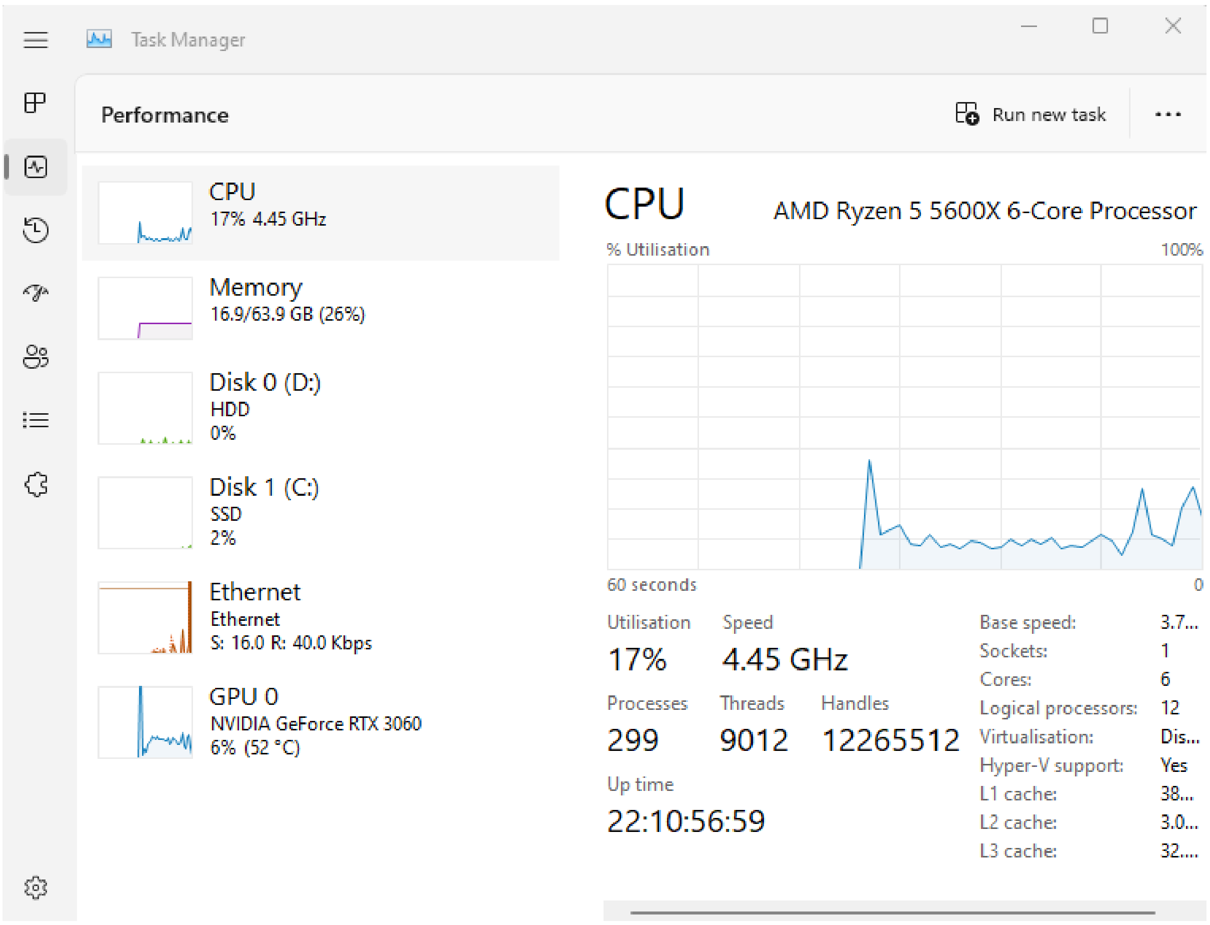
   b. The model is also from the link above.
   Here are some video that briefly explains the occupancy network:
   https://www.youtube.com/watch?v=Nu3LUB8wolc

https://www.youtube.com/watch?v=fCsfLCc4UzQ

cons: the size of the pre-processed dataset is humongous.
pros: powerful to use. Has a comprehensive tutorial.
There also other models but this one has a better tutorials and have pre-processed dataset.
https://kevintshoemaker.github.io/NRES-746/Occupancy.html#Model_Comparison
c.
evaluation data will be generated by scripts in this repository. https://github.com/autonomousvision/occupancy_networks
"The script takes the meshes generated in the previous step and evaluates them using a standardized protocol. The output will be written to .pkl/.csv files in the corresponding generation folder which can be processed using pandas."

3. Set up cameras and visualizing the outcome of the 3D vector space.

P.S. For these two projects above, here is my Desktop info. I have installed CUDA so my GPU can also be used for training


Task Manager

Performance

Run new task

CPU
17% 4.45 GHz

Memory
16.9/63.9 GB (26%)

Disk 0 (D:)
HDD
0%

Disk 1 (C:)
SSD
2%

Ethernet
Ethernet
S: 16.0 R: 40.0 Kbps

GPU 0
NVIDIA GeForce RTX 3060
6% (52 °C)

CPU AMD Ryzen 5 5600X 6-Core Processor

% Utilisation 100%

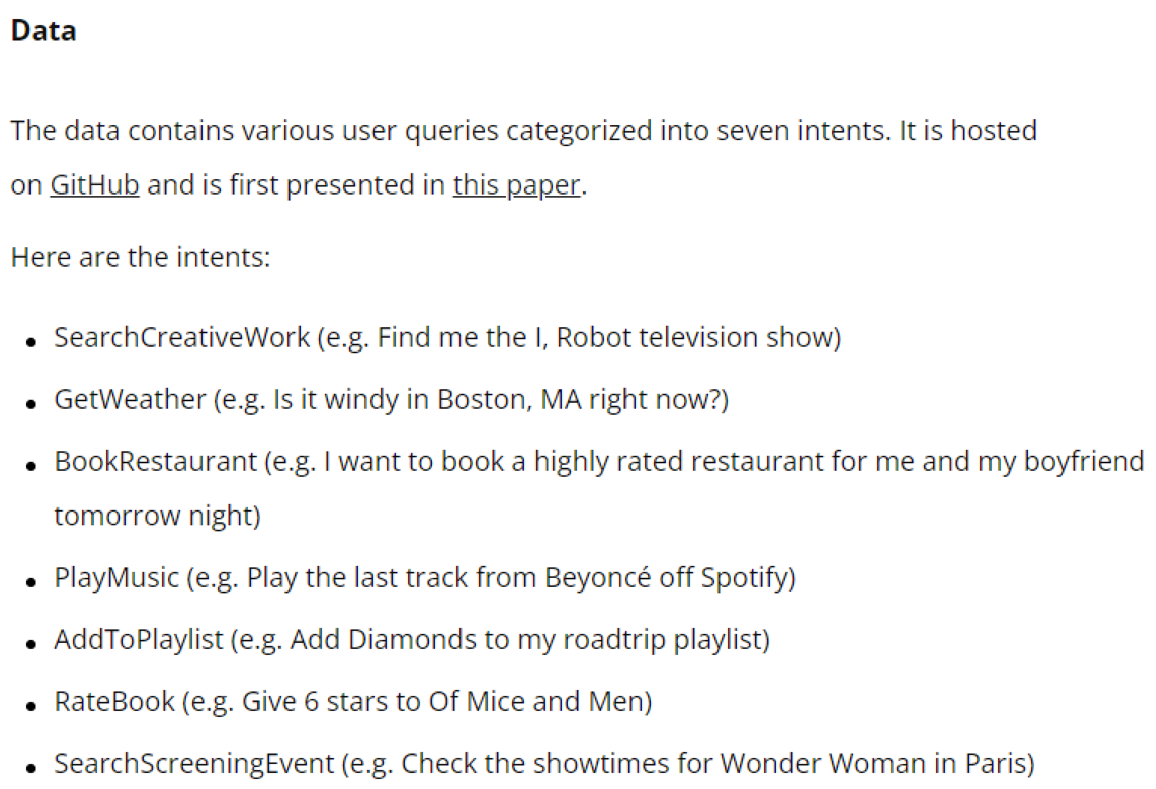
60 seconds 0

Utilisation	Speed		Base speed:	3.7...
17%	4.45 GHz		Sockets:	1
			Cores:	6
Processes	Threads	Handles	Logical processors:	12
299	9012	12265512	Virtualisation:	Dis...
Up time			Hyper-V support:	Yes
22:10:56:59			L1 cache:	38...
			L2 cache:	3.0...
			L3 cache:	32....

WORST CASE: (Didn't really have time to prepare for this. Was asked on the last date that my previous project may be too difficult for TAs and myself.)
Project: text interfaced Intent Recognition

Here are the papers I took screen shots and links from.
https://www.kdnuggets.com/2020/02/intent-recognition-bert-keras-tensorflow.html

1. Dataset:



Data

The data contains various user queries categorized into seven intents. It is hosted on GitHub and is first presented in this paper.

Here are the intents:

- SearchCreativeWork (e.g. Find me the I, Robot television show)
- GetWeather (e.g. Is it windy in Boston, MA right now?)
- BookRestaurant (e.g. I want to book a highly rated restaurant for me and my boyfriend tomorrow night)
- PlayMusic (e.g. Play the last track from Beyoncé off Spotify)
- AddToPlaylist (e.g. Add Diamonds to my roadtrip playlist)
- RateBook (e.g. Give 6 stars to Of Mice and Men)
- SearchScreeningEvent (e.g. Check the showtimes for Wonder Woman in Paris)

These look like a comprehensive dataset. Maybe I can create some more if there is a need.

2.
   a) There is no need but just in case:

Preprocessing

We need to convert the raw texts into vectors that we can feed into our model. We'll go through 3 steps:

- Tokenize the text
- Convert the sequence of tokens into numbers
- Pad the sequences so each one has the same length

Let's start by creating the BERT tokenizer:

```
tokenizer = FullTokenizer(
  vocab_file=os.path.join(bert_ckpt_dir, ”vocab.txt”)
)
```

Let's take it for a spin:

```
tokenizer.tokenize(”I can’t wait to visit Bulgaria again!”)
```

```
['i', 'can', "'", 't', 'wait', 'to', 'visit', 'bulgaria', 'again', '!']
```

The tokens are in lowercase and the punctuation is available. Next, we'll convert the tokens to numbers. The tokenizer can do this too:

```
tokens = tokenizer.tokenize(”I can’t wait to visit Bulgaria again!”)
tokenizer.convert_tokens_to_ids(tokens)
```

```
[1045, 2064, 1005, 1056, 3524, 2000, 3942, 8063, 2153, 999]
```

We'll do the padding part ourselves. You can also use the Keras padding utils for that part.

We'll package the preprocessing into a class that is heavily based on the one from this notebook:

b) Model: BERT? This is the name I found in the article. https://github.com/kpe/bert-for-tf2
   NOTE: the version of TensorFlow that is compatible with my python version is tf2

c) Evaluation Metric
   Didn't find specific type of metrics but tensorflow has a package named tensorboard that can generate eval metrics, regularization loss, learning rate and etc..

3. Make a primitive “voice” assistance but in text interface so I don’t need voice recognition model. It can be any platform. Maybe on web since it is easier?