**MỤC LỤC**

[PHẦN 1: CƠ SỞ LÝ THUYẾT 2](#_Toc56995115)

[I. Các khái niệm về xử lý ảnh 2](#_Toc56995116)

[1.1 Xử lý ảnh 2](#_Toc56995117)

[1.2 Các quá trình xử lý ảnh 2](#_Toc56995118)

[1.3 Điểm ảnh 3](#_Toc56995119)

[1.4 Độ phân giải của ảnh 3](#_Toc56995120)

[1.5 Mức xám của ảnh 4](#_Toc56995121)

[II. Công nghệ 4](#_Toc56995122)

[2.1 Ngôn ngữ python 4](#_Toc56995123)

[2.2 TensorFlow 5](#_Toc56995124)

[2.3 Anaconda 5](#_Toc56995125)

[2.4 VSCode 6](#_Toc56995126)

[PHẦN 2: CÁC BƯỚC THỰC HIỆN 6](#_Toc56995127)

[I. Thu thập dữ liệu. 6](#_Toc56995128)

[II. Gán nhãn. 8](#_Toc56995129)

[III. Đào tạo mô hình phát hiện đối tượng. 9](#_Toc56995130)

[IV. Chạy model trên anaconda. 16](#_Toc56995131)

[PHẦN 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 22](#_Toc56995132)

[3.1 Kết luận 22](#_Toc56995133)

[3.2 Hướng phát triển 22](#_Toc56995134)

[TÀI LIỆU THAM KHẢO 23](#_Toc56995135)

# PHẦN 1: CƠ SỞ LÝ THUYẾT

## I. Các khái niệm về xử lý ảnh

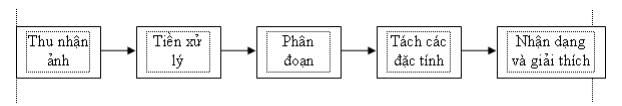
### 1.1 Xử lý ảnh

Xử lý ảnh(XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh. Cũng như xử lý dữ liệu bằng đồ hoạ, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên.

* Mục đích của xử lý ảnh gồm:
* Biến đổi ảnh làm tăng chất lượng ảnh.
* Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có thể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh,… Kỹ thuật này được dùng nhiều trong y học(xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

### 1.2 Các quá trình xử lý ảnh



Thu nhận ảnh: Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình XLA. Ảnh đầu vào sẽ đƣợc thu nhận qua các thiết bị như camera, sensor, máy scanner,v.v… và sau đó các tín hiệu này sẽ được số hóa. Việc lựa chọn các thiết bị thu nhận ảnh sẽ phụ thuộc vào đặc tính của các đối tượng cần xử lý. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.

Tiền xử lý: Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch,v.v… với mục đích làm cho chất lượng ảnh trở lên tốt hơn nữa, chuẩn bị cho các bước xử lý phức tạp hơn về sau trong quá trình XLA. Quá trình này thường được thực hiện bởi các bộ lọc.

Phân đoạn ảnh: Phân đoạn ảnh là bước then chốt trong XLA. Giai đoạn này phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng mức xám v.v… Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều phần tử khác nhau cấu tạo lên ảnh thô. Vì lượng thông tin chứa trong ảnh rất lớn, trong khi đa số các ứng dụng chúng ta chỉ cần trích một vài đặc trƣng nào đó, do vậy cần có một quá trình để giảm lượng thông tin khổng lồ đó. Quá trình này bao gồm phân vùng ảnh và trích chọn đặc tính chủ yếu.

Tách các đặc tính: Kết quả của bước phân đoạn ảnh thường được cho dưới dạng dữ liệu điểm ảnh thô, trong đó hàm chứa biên của một vùng ảnh, hoặc tập hợp tất cả các điểm ảnh thuộc về chính vùng ảnh đó. Trong cả hai trường hợp, sự chuyển đổi dữ liệu thô này thành một dạng thích hợp hơn cho việc xử lý trong máy tính là rất cần thiết. Để chuyển đổi chúng, câu hỏi đầu tiên cần phải trả lời là nên biểu diễn một vùng ảnh dưới dạng biên hay dưới dạng một vùng hoàn chỉnh gồm tất cả những điểm ảnh thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ nhƣ các góc cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu trúc xương của nó. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.

Nhận dạng và giải thích: Đây là bước cuối cùng trong quá trình XLA. Nhận dạng ảnh có thể được nhìn nhận một cách đơn giản là việc gán nhãn cho các đối tượng trong ảnh. Ví dụ đối với nhận dạng chữ viết, các đối tượng trong ảnh cần nhận dạng là các mẫu chữ, ta cần tách riêng các mẫu chữ đó ra và tìm cách gán đúng các ký tự của bảng chữ cái tương ứng cho các mẫu chữ thu được trong ảnh. Giải thích là công đoạn gán nghĩa cho một tập các đối tƣợng đã được nhận biết. Chúng ta cũng có thể thấy rằng, không phải bất kỳ một ứng dụng XLA nào cũng bắt buộc phải tuân theo tất cả các bước xử lý đã nêu ở trên, ví dụ như các ứng dụng chỉnh sửa ảnh nghệ thuật chỉ dừng lại ở bước tiền xử lý. Một cách tổng quát thì những chức năng xử lý bao gồm cả nhận dạng và giải thích thường chỉ có mặt trong hệ thống phân tích ảnh tự động hoặc bán tự động, được dùng để rút trích ra những thông tin quan trọng từ ảnh, ví dụ như các ứng dụng nhận dạng ký tự quang học, nhận dạng chữ viết tay v.v…

### 1.3 Điểm ảnh

Điểm ảnh(Pixel) là một phần tử của ảnh số tại toạ độ(x, y) với độ xám hoặc màu nhất định. Kích thước và khoảng cách giữa các điểm ảnh đó được chọn thích hợp sao cho mắt người cảm nhận sự liên tục về không gian và mức xám (hoặc màu) của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận được gọi là một phần tử ảnh.

### 1.4 Độ phân giải của ảnh

Độ phân giải(Resolution) của ảnh là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị. Theo định nghĩa, khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bổ, đó chính là độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

Ví dụ: Độ phân giải của ảnh trên màn hình CGA(Color Graphic Adaptor) là một lưới điểm theo chiều ngang màn hình: 320 điểm chiều dọc \* 200 điểm ảnh (320\*200). Rõ ràng, cùng màn hình CGA 12” ta nhận thấy mịn hơn màn hình CGA 17” độ phân giải 320\*200. Lý do: cùng một mật độ(độ phân giải) nhưng diện tích màn hình rộng hơn thì độ mịn(liên tục của các điểm) kém hơn. Như hình ảnh dưới đây.

### 1.5 Mức xám của ảnh

Mức xám là kết quả của sự biến đổi tương ứng 1 giá trị độ sáng của 1 điểm ảnh với 1 giá trị nguyên dương. Thông thường nó xác định trong[0, 255] tuỳ thuộc vào giá trị mà mỗi điểm ảnh được biểu diễn. Các thang giá trị mức xám thông thường: 16, 32, 64, 128, 256(Mức 256 là mức phổ dụng. Lý do: từ kỹ thuật máy tính dùng 1 byte(8 bit) để biểu diễn mức xám. Mức xám dùng 1 byte biểu diễn: 28 =256 mức, tức là từ 0 đến 255).

## II. Công nghệ

### 2.1 Ngôn ngữ python

* **Khái niệm**
  + Python là ngôn ngữ lập trình hướng đối tượng thông dụng dùng để viết các tiện ích hệ thống. Nó cũng được sử dụng như ngôn ngữ kết dính đóng vai trò tích hợp C và C++.
  + Được tạo ra bởi Guido van Rossum tại Amsterdam năm 1990, Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.
  + Python là ngôn ngữ có hình thức khá đơn giản và rõ ràng, do đó tạo nên sự dễ dàng tiếp cânh cho những lập trình viên mới bắt đầu.
  + Ban đầu, Python được phát triển để chạy trên nền Unix, nhưng rồi theo thời gian, nó đã được mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.
* **Một số tính năng của python**
  + Interpreted: nhờ chức năng thông dịch mà trình thông dịch (Interpreter) của Python có thể xử lý lệnh tại thời điểm chạy chương trình (runtime). Nhờ đó mà không cần biên dịch chương trình trước khi thực hiện nó (tương tự như Perl và PHP).
  + Interactive: tính năng tương tác của Python giúp tương tác trực tiếp với trình thông dịch của nó ngay tại dấu nhắc lệnh. Cụ thể: Có thể thực hiện lệnh một cách trực tiếp tại dấu nhắc của Python.
  + Object-Oriented: Python hỗ trợ mạnh cho phong cách lập trình hướng đối tượng và kỹ thuật lập trình gói mã trong đối tượng.
  + Beginner's Language: Python được xem là ngôn ngữ lập trình dành cho những ai mới làm quen với việc lập trình trên máy tính, nhưng nó hỗ trợ mạnh cho việc phát triển nhiều loại ứng dụng khác nhau, từ các chương trình xử lý văn bản đơn giản đến các ứng dụng web, đến các chương trình game, …
* **Tính năng**
  + Python là một ngôn ngữ lập trình dễ học, dễ đọc. Python tăng cường sử dụng từ khóa tiếng Anh, hạn chế các ký hiệu và cấu trúc cú pháp so với các ngôn ngữ khác.
  + Mã nguồn của Python tương đối dễ để bảo trì và duy trì và có khả năng mở rộng.
  + Python có một tiêu chuẩn thư viện rộng, nền tảng tương thích trên nhiều hệ điều hành hiện nay như: UNIX, Windows, và Macintosh. Đây cũng là một trong những điểm mạnh đáng kể của Python.
  + Python có thể được sử dụng như ngôn ngữ script, hoặc ngôn ngữ biên dịch, nhờ đó có thể build các chương trình lớn trên nó.
  + Khi chế độ Interactive, có thể nhập kết quả từ các đầu cuối khác nhau vào chương trình Python, do đó mà việc test hay debug lỗi code trở nên đơn giản hơn.
  + Python cho phép người dùng tích hợp vào các module để có thể sử dụng trong các chương trình khác. Nó cũng cung cấp sẵn một tập hợp các modules chuẩn mà lập trình viên có thể sử dụng lại trong chương trình của họ. Các module này cung cấp nhiều chức năng hữu ích, như các hàm truy xuất tập tin, các lời gọi hệ thống, hỗ trợ lập trình mạng (socket),…
  + Python cung cấp giao diện cho tất cả các cơ sở dữ liệu thương mại lớn.
  + Có thế dễ dàng tích hợp với C, C++, COM, CORBA, ActiveX, Java.

### 2.2 TensorFlow

Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lí tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu, trong đó các node là các phép tính toán học còn các cạnh biểu thị luồng dữ liệu.

### 2.3 Anaconda

Anaconda là nền tảng mã nguồn mở về Khoa học dữ liệu trên Python thông dụng nhất hiện nay. Anaconda với hơn 11 triệu người dùng, Anaconda là cách nhanh nhất và dễ nhất để học Khoa học dữ liệu với Python hoặc R trên Windows, Linux và Mac OS X. Lợi ích của Anaconda:

* Dễ dàng tải 1500+ packages về Python/R cho data science
* Quản lý thư viện, môi trường và dependency giữa các thư viện dễ dàng
* Dễ dàng phát triển mô hình machine learning và deep learning với scikit-learn, tensorflow, keras
* Xử lý dữ liệu tốc độ cao với numpy, pandas
* Hiện thị kết quả với Matplotlib, Bokeh

Anaconda là một Distribution miễn phí và mã nguồn mở của Python và R. Anaconda giúp đơn giản hóa việc cài đặt, quản lý và triển khai packages(numpy, scipy, tensorflow,…).

Anaconda phục vụ cho nhiều mục địch, đặc biệt trong Data Science(Khoa học dữ liệu), Machine learnig(Máy học), Big Data(Dữ liệu lớn), Image Processing(Xử lý ảnh),…

Anaconda hiện nay đã có hơn 6 triệu người dùng và hơn 1400 packages khoa học dữ liệu dành cho Windows, Linux và MacOS.

### 2.4 VSCode

VSCode Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

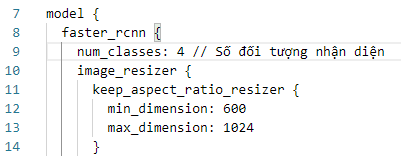
Lý do nên sử dụng Visual Studio Code là vì nó có rất nhiều ưu điểm vượt trội so với bất kỳ IDE nào khác:

* Hỗ trợ đa nền tảng: Windows, Linux, Mac
* Hỗ trợ đa ngôn ngữ: C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript, JSON
* Ít dung lượng
* Tính năng mạnh mẽ
* Intellisense chuyên nghiệp
* Giao diện thân thiện

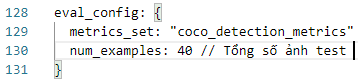
# PHẦN 2: CÁC BƯỚC THỰC HIỆN

## I. Thu thập dữ liệu.

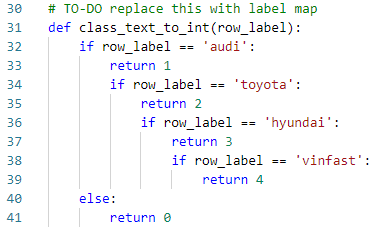
* Tạo một thư mục images có hai thư mục test và train. Thư mục test có tầm 25% ảnh của thư mục train.
* Tạo các file faster\_rcnn\_inception\_v2\_logo.config, generate\_tfrecord.py, labelmap.pbtxt, train.py (<https://github.com/DataExplained/How-to-train-Custom-Object-Detection-Model-Using-Google-Colab>)
* Trong file faster\_rcnn\_inception\_v2\_pets.config thay đổi:
* num\_classes



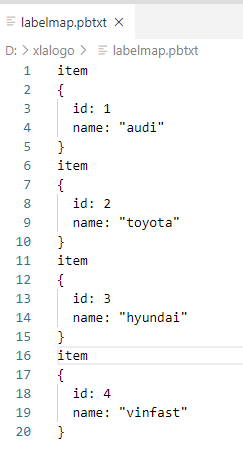
* num\_examples



* Trong file generate\_tfrecord.py thay thế các nhãn đối tượng.



* Trong file labelmap.pbtxt là các nhãn đối tượng.



## II. Gán nhãn.

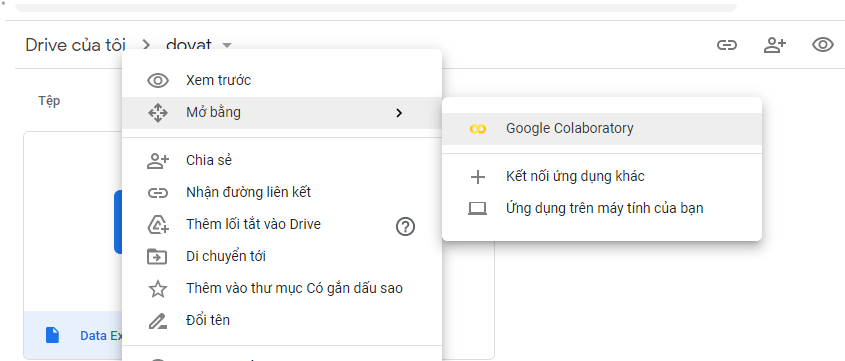
* Tôi sẽ giới thiệu LabelImg là công cụ chú thích hình ảnh đồ họa, được viết bằng python và sử dụng Qt cho giao diện đồ họa. Về cơ bản, đây là vẽ các hộp xung quanh đối tượng của bạn. Chương trình nhãn sẽ tạo một tệp xml có chứa thông tin đối tượng trong hình ảnh tức là. Đường dẫn đối tượng, tên đối tượng, tọa độ đối tượng (Xmin, Ymin, Xmax và Ymax) trong hình ảnh.Bạn có thể lưu tệp xml của mình ở định dạng Pascal VOC được sử dụng bởi định dạng Imagenet hoặc YoLo.
* Cài đặt LabelImg (<https://github.com/maihaan/XLA_K62>). Gán nhãn:
  + Mở phần mềm vào View-> Auto Save Mode
  + Vào Change Save Dir chọn chỗ lưu ảnh đã gán nhãn và Open Dir chọn chỗ ảnh bạn đã thu thập.
  + Vẽ một hình hộp trên hình ghi tên đối tượng và nhấn Save.



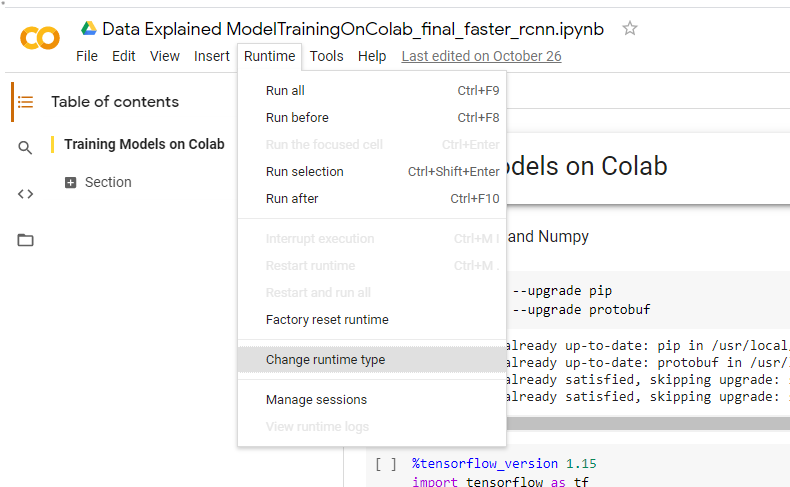
* Gán nhãn xong nén thư mục chứa thư mục images, labelmap.pbtxt, train.py, generate\_tfrecord.py, faster\_rcnn\_inception\_v2\_logo.config. Và tải lên google drive cùng với file Data Explained ModelTrainingOnColab\_final\_faster\_rcnn.ipynb.

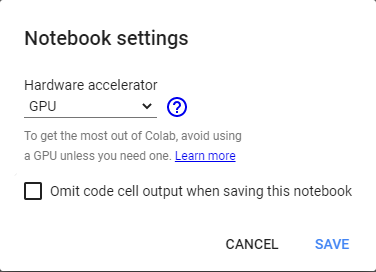
## III. Đào tạo mô hình phát hiện đối tượng.

* Nhấn file Data Explained ModelTrainingOnColab\_final\_faster\_rcnn.ipynb mở Google Colaboratory.



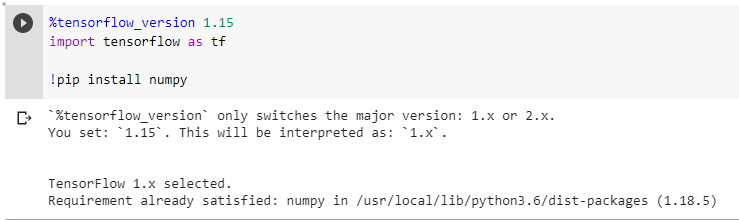
* Vào Runtime => Change Runtime Type và chọn GPU để tăng tốc phần cứng. Nhấn Save.



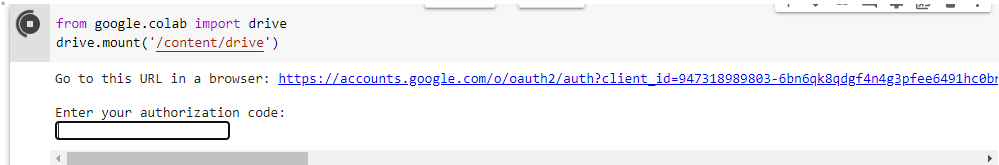


* Cài đặt Tensorflow và Numpy. Chạy các mã bên dưới.

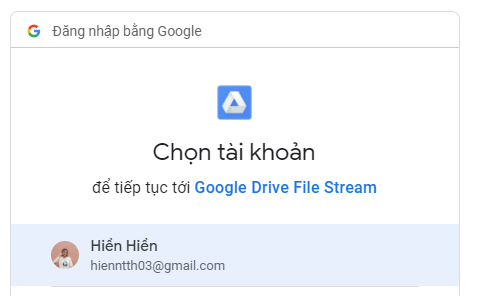




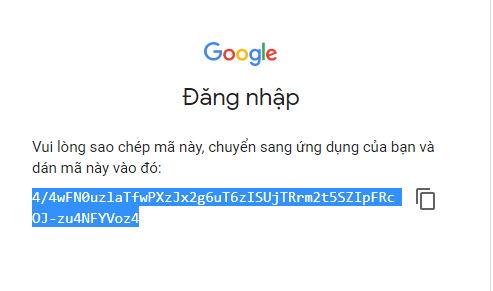
* Nhấn vào đường link.



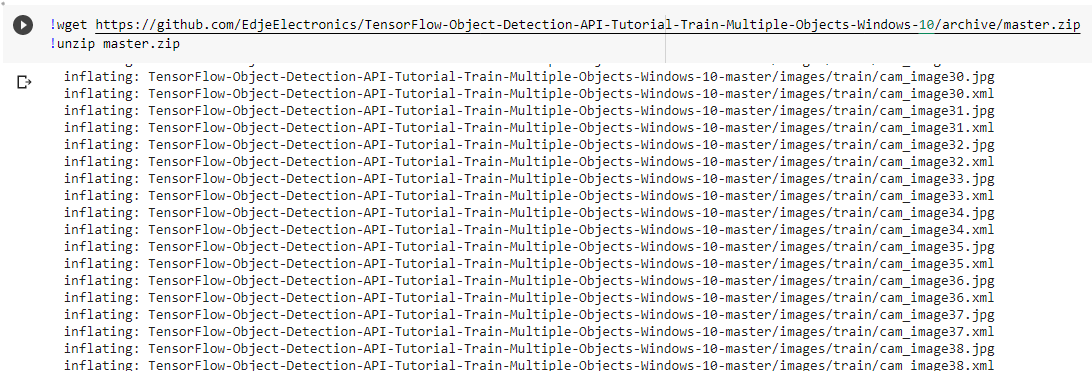
* Chọn tài khoản bạn dùng để đăng nhập vào Google Colab.

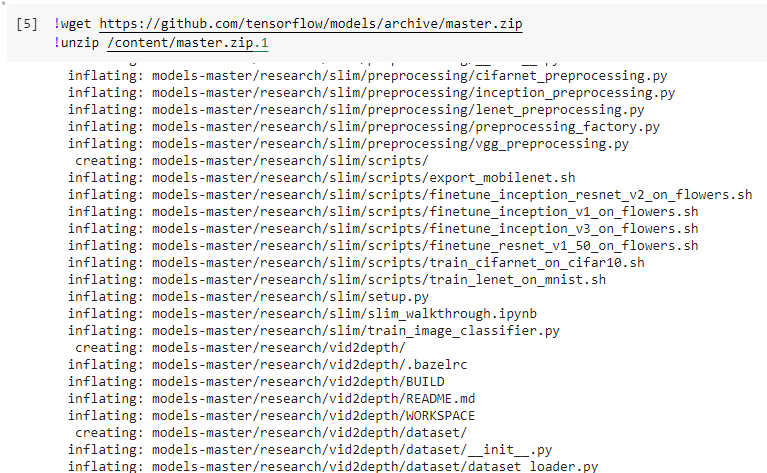


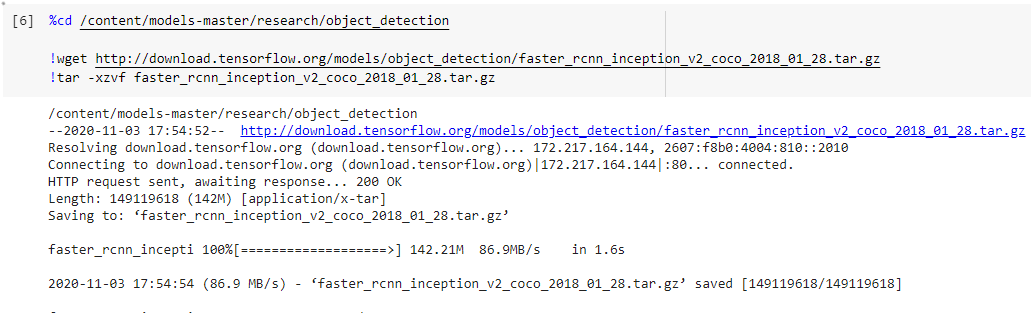
* Sao chép mã bên dưới Paste vào phiên Colab đang chạy rồi nhấn Enter.

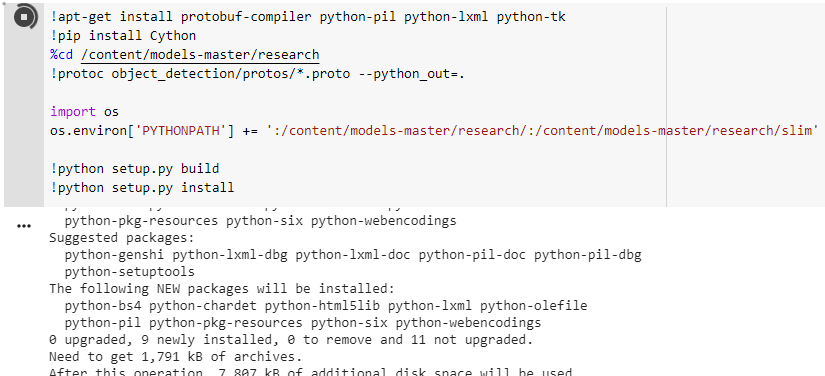


* Chạy tiếp các mã dưới.

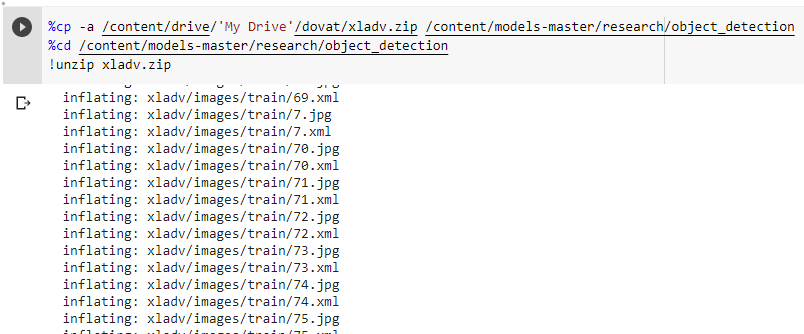








* Sửa đến đúng đường dẫn đến thư mục để các file tải lên Google Drive.

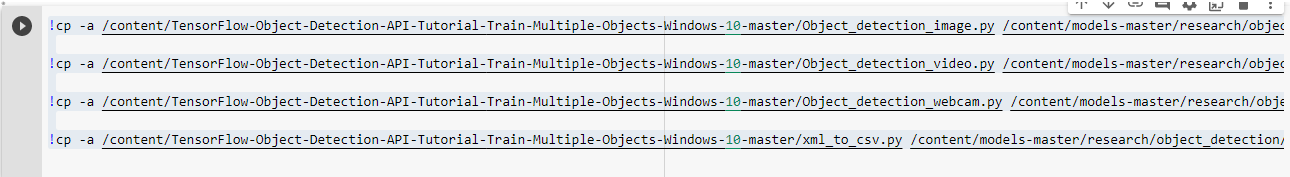


!cp -a /content/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10-master/Object\_detection\_image.py /content/models-master/research/object\_detection/

!cp -a /content/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10-master/Object\_detection\_video.py /content/models-master/research/object\_detection/

!cp -a /content/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10-master/Object\_detection\_webcam.py /content/models-master/research/object\_detection/

!cp -a /content/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10-master/xml\_to\_csv.py /content/models-master/research/object\_detection/xladv/

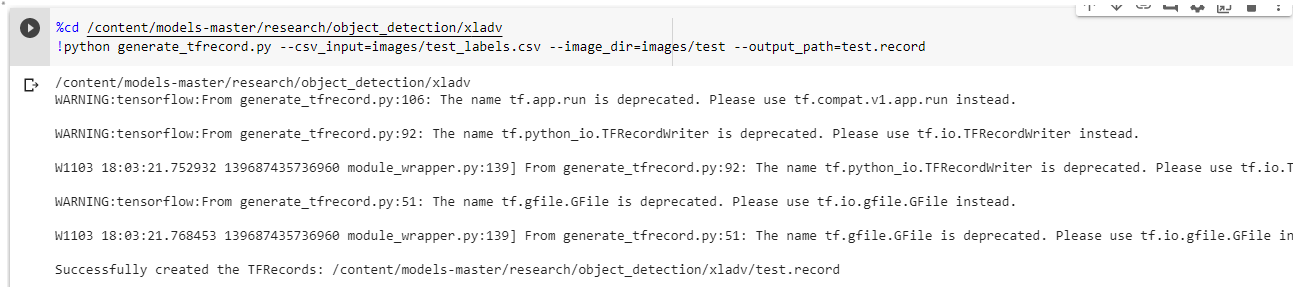


* Chuyển images từ xml sang csv.

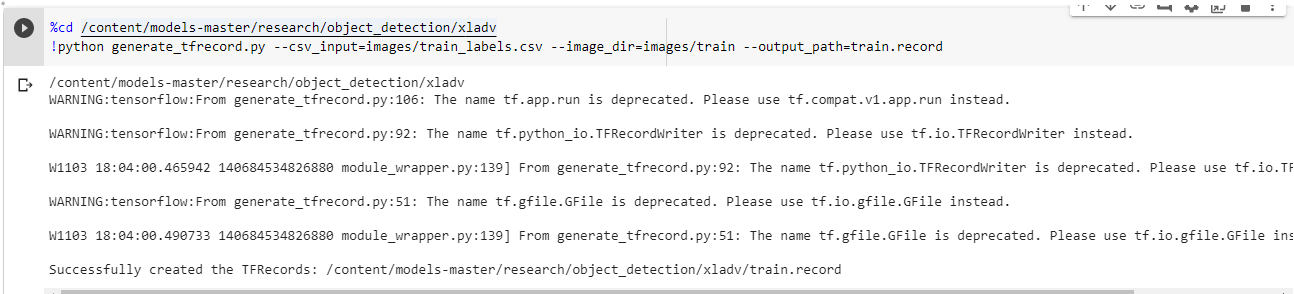




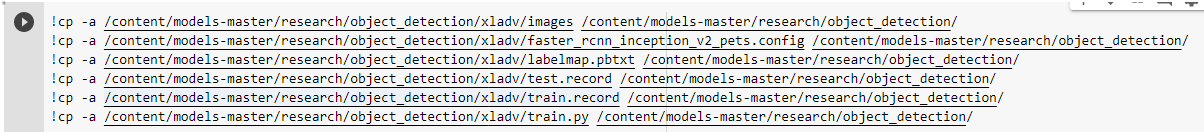
* Tạo dữ liệu kiểm tra.



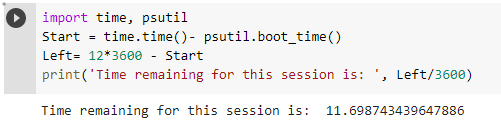
* Tạo dữ liệu đào tạo.



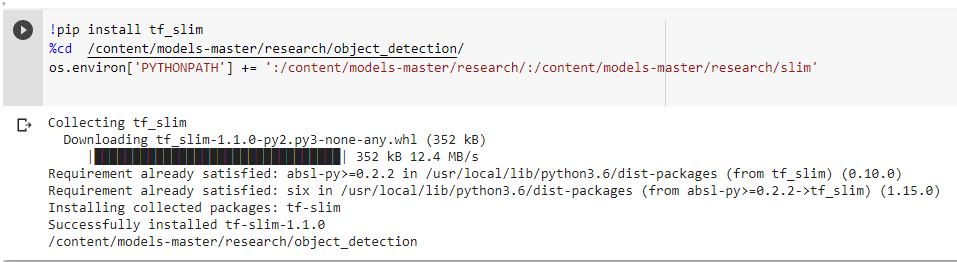
* Sửa đúng đường dẫn đến thư mục.

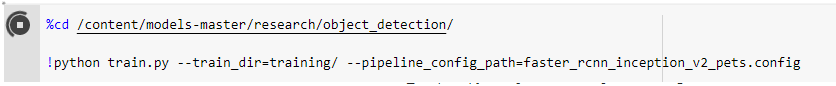


* Kiểm tra thời gian còn bao nhiêu ( Một phiên hoạt động của Google Colab là 12 tiếng, hết thời gian là tự động xoá hết các dữ liệu nên khi đào tạo thì chạy trong 12 tiếng).

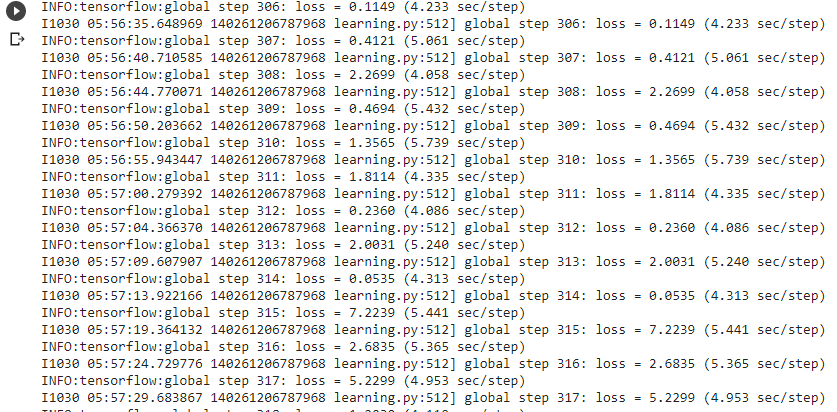


* Bắt đầu đào tạo. Chạy các mã dưới để đào tạo mô hình.

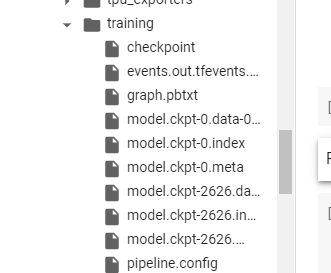




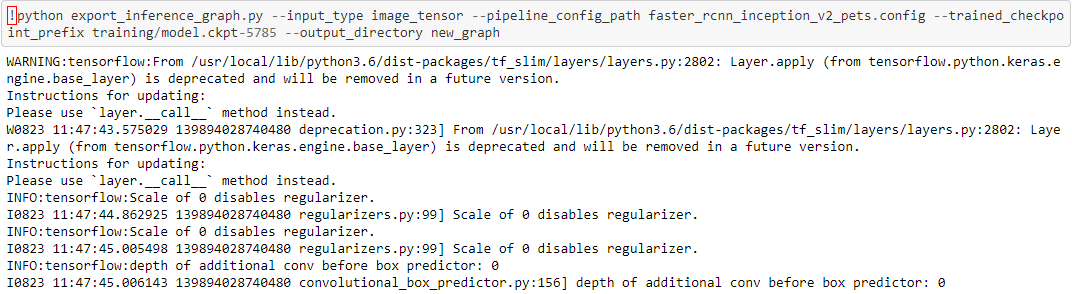
* Đào tạo cho đến khi global step hơn 50000 và loss nhỏ hơn 0.05 (Lúc thấy điểm kiểm tra và bạn thấy thích hợp thì dừng lại nhấn ctrl+M)



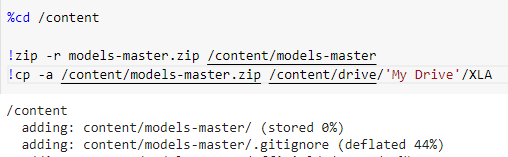
* Vào models-master -> research -> object\_detection -> training lấy điểm kiểm tra cao nhất.



* Lấy điểm kiểm tra thay vào đuôi ckpt.

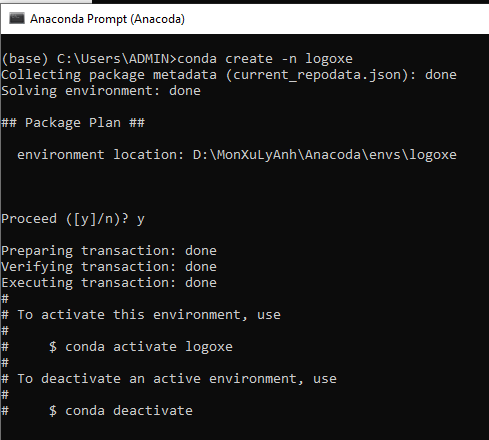


* Chạy mã bên dưới và thay đổi đường dẫn để nén lại model và tải lên google drive.



## IV. Chạy model trên anaconda.

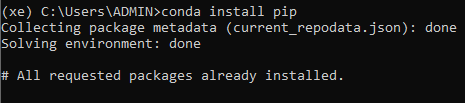
* Tải model về máy. Vào thư mục object\_detection copy file labelmap.pbtxt dán vào thư mục training.
* Mở anaconda gõ câu lệnh:
  + conda create –n <tên mình đặt> => chọn y



* conda activate <tên mình đặt>

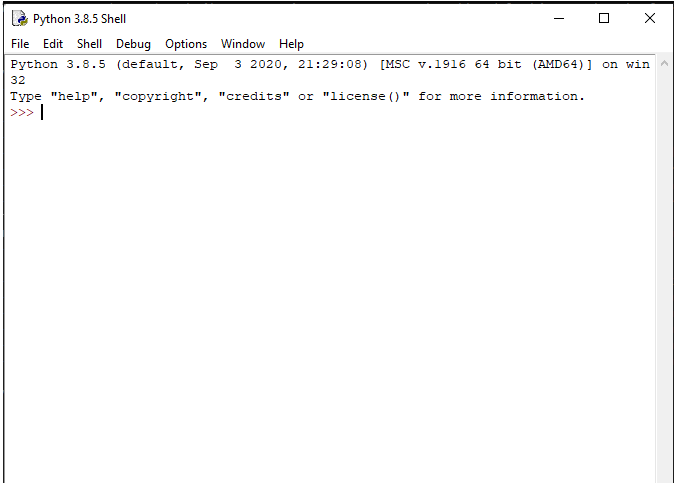


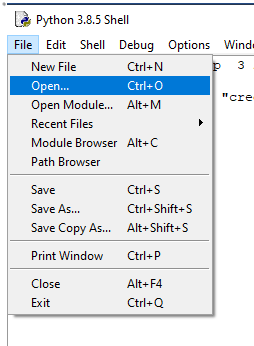
* conda install pip



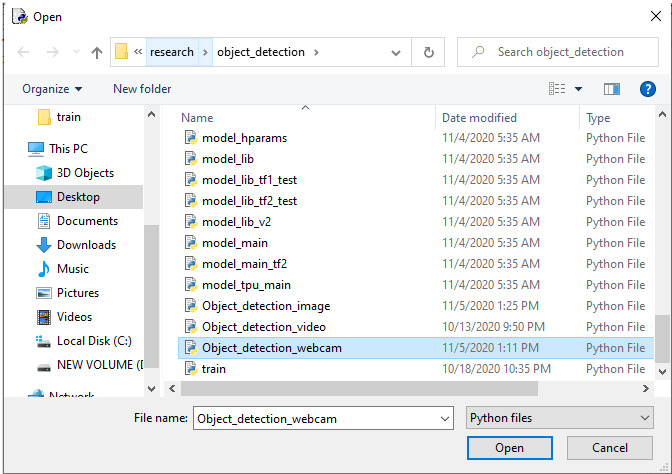
* Chuyển đến thư mục research bằng câu lệnh cd.



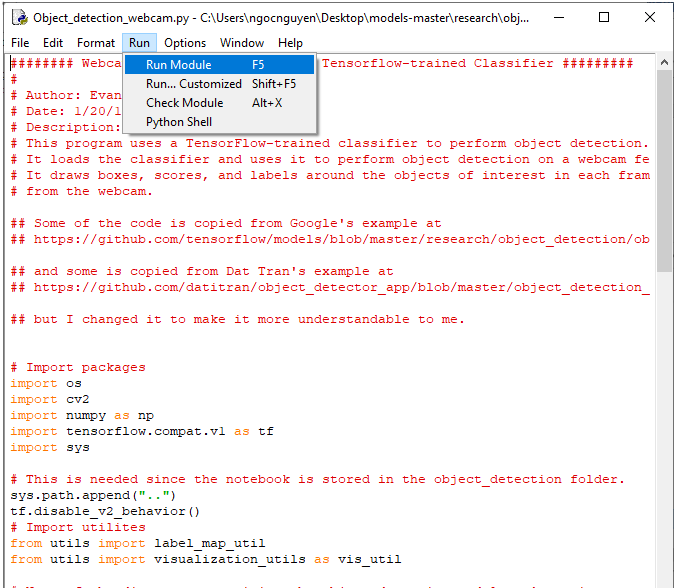
* Gõ câu lệnh protoc object\_detection/protos/\*.proto --python\_out=.
* Python setup.py build
* python setup.py install
* Gõ idle sẽ xuất hiện như hình dưới.
* Nhấn File -> Open.



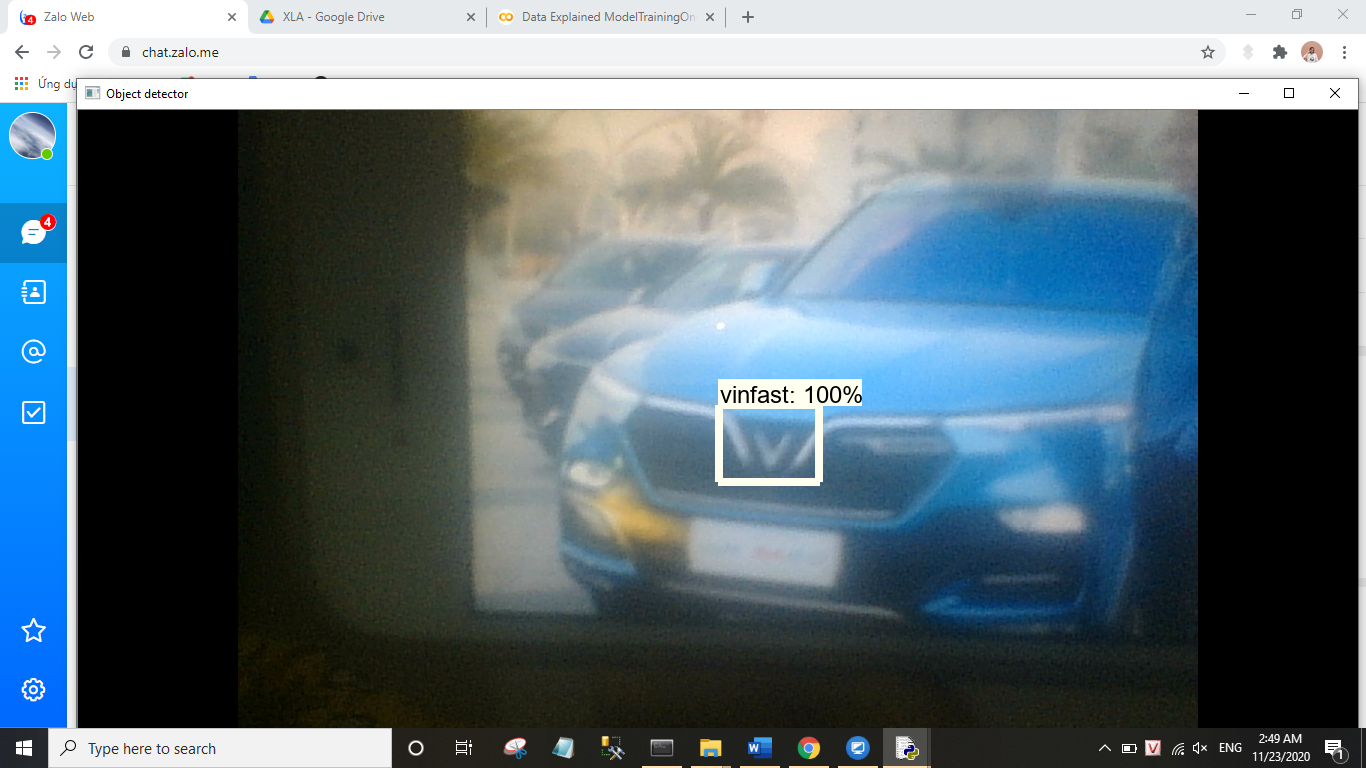
* Chọn Object\_detection -> Object\_detection\_webcam.



* Chọn Run -> Run Module.



* Hiển thị ra màn hình webcam như dưới.



# PHẦN 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 3.1 Kết luận

* Đã hoàn thành được mục tiêu đã đề ra.
* Hiện tại đã xây dựng được mô hình đào tạo.

## Hướng phát triển

* Hoàn thiện mô hình thu thập dữ liệu nhiều hơn.
* Quá trình đào tạo sẽ được kỹ hơn.
* Do thời gian và kiến thức còn hạn hẹp nên không thể tránh khỏi những thiếu sót trong quá trình thực hiện đề tài. Rất mong nhận được những góp ý của Thầy và các bạn để bài báo cáo được hoàn thiện hơn.

# TÀI LIỆU THAM KHẢO

[1]

https://github.com/DataExplained/How-to-train-Custom-Object-Detection-Model-Using-Google-Colab

[2]

https://github.com/maihaan/XLA\_K62

[3]

https://medium.com/analytics-vidhya/training-an-object-detection-model-with-tensorflow-api-using-google-colab-4f9a688d5e8b