**PEMBUATAN MODEL DEEP LEARNING UNTUK SIMULATOR SELF DRIVING CAR YANG DITRAINING MENGGUNAKAN GPU**

**SKRIPSI**

**Oleh:**

**ADWI ARIFIN**

**115090300111008**

****

**JURUSAN FISIKA**

**FAKULTAS MATEMATIKA DAN   
ILMU PENGETAHUAN ALAM**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2017**

BAB I  
PENDAHULUAN

1. BAB I PENDAHULUAN
   1. Latar Belakang

Perkembangan self-driving car semakin lama semakin baik, banyak perusahaan berlomba-lomba membuat mobilnya sendiri untuk dapat dikendarai tanpa pengemudi. Hal salah satunya disebabkan oleh kemajuan di bidang teknik Machine Learning, yaitu Deep Neural Network. Pada dasarnya Deep Neural Network adalah Artificial Neural Network dengan jumlah Hidden Layer yang sangat banyak, maka dari itu disbut “deep”. Teknik ini pertama kali dikenalkan sudah sejak lama, namun tidak banyak digunakan karena membutuhkan proses komputasi yang cukup besar. Seiring dengan perkembangan jaman, pada tahun 2011 perkembangan GPU meningkat secara signifikan, yang salah satu efeknya adalah memungkinkan melatih neural network dengan menggunakan GPU. Sejak saat itu banyak penelitian tentang Deep Neural Network, beberapa aplikasi yang digunakan antara lain Image Recognition, Voice Recognition, Pattern Recognition, Natural Languange Processing, dan lain-lain. Perkembangan Deep Neural Network di bidang Image Recognition, dalam beberapa aspek dapat mengimbangi dan bahkan lebih baik dibanding dengan Image Recognition Tradisional.

Dari sini penulis ingin mencoba menunjukkan bagaimana Image Recognition dengan Menggunakan Deep Neural Network spesifik pada kasus Self Driving Car. Kendala dari topik tersebut membutuhkan kompleksitas dan biaya yang cukup tinggi, karena membutuhkan mobil dengan spesifikasi yang memadai, lalu dibutuhkan modifikasi pada mobil itu, hingga memasangkan perangkat lunak untuk mengontrol mobil tersebut. Pada bulan mei 2017, salah satu perusahaan Online Course yang bernama Udacity, meng-open-source kan salah satu produknya yaitu Udacity Self Driving Simulator. Simulator ini ada program yang bisa dijalankan di komputer, lalu bertindak selayaknya mobil yang akan kita program agar dapat mengendarai dirinya sendiri. Simulator ini dapat menyediakan / menghasilkan data training, sekaligus dapat diberikan perintah untuk mengontrol mobil yang ada di dalamnya. Oleh karena itu penelitian ini tidak akan menggunakan mobil nyata, namun menggunakan simulator sebagai penggantinya.

* 1. Rumusan Masalah Penelitian

Berdasrkan latar belakang yang telah disebutkan, dapat dirumuskan beberapa rumusan masalah pada tugas akhir ini adalah:

1. Bagaimana membuat program untuk dapat mengontrol simulator dengan menggunakan Deep Neural Network?
2. Bagaimana proses training Deep Neural Network?
3. Bagaimana perbandingan proses training menggunakan CPU dan GPU?
4. Bagaimana proses testing pada simulator? Seberapa akurat?
   1. Batasan Masalah

Ruang lingkup penelitian tugas akhir ini dibatasi beberapa hal yaitu:

1. Simulator digunakan sebagai pengganti objek mobil yang sebenarnya.
2. Hanya mengamati bagaimana hasil training maupun testing, tidak membahas simulator.
   1. Tujuan Penelitian

Tujuan penelitian pada tugas akhir ini adalah:

1. Membuat program training self-driving car pada simulator.
2. Membandingkan training menggunakan CPU maupun GPU.
3. Menganalisa tingkat presisi hasil training self-driving car pada simulator
   1. Manfaat Penelitian

Manfaat penelitian tugas akhir ini adalah:

1. Sebagai tahap awal penelitian tentang self-driving car, agar kedepannya nanti dapat dibuat model-model yang lebih baik.
2. Terciptanya model self-driving car yang aman dan nyaman. Setidaknya pada tingkat simulator.

BAB II  
TINJAUAN PUSTAKA

1. BAB IV HASIL DAN PEMBAHASAN
   1. Artificial Intelegence

Artificial Intelegence (AI) atau dalam bahasa indonesia disebut dengan Kecerdasan Buatan adalah suatu teknik pengembangan mesin yang dapat berpikir ataupun bertindak layaknya manusia. Cakupan Kecerdasan Buatan cukup luas, meliputi perencanaan (planning), dapat memahami bahasa, dapat mengenali objek ataupun suara, penyelesaian masalah, dan bahkan pembelajaran.

Secara umum kecerdasan buatan dapat dibagi menjadi dua kategori, yaitu kecerdasan umum (General AI) yaitu kecerdasan buatan yang dapat melakukan semua hal yang telah disebutkan. Kategori berikutnya yaitu kecerdasan khusus (Spesific AI) yaitu kecerdasan yang dapat melakukan satu tugas dengan sangat baik, namun tidak dapat melakukan tugas yang lain. Contohnya ada kecerdasan yang dapat mengenali objek dengan baik, namun tidak dapat mengenali suara sama sekali.

* 1. Machine Learning

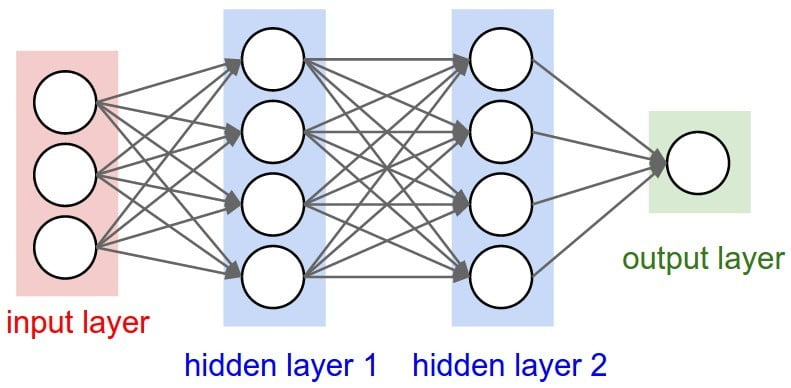
Machine Learning (ML) atau dalam bahasa indonesia disebut dengan pembelajaran mesin, adalah suatu pendekatan untuk dapat mewujudkan kecerdasan buatan. Pada tahun 1959, Arthur Samuel pernah mengemukakan pendapatnya tentang Machine Learning yaitu “suatu kemampuan untuk belajar tanpa harus diprogram secara eksplisit”. Dari pernyataan Arthur dapat kita ketahui bahwa pada dasarnya kita bisa mendapatkan kecerdasan buatan tanpa menggunakan pembelajaran mesin, yaitu dengan cara membuat perintah percabangan yang cukup banyak dan kompleks.

Dibanding dengan cara percabangan yang cukup banyak, pembelajaran mesin hadir dengan cara melatih program agar dapat mengerti input ataupun output yang diberikan. Dalam proses ini biasanya kita memberikan sejumlah input ataupun output yang sangat banyak, lalu membiarkan program untuk menyesuaikan sendiri apa korelasi antara input dan output yang diberikan.

Salah satu contoh dari dari pembelajaran mesin yaitu untuk mengenali suatu objek dalam gambar. Misalnya kita memberikan sejumlah gambar dan melabeli beberapa gambar tersebut dengan gambar yang memiliki objek kucing dan yang tidak memiliki kucing. Pada akhirnya algoritma pembelajaran mesin ini dapat membentuk suatu model yang dapat mengenali ada atau tidaknya kucing dalam suatu gambar.

* 1. Artificial Neural Network

Artificial Neural Network tersusun atas beberapa elemen, yaitu node, network, dan layer. Node disimbolkan dengan simbol lingkaran, yaitu merepresentasikan suatu informasi. Network disimbolkan dengan tanda panah, yaitu menghubungkan antar satu node dengan node yang lainnya, satu tanda panah berarti node awal akan mempengaruhi node yang ditunjuk. Dalam network ini terdapat properti yang disebut dengan “weight” atau “bobot” dan juga “bias”. Layer merepresentasikan sekumpulan node yang tergabung dalam satu kelompok. Dalam satu Artificial Neural Network dipastikan memiliki satu input layer, satu output layer dan berapapun hidden layer. Jumlah hidden layer yang sedikit sering disebut dengan Shallow Neural Network, sedangkan jumlah hidden layer yang cukup banyak disebut dengan Deep Neural Network.



Gambar 2.1 Artificial Neural Network

* 1. Deep Learning

Deep Learning adalah salah satu teknik agar dapat mengimplementasikan pembelajaran mesin. Selain Deep Learning, terdapat beberapa teknik lain seperti Decision Tree Learning, Inductive Logic Programming, Clustering, Reinforcement Learning, Bayesian Network, dan masih banyak lagi yang lainnya.

Deep Learning merupakan suatu teknik pembelajaran mesin dengan menggunakan Deep Neural Network. Teknik Deep Learning ini awalnya tidak cukup populer karena dengan jumlah hidden layer yang cukup besar, maka kompleksitas program semakin besar, sehingga waktu komputasi yang dibutuhkan pun cukup besar. Namun dengan seiring perkembangan teknologi perangkat keras yang memadai, teknik Deep Learning kembali diminati. Karena dapat menyelesaikan permasalahan yang cukup kompleks.

* 1. Python

Python merupakan suatu bahasa pemrograman tingkat tinggi yang bersifat open source. Python memiliki struktur data tingkat tinggi yang efisien dan pendekatan terhadap pemrograman berorientasi object (OOP) yang sederhana namun efektif. Python telah banyak digunakan di pengembangan perangkat lunak, bisnis, pendidikan, pemerintahan, sains dan teknik. Kelebihan python yang lain yaitu didukung library yang cukup banyak dan terjamin karena di maintance oleh komunitasnya masing-masing.

* 1. Keras

Keras merupakan salah satu library python yang ditujukan untuk pengolahan neural network. Keras mendukung banyak algoritma termasuk di dalamnya yaitu deep neural network. Kelebihan keras yaitu dengan sintaks yang sederhana, keras dapat menjalankan library lain antara lain TensorFlow, Theano, ataupun Microsoft Cognition Toolkit.

* 1. TensorFlow

Tensor Flow sendiri adalah machine learning engine yang dipakai Google di banyak aplikasi mereka, mulai dari pengenalan suara, SmartReply yang membantu pengguna dengan mengidentifikasi email penting sekaligus memberikan usulan balasannya, pengenalan gambar yang memungkinkan kita melakukan pencarian berdasarkan foto, mengenali dan menerjemahkan tulisan dari sebuah foto, dan lain-lain.

Menurut Google, engine ini dapat digunakan baik dalam riset maupun komersial, mulai dari mesin besar sampai telepon genggam. TensorFlow menggunakan metode deep learning dalam prosesnya, namun dapat juga menggunakan reinforcement learning and logistic regression.

Tensorflow mendukung metode learning dengan menggunakan CPU ataupun GPU. Dengan melakukan learning menggunakan GPU, maka waktu yang dibutuhkan untuk melakukan proses training akan lebih cepat. Untuk saat ini GPU yang didukung hanya GPU dari NVIDIA saja.

* 1. GPU

GPU (singkatan dari Graphics Processing Unit) adalah sebuah prossesor khusus untuk untuk bagian grafis 3D dari microprocessor. Alat ini digunakan di sistem benam, telepon genggam, komputer pribadi, workstation, dan konsol game. GPU Modern sangat efisien dalam memanipulasi komputer grafis dan struktur paralel, membuatnya lebih efektif dari fungsi umum CPU yang digunakan untuk bebagai perhitungan alogaritma. Pada komputer pribadi (PC), GPU biasanya terdapat di video card atau di motherboard. Lebih dari komputer desktop dan notebook mempunyai GPU yang terintegrasi, yang biasanya jauh daripada yang ada di video card.



Gambar 2.2 Chip GPU

* 1. Mean Squared Error

kljkl

BAB III  
METODE PENELITIAN

1. BAB IV HASIL DAN PEMBAHASAN
   1. Lokasi dan Waktu

Penelitian ini dilakukan pada bulan Februari sampai bulan April 2018 bertempat di rumah penulis dan Student Activity Center Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya Malang.

* 1. Alat dan Bahan

Alat yang digunakan dalam penelitian kali ini adalah seperangkat komputer dengan spesifikasi Processor Intel i3 6100 3.7 GHz, Motherboard Colorful Battle Axe C.B150M-D V23, Memory Corsair Value Select 1x8 GB, VGA Zotac GTX 1050 2 GB, Hard Drive 1 TB, Power Supply 450 Watt, serta Monitor AOC 19 inch dengan Resolusi 1366x768 pixel.

* 1. Tahapan Penelitian

Penelitian ini dilakukan dengan beberapa tahap sebagai berikut:

1. Install Software Dependency
2. Generate Data
3. Training
4. Testing
   * 1. Install Software Dependency

Pada Tahap ini, akan dilakukan instalasi software-software pendukung yang akan digunakan untuk membuat script training ataupun testing. Software akan ditulis dalam bahasa python, sehingga kita perlu menginstall python interpreter, versi python yang akan digukanan adalah python versi 3.5.2.

Untuk memudahkan proses instalasi software pendukung / library python, digunakan aplikasi conda. Pustaka yang diperlukan antara lain, numpy, matplotlib, jupyter, opencv3, pillow, scikit-learn, scikit-image, scipy, h5py, eventlet, flask-socketio, seaboarn, pandas, imageio, moviepy, tensoflow, dan keras. Library yang telah disebutkan disimpan kedalam satu file yang disebut environment-gpu.yml, lalu diinstal dengan perintah

conda env create -f environment-gpu.yml

* + 1. Generate Data

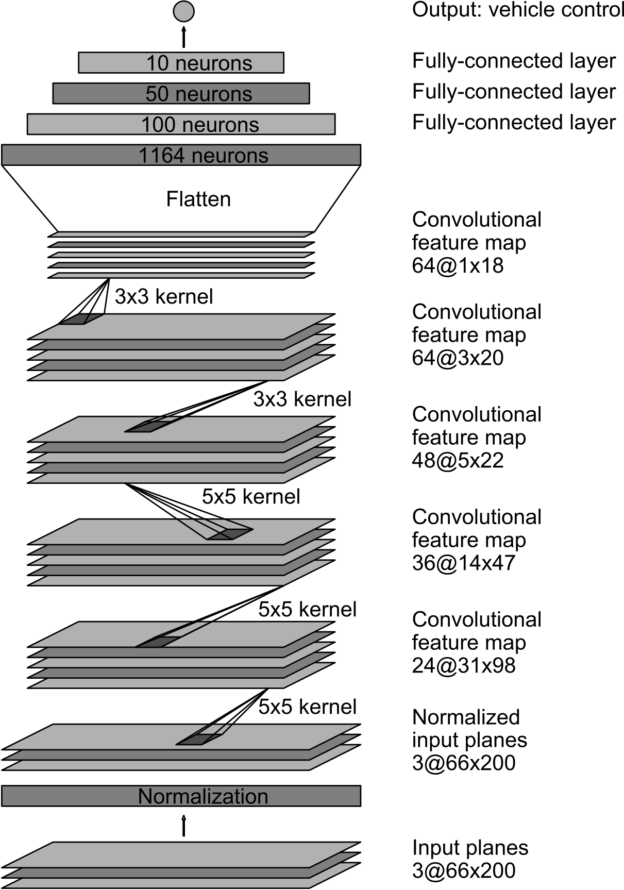
Pada tahap ini diperlukan sejumlah data yang fungsinya nanti sebagai input pada di Deep Neural Network. Input yang dibutuhkan yaitu gambar tampak depan dari mobil serta aksi apa yang diberikan kepada mobil. Dalam penilitan ini gambar diambil dari 3 sudut, yaitu dari kiri, kanan serta tengah. Aksi yang direkam antara lain, steering-wheele, throttle, reverse, dan speed.

Metode pengambilan data ini dilakukan dengan cara membuka program simulator yang disediakan oleh Udacity Self Driving Simulator, saat telah masuk dalam progam, klik Training Mode, setelah itu akan muncul tampilan mobil, lalu tekan tombol [R] untuk melakukan perekaman. Setelah itu kita diminta untuk mengendarai mobil, diusahakan minimal mengendarai 1 putaran agar mobil dapat mengenali berbagai kondisi pengendaraan. Setelah dikendarai sejauh 1 putaran, ditekan lagi tombol [R] untuk berhenti melakukan perekaman.

* + 1. Training

Pada tahap ini dilakukan penulisan program training untuk dapat membangun model dari Deep Learning. Pada tahap training ini, program harus dapat membaca file gambar dan aksi yang telah kita simpan, lalu dapat menyimpan hasil training dalam suatu file. Dalam penelitian ini digunakan file h5 sebagai penyimpanan hasil training dari Deep Learning, hal ini dikarenakan h5 merupakan file binary yang terkompresi, sehingga tidak membutuhkan space yang cukup banyak.

Konfigurasi jaringan yang digunakan dalam penelitian ini adalah Convolutional Neural Network dengan 10 hidden layer. Konfigurasi ini diinspirasi dari konfigurasi yang telah dipaparkan oleh NVIDIA, dan susunannya adalah sebagai berikut:



Pada layer pertama terdapat layer input yang berupa citra gambar dengan panjang 200 pixel, lebar 66 pixel dan 3 channel (red, green, blue). Setelah itu layer input dinormalisasi terlebih dahulu dengan persamaan x = x / 127.5 - 1.0. Setelah dilakukan normalisasi, masuk ke layer convolutional ke-1 dengan kernel sebesar 5x5, stride 2x2, dan memiliki 24 output convolutional, sehingga didapatkan output dari layer ini sebesar 24x98x31. Setelah itu masuk ke layer convolutional ke-2 dengan kernel sebesar 5x5, stride 2x2 dan memiliki output 36, sehingga didapatkan output dari layer ini sebesar 36x47x41. Setelah itu masuk ke layer convolutional ke-3 dengan kernel sebesar 5x5, stride 2x2 dan memiliki output 48 node, sehingga didapatkan output dari layer ini sebesar 48x22x5. Setelah itu masuk ke layer convolutional ke-4 dengan kernel sebesar 3x3, stride 1x1 dan memiliki output 64 node, sehingga didapatkan output dari layer ini sebesar 64x20x3. Setelah itu masuk ke layer convolutional ke-5 dengan kernel sebesar 2x2, stride 1x1 dan memiliki output 64 node, sehingga didapatkan output dari layer ini sebesar 64x18x1. Setelah itu diberikan fungsi flatten yang merubah bentuk data yang semula 64x18x1 menjadi satu baris dengan 1152 data. Setelah itu masuk ke layer selanjutnya yaitu fully connected layer dengan output sebesar 100 node. Setelah itu masuk ke layer selanjutnya yaitu fully connected layer ke-2, dengan output sebesar 50 node. Setelah itu masuk ke layer selanjutnya yaitu fully connected layer ke-3, dengan output sebesar 10 node. Setelah itu masuk ke layer output 1 node yang berupa hasil steering wheel.

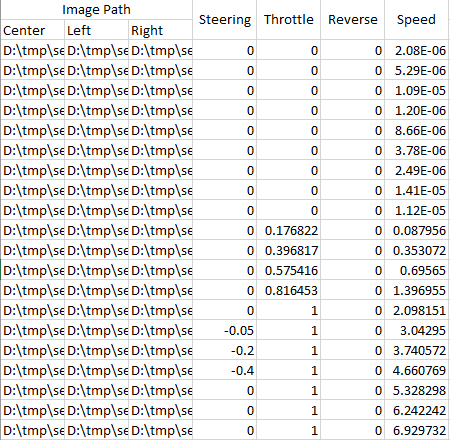
* + 1. Testing

Testing dilakukan dengan cara membuka program simulator yang disediakan oleh udacity, lalu masuk ke mode Autonomous Drive, lalu dijalankan program testing yang telah ditulis dalam python dan mengambil hasil training dengan loss terkecil agar didapatkan performa maksimal. Pembelajaran dikatakan berhasil jika mobil dapat mengendarai dirinya sendiri dan dapat melewati 1 lap penuh tanpa menabrak halangan sama sekali.

BAB IV  
HASIL DAN PEMBAHASAN

1. BAB IV HASIL DAN PEMBAHASAN
   1. Hasil Generate Data

Data yang dihasilkan pada tahap ini berupa 1 file csv serta kumpulan gambar yang terdapat dalam folder images. Berikut cuplikan file csv dan cuplikan hasil gambar yang dihasilkan.



Gambar 4.1 File CSV hasil training.

Dari gambar diatas proses training menghasilkan 7 macam data. Tiga data pertama merupakan data alamat gambar disimpan, gambar yang dihasilkan juga terdapat 3 macam gambar, yaitu gambar dari sisi kiri mobil, gambar dari sisi tengah mobil dan gambar dari sisi kanan mobil, untuk lebih jelasnya lihat gambar 4.2.

(a) (b) (c)

Gambar 4.2 Gambar yang diambil dari tampak

(a) kiri (b) tengah (c) kanan

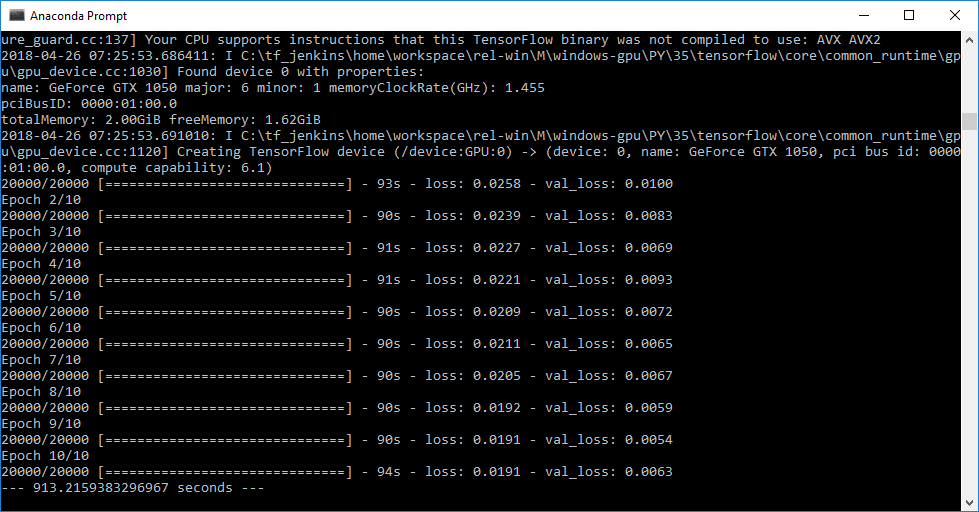
Tiga data berikutnya adalah data aksi yang kita berikan ke simulator yaitu steering yang mempunyai 3 macam nilai yaitu 0, bilangan positif atau bilangan negatif. Angka nol mengartikan bahwa steering tidak digerakkan ke arah kanan ataupun kiri, sedangkan bilangan positif menunjukkan steering digerakkan ke arah kanan dan bilangan negatif menunjukkan steering digerakkan ke arah kiri. Steering dalam simulator diwakili dengan pemberian input keyboard [A] untuk kiri dan [D] untuk kanan, atau input drag mouse. Aksi yang berikutnya yaitu throttle yang mewakili dari pedal gas yang ditekan, dalam simulator diwakili dengan penekanan tombol keyboard [W]. Aksi berikutnya yang direkam adalah reverse, mewakili apakah mobil dalam keadaan mundur atau tidak, nilai ini bernilai 0 untuk False dan 1 untuk True, dalam simulator aksi ini berkorelasi dengan penekanan tombol keyboard [S]. Kolom terakhir yang direkam yaitu kecepatan mobil dalam simulator, data ini bukanlah merupakan input melainkan output yang mana dari aksi-aksi yang kita berikan menghasilkan kecepatan seperti yang tertera pada kolom ini.

* 1. Hasil Proses Training

Proses Training dilakukan dengan dua cara yaitu dengan menggunakan CPU dan GPU. Apakah training dengan dengan menggunakan GPU meningkatkan proses training secara signifikan? Apakah training menggunakan GPU mendapatkan hasil yang pasti lebih baik dibandingkan dengan CPU?

* + 1. Training dengan GPU

Training yang pertamakali dilakukan oleh penulis adalah training dengan menggunakan GPU. Dan hasilnya adalah sebagai berikut:

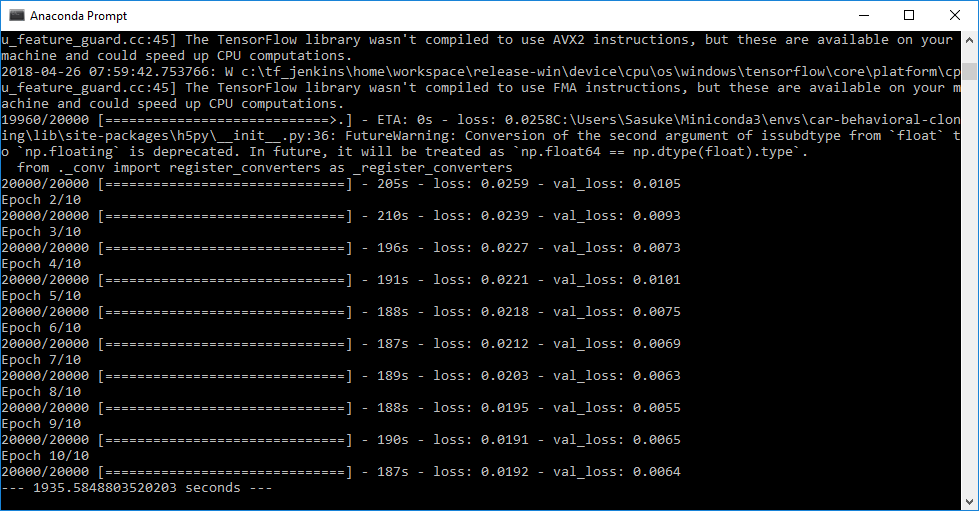


Gambar 4.3 Proses Training dengan GPU

Dari gambar diatas, dapat kita ketahui bahwa proses training dengan menggunakan GPU memakan waktu sebanyak 913 detik atau 15 menit 13 detik. Training menggunakan GPU ini juga dibatasi hingga 10 epoch atau 10 kali proses training, dari kesepuluh data tersebut, didapatkan data terbaik pada epoch ke-9 dimana pada epoch tersebut didapatkan nilai loss function 0.191 pada data training, dan 0.0054 pada data test.

* + 1. Training dengan CPU

Training berikutnya adalah training dengan menggunakan CPU, dan hasilnya dapat dilihat pada gambar berikut.



Gambar 4.4 Proses Training dengan CPU

Dari gambar diatas, dapat kita ketahui bahwa proses training dengan menggunakan CPU memakan waktu sebanyak 1935 detik atau 32 menit 15 detik. Training menggunakan CPU ini juga dibatasi hingga 10 epoch atau 10 kali proses training, dari kesepuluh data tersebut, didapatkan data terbaik pada epoch ke-8 dimana pada epoch tersebut didapatkan nilai loss function 0.195 pada data training, dan 0.0055 pada data test.

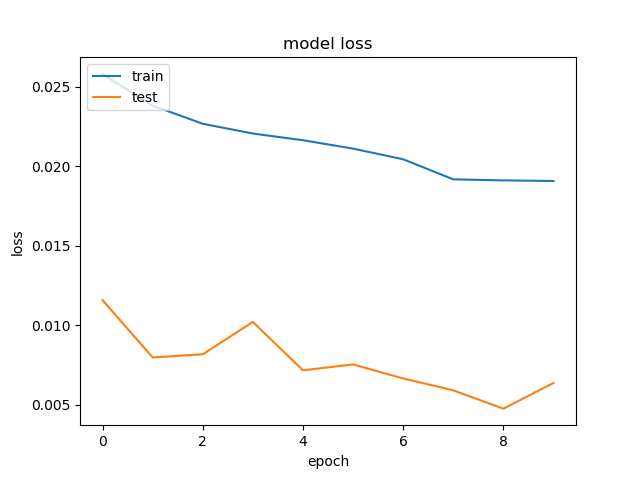
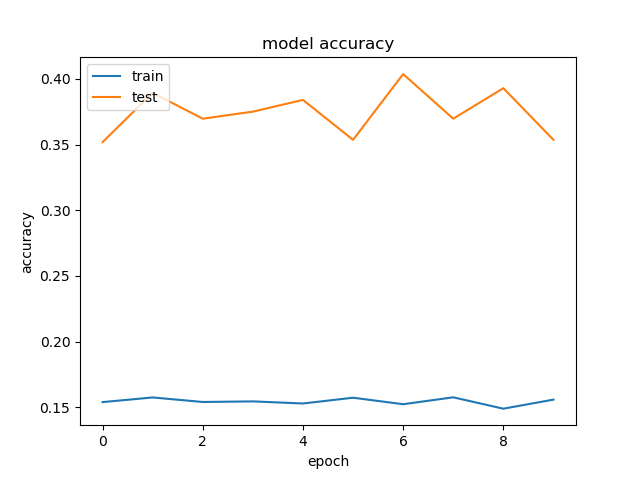
* + 1. Perbandingan Training CPU dan GPU

Secara umum training dengan CPU maupun GPU menghasilkan data yang tidak jauh berbeda, dapat dilihat pada loss function yang dihasilkan berikas pada kisaran angka yang sama, yaitu 0.0054 pada training dengan GPU sedangkan 0.0055 pada training dengan menggunakan CPU. Namun dari kedua proses training tersebut, waktu yang dibutuhkan berbeda. Pada proses training dengan CPU memakan waktu sebanyak 32 menit 15 detik, sedangkan pada GPU memakan waktu sebanyak 15 menit 13 detik. Dapat dilihat bahwa dengan menggunakan GPU terdapat peningkatan 2 kali lebih cepat dibanding dengan menggunakan CPU. Artinya proses training dengan menggunakan GPU lebih efisien dibanding dengan CPU tanpa mengorbankan hasil training yang dihasilkan.

* 1. Hasil Testing

Data testing yang diperhatikan pada penelitian ini dibagi menjadi 2 macam, yaitu testing berdasarkan data testing, serta testing dengan memperhatikan pada simulator apakah model yang telah dibangun memiliki performa yang baik atau tidak.

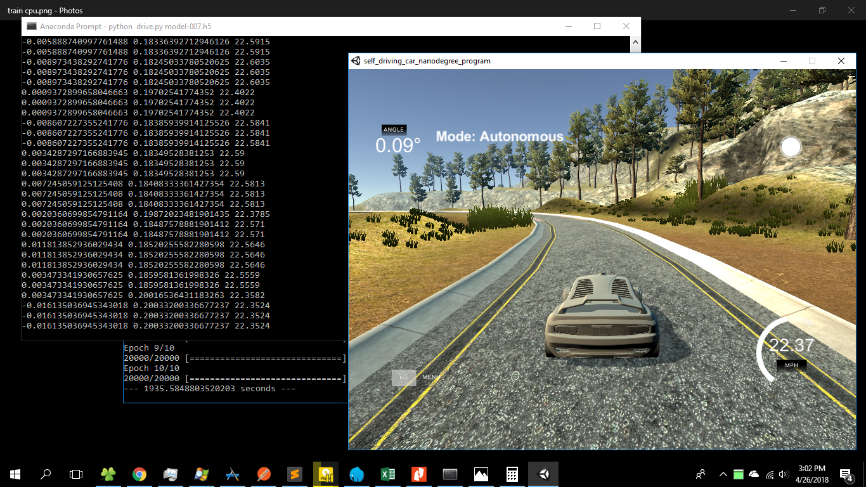
* + 1. Data Testing



Gambar 4.5 Akurasi dan Loss yang dihasilkan oleh model

* + 1. Visual Testing

Pada testing ini, kita memperhatikan pada simulator apakah mobil yang dikendarai memiliki performa yang baik atau tidak. Definisi memiliki performa yang baik adalah mobil yang dapat mengendarai tanpa menabrak ataupun keluar dari jalur. Skema testing yang dilakukan yaitu memberikan pinalti setiap mobil menabrak pembatas jalan atau keluar dari jalur.



Gambar 4.6 Visual Testing