

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

CAMPUS CIUDAD DE MÉXICO

ESCUELA DE DISEÑO, INGENIERÍA Y ARQUITECTURA

DEPARTAMENTO DE INGENIERÍA MECATRÓNICA

INGENIERÍA EN TECNOLOGÍAS COMPUTACIONALES



**Tecnológico
de Monterrey**

Reporte Servicio Social

PROYECTO SISTEMAS INTELIGENTES DE ASISTENCIA VIAL

GUILLERMO DAVID AGUILAR CASTILLEJA

A01337242

ASESORES:

DR. MARTÍN ROGELIO BUSTAMANTE BELLO

ING. LUIS ALBERTO CURIEL RAMIREZ

ING. JAVIER IZQUIERDO REYES

MAYO DE 2018

INTRODUCCIÓN

En el periodo de servicio social que he realizado he realizado una representación del Tecnológico de Monterrey Campus Ciudad de México actualmente, entrenamiento y pruebas del modelo de una red neuronal para diferentes ambientes, incluyendo el diseñado en este periodo (CCM). Este modelo es de gran importancia dado que es un ambiente seguro de desarrollo en el que se pueden representar situaciones diferentes y que podemos adaptar a múltiples técnicas y modelos de Machine y Deep learning, lo cual nos ayudará a incrementar la velocidad de desarrollo del proyecto, a obtener una mayor seguridad de que el modelo sirve correctamente antes de implementarlo al automóvil real y simular situaciones de riesgo a las que se tiene que enfrentar el modelo implementado.

OBJETIVOS

Objetivo General

Optimización del modelo de la red neuronal a través de herramientas de open source y ambientes diseñados a la medida para simular casos reales, corrección de errores al comportamiento que tenga el modelo en el simulador. Representar el campus ciudad de México en un modelo 3D para implementarlo en un sistema de simulación donde se puedan hacer pruebas seguras y confiables.

Objetivos Especificos

1. Modificación del Menú principal para poder acceder al modelo.
2. Diseño de la maqueta en formas 3D del Tecnológico de Monterrey Campus Ciudad de México.
3. Realización del modelo en Unity de la maqueta para fácil modificación.
4. Creación de un ejecutable y unitypackage para poder realizar los entrenamientos de los diferentes modelos de redes neuronales.

ENTREGABLES

El entregable principal se trata del Unitypackage y ejecutable del proyecto con todos los assets, código en c#, animaciones, prefabs y objetos utilizados para la realización de la representación 3D del Tecnológico de Monterrey Campus Ciudad de México y el recorrido vehicular con el que cuenta la institución. También este proyecto contiene el modelo de un automóvil que tiene los funcionamientos básicos de un vehículo, integrado con una cámara. Ejecutable .exe para Windows x86 y x64 para que pueda ser utilizada en los entrenamientos.

ACTIVIDADES

Revisar la documentación de Udacity acerca de Self Driving Car Simulator project que es libre para la comunidad y descargar el game build actualizado.

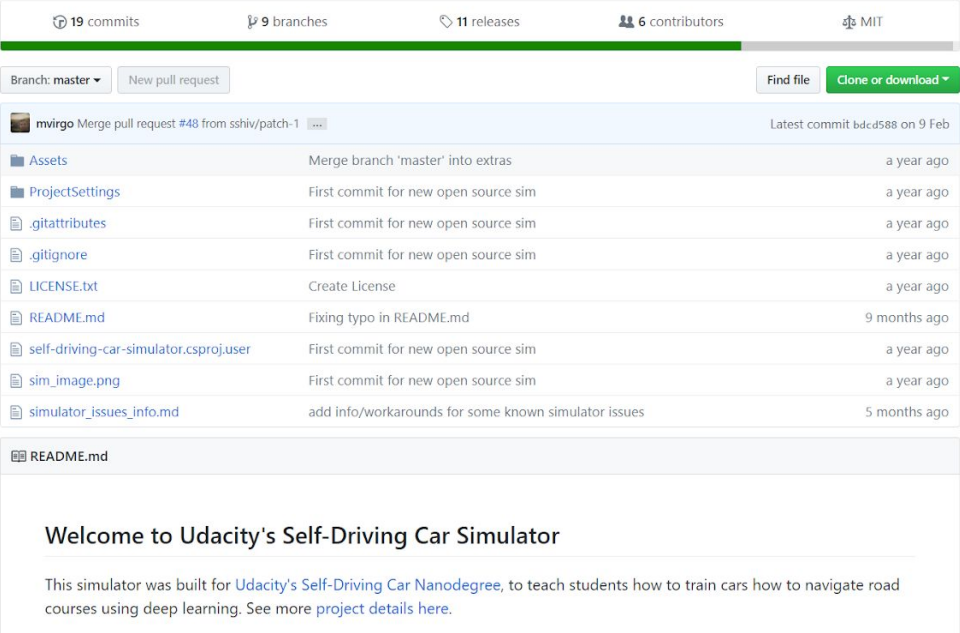
Instalación de Unity, para el desarrollo de la maqueta en el proyecto libre de entrenamiento de redes neuronales.

Investigación del funcionamiento del proyecto, para poder modificarlo fue necesario leer toda la documentación acerca del proyecto.

Instalar las dependencias necesarias para que el proyecto corriera en el ordenador y también para determinar si este modelo es útil para el proyecto y nuestros modelos de redes neuronales del proyecto ADMAS.

DESARROLLO

Fue necesario la instalación de Unity en el ordenador para el desarrollo del proyecto junto con programas como Git y Git LFS para la descarga del Game Build ya que es un archivo bastante grande, instalación de dependencias de Unity para poder utilizar este recurso.



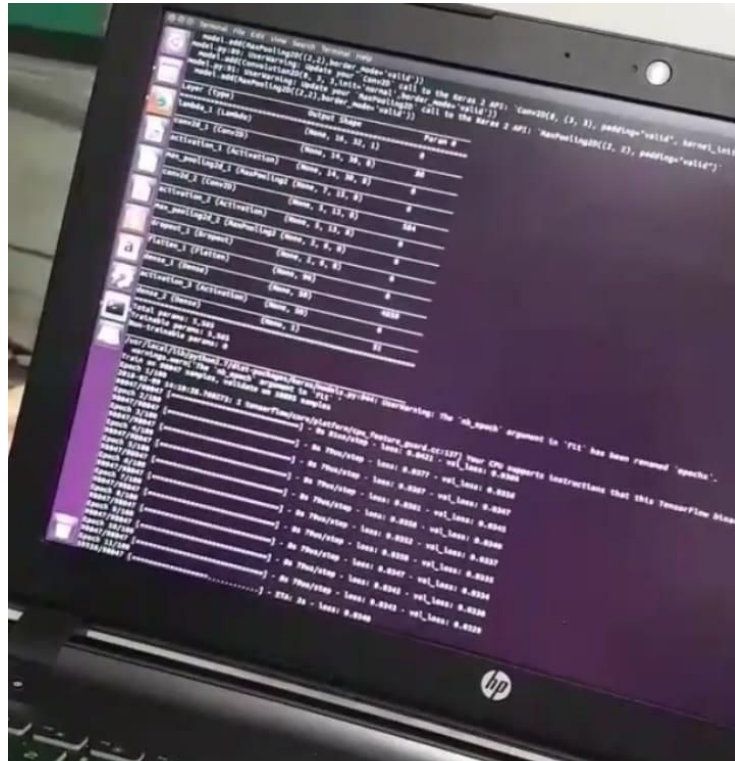
The screenshot shows the GitHub interface for the repository 'Udacity's Self-Driving Car Simulator'. At the top, it displays statistics: 19 commits, 9 branches, 11 releases, 6 contributors, and the MIT license. Below this, there are buttons for 'Branch: master', 'New pull request', 'Find file', and 'Clone or download'. A merge pull request #48 is highlighted, showing a commit history table with columns for file name, commit message, and time ago. The files listed include Assets, ProjectSettings, .gitattributes, .gitignore, LICENSE.txt, README.md, self-driving-car-simulator.csproj.user, sim_image.png, and simulator_issues_info.md. Below the table, the README.md content is visible, starting with 'Welcome to Udacity's Self-Driving Car Simulator' and providing a link to project details.

File	Commit Message	Time Ago
Assets	Merge branch 'master' into extras	a year ago
ProjectSettings	First commit for new open source sim	a year ago
.gitattributes	First commit for new open source sim	a year ago
.gitignore	First commit for new open source sim	a year ago
LICENSE.txt	Create License	a year ago
README.md	Fixing typo in README.md	9 months ago
self-driving-car-simulator.csproj.user	First commit for new open source sim	a year ago
sim_image.png	First commit for new open source sim	a year ago
simulator_issues_info.md	add info/workarounds for some known simulator issues	5 months ago

Hki wt c "3Q'F guæcti c 'f'g'Tgewt uqu0

Para el poder utilizar el simulador descargado fue necesario de la lectura de la documentación donde señalaba como esta estructurado el proyecto. El siguiente paso fue probarlo para

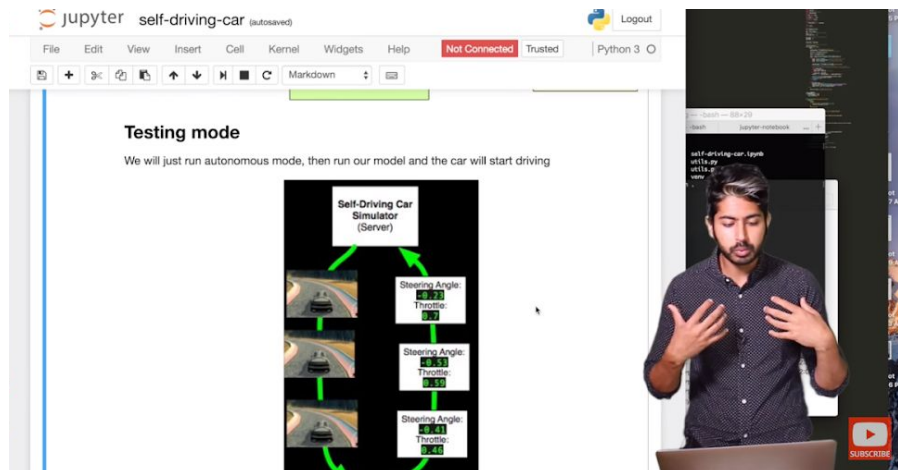
determinar si este simulador se podía utilizar para las redes neuronales que se utilizan para ADMAS, para esto se hecho a andar el simulador y se hicieron entrenamientos con el modelo que viene por defecto en el proyecto, teniendo que manejar el simulador para obtener los datos. Esta toma de datos y entrenamiento se realizó con más de 65,000 imágenes y datos de ángulo y velocidad. Se realizaron varios entrenamientos con diferentes parámetros en el modelo para poder tener un manejo del vehículo automático óptimo.



Hki wt c'4Q'Gpvt gpc o kgpvq'f gl'Uko wxf qt O'

"

Se revisó acerca de la estructura de la red y el método para entrenarla con contenido libre en internet por parte del experto en ciencias computacionales Siraj Raval en su canal especializado en inteligencia artificial.



Hki wt c '5Q' Ewtuq' rqt 'Uk cl' Texc r0'

''

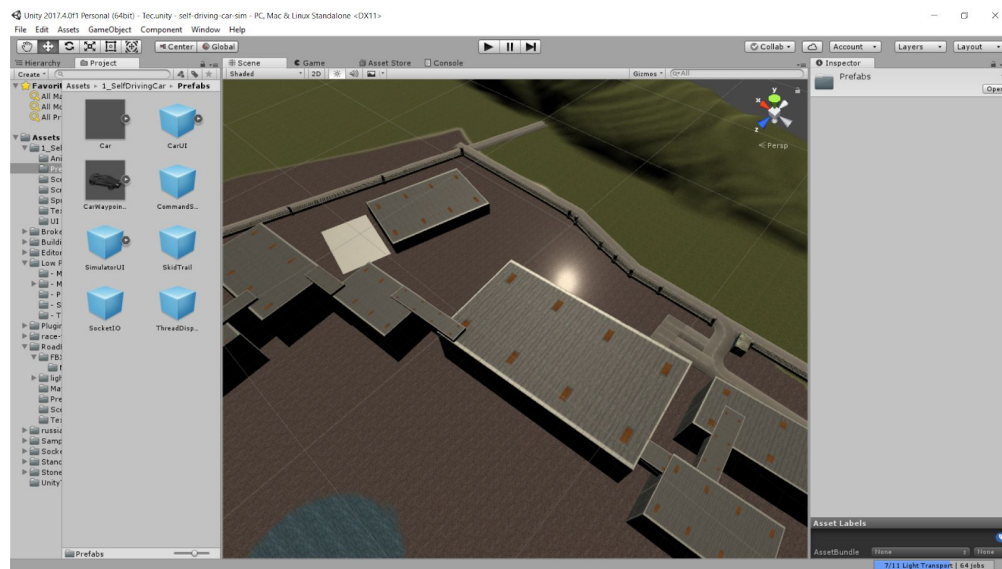
En este curso obtuve los fundamentos para poder echar a andar el simulador y empezar a recopilar datos a través de conducción en el simulador y un sistema de toma de imágenes que guardaba a su vez los datos del ángulo y velocidad. Después se utilizó para entrenar el modelo en el que utilizamos una variación de veces que realiza el entrenamiento, cuidando que los porcentajes de error fueran bajos. Al finalizar con cada uno de los entrenamientos se realizaron pruebas de manejo automático con el uso de un servidor en Node que diera instrucciones a vase del modelo previamente entrenado. Algunas fallaron y otras fueron exitosas lo que rascaba en el números de datos con lo que se entrenaba el modelo.



Hki wt c '6Q' Gpvt gpc o kgpvqu'f gn'uko wrcf qt 0'

”

Después de realizar los entrenamientos de prueba y conocer el funcionamiento del simulador la fase siguiente consistió en el diseño del Tecnológico de Monterrey Campus Ciudad de México en un modelo 3D para que pudiese ser implementado en el simulador para después desarrollar y entrenar modelos de redes neuronales basados en un ambiente conocido para poder simular de forma segura el manejo de un vehículo autónomo en escenarios reales.



Hki wt c "7Q" F guct t qmq" f grlVgepqm»i keq" f g" O qpvgt t gl "EEOO"

Por último se detalló la pista para que pudiese representar de mejor manera el campus y pudiese ser utilizado para los entrenamientos de los modelos. El modelo en unity está listo para ser desplegado en múltiples plataformas como Windows, Linux y Mac.

Conclusiones

Es de vital importancia de tener un ambiente de pruebas para el desarrollo de redes neuronales dado que antes de poder implementar una solución en el mundo real debemos pasar por rigurosas pruebas para asegurar la seguridad de los usuarios que van dentro del automóvil, esto nos lleva a que el ambiente de desarrollo debe representar situaciones reales en un ambiente seguro que es el virtual, sin sacrificar la calidad e importancia de las pruebas. Esto asegura que el

usuario no será expuesto en riesgo al ser implementada la red neuronal en un vehículo que esté ocupará, dará una asistencia vial de calidad y disminuye los riesgos de un conductor promedio como está planteado en el proyecto de ADMAS.

Bibliografía

Udacity. (2018). self-driving-car. 30 de abril del 2018, de Github Sitio web:

<https://github.com/udacity/self-driving-car>

Oliver Cameron. (2016). We're Building an Open Source Self-Driving Car. 30 de abril del 2018, de Medium Sitio web:

<https://medium.com/udacity/were-building-an-open-source-self-driving-car-ac3e973cd163>

Harrison Kinsley. (2016). Introduction to Neural Networks. 11 de octubre, de Python Programming Sitio web:

<https://pythonprogramming.net/neural-networks-machine-learning-tutorial/>

Thomas Gallice y Aymeric Damien. (2016). Alexnet. 1 de noviembre del 2017, de GitHub Sitio web:

<https://github.com/tflearn/tflearn/blob/master/examples/images/alexnet.py>

Karol Zieba. (2016). End to End Learning for Self-Driving Cars. 4 de octubre, de NVIDIA Corporation Sitio web: <https://arxiv.org/pdf/1604.07316.pdf>

Avinash Sharma. (2015). Understanding Activation Functions in Neural Networks. 5 de noviembre del 2017, de Medium Sitio web:

<https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>

Anexos

