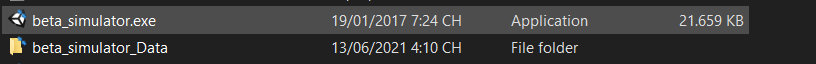
# SELF-DRIVING CAR

1. **Mở đầu**

Tôi đã làm xe tự lái như thế nào? Quá trình làm project này không hề đơn giản. Chúng ta phải đi từ những thứ cơ bản, sau đó mới nâng dần lên kết hợp những thứ cơ bản đó thành 1 dự án hoàn thiện. Những thứ cơ bản mà tôi muốn đề cập: kiến thức về Python cơ bản và kĩ năng đọc hiểu code; những kiến thức xử lí ảnh; setup môi trường làm việc đúng để các phần mềm có thể kết nối với nhau; và cuối cùng là DEBUG!!!

1. **Cấu trúc của bài viết**
2. **Môi trường vận hành**
3. **Phần mềm giả lập**

Trong project, môi trường mô phỏng được dùng là beta\_simulator, bao gồm 1 file exe và 1 folder chứa các tệp tin cần thiết cho vận hành.





Có 2 chế độ:

+) Chế độ Training mode – cho dữ liệu về xe tự lái để train mô hình; Ở mỗi vị trí ô tô di chuyển, ta sẽ thu về vị trí của 3 góc camera (trái, giữa, phải), góc lái vô lăng, tốc độ xe, độ giảm tốc (throttle) và phanh (brake).

+) Chế độ Autonomous mode – sau khi đã hoàn thiện model thì đây sẽ là chế độ xe tự lái.

1. **Các thư viện cần thiết cho Python**

Đi vào nơi vận hành gốc của Python trong ổ C để tìm. Sau đó copy và paste vào nơi thực hiện project (cụ thể là ổ D) để tránh các lỗi sau này.



1. **Quá trình tiến hành project**
2. **Xác định bài toán**
3. *Ý tưởng:*

Xe tự lái sẽ thu về dữ liệu ***input***: từ cảm biến trước sau, từ cảm biến hai bên, từ camera quay 4 phía quanh xe để có thông tin về tốc độ, góc lái... ***Output*** trả về là các hành động của xe: tăng ga, đạp phanh, điều chỉnh vô lăng...

* Như vậy, ý tưởng sẽ là train cho xe các model để nó tự học và hiểu. Sau đó, khi để xe tự chạy thì nó sẽ dựa vào model, xác định các góc lái đã học để tự điều chỉnh output. Nôm na ra khi gặp phân cảnh A (đại khái là khúc cua nào đó), camera từ xe thu được input B. Vì xe đã có kiến thức trong bộ não của nó (thứ ta sẽ xây dựng sau này trong phần software) nên gặp input B nó sẽ cho ra output C tương ứng, ví dụ như đạp 1/3 phanh, bẻ vô lăng 30 độ sang trái, giảm ga về tốc độ tối đa 25km/h.

1. *Thuật toán:*

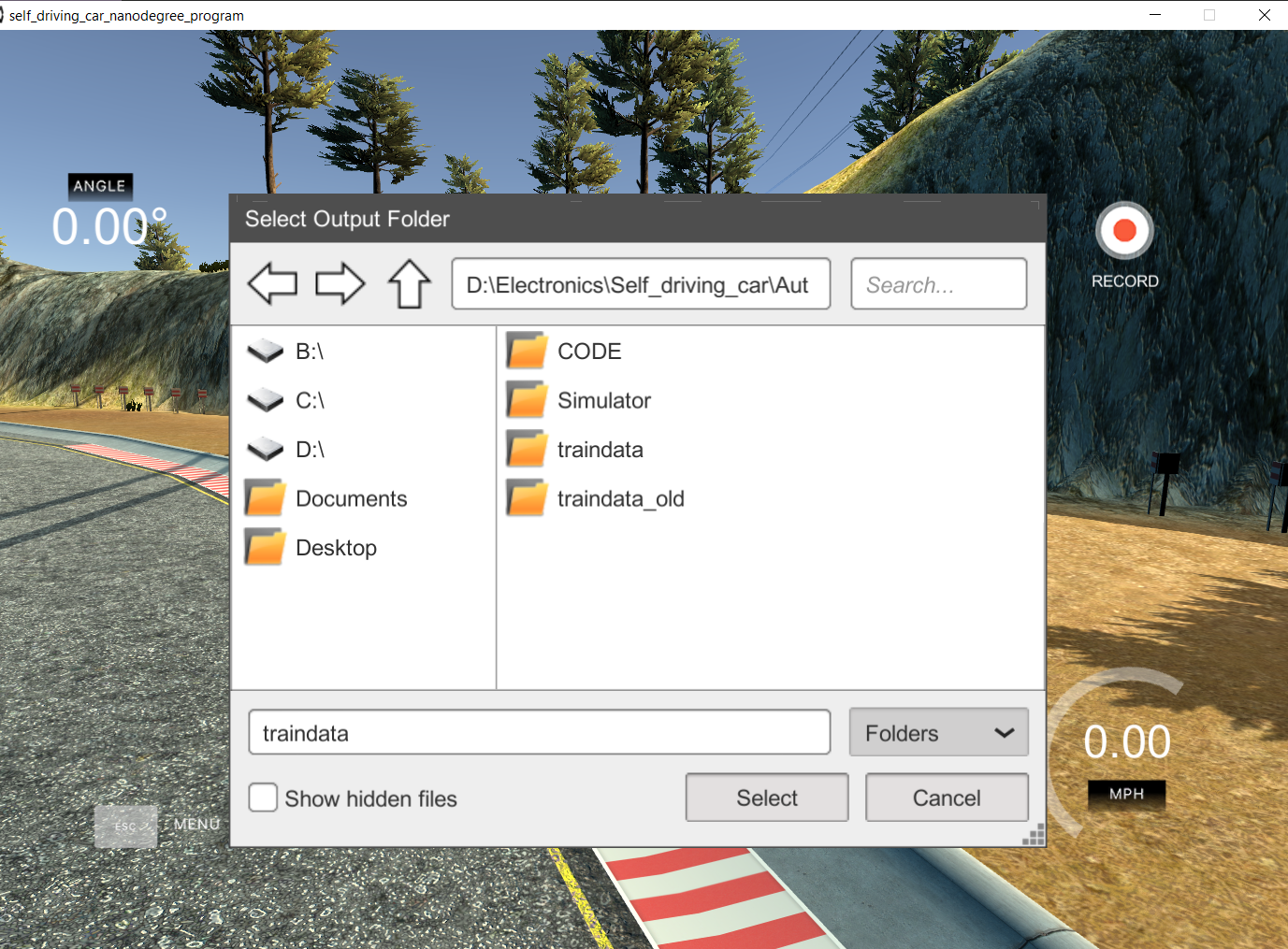
Vì ta cần dự đoán góc lái bằng ảnh của camera, nghĩa là input là ảnh nên nghĩ ngay đến việc dùng CNN (Convolutional Neutral Network). CNN là một trong những mô hình deep learning tiên tiến, là một dạng mô hình notron. Nó được dùng chủ yếu trong các bài toán xử lí ảnh và được xây dựng thành một số thư viện. Trong project này, thư viện được sử dụng là keras. Vì dữ liệu ảnh thu về đều ở dạng thô, nên ta cần xử lí ảnh để tìm ra những nét đặc trưng nhất của ảnh đó. Từ đó, xây dựng model và dạy cho máy tự học và áp dụng vào chế độ tự lái. Lưu ý project chỉ mới phát triển ở phần mô phỏng nên những vấn đề về cơ khí (cấu tạo xe), điện tử (vi mạch, kết nối linh kiện điện tử xe) và nhiễu chưa được bàn đến.

Tham khảo về CNN của tác giả Tuan Nguyen tại link: <https://nttuan8.com/bai-6-convolutional-neural-network/> (30-03-2019)

1. **Thu thập dữ liệu để train model**



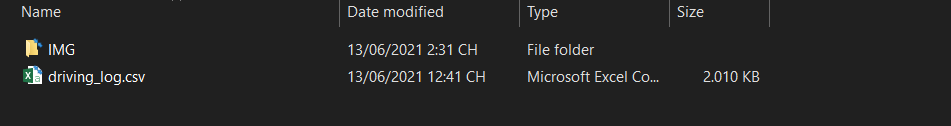
* Vào Training mode để thực hiện việc thu thập dữ liệu. Ta nên tập lái thử bằng các nút mũi tên trước để cho xe chạy mượt. Sau đó mới nhấn Record để thu thập dữ liệu. Lưu ý rằng dữ liệu càng nhiều, cộng với xe chạy càng mượt thì sau này xe sẽ tự lái tốt hơn. Ta nên record trong tầm 5-10 phút để dữ liệu thu về đa dạng.
* Khi nhấn record, phần mềm sẽ chọn nơi lưu data thu về. Ở đây mình chọn lưu trong thư mục traindata.

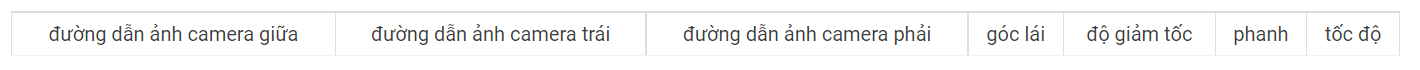


* Trong traindata bao gồm

+) Folder IMG chứa ảnh từ camera của 3 góc lái trái, giữa, phải

+) File driving\_log.csv lưu thông tin cần thiết của data: đường dẫn đến 3 file ảnh camera; góc lái, độ giảm tốc, phanh và tốc độ hiện tại của xe.





1. **Xây dựng model**

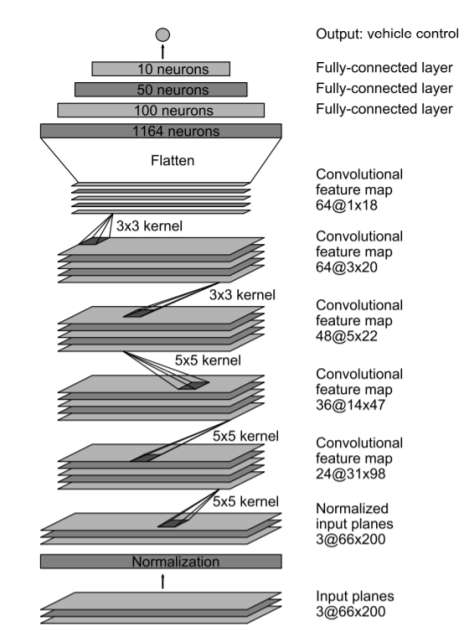
Trước tiên, ảnh đầu vào input, sau khi trải qua tiền xử lý sẽ có kích thước là 66\*200. Vì ảnh màu được cấu tạo từ 3 thành phần màu cơ bản là đỏ, xanh lam, xanh dương nên ta tách ảnh màu thành 3 ảnh thành phần và biểu diễn ở dạng ma trận như sau:



Nguồn: <https://www.slideshare.net/BertonEarnshaw/a-brief-survey-of-tensors>

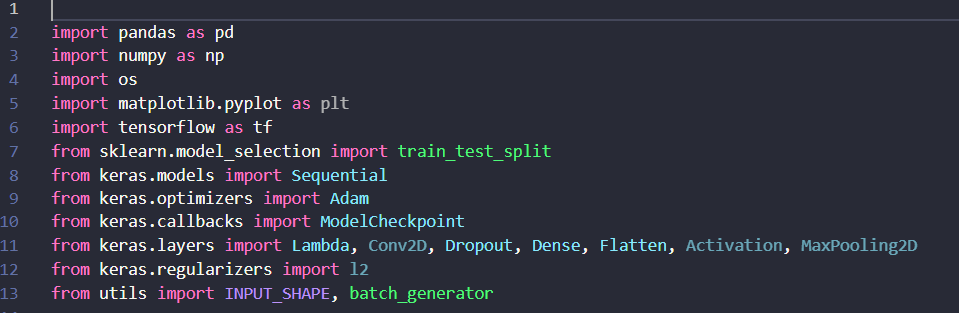
Để hiểu thêm về xử lý ảnh và những phần code được đề cập ở sau, tham khảo bài viết: <https://nttuan8.com/bai-5-gioi-thieu-ve-xu-ly-anh/>

Sau đó, áp dụng thuật toán CNN, đại khái là dùng tích chập để tìm ra những đặc trưng của ảnh, ta sẽ tìm ra được node cuối cùng mang nội dung ‘Vehicle control’Sa, giúp cho xe dựa vào đó tự lái sau này.



Nguồn: <https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf>

1. **Train model** 
   1. ***Các thư viện cần thiết***

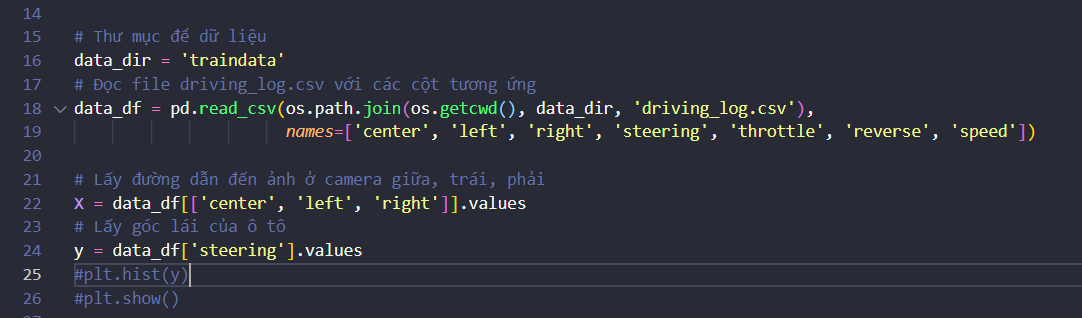


* Matplotlib.pyplot: dùng để vẽ đồ thị, tham khảo thêm tại: <https://viblo.asia/p/gioi-thieu-ve-matplotlib-mot-thu-vien-rat-huu-ich-cua-python-dung-de-ve-do-thi-yMnKMN6gZ7P>
* Tensorflow: thư viện mã nguồn mở dành cho máy học trong nhiều loại hình tác vụ nhận thức và hiểu ngôn ngữ. Tham khảo thêm: <https://topdev.vn/blog/tensorflow-la-gi/>
* Keras: là thư viện mã nguồn mở (open source), dùng để xây dựng model với ưu điểm nhanh, dễ sử dụng, có thể run trên cả CPU lẫn GPU. Tham khảo thêm:

+)<https://thorpham.github.io/blog/2018/05/25/keras/?fbclid=IwAR3cAG2mRdGiVYRnq-GLBUNdwxfTRufAJ0R0zspbo6_KYjfXUHHT58LjAPE>

+)<https://viblo.asia/p/lam-quen-voi-keras-gGJ59mxJ5X2?fbclid=IwAR0sMd3W39AqQYbPRaQLrxRQBs19kaFWlkvh0wdXLZ2S28z-JxDO04nDeFQ>

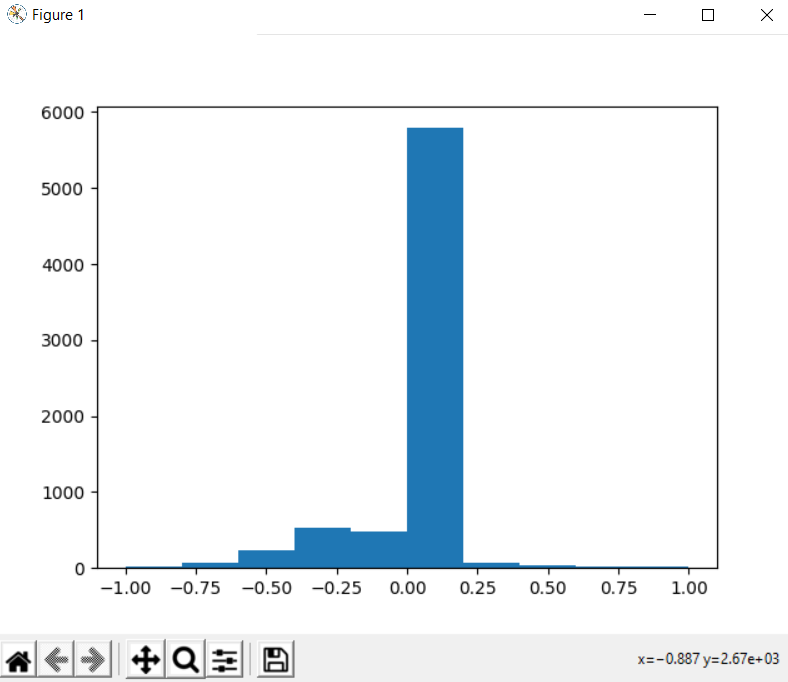
* 1. ***Tiền xử lý dữ liệu***



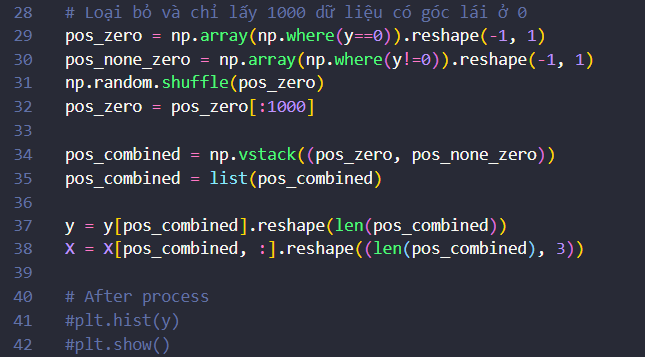
* Thư mục traindata chứa dữ liệu xe gửi về và train.py nên nằm cùng folder để python có thể dễ dàng tìm được.
* data\_df trả về sẽ ở dạng ma trận.
* Hai biến x và y:

+) x: là ảnh góc lái của camera lần lượt theo thứ tự giữa, trái, phải

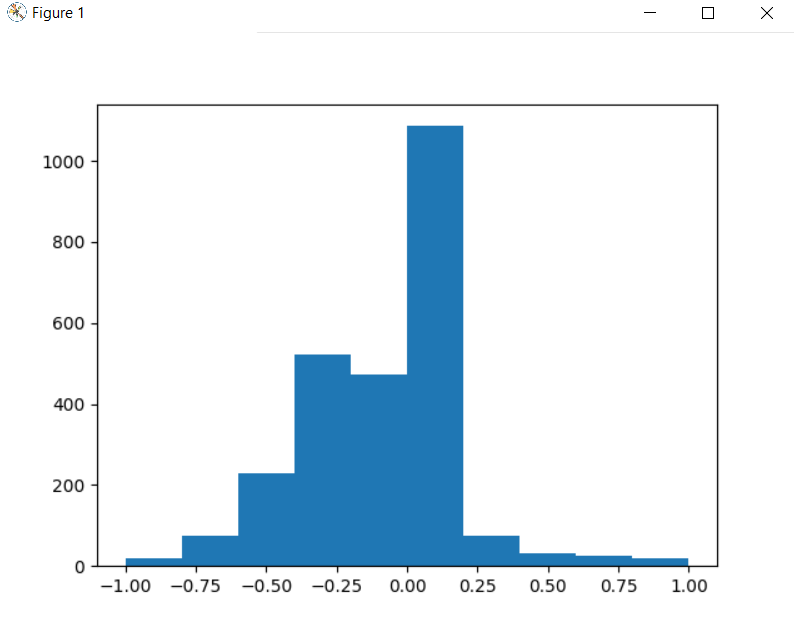
+) y: góc lái của xe (thể hiện ở ảnh bên dưới)



* Ta nhận thấy, dữ liệu ở góc lái số 0 (trường hợp xe đi thẳng) là quá nhiều nên ta cần loại bỏ bớt để tránh việc model hiểu sai và có xu hướng dự đoán góc lái thiên về 0

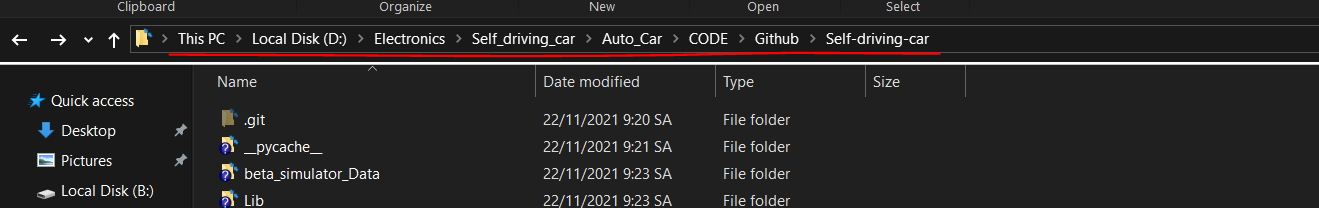


* Kết quả: Chỉ còn hơn 1000 kết quả ở xung quanh góc lái số 0. Góc lái này vẫn nhiều nhất vì đa phần thời gian trong map xe đi thẳng. Đến đây ta đã có đủ dữ liệu cần thiết của xe (camera, góc lái) để chuẩn bị bước vào train model.

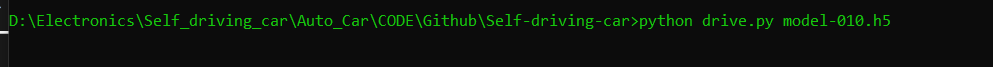


How to run the file:

1. Type the cmd here



1. Turn the Unity on in auto mode
2. Command



1. Enjoy