ECE-MAE-148 – 2019 Summer Team4

SUMMARY

Objective:

Our course project's main objective is to create an autonomous public transport system that responds to calls for bus service.

Team Members:

Nikhil Challa
Richard Lai
Edison Chiu
Jiaye Wang

Video Recording Links:

Video of Drive Success – Controlled via Joystick [Week 1]	https://youtu.be/5q-gwsqywFY
Video of Drive Success – Autonomous [Week 2]	https://youtu.be/NTa6CsGERUc
Video Initial Socket Comm & adding to database [Week 4]	https://youtu.be/R0rohTYQhwE
Video of Drive Success – stop at circle color detection [Week 4]	
• *Case R-B [Stop at both Red & Blue]*	https://youtu.be/NmuKnkXDw9I
• *Case B-B [Don't stop at Red]*	https://youtu.be/wx5Gt8GrVPU
Video of Text Recognition [Week 4]	https://youtu.be/RXwLU-Pcnjo

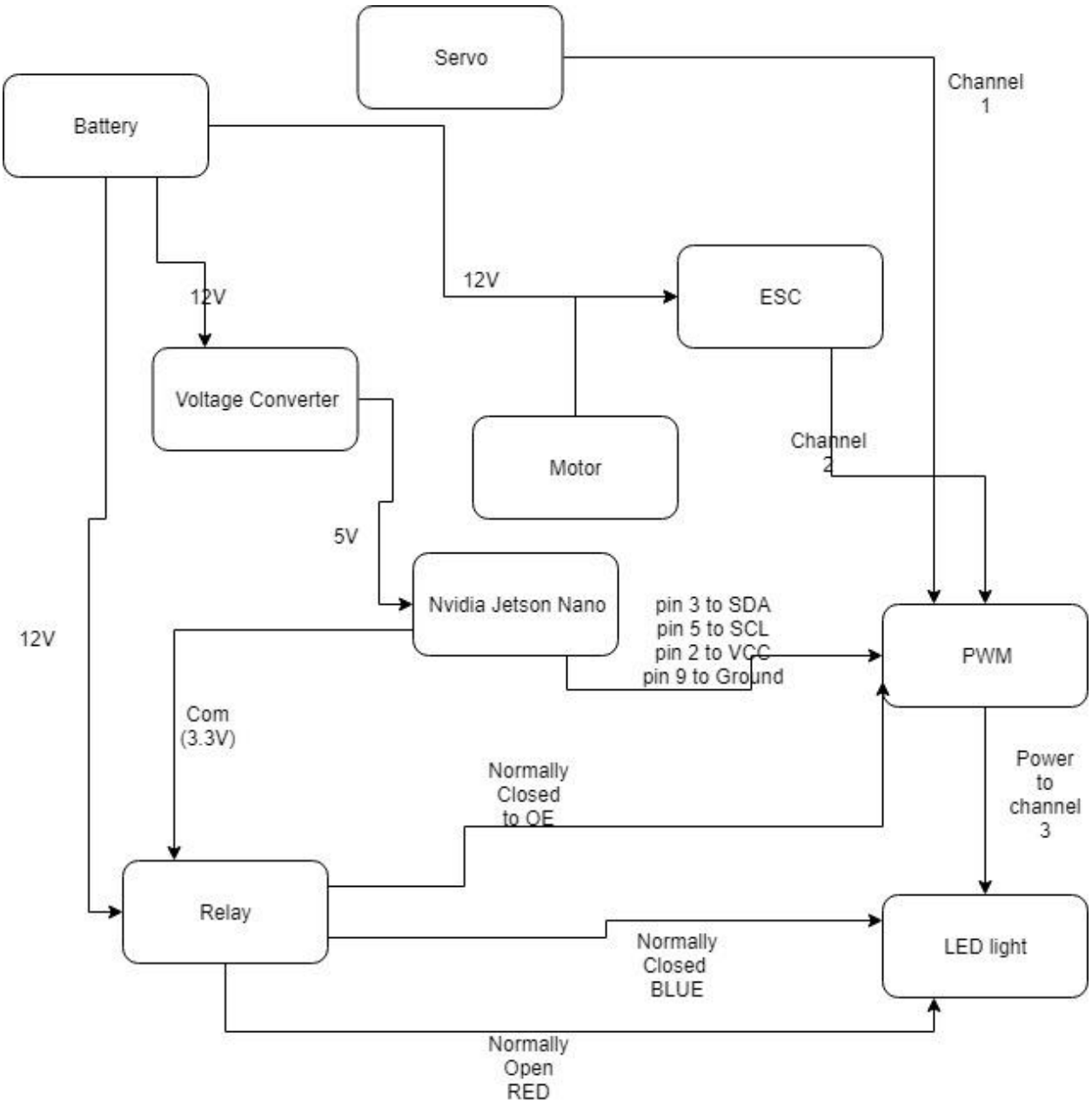
Pre-Course competition Presentation: [Week 3]

https://docs.google.com/presentation/d/1H2Ewvw5qM86MhphtP_fhXTXplnDbM_vN94Hb5jxKkR0/edit?fbclid=IwAR1CnCcgW3FtmE903dmpLGUnmoypTPhHH8tSun2QQTntF24pOASz90ppXwc#slide=id.g6124099362_0_4

Final Presentation:

https://docs.google.com/presentation/d/1utyLcdTrobHgFCBdYeFvQQH26zS5eyX6AysoQm0Fuwk/mobilepresent?slide=id.g41cb0863a2_10_3

Basic Circuit Diagram



Basic Proposal of Final Project: Autonomous Public Transportation

Features:

Must-Have:

a. Autonomous Driving Around Track.
b. Multiple Bus Stands along the track [3 in Demo, will need OpenCV to extract 'ONE' sign; details below*], with Sign placements as 'TWO', 'THREE' Etc.
c. Running Socket process to accept any command from an external device to request for pick up; drop off. [We will use a laptop and have a multi-thread server to accept connections from multiple clients**]
d. Running Command Center process to change between different modes. [Demo2 modes, 'School Bus' mode and 'Normal Passenger' mode; details below***]

Optional: [We could not complete any of the below features to focus on mandatory ones**]**

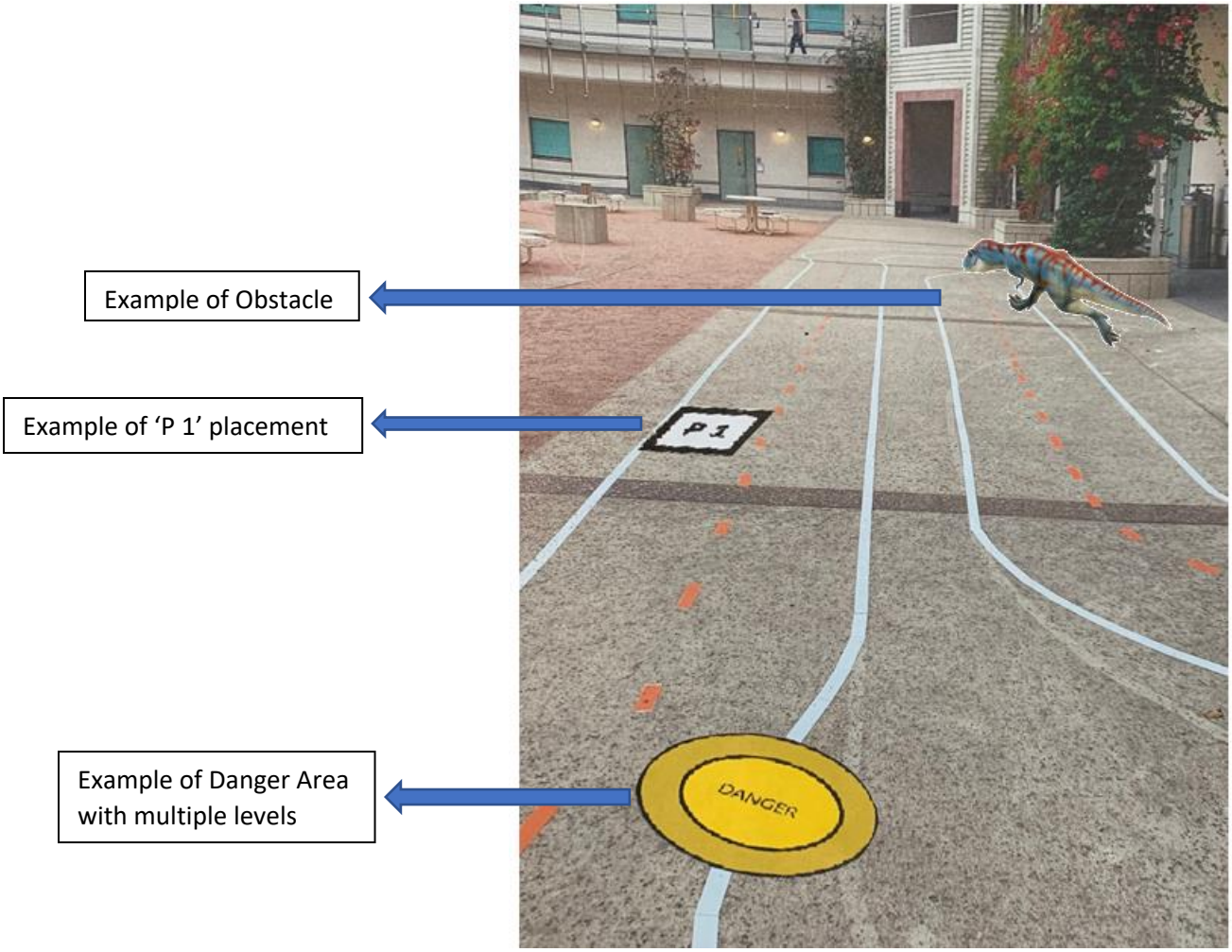
a. Collision Avoidance
b. GPS Tracking to avoid predetermined locations/danger zones

* OpenCV OCR using Tesseract, we would plan to extract the words 'ONE', 'TWO', 'THREE' and thereby making a stop if there is an open request or operating in 'School Bus' mode. Feedback to Throttle and Steering is required to overwrite the commands from model output until wait time is over.

** The Socket communication is to replicate a real-world scenario using apps and is used for the customer to place a request to drop and pickup. Addition code is required to erase requests upon stopping at the required point and mutex to handle the same file access issues.

*** Control process is to toggle between 'School Bus' mode and 'Normal Passenger' Mode.

Sample image for better understanding



Towards the end of the course, we have decided to conclude the course fulfilling the mandatory requirements due to time constraints and complexity in exploring new areas like socket programming/MySQL/OpenCV OCR/Tesseract. Further down are complete details including the base/reference code locations and learnings of different tools we utilized to fulfill the requirements.

Tables & Access Permissions

Tables:

VehicleMode Table

Mode Type	Max Speed	Danger Zone	Wait Time	Stop profile	External Request	Active
Normal Passenger	1meter/sec	0.25meter	2 secs	Upon request	Yes	NULL
School Bus	0.5meter/sec	0.5meter	5 secs	All Stops	No/Disable	N

Not all the fields will be used for Demo, this is just an example of potential uses

TransferRequest Table

Passenger Name	Start-Stop	End Stop
XX	B	R
YY	NULL	R

Permission Details:

server_process

- ⇨ Read permission for VehicleMode Table [Under USCDrobocar04_VehicleMode Database in mySQL]
- ⇨ Write permission for TransferRequest Table [Under UCSDrobocar04_TransferRequest in mySQL]

manage_process

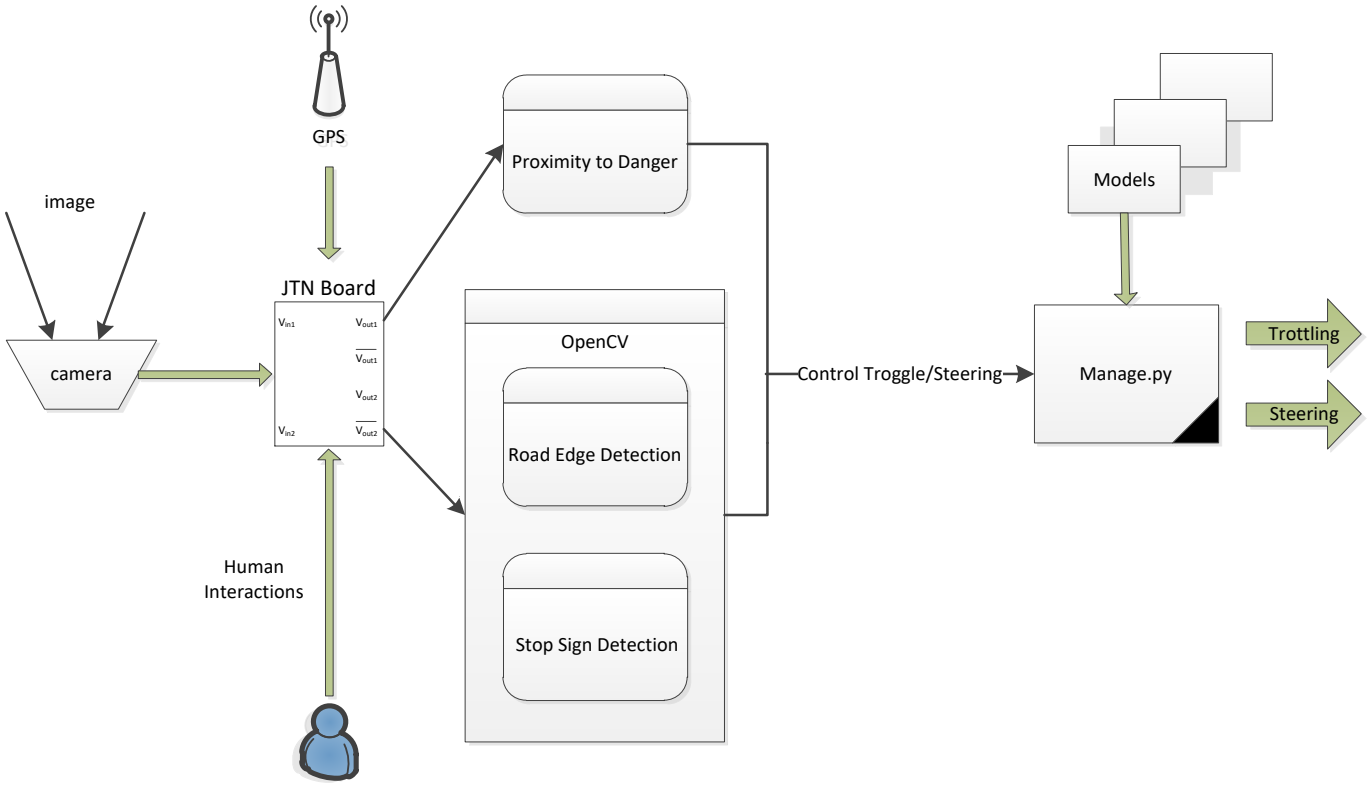
- ⇨ Read permission for VehicleMode Table [Under USCDrobocar04_VehicleMode Database in mySQL]
- ⇨ Write permission for TransferRequest Table [Under UCSDrobocar04_TransferRequest in mySQL]

control_process

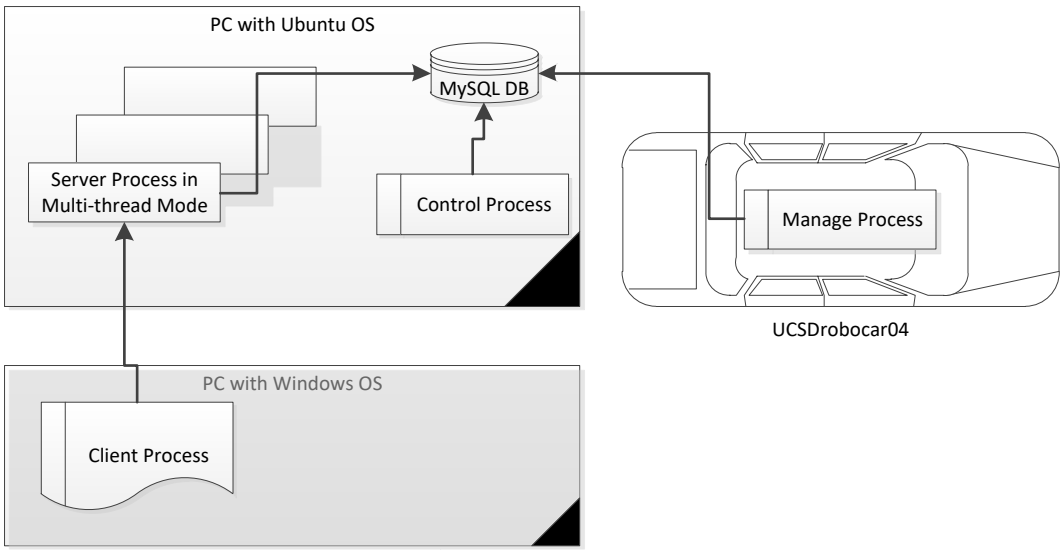
- ⇨ Write permission for VehicleMode Table [Under USCDrobocar04_VehicleMode Database in mySQL]

Process Flow Chart

Overall Flow:



Process Location:



MySQL Details

Reference Link: https://support.rackspace.com/how-to/installing-mysql-server-on-ubuntu/

We need MySQL connection to link python to MySQL database for editing/reading
```
pip install mysql-connector-python-rf
```

Install the MySQL server by using the Ubuntu package manager:
```
sudo apt-get update
sudo apt-get install mysql-server
```

Allow remote access, if you have ufw on your machine. If not, this step can be ignored. We have encountered issues with python socket programming communication with ufw but didn't get time to identify how to resolve it.
```
sudo ufw allow mysql
```

Start the MySQL service
```
systemctl start mysql
```

Launch at reboot
```
systemctl enable mysql
```

Make sure to bind MySQL server to any ipaddress
For ubuntu 18.04, edit /etc/mysql/my.conf.d/mysql.cnf; go to 'bind-address' and change from 127.0.01 to 0.0.0.0 to listed to all address. If security is concern, list the exact ip address

Note, once you have changed the bind_address to a value other than 0.0.0.0, you will need to add the host IP address even to local processes.

After saving the changes, do 'systemctl restart mysql'

To check if MySQL is fine, you can root to mysql

Start MySQL shell; note you need to do sudo for root; for other uses, you don't need to add 'sudo'
```
sudo /usr/bin/mysql -u root -p
```

Once the shell is open, you can enter the below command while in MySQL shell…
```
CREATE DATABASE UCSDrobocar04_VehicleMode;
```
We will have to create 2 databases, instead of a single database, as each user can control user access on the entire database and not just a single table. If we have a single database; a user write will block every table in the database which is not good.

```
CREATE DATABASE UCSDrobocar04_TransferRequest;
```
to keep track of requests via a client process

```
SHOW DATABASES;
```

Add a database user – Add a new user instead of using root, since for root I am not able to edit on my machine. By creating a new user, you can assign/reassign privileges or experiment without issues.

```
INSERT INTO mysql.user (User,Host,authentication_string,ssl_cipher,x509_issuer,x509_subject)
VALUES('server_process','localhost',PASSWORD('team4ucsd'),'','','');

INSERT INTO mysql.user (User,Host,authentication_string,ssl_cipher,x509_issuer,x509_subject)
VALUES('maange_process','localhost',PASSWORD('team4ucsd'),'','','');

INSERT INTO mysql.user (User,Host,authentication_string,ssl_cipher,x509_issuer,x509_subject)
VALUES('control_process','localhost',PASSWORD('team4ucsd'),'','','');
```

For any major activity, always use the below command, to ensure all users connected to DB are in sync

```
FLUSH PRIVILEGES;
```

To see if the user has been added

```
SELECT User, Host, authentication_string FROM mysql.user;
```

To access a database, you will need to access the database as 'root' first and start adding tables and privileges before trying to access the table as a non-root user

To latch on to the database

```
USE UCSDrobocar04_VehicleMode;
```

Below is an example of how to create table//PRIMARY KEY is to assign this as the primary key for replacement function

```
CREATE TABLE VehicleMode (ModeType VARCHAR(20), MaxSpeed FLOAT(2,1), DangerZoneRadius FLOAT(3,2), WaitTime
SMALLINT(4), StopProfile VARCHAR(20), ExternalRequest VARCHAR(20), Active VARCHAR(1), PRIMARY KEY(ModeType));
```

Do the same for the other table in UCSDrobocar04_TransferRequest.

```
CREATE TABLE TransferRequest (PassengerName VARCHAR(20), StartLocation ENUM('P1','P2','P3'), EndLocation
ENUM('P1','P2','P3'), PRIMARY KEY (PassengerName));
```

In case you forgot to add a primary key, use the below command as an example

```
ALTER TABLE PriceList ADD PRIMARY KEY(PassengerName);
```

To add privileges for the entire Database example below; we are going to add to the per table instead of the entire database

```
GRANT ALL ON UCSDrobocar04_TransferRequest.TransferRequest TO server_process@localhost IDENTIFIED BY 'team4ucsd'
WITH MAX_USER_CONNECTIONS 1;

GRANT ALL PRIVILEGES ON UCSDrobocar04_TransferRequest.* to manage_process@localhost IDENTIFIED BY 'team4ucsd'
WITH MAX_USER_CONNECTIONS 1;


GRANT ALL PRIVILEGES ON UCSDrobocar04_VehicleMode.* to control_process@localhost IDENTIFIED BY 'team4ucsd' WITH
MAX_USER_CONNECTIONS 1;

GRANT SELECT, SHOW VIEW ON UCSDrobocar04_VehicleMode.* TO server_process@localhost IDENTIFIED BY 'team4ucsd'
WITH MAX_USER_CONNECTIONS 1;

GRANT SELECT, SHOW VIEW ON UCSDrobocar04_VehicleMode.* TO manage_process@localhost IDENTIFIED BY 'team4ucsd'
WITH MAX_USER_CONNECTIONS 1;
```

The MAX_USER_CONNECTIONS is to ensure only 1 thread from the same multithread process, or 1 separate process can access the table

Below command to ensure mutual exclusion in file/table access. There is a limitation to locking a particular table, instead, all tables will be locked.

```
GRANT LOCK TABLES, SELECT ON UCSDrobocar04_TransferRequest.* TO server_process@localhost IDENTIFIED BY 'team4ucsd';

GRANT LOCK TABLES, SELECT ON UCSDrobocar04_TransferRequest.* TO manage_process@localhost IDENTIFIED BY 'team4ucsd';
```

Below to query on details on tables. doesn't show the entries just column/row headers

```
SHOW TABLES;
```

Below is a query on how to get columns assigned to Table; doesn't show table entries

```
DESCRIBE VehicleMode
```

To insert an entry to the table

```
INSERT INTO VehicleMode VALUES ('Normal Passenger',1.0,0.25,2,'Upon Request','Yes', 'Y');
INSERT INTO VehicleMode VALUES ('School Bus',0.5,0.5,5,'All Stops','No/Disable', 'N');
```

Replace 'INSERT' with 'REPLACE' with the above same syntax to allow replacement or addition if the entry doesn't exist. REPLACE will only if PRIMARY KEY is provided

To print out all tables

```
SELECT * FROM VehicleMode;
```

To test the remote connection, from windows OS

[install from https://dev.mysql.com/downloads/installer/ or other similar sites]

```
c:\Program Files\MySQL\MySQL Shell 8.0\bin>mysqlsh.exe -u manage_process -h <mqsql machine ipaddr> -p
```

After entering the password and connecting to SQL, it will be in JS mode, change the mode to SQL by entering `\sql`
Then any command like Linux/ubuntu will work, except all lower-case commands.

To quit: `\q` [lower case only]

To test the remote connection, from MAC OS

```
mysql -u manage_process -h <mqsql machine ipaddr> -p
```

For more references, use the below links
https://pythonspot.com/mysql-with-python/
https://dev.mysql.com/doc/refman/5.7/en/creating-tables.html
https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html

OpenCV OCR using Tesseract

You need to install 'imutils' to exploit the great benefits of image processing.

`pip install imutils` or `pip install –upgrade imutils` if you already have 'imutils'

We can use Tesseract for text recognition.

```
sudo apt-install tesseract-ocr
```

Now we need to install python bindings.

```
pip install pillow
pip install pytesseract
```

Reference Link:

https://www.pyimagesearch.com/2018/09/17/opencv-ocr-and-text-recognition-with-tesseract/

Problems we encountered with OpenCV OCR using Tesseract

a) Text Size for processing; too large or too small font size doesn't yield the right results.
The best font size for OCR is 10 to 12 points. The use of smaller font sizes has led to bad OCR. Font sizes greater than 72 are considered images, and thus should be avoided.

b) Tesseract doesn't work well with high contrast borders, identifies them as alphabets. Also considers most black borders as text. Used Black Text on white paper

c) Tesseract doesn't recognize similar characters, like P is recognized as D, 1 as 4, or l.

d) Tesseract works very well if the text is linguist recognized word, eg 'P1' doesn't work but 'ONE' works very well. Replaced 'Px' with 'ONE', 'TWO', and 'THREE', and detection was 100% accurate.

e) Running image through deep learning model taking time, in order of secs. Shorter time if higher processing power is available.

f) Range of Detection of text, 1 foot to 5 feet reliably. A longer-range is possible if a higher resolution camera was used.

g) Image recognition was quite poor if the image was kept on the floor instead of vertical due to dimension stress of the image that Tesseract couldn't recognize very well. We decided to keep the image in a vertical position instead of on the floor.

To Demo our case and create a video of success, we decided to use smaller functions that take less time to process. We used Circle filled with single Color [Red and Blue].

At this time, we are exploring other solutions; some examples are listed below. We are not certain of success at the time document was written

a. Forking the main process to another process periodically to run separately for image processing while using the main thread for driving only. The separate process to write to file which is to be read by the main process after predetermined time [1-2 sec in our case]
b. Off-loading processing to supercomputer instead of onboard if step a doesn't help.
c. Driving car at very low speeds to ensure enough time to get results from image processing

Conclusion - What did we learn!

A Lot in just 5 weeks!

a. Understanding how to interface primary components to drive a Car.
b. Understanding Syntax of Linux/Ubuntu operating system
c. Understanding the basis of now Training model works
d. Observing for real, ML application
e. OpenCV and OCR
f. Working of Tesseract and identifying limitations
g. MySQL
h. Python Socket programming with multi-threading
i. Python mysql.connector

Reference Source Code Locations in GitHub

https://github.com/niil87/believe/tree/master/pythonScripts/For_UCSD_RoboCar

https://github.com/wingz0c/ucsdrobocar04