

PHÂN LOẠI XE SỐ HOẶC XE TAY GA BẰNG HÌNH ẢNH

NGUYỄN PHÚ QUỐC – 18520343 -
CS114.K21.KHTN

Link Github:

<https://github.com/KicksxNPQ/CS114.K21.KHTN>

Tóm tắt

- Tên đề tài
- Tóm tắt về đề án và kết quả đạt được
- Ảnh của các thành viên của nhóm

Bài toán

Nhận diện xe máy hiện nay là bài toán đang rất hot trên thế giới. Hiện nay các ứng dụng máy học, trí tuệ nhân tạo dần đang được áp dụng trong đời sống của chúng ta. Và việc xử lý vi phạm giao thông hiện nay đang rất cần thiết khi các loại xe phân khối lớn càng ngày càng phát triển và vi phạm giao thông của mọi người hiện nay còn rất nhiều. Và bên cạnh đó, bài toán phân loại xe, nhận diện xe sẽ là các bài toán chính trong đời sống hiện nay.

Cách giải

Sử dụng Machine Learning để phân loại xe đó có phải loại xe nào, có thể giúp cho cảnh sát giao thông có thể quản lý được là nhận diện được thông qua camera với số lượng lớn phương tiện giao thông di chuyển trong màn hình, và có thể phân loại được dòng xe nào là dòng xe có thể chạy được với tốc độ cao.

Cách giải

Input: Một hình ảnh bao gồm 1 chiếc xe (không có object khác) vì hạn chế của dữ liệu.

Output: dòng xe đó là dòng xe nào (tay ga hay xe số).

Mô tả dữ liệu

Sử dụng dữ liệu được lấy từ internet.

Kích thước dữ liệu:

- Xe số 107 ảnh
- Xe tay ga 107 ảnh

Mô tả dữ liệu

Chú thích:

Hình ảnh của dữ liệu chỉ bao gồm 1 chiếc xe và nền trơn.

Không bao gồm các vật thể khác trong hình ảnh vì không trải qua quá trình object detection.

Kết quả đạt được

Chạy với 20 lần
epochs để thực
hiện Gradient
Descent

```
Epoch 1/20
25/25 [=====] - 46s 2s/step - loss: 0.6898 - accuracy: 0.5904 - val_loss: 0.6878 - val_accuracy: 0.5714
Epoch 2/20
25/25 [=====] - 42s 2s/step - loss: 0.6762 - accuracy: 0.6000 - val_loss: 0.6890 - val_accuracy: 0.5714
Epoch 3/20
25/25 [=====] - 42s 2s/step - loss: 0.6728 - accuracy: 0.6000 - val_loss: 0.6861 - val_accuracy: 0.5714
Epoch 4/20
25/25 [=====] - 42s 2s/step - loss: 0.6725 - accuracy: 0.6000 - val_loss: 0.6867 - val_accuracy: 0.5714
Epoch 5/20
25/25 [=====] - 42s 2s/step - loss: 0.6723 - accuracy: 0.6000 - val_loss: 0.6864 - val_accuracy: 0.5714
Epoch 6/20
25/25 [=====] - 42s 2s/step - loss: 0.6722 - accuracy: 0.6000 - val_loss: 0.6866 - val_accuracy: 0.5714
Epoch 7/20
25/25 [=====] - 43s 2s/step - loss: 0.6721 - accuracy: 0.6000 - val_loss: 0.6855 - val_accuracy: 0.5714
Epoch 8/20
25/25 [=====] - 42s 2s/step - loss: 0.6719 - accuracy: 0.6000 - val_loss: 0.6864 - val_accuracy: 0.5714
Epoch 9/20
25/25 [=====] - 42s 2s/step - loss: 0.6719 - accuracy: 0.6000 - val_loss: 0.6860 - val_accuracy: 0.5714
Epoch 10/20
25/25 [=====] - 42s 2s/step - loss: 0.6717 - accuracy: 0.6000 - val_loss: 0.6867 - val_accuracy: 0.5714
Epoch 11/20
25/25 [=====] - 42s 2s/step - loss: 0.6715 - accuracy: 0.6000 - val_loss: 0.6869 - val_accuracy: 0.5714
Epoch 12/20
25/25 [=====] - 42s 2s/step - loss: 0.6714 - accuracy: 0.6000 - val_loss: 0.6875 - val_accuracy: 0.5714
Epoch 13/20
25/25 [=====] - 42s 2s/step - loss: 0.6713 - accuracy: 0.6000 - val_loss: 0.6866 - val_accuracy: 0.5714
Epoch 14/20
25/25 [=====] - 42s 2s/step - loss: 0.6712 - accuracy: 0.6000 - val_loss: 0.6853 - val_accuracy: 0.5714
Epoch 15/20
25/25 [=====] - 45s 2s/step - loss: 0.6710 - accuracy: 0.6000 - val_loss: 0.6850 - val_accuracy: 0.5714
Epoch 16/20
25/25 [=====] - 42s 2s/step - loss: 0.6707 - accuracy: 0.6000 - val_loss: 0.6843 - val_accuracy: 0.5714
Epoch 17/20
25/25 [=====] - 42s 2s/step - loss: 0.6706 - accuracy: 0.6000 - val_loss: 0.6877 - val_accuracy: 0.5714
```


Khó

Kết quả thấp, độ chính xác chỉ ở mức độ khoảng 60% - chưa thực sự có tính ứng dụng Mô hình máy học ổn định

- Nguyên nhân có thể là do nhiều yếu tố sau :
- Mô hình chưa đủ tốt
- Bộ dữ liệu chưa đủ
- Tiền xử lý dữ liệu chưa tốt
- Kỹ thuật rút trích đặc trưng chưa hợp lý

Kết luận

Sau khi train model với số lượng epochs là 20, tuy nhiên thì có thể do data nên accuracy train vẫn không được tốt hơn những lần đầu tiên train nên gradient descent trong Model VGG không được phát huy tác dụng.

Bộ dữ liệu cần được cập nhật thêm để có thể phát huy hết được model.\

Hướng phát triển

Sử dụng deep model để phát triển thêm:

- Tiền xử lý dữ liệu tốt hơn
- Xây dựng bộ dữ liệu tốt hơn
- Sử dụng nhiều model để có thể đạt được kết quả tốt nhất.

Thanks!

Contact us:

Big Data &
Deep Learning
Lab
Ho Chi Minh City,
Vietnam

b2dl.uit@gmail.com

