Các lỗi thường gặp

1. Lỗi về .proto

Lý do: là protobuf bản khác 3.4 lỗi cmn hết

Cách sửa: tải bản 3.4 về, rồi chạy lệnh

Cách 1:

|  |
| --- |
| (py36) C:\tensorflow1\models\research> C:\tensorflow1\protoc-3.4.0-win32\bin\ protoc object\_detection/protos/\*.proto --python\_out=. |

Cách 2: (viết cùng 1 hàng)

|  |
| --- |
| (envir) "C:\ Desktop\Tensorflow\protbuf\bin\protoc.exe" object\_detection/protos/\*.proto --python\_out=. |

1. Lỗi về numpy và pandas

Lý do: trong thư viện pandas có numpy nhưng khi cập nhật nó méo tự up numpy lên bản mới -\_- và thế là gây ra lỗi

Cách sửa: tháo ra hết và cài lại

|  |
| --- |
| (py36) C:\> pip uninstall pandas  (py36) C:\> pip uninstall numpy  (py36) C:\> pip install pandas  (py36) C:\> pip uninstall numpy  (py36) C:\> pip install numpy |

1. ModuleNotFoundError: No module named 'deployment'

Lý do: đường dẫn PYTHONPATH gặp vấn đề, kiểm tra

Nhập *echo %PYTHONPATH%* ở bất kỳ thư mục nào để kiểm tra

|  |
| --- |
| (py36) C:\> echo %PYTHONPATH% |

Cách sửa: làm lại các bước cài PYTHONPATH và chạy setup.py ở thư mục models\research (note: bất kỳ chỉnh sửa gì ở các thư mục thực thi đều phải chạy lại *setup.py build* và *install*)

|  |
| --- |
| setup.py build  setup.py install |

Bước 1: Tải các thư viện và tool cần thiết

* tải anaconda 3
* tải models-master

|  |
| --- |
| https://github.com/tensorflow/models |

* Tải trained model

|  |
| --- |
| https://github.com/tensorflow/models/blob/master/research/object\_detection/g3doc/detection\_model\_zoo.md |

* Tải protobuf (Lưu ý tải bản 3.4, mấy bản sau lỗi cmn hết r -\_-)

|  |
| --- |
| https://github.com/protocolbuffers/protobuf/releases |

Ví dụ: faster\_rcnn\_inception\_v2\_coco

Tải về thư mục object\_detection trong models-master và giải nén

* Tải chương trình mẫu về

|  |
| --- |
| https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10 |

* Lưu ý: để phù hợp với các đường dẫn phía dưới thì tạo thư mục C:\tensorflow1 quăng hết đống file trên vào rồi giải nén, đổi tên thư mục models-master thành models, quăng hết tất cả các file trong chương trình mẫu vào models\research\object\_detection
* Có thế sửa đường dẫn theo ý muốn.

Bước 2: Tải thư viện

* Tạo môi trường python 3.6

|  |
| --- |
| C:\> conda create -n tensorflow1 pip python=3.6  C:\> activate tensorflow1 |

* Cài các thư viện

Install tensorflow:

|  |
| --- |
| (tensorflow1) C:\> pip install --ignore-installed --upgrade tensorflow |

Install packages:

|  |
| --- |
| (tensorflow1) C:\> conda install -c anaconda protobuf  (tensorflow1) C:\> pip install pillow  (tensorflow1) C:\> pip install lxml  (tensorflow1) C:\> pip install Cython  (tensorflow1) C:\> pip install jupyter  (tensorflow1) C:\> pip install matplotlib  (tensorflow1) C:\> pip install pandas  (tensorflow1) C:\> pip install opencv-python |

* Set PYTHOPATH.

|  |
| --- |
| (tensorflow1) C:\> set PYTHONPATH=C:\tensorflow1\models;C:\tensorflow1\models\research;C:\tensorflow1\models\research\slim |

|  |
| --- |
| (tensorflow1) C:\> set PATH=%PATH%  (tensorflow1) C:\> echo %PYTHONPATH% |

**Lưu ý**: cài đặt này sẽ mất khi tắt máy, để khắc phục

This PC -> Properties -> Advanced system settings -> Enviroment Variable -> tại khung System variables -> PYTHONPATH (nếu không có thì tạo luôn) -> add vào các đường dẫn đến thư mục



* Compile Protobuf

Bước này để compile file .proto sang .py trong thư mục C:\tensorflow1\models\research\object\_detection\protos

**Notes:** xem *Lỗi về .protos*

|  |
| --- |
| protoc --python\_out=. .\object\_detection\protos\anchor\_generator.proto .\object\_detection\protos\argmax\_matcher.proto .\object\_detection\protos\bipartite\_matcher.proto .\object\_detection\protos\box\_coder.proto .\object\_detection\protos\box\_predictor.proto .\object\_detection\protos\eval.proto .\object\_detection\protos\faster\_rcnn.proto .\object\_detection\protos\faster\_rcnn\_box\_coder.proto .\object\_detection\protos\grid\_anchor\_generator.proto .\object\_detection\protos\hyperparams.proto .\object\_detection\protos\image\_resizer.proto .\object\_detection\protos\input\_reader.proto .\object\_detection\protos\losses.proto .\object\_detection\protos\matcher.proto .\object\_detection\protos\mean\_stddev\_box\_coder.proto .\object\_detection\protos\model.proto .\object\_detection\protos\optimizer.proto .\object\_detection\protos\pipeline.proto .\object\_detection\protos\post\_processing.proto .\object\_detection\protos\preprocessor.proto .\object\_detection\protos\region\_similarity\_calculator.proto .\object\_detection\protos\square\_box\_coder.proto .\object\_detection\protos\ssd.proto .\object\_detection\protos\ssd\_anchor\_generator.proto .\object\_detection\protos\string\_int\_label\_map.proto .\object\_detection\protos\train.proto .\object\_detection\protos\keypoint\_box\_coder.proto .\object\_detection\protos\multiscale\_anchor\_generator.proto .\object\_detection\protos\graph\_rewriter.proto |

* Chạy file setup.py

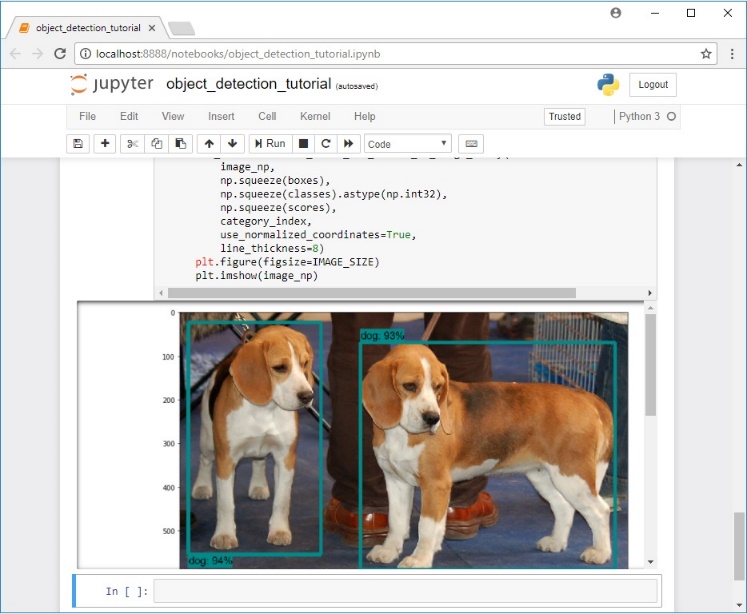
Vào đường dẫn C:\tensorflow1\models\research directory:

|  |
| --- |
| (tensorflow1) C:\tensorflow1\models\research> python setup.py build  (tensorflow1) C:\tensorflow1\models\research> python setup.py install |

* Test Tensorflow chạy tốt! well bước này làm cũng được không làm cũng không sao tại nó chỉ cho mình biết Tensorflow có lỗi hay không thôi, cuối cùng mình cũng phải tự debug -\_-

|  |
| --- |
| (tensorflow1) C:\tensorflow1\models\research\object\_detection> jupyter notebook object\_detection\_tutorial.ipynb |

Nhấn run nếu nó hiện [\*] thì đợi nó load 1 lúc, nếu chạy tốt thì cuối cùng được



Không thì thôi =))

* Test TensorFlow : có thể chạy như trên hoặc :

Tại thư mục *models/research/*

Chạy lệnh từ Anaconda Prompt :

|  |
| --- |
| python object\_detection/builders/model\_builder\_test.py |

Nếu test thành công và không lỗi sẽ nhận được những dòng này ở Anaconda Promt :

|  |
| --- |
|  |

Bước 3: Thu thập dữ liệu và gắn label.

* Chụp hình mấy thứ cần detect, càng nhiều càng tốt, train càng lâu
* Dùng tool LabelImg [https://www.dropbox.com/s/tq7zfrcwl44vxan/windows\_v1.6.0.zip?dl=1 ] khoang vùng và gắn label cho nó để tạo ra file xml.
* Lưu ý để các đường dẫn dưới đây chính xác theo kiểu copy xong chạy thì Tạo thư mục images\train, images\test và quăng hết vào cả file hình lẫn file xml. Dĩ nhiên tự sửa link cho đẹp mắt cũng được =))
* Copy thư mục images vào object\_detection.

Bước 4: tạo training data

* Vào link C:\tensorflow1\models\research\object\_detection chạy xml\_to\_csv.py còn nó thế nào thì tự mở ra mà xem :v

|  |
| --- |
| (tensorflow1) C:\tensorflow1\models\research\object\_detection> python xml\_to\_csv.py |

Chương trình này sẽ tạo ra file *train\_labels.csv* và *test\_labels.csv* trong thư mục images.

* Tiếp theo mở generate\_tfrecord.py lên và sửa lại def class\_text\_to\_int theo số nhãn của mình

|  |
| --- |
| def class\_text\_to\_int(row\_label):  if row\_label == 'person':  return 1  elif row\_label == 'not person 1':  return 2  elif row\_label == 'not person 2':  return 3  else:  return None |

* Tạo train.record và test.record

|  |
| --- |
| python generate\_tfrecord.py --csv\_input=images\train\_labels.csv --image\_dir=images\train --output\_path=train.record  python generate\_tfrecord.py --csv\_input=images\test\_labels.csv --image\_dir=images\test --output\_path=test.record |

Với :

*csv\_input* : nơi chứa file *train\_label.csv* và file *test\_label.csv*

*image\_dir* : nơi chứa file hình ảnh cho train và test

*output\_path* : nơi xuất ra file *train.record* và *test.record*

Bước 5: Tạo label map và Configure Training

* Tạo 1 file tên labelmap.pbtxt vào C:\tensorflow1\models\research\object\_detection\training
* Ghi vào file đó tương tự như class\_text\_to\_int

|  |
| --- |
| item {  id: 1  name: 'person'  }  item {  id: 2  name: 'not person1'  }  item {  id: 3  name: 'not person2'  } |

* Vào thư mục  C:\tensorflow1\models\research\object\_detection\samples\configs copy file .config cùng tên với model mình định làm, ví dụ faster\_rcnn\_inception\_v2\_pets.config copy và quăng nó vào cùng chỗ với label map (C:\tensorflow1\models\research\object\_detection\training)

Lưu ý: copy đi đâu cũng được nhưng đừng quăng nó vào chung với thư mục model( ở *Bước 1* tải *trained model* ) tải về, bị lỗi rồi lôi ra cũng vậy à -\_-

* Mở file config đó lên và sửa các dòng sau:

|  |
| --- |
| num\_classes: 3 (theo số lượng nhãn label của chú =))  fine\_tune\_checkpoint: "C:/tensorflow1/models/research/object\_detection/faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28/model.ckpt"  input\_path: "C:/tensorflow1/models/research/object\_detection/train.record"  label\_map\_path: "C:/tensorflow1/models/research/object\_detection/training/labelmap.pbtxt"  eval\_config: {  metrics\_set: "coco\_detection\_metrics"  num\_examples: 123 #theo số lượng ảnh test của chú  }  input\_path: "C:/tensorflow1/models/research/object\_detection/test.record"  label\_map\_path: "C:/tensorflow1/models/research/object\_detection/training/labelmap.pbtxt" |

Bước 6. Chạy file training

* Vào C:\tensorflow1\models\research\object\_detection \legacy copy file *train.py* ra research/object\_detection/, rồi chạy

|  |
| --- |
| python train.py --logtostderr --train\_dir=training/ --pipeline\_config\_path=training/faster\_rcnn\_inception\_v2\_pets.config |

* Cách để dừng khi train:

*Ctrl + C* “cổ điển”.

* Nếu dừng và train lại sẽ bắt đầu từ đâu:

Sẽ bắt đầu từ *step* đã đc *Recorded* trước đó:

|  |
| --- |
|  |

Tất cả các *step* được *Recorded*  sẽ được lưu tại *train\_dir*.

* Nếu train xong sẽ cho ra gì để mang đi xử lý:

Sẽ cho ra file *model.ckpt* tại *fine\_tune\_checkpoint* ,còn những file *model.ckpt* khác được lưu lại ở *train\_dir* tại ở những *step cố định* :

|  |
| --- |
|  |

File ckpt sau khi đã train xong:

|  |
| --- |
|  |