Báo cáo đồ án

I. Tính cấp thiết của đề tài:

Thế giới vừa mới trải qua những tổn thất nghiêm trọng do chủng virus corona mang đến. Trong giai đoạn dịch bệnh căng thẳng, việc các cá nhân tự bảo vệ bản thân bằng các biện pháp phòng tránh như là một điều kiện bắt buộc và quan trọng nhất trong việc giảm thiểu các hậu quả của virus mang lại cho bản thân cũng như mọi người xung quanh. Với việc lây nhiễm chính bằng đường không khí, việc trang bị cho bản thân những chiếc khẩu trang như một lớp giáp bảo vệ bản thân với sự xuất hiện của virus trong không khí chưa bao giờ là thừa lúc này. Việc tiếp xúc giữa các cá thể với nhau trong những khu vực có mật độ cá thể lớn là điều kiện tuyệt vời đề cho dịch bệnh được lây lan, khi mà một số chủng virus đã được chứng minh là có thể lây lan trong không khí ở một khoảng cách rất xa. Chính vì vậy mà mỗi cá nhân trong thời đại dịch bệnh luôn được khuyến khích, thậm chí là bắt buộc phải trang bị cho cá nhân một chiếc khẩu trang khi tiếp xúc với mọi người trong cộng đồng, đó như là biện pháp để bảo vệ cho an toàn của bản thân cũng như an toàn của mọi người. Tuy vậy, có nhiều cá nhân vẫn chưa nhận thức được sự quan trọng của khẩu trang trong việc phòng chống lây nhiễm virus, và tiếp xúc với mọi người trong cộng đồng với tình trạng không đeo khẩu trang. Việc phát hiện và cảnh báo các cá nhân đó ở những nơi công cộng có thể không xảy ra kịp thời được, từ đó có thể tạo điều kiện cho virus được lây lan nhiều hơn nếu như những cá thể không mang khẩu trang đó dương tính.

Máy tính với sức mạnh tính toán vượt trội có thể hỗ trợ con người trong việc phát hiện và cảnh báo các cá thể trong cộng đồng trong việc mang khẩu trang một cách kịp thời. Nhận thấy được tính thiết thực của vấn đề này và sự hiệu quả của mô hình máy tính vào việc phát hiện và cảnh bảo đeo khẩu trang, nhóm quyết định chọn đề tài “Ứng dụng deep learning vào phát hiện cá thể mang khẩu trang”.

II. Nội dung của đề tài:

1. Phát hiện vật thể

Object detection, hay còn gọi là phát hiện vật thể, là một lĩnh vực trong thị giác máy tính và trí tuệ nhân tạo. Thị giác máy tính là một ngành khoa học đã và đang phát triển rất mạnh, đặc biệt là trong những như sau này, khi có sự ra đời của các cấu trúc mạng neural. Trong nội dung của đồ án, nhóm tìm hiểu một nhánh của thị giác máy tính, object detection với hướng tiếp cận theo mạng neural. Hơn 90% thông tin mà con người xử lý hằng ngày được tiếp nhận thông qua kênh thị giác, bởi vì đặc thù của thị giác nên tốc độ xử lý của con người khi tiếp nhận những thông tin này cũng nhanh hơn nhiều so với thông tin được tiếp nhận từ những kênh khác. Bằng việc áp dựng cơ chế tương tự như vậy vào máy tính, các nhà khoa học đã phát minh ra hướng tiếp cận theo neural, đây là một hướng tiếp cận khác, mạnh mẽ hơn nhiều, và đánh bại các hướng tiếp cận cũ trong việc xử lý bài toán với độ phức tạp cao với thời gian xử lý nhanh. Thông tin được nhận vào từ thị giác được xử lý qua hệ các dây thần kinh neural được nối với nhau một cách phức tạp, từ đó trích xuất ra được các thông tin trừu tượng được xử dụng để xử lý và tính toán ra các thông tin mới theo mục đích của bài toán.

2. Áp dụng vào bài toán nhận diện khẩu trang:

Con người nhìn vào một cá thể liền có thể lập tức nhận ra được là cá thể đó có đang mang khẩu trang hay không mà không cần phải mất thời gian nhiều cho việc tính toán và tiếp nhận. Và nếu như một mô hình học máy, được xây dựng với một cấu trúc hợp lý và được cung cấp cho một nguồn kiến thức đủ phong phú để máy có thể học được cách nhận biết và phân biệt các cá thể có mang khẩu trang hay không, thì cộng với tốc độ xử lý cực nhanh của máy tính, máy tính hoàn toàn có thể hỗ trợ con người trong việc hỗ trợ phát hiện nhận dạng khẩu trang trong tầm nhìn của máy tính.

Một bức ảnh được tiếp nhận vào mắt người, thì ta có thể có biết được khẩu trang nằm ở đâu trên bức ảnh, và khẩu trang có được người trước mặt đeo không thông qua những thông tin trừu tượng hoặc cụ thể mà não ta trích xuất được, ví dụ như: khẩu trang có che được miệng, hay khẩu trang có che được mũi hay không. Khẩu trang có được đeo vào vị trí hợp lý trên khuông mặt hay không,… Những thông tin đó hoàn toàn có thể được biểu diễn trong máy tính dưới dạng các đặc trưng, và máy tính có thể học được và trích xuất được thông tin từ việc sử dụng những đặc trưng đó trong việc tính toán.

Trong hướng tiếp cận theo mạng neural, bức ảnh được biểu diễn dưới dạng một ma trận với các giá trị tương ứng cho từng vị trí trên bức ảnh, giá trị của mỗi vị trí trong ma trận đặc trưng cho mức độ mạnh yếu về sắc độ trong bức ảnh. Việc biểu diễn thông tin như thế này làm cho bức ảnh trở thành một ma trận mang thông tin với giá trị có thể tính toán được.

3. Hướng tiếp cận của đồ án:

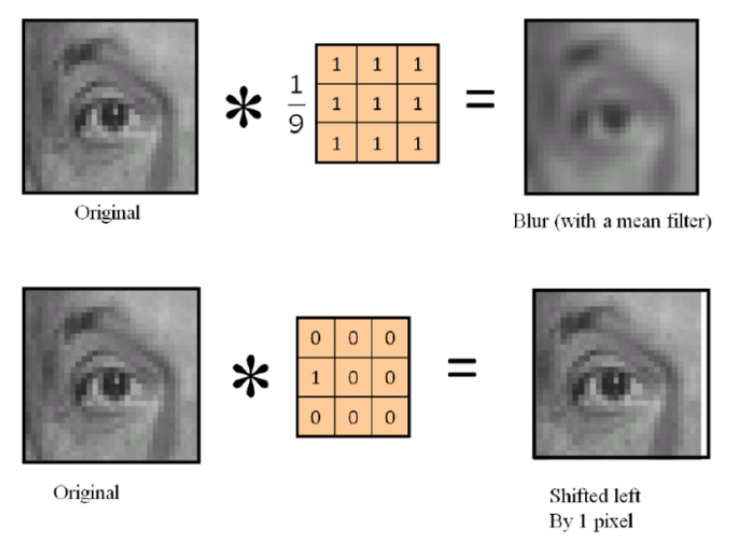
Trong khi liệt kê ra các yêu cầu chức năng phi chức năng của mô hình hiện thực, nhóm nhận thấy mô hình như là một phần hỗ trợ con người trong việc giám sát và phát hiện các cá thể đeo khẩu trang trong cộng đồng, cho nên tốc độ xử lý là yếu tố được ưu tiên, nhưng vẫn phải đảm bảo được độ chính xác tương đối. Có một số hướng tiếp cận theo mô hình neural mà nhóm tìm hiểu được bao gồm:

* Mô hình multi feature map detection, ví dụ như SSD. Với việc phát hiện và nhận dạng vật thể thông qua một lần forward, nên mô hình SSD mang lại tốc độ xử lý khá cao, nếu được thiết kế hợp lý, có thể áp dụng việc nhận diện theo thời gian thực, bên cạnh đó độ chính xác cũng khá cao, có thể nhận dạng được chính xác cács trường hợp dễ.
* Mô hình Region based convolutional neural network, R-CNN. Việc nhận dạng và tính toán có sự hỗ trợ của giải thuật trong việc xác định vùng có tiềm năng chứa vật thể trong bức ảnh. Độ chính xác của mô hình khá cao, và dự đoán khá chính xác vị trí của vật thể trong khung ảnh. Tuy vậy, tốc độ của mô hình trong việc tính toán khá lâu, việc áp dụng vào thực tế ở một bài toán yêu cầu xử lý theo thời gian thực như việc nhận dạng đeo khẩu trang là khá khó khăn và bất khả thi, vì độ trễ có thể làm cho mô hình mất đi nhiều thông tin có thể nhận được trong thời gian tính toán, cộng với khối lượng tính toán lớn như vậy là cản trở trong việc áp dụng ở trong các thiết bị nhúng hay thiết bị di động với cấu hình thấp.
* Mô hình YOLO, thay đổi trong việc chia bức ảnh ra các vùng đặc trưng với các bounding box khác nhau, mô hình bao gồm phần cnn để trích xuất thông tin từ bức ảnh mà chỉ gồm các lớp fully connected, nên tốc độ xử lý của mô hình YOLO rất nhanh, có thể đáp ứng được các bài toàn yêu cầu xử lý theo thời gian thực.

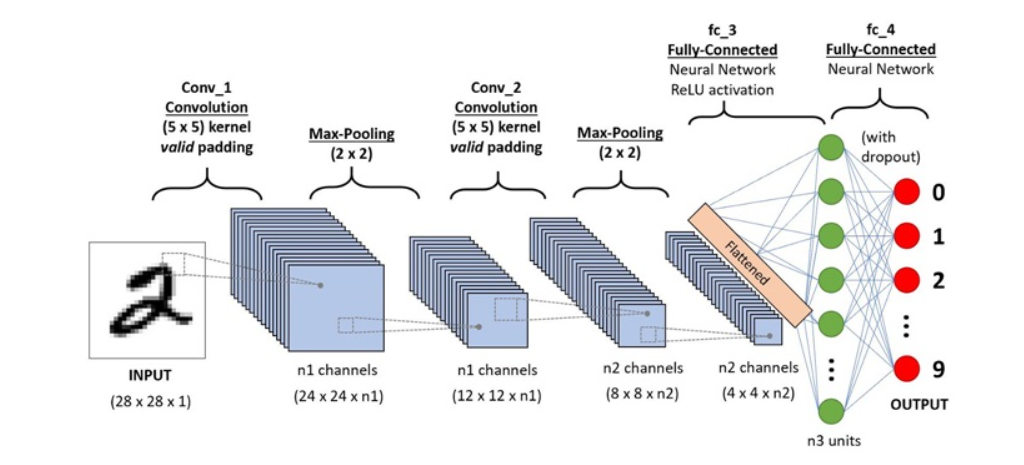
Trong nội dung hiện thực của đồ án, nhóm quyết định hiện thực và sử dụng mô hình theo cấu trúc SSD, cụ thể là mô hình SSD300 để áp dụng vào bài toán nhận dạng khẩu trang.

4. Một số kiến thức cơ bản trong bài toán object detection theo hướng tiếp cận mô hình neural:

Convolutional layer, pooling layer: Là một lớp tích chập, thành phần chính của một lớp tích chập chính là một cửa sổ nhân (kernel window) với kích thước cố định, là một ma trận với các giá trị. Cửa số này sẽ trượt trên bức ảnh, tại mỗi vị trí trong bức ảnh sẽ thực hiện các tác vụ, nhằm biến đổi ma trân ảnh đầu vào thành một ma trận đặc trưng ở sau, với mục đích trích xuất được đặc trưng từ bức ảnh.



Một cửa số nhân với các thông số khác nhau có thể trích xuất được các đặc trưng khác nhau trong bức ảnh, như là các đường thẳng, các đường chéo, các đường ngang, đây là các đặc trưng sơ cấp trong bức ảnh, và càng về sau của mô hình, các thông tin được nhận vào càng trừu tượng, và các lớp này càng trích xuất được các thông tin cao cấp hơn nữa.



Ví dụ như, với một bức ảnh bản đầu, trong các convolutional layer đầu tiên, thông tin trích xuất ra được chỉ là các đường nét thẳng và ngang xuất hiện trong bức ảnh, thì càng về cuối của mô hình, các layer càng sau càng nhận được các thông tin trừu tượng hơn nữa, như là một đường nét bao quanh khuông mặt, mũi, miệng, chiếc lá,.. Thực chất các thông tin trừu tượng này cũng chỉ là thông tin được trích xuất ra từ những thông tin sơ khai nhất. Ví dụ như là con mắt của con người sẽ là tập hợp của các đường nét tạo thành viền mắt, và các đường nét tạo thành con ngươi. Các thức xử lý của convolutional lấy cảm hứng từ các mạng thần kinh neural ở trong não bộ con người.

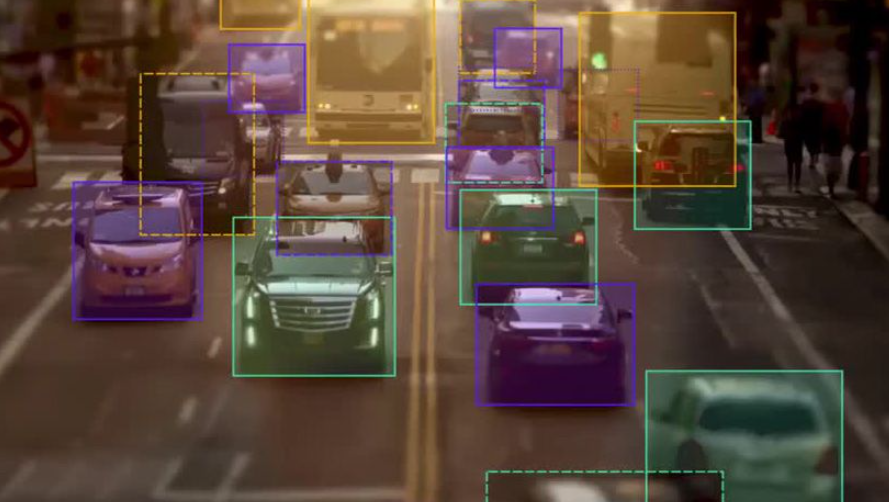
Bounding box:

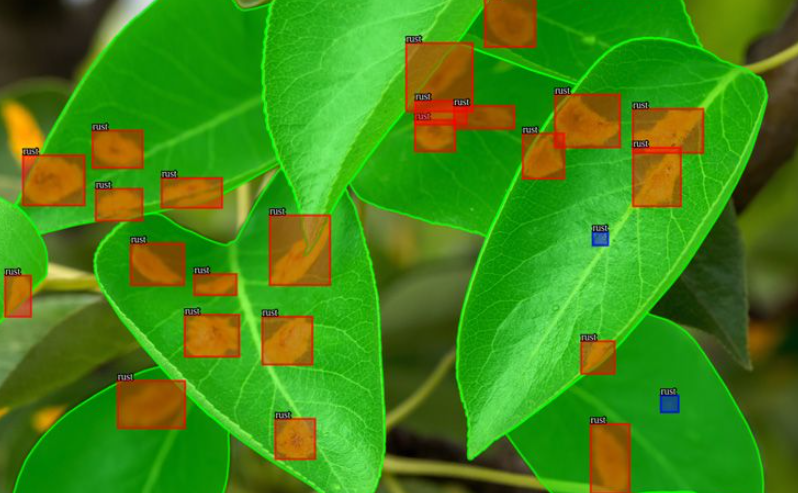
Sau khi trích xuất ra các đặc trưng trừu tượng bậc cao từ bức ảnh, để giải quyết bài toán object detection, mô hình phải học được cách sử dụng các đặc trưng đó để phát hiện vật thể và dự đoán lớp mà vật thể đó thuộc về.

Vị trí của vật thể được giới hạn trong một phần không gian ở khung hình, được gọi là bounding box. Có hai cách biểu diễn bounding box chính được sử dụng trong các mô hình:

* Biểu diễn thông qua một cặp hai toạ độ của các điểm ở góc trên bên trái và góc dưới bên phải của bounding box.
* Biểu diễn thông qua toạ độ của trung tâm bounding box và chiều dài, rộng của bounding.

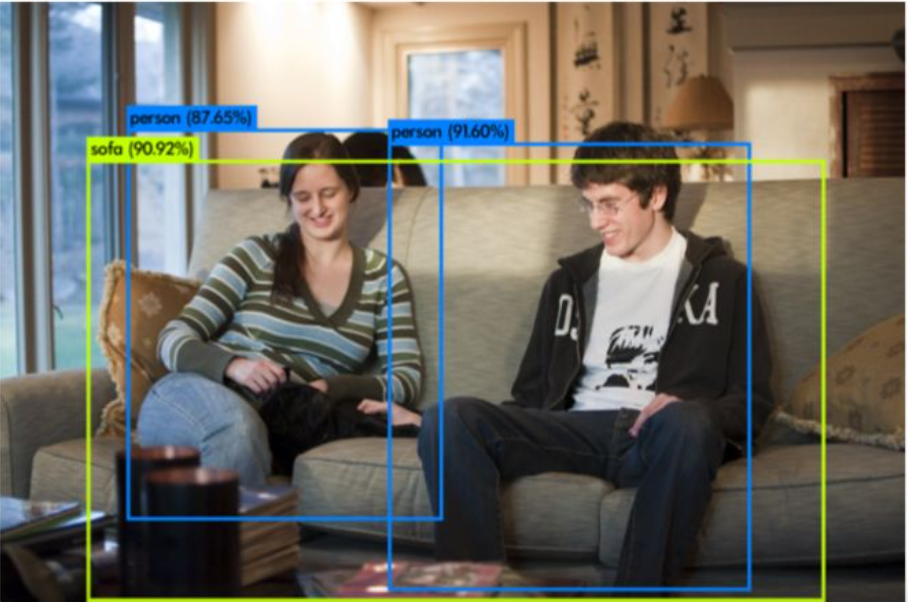
Trong bài toán nhận dạng vật thể của nhóm, các vật thể được xác định bởi một bounding box 2d, nên sẽ cần tối thiểu 4 thông tin để xác định được một bounding box cụ thể.





Confidence score:

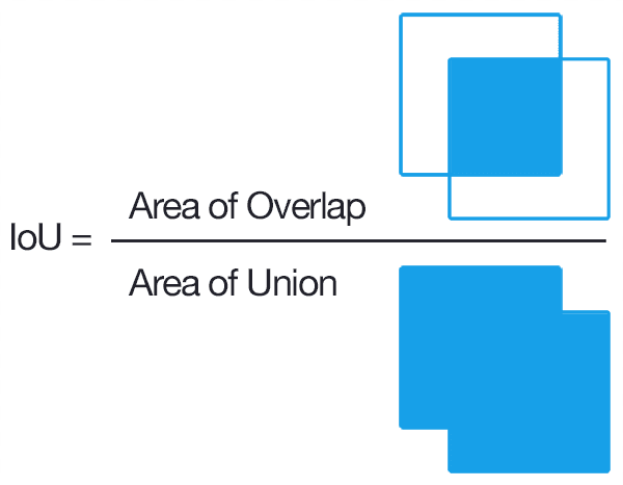
Việc chỉ dự đoán ra được thông tin về vị trí của bounding box của vật thể là chưa đủ, khi mà bài toán object detection còn bao gồm bài toán classsification ở trong đó. Cần phải có một thông số để đánh giá khả năng mà vật thể xuất hiện trong bounding box thuộc về một lớp nào đó. Đó là nhiệm vụ của confidence score. Một object có confidence score càng lớn, thì khả năng mà bounding box dự đoán vật thể thuộc về lớp tương ứng càng chính xác cao.



Đây chính là một kết quả cần được dự đoán bởi mô hình. Confidence score có thể được biểu diễn bởi nhiều hình thức, một hình thức khá phổ biến để biểu diễn confidence score chính là cross entropy. Gồm một dãy các xác xuất mà biểu diễn khả năng vật thể thuộc về một lớp nào đó tương ứng. Tổng các xác xuất này bằng 1.

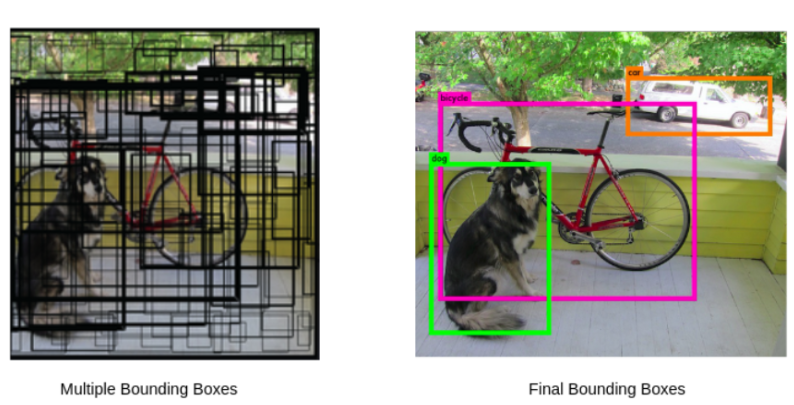
IoU:

Các phần không gian xuất hiện trong bức ảnh nếu bị nằm đè lên nhau thì ta cần phải có một chỉ số để phân biệt được mức độ trùng lặp không gian của các phần không gian này. Trong bài toán object detection và các bài toán khác, IoU chính là một số đo, biểu diễn mức độ trùng nhau của hai phần không gian. Chỉ số này chính là tỉ số giữa phần không gian giao nhau chia cho phần không gian chung chứa chúng.



Hai phần không gian có chỉ số IoU càng lớn, thì có nghĩa là chúng trùng nhau càng nhiều, và nếu iou = 1 thì đây chính là hai phần không gian trùng nhau.

Non-maximum supression:

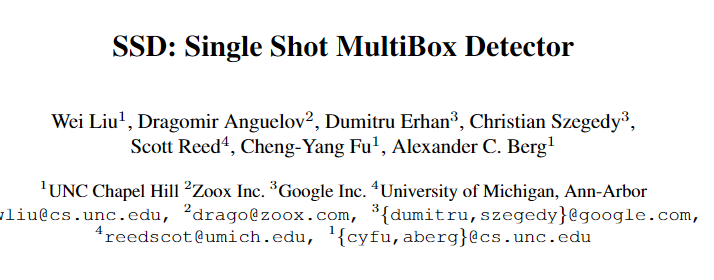


Việc dự đoán sự xuất hiện của vật thể thông qua các multi scale anchor box có thể làm xuất hiện tình trạng có nhiều bounding box xuất hiện trong khung hình, cùng dự đoán về một vật thể nằm trong phần không gian nào đó. Cụ thể, nếu có nhiều bounding box được dự đoán ra nằm chồng lên nhau trong cùng một phần không gian của bức ảnh, thì khả năng các bounding box dự đoán cùng một vật thể là rất cao. Nếu các bounding box bị trùng nhau như vậy thì cần có một giải thuật để lọc và loại bỏ các bounding box kém chính xác hơn, và giữ lại các bounding box dự đoán vật thể một các chính xác nhất.

Đó chính là nhiệm vụ của giải thuật non maximum supression. Cụ thể trong giải thuật non maximum supression, giải thuật sẽ nhận vào danh sách các bonding box được dự đoán từ mô hình và confidence score tương ứng của chúng. Giải thuật sẽ tiến hành lặp lại, và chọn lọc ra các bounding box theo trình tự như sau:

* Chọn ra một bounding box có confidence score lớn nhất, tức là bouding có khả năng dự đoán vật thể đúng nhất làm bounding box thông tin.
* Tiến hành so sánh các bounding box còn lại với bounding box trung tâm, và loại bỏ đi các bounding box trùng với bounding box thông tin hiện tại nếu mức độ trùng lớn hơn một ngưỡng nào đó được định nghĩa trước. Thông thường nếu mức độ giao nhau lớn hơn 50-60% thì ta có thể xem rằng hai bounding box đó có khả năng dự đoán cùng một vật thể là rất cao.
* Tiến hành lặp lại quá trình trên cho đến lúc nào thu được số lượng bounding box cần thiết.

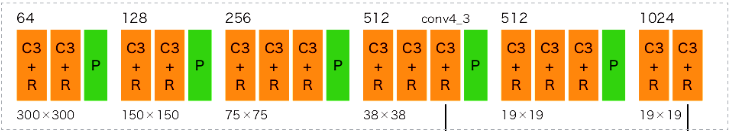
5. Thiết kế mô hình  
Mô hình SSD300 mà nhóm hiện thực dựa trên thiết kế theo mô hình được publish ở paper của nhóm nghiên cứu:



Mô hình có thể chia ra thành những module chính là: module học thông tin từ bức ảnh, module trích xuất đặc trưng từ bức ảnh, module dự đoán vị trí của vật thể trong bức ảnh, module dự đoán loại vật thể ở trong vị trí tương ứng.

**VGG16 Module**

Phần module học thông tin từ bức ảnh, nhóm sử dụng kiến trúc vgg16:

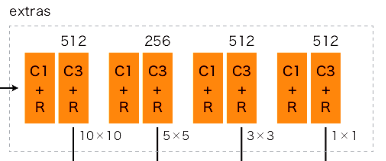


Gồm các lớp tích chập với kích thước nhân tích chập và lớp gộp nhằm giảm kích thước mỗi kênh bản đồ đặc trưng sau mỗi khối, nhằm tăng số kênh thông tin nhận vào ở khối sau, có nghĩa là thông tin nhận được ít cụ thể hơn nhưng ở phạm vi tổng quát hơn. Càng về cuối mô hình thì thông tin học được càng trừu tượng và tổng quát hơn so với những lớp đầu.

Ý tưởng để hiện thực module theo kiến trúc vgg16 đó chính là, đây là một kiến trúc dễ hiện thực khi bao gồm các lớp tích chập kết hợp với nhau một cách có thứ tự tạo thành một khối vgg và sau đó kết nối các khối với nhau để tạo thành một mô hình vgg16.

Cụ thể trong mô hình nhóm hiện thực, mỗi khối vgg gồm 2 lớp tích chập 3 \* 3 với padding = 1 và stridge = 1, nhằm giữ lại thông tin về chiều cao và chiều rộng của feature map, và sau cùng của một khối vgg chính là một lớp down sampling, ở đây nhóm sử dụng lớp Max Pooling với size = 2 \* 2 và stride = 2. Sau mỗi lần đi qua một khối vgg thì kích thước của feature map sẽ giảm đi gấp đôi, bù lại số channels sẽ được tăng lên gấp đôi, có thể hình dung, việc này sẽ giúp cho các layer ở càng về sau của mô hình thì sẽ nhận được bản đồ đặc trưng càng tổng quát hơn, và càng có nhiều đặc trưng được trích xuất hơn

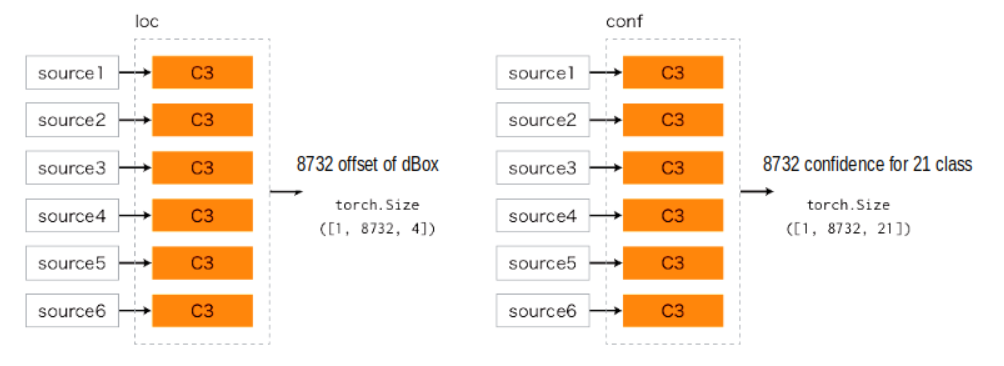
Phần module trích xuất đặc trưng từ bức ảnh, nhóm sử dụng kiến trúc như sau:



Module extras có nhiệm vụ sẽ học từ các đặc trưng được trích xuất từ module vgg và nhận biết được các object mà mô hình cần phát hiện, cụ thể ở trong đề tài này chính là các đối tượng thuộc về 4 lớp chính. Module extras gồm 4 khối, với mỗi khối gồm 2 lớp nhân tích chập. Lớp đầu tiên là một lớp convolutional layer với kernel size là 1 \* 1, do đây là tích chập theo từng pixel trong feature map nên tác dụng của lớp này cũng tương tự như một lớp fully connected layer. Lớp tiếp theo là một lớp tích chập với kernel size 3 \* 3, stride = 2 và padding = 1. Feature map sau khi qua lớp này sẽ bị giảm kích thước đi một nửa và tăng số lượng kênh thông tin lên.

Feature map được trích xuất ra từ module extras sẽ được cho vào hai module tiếp theo để dự đoán các thông tin cho việc nhận dạng và phát hiện đối tượng.

Sau hai lớp trên sẽ là hai module dự đoán vị trí và lớp của đối tượng phát hiện được.



Thông tin được sử dụng để nhận dạng các vật thể và dự đoán không phải nhất thiết lúc nào là thông tin được trích xuất ra cụ thể nhất là cũng mang lại độ chính xác cao, một số trường hợp, tương tự như sự suy nghĩ của não bộ con người khi nhận dạng vật thể, ta có thể dễ dàng nhận ra được đặc trưng và phân biệt được vật thể khi chỉ mới nhìn qua, hay là chỉ mới có thông tin một cách tổng quát mà chưa hề tính toán phân tích các thông tin đó. SSD hiện thực hai module dự đoán lớp và vị trí của vật thể cũng dựa trên các cảm hứng như vậy. Cụ thể việc dự đoán được thực hiện qua hai module chính là loc và conf.

Một cách tổng quát, việc nhận dạng và phát hiện vật thể bao gồm hai nhiệm vụ chính là phát hiện được sự xuất hiện của vật thể trong khung ảnh, bao gồm vị trí của vật và phần không gian mà vật chiếm ở trong không gian khung ảnh, nhiệm vụ thứ hai chính là, khi phát hiện được vị trí của vật thể thì còn phải dự đoán vật thể xuất hiện trong phần không gian đó là vật thể thuật lớp nào. Đây chính là bài toán localiztion và classification.

Mỗi nhiệm vụ trong bài toán nhận dạng vật thể được thực hiện bởi mỗi module trong hai module trên. Thông tin mà hai module nhận vào chính là các feature map với kích thước khác nhau và số lượng kênh khác nhau được trích xuất ra từ các lớp riêng biệt trong mô hình từ module vgg cho đến module extras.

Cụ thể, thông tin dự đoán ra từ module localization có kích thước là (n,8732,4). Trong đó, n chính là batch-size, tức là số lượng ảnh được sử dụng trong một batch. 8732 chính là số lượng anchor box dự đoán được từ mô hình, 4 chính là số lượng offset so với grounding box mà lớp dự đoán được. Từ các thông tin về offset này, nhóm sử dụng một hàm để giải mã thông tin về offset thành thông tin về toạ độ xác định của bounding box.

Thông tin dự đoán ra từ module classification có kích thước là (n,8732,4). Trong đó, n chính là batch-size, tức là số lượng ảnh được sử dụng trong một batch. 8732 chính là số lượng anchor box dự đoán từ mô hình, 4 chính là confidence score được tính toán cho mỗi class từ mô hình, confidence score được biểu diễn dưới dạng cross entropy, tức là index tại vị trí có confidence score lớn nhất chính là index của lớp có khả năng cao nhất là chính xác.

Số lượng anchor box của mô hình SSD của nhóm là 8732 bởi vì: Tại mỗi feature map, mỗi phần pixel của feature map chính là một dãy có kích thước là (batch\_size, channels, 1). Tức là phần thông tin tương ứng với phần không gian nào đó từ bức ảnh ban đầu, chính vì thể các anchor box sẽ được tạo tương ứng với từng vị trí của pixel ở trên feature map. Với mỗi source khác nhau được lấy ra từ các lớp layer, đều có kích thước của feature map khác nhau, dẫn đến số lượng pixel trong mỗi feature map cũng khác nhau. Và với mỗi source đó, mô hình ssd cũng tính các anchor box với số lượng khác nhau. Cụ thể trong mô hình của nhóm, tại source 1,5,6, mỗi pixel sẽ có 4 bounding box và các source khác thì tại mỗi pixel sẽ có 6 bouding box. Các cấu hình như này vẫn luôn là một câu hỏi mở, tuỳ thuộc vào người thiết kế, và có thể thay đổi theo mục đích của mô hình.

Lúc này số lượng bounding box là 38 \* 38 \* 4 + 19 \* 19 \* 6 +10 \* 10 \* 6 + 5 \* 5 \*6 +3 \* 3 \* 6 +1 \* 1 \* 4 = 8732.

Phần cấu hình chi tiết của mỗi layer nhóm hiện thực giống như paper thiết kế:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên | Kernel size | Padding | Stridge | Type |
| C3 | 3 \* 3 | 1 | 1 | Convolutional layer |
| C1 | 1 \* 1 | 0 | 1 | Convolutional layer |
| R |  |  |  | RELU |
| P | 2 \* 2 | 0 | 2 | Max pooling |

Hàm đánh giá:

Việc hiện thực mô hình và huấn luyện cần phải có một module để đánh giá mức độ chính xác của mô hình với tập dữ liệu và các thông số trong mô hình hiện tại. Thông số đánh giá này sử dụng để tối ưu các thông số trong mô hình để cho mô hình có khả năng học tốt hơn với bộ dữ liệu.

Trong bài toán object detection, việc đánh giá mức độ chính xác của việc nhận diện và phát hiện vật thể có thể thông qua nhiều thông số, ví dụ như là mAp, lượng mất mát khi dự đoán,…

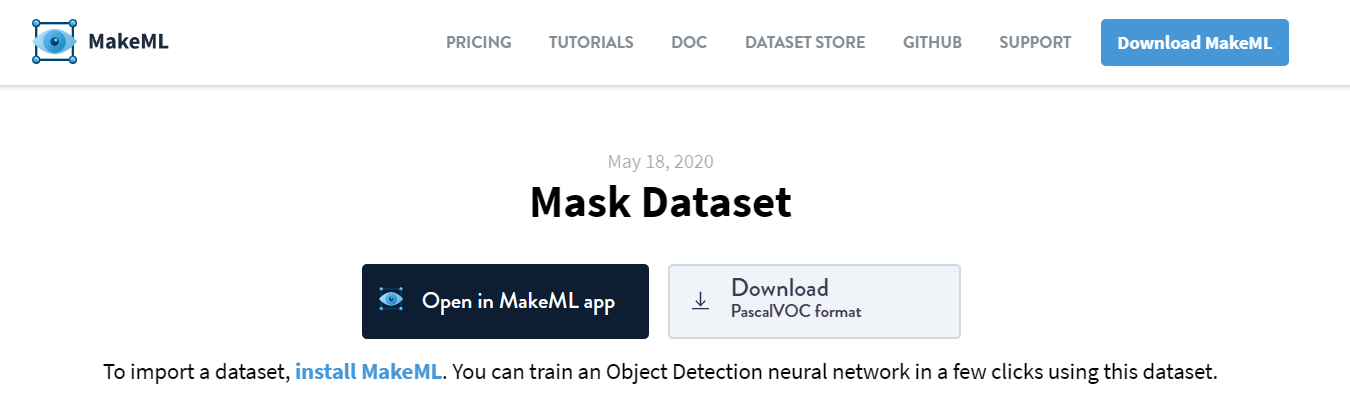
Trong phần mất mát (Loss), với bài toán object detection, có hai loại loss chính.

* Loss liên quan đến việc xác định vị trí vật thể: mất mát này thể hiện mức độ chênh lệch giữa vị trí của bounding box mà mô hình dự đoán được so với vị trí đúng của các bounding box. Có thể sử dụng các hàm đánh giá khoảng cách thông dụng như hàm loss normL1, lossnorm L2. Giá trị loss càng nhỏ, thì càng chứng tỏ mô hình dự đoán được vị trí của vật thể càng đúng với dữ liệu được huấn luyện.
* Loss liên quan đến việc nhận dạng vật thể trong bounding box: mất mát này thể hiện mức độ chính xác mô hình dự đoán các vật thể trong bounding box là chính xác đến mức nào. Giá trị của loss này càng nhỏ thì càng chứng tỏ mô hình dự đoán loại vật thể càng chính xác. Hàm loss có thể sử dụng một số cách đánh giá thông dụng, như trong mô hình SSD của tác giá, nhóm sử dụng loss cross entropy.

Kết hợp của hai hàm mất mát, tạo thành mất mát của mô hình object detection.

6. Tập dữ liệu huấn luyện:

Nguồn dữ liệu:



Dữ liệu gồm 853 bức ảnh được đánh định dạng theo tập dữ liệu VOC. Annotation là một file xml, với các key và field cần thiết như là kích thước của bức ảnh và số kênh màu trong bức ảnh. Các vật thể trong bức ảnh với thông tin vị trí của đối tượng đó, lớp mà vật thể đó thuộc về.

Mô hình của nhóm sẽ trích xuất thông tin từ các file annotation này về vị trí toạ độ của các đối tượng trong bức ảnh của như lớp mà vật thể thuộc về. Vị trí của vật thể được biểu diễn dưới dạng một bounding box với các thông tin về toạ độ của hai điểm ở góc trên bên trái và góc dưới bên phải.

Các vật thể trong tập dữ liệu được đánh nhãn thuộc vào 3 lớp chính là “with\_mask”, “without\_mask”, “mask\_weared\_incorrectly”, tương ứng với nhãn đeo khẩu trang, không đeo khẩu trang, hay đeo khẩu trang sai cách. Tổng số vật thể để học từ trong tập dữ liệu tầm 3-4000 vật thể, tuy vậy dataset gây ra khó khăn cho nhóm khi dữ liệu bị imbalance, cụ thể là số label được đánh nhãn “with\_mask” lớn hơn số label được đánh nhãn “without\_mask” vài lần, và lớn label “mask\_weared\_incorrectly”.

**Vấn đề với tập dữ liệu:**

Chính vì sự mất cân đối về số lượng của các lớp trong tập dữ liệu huấn luyện, làm cho mô hình mà nhóm hiện thực có xu hướng đoán chính xác các vật thể thuộc các lớp đa số như là lớp “with\_mask” và “without\_mask” mà bỏ qua các vật thể thuộc lớp còn lại. Điều này xảy ra trong quá trình tối ưu, nhằm nâng cao độ chính xác lúc huấn luyện mà mô hình có xu hướng dự đoán vật thể thuộc vào hai lớp đa số đó mà bỏ qua lớp thiểu số còn lại.

**Chèn ảnh số lượng dữ liệu của từng lớp.**

7. Hiện thực:

Trong phần hiện thực mô hình, nhóm hiện thực các module code theo quá trình bắt đầu chuẩn bị cho mô hình cho đến lúc kết thúc quá trình huấn luyện và kiểm thử. Theo đó, mỗi một mô hình object detection thường được hiện thực thông qua các giai đoạn:

* Tìm kiếm kiến trúc mô hình, dataset
* Chuẩn bị dữ liệu cho mô hình: bao gồm đọc dữ liệu ảnh, thông tin nhãn, tạo dataloader, xử lý ảnh cơ bản
* Hiện thực các module của mô hình.
* Hiện thực các hàm đánh giá, module huấn luyện mô hình.
* Huấn luyện mô hình với tập dataset đã chuẩn bị.
* Kiểm thử mô hình, sử dụng mô hình cho việc tham chiếu.

7.1 Tìm kiếm kiến trúc cho mô hình, và tìm kiếm dataset:

Quá trình tìm kiếm kiến trúc cho mô hình và tìm kiếm dataset cho mô hình kéo dài trong vài ngày, trong giai đoạn này, chủ yếu các thành viên trong nhóm tìm hiểu các mô hình có thể đáp ứng được các yêu cầu đề ra trong bài toán, và tìm kiếm các bộ dữ liệu phù hợp để huấn luyện. Kết quả mà nhóm chọn được sau khi phân tích và cân nhắc chính là mô hình single shot detection (SSD) và cụ thể là SSD300, cùng với bộ dữ liệu từ makeML như các phần trình bày trước đó. Đây cũng chính là giai đoạn mà nhóm lên ý tưởng cho việc hiện thực các module tiếp theo.

7.2 Chuẩn bị dữ liệu cho mô hình:

Sau khi có được dữ liệu cho việc huấn luyện thì đầu tiên nhóm sẽ hiện thực các module cần thiết cho mô hình, bao gồm việc trích xuất các thông tin từ các file trong dataset để tạo thành các nhãn cho object cần học.



Đầu tiên là hàm make\_datapath\_list, đây là một hàm đọc các file từ một đường dẫn cho trước và lấy ra thông tin về tên và đường dẫn của các file tương ứng trong dataset. Bộ dataset gồm 2 folder chính là phần images và phần annotations tương ứng chứa các file ảnh và các file metadata của các ảnh đó.

Sau khi có danh sách đường dẫn các file annotations, thì mô hình cần tách ra các thông tin cần quan tâm trong dataset, bao gồm thông tin của các object ở trong từng bức ảnh (toạ độ của bounding box và class của bounding box thuộc về).



Kết quả trả về của hàm trên chinh là danh sách các label được đánh nhãn, bao gồm toạ độ của bounding box được đánh theo định dạng của cặp toạ độ góc trên bên trái và góc dưới bên phải, cùng với nhãn của bounding box đó.

|  |  |
| --- | --- |
| Label ID | Label |
| 0 | Background |
| 1 | with\_mask |
| 2 | without\_mask |
| 3 | mask\_wear\_incorrect |

Sau khi có danh sách các đường dẫn file ảnh và thông tin về label của các object trong dataset thì nhóm tạo class MyDataset, đây là một class hiện thực kế thừa từ class Dataset của pytorch với mục đích sử dụng cho dataset custom mà nhóm đã chuẩn bị.



Khi sử dụng ảnh để train, do số lượng ảnh rất lớn cũng như cấu hình máy dùng để train có giới hạn nên việc truyền tất cả ảnh vào để huấn luyện trong một lần là một việc bất khả thi, vì số lượng tính toán diễn ra là một con số khổng lồ. Chính vì thể, cần phải tạo dataloader, như là một con trỏ có khả năng lặp qua bộ dataset, và ta có thể lấy ra từ dataset từng cụm dữ liệu với số lượng cụ thể và huấn luyện từ từ. Do ta cần kết hợp data của ảnh và data của file annotation thành một mẫu trong tập huấn luyện, nên cần phải có một hàm collate custom để thực hiện việc đó.



Sau khi có classDataset và hàm collate custom, thì nhóm có thể sử dụng dataloader hiện thực sẵn bởi pytorch với bộ dữ liệu và hàm custom vừa mới tạo.



Sau khi chuẩn bị các file cần thiết cho việc đọc ảnh, trích xuất thông tin thì nhóm đã có thể đọc các ảnh từ dataset với một batch\_size cụ thể và cho vào mô hình để tính toán. Tuy vậy, dữ liệu ảnh nếu không qua xử lý mà cho thẳng vào mô hình có thể khiến cho mô hình học một cách thiên vị cho các dữ liệu trong dataset, như vậy để tăng độ chính xác của mô hình, nhóm hiện thực thêm một module transform dữ liệu đầu vào để tạo ra dữ liệu sẵn sàng cho vào mô hình.



Tuỳ theo pha sử dụng dữ liệu là để huấn luyện hay để đánh giá và sử dụng thì các bước trong transform data cũng khác nhau. Cụ thể khi huấn luyện mô hình, thì dữ liệu được biển đổi qua tuần tự các bước:

* Chuyển giá trị của các pixel thành số thực để có thể thực hiện các phép tính toán với số thực.
* Chuyển giá trị tại từng pixel từ khoảng (0,255) thành (0,1).
* Chuyển đổi từng mảng màu thành màu sắc khác ngẫu nhiên
* Cắt ảnh.
* Đối xứng ảnh.
* Chỉnh kích thước của ảnh thành (300\*300) để cho vào mô hình

Khi sử dụng mô hình để tính toán, thì ảnh trước khi được cho vào chỉ cần chuyển về giá trị dưới dạng (0,1) và chuyển kích thước về 300 \* 300.

Sau khi đã chuẩn bị đầy đủ các thao các cần xử lý để thu được data sẵn sàng để cho vào mô hình học thì bước tiếp theo trong quá trình hiện thực chính là hiện thực mô hình, cụ thể là mô hình SSD.

7.3 Hiện thực các module của mô hình:



Module nằm đầu tiên trong mô hình chính là module vgg. Nhóm hiện module vgg với các thông số như tác giả đã cung cấp. Việc sử dụng lại cấu hình có sẵn cũng là một hướng tiếp cận tốt, khi cấu hình như trên đã thể hiện được độ hiệu quả tương đối cao sau khi thử nghiệm với một số cấu hình khác nhau. Ở đây nhóm sử dụng framework pytorch, cụ thể là các lớp convolutional và các lớp pool đã được hiện thực sẵn trong framework.



Module tiếp theo, là module extras. Cấu hình và công dụng của module này tương tự như paper mà nhóm tác giả của mô hình SSD đã công bố.

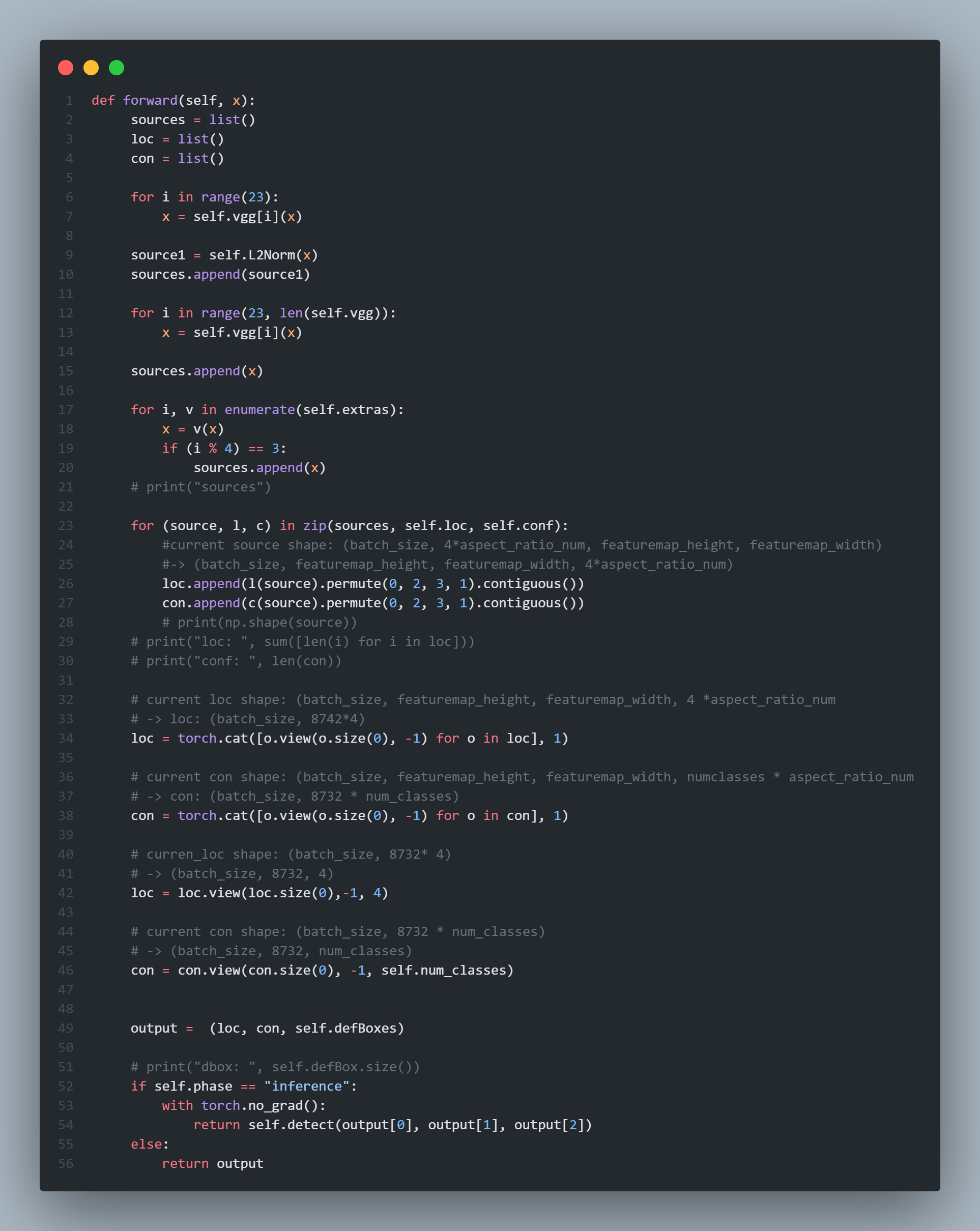


Cuối cùng là hai module localization predict và classification predict, hai module này có cấu hình tương tự nhau về số kênh cũng như thông tin đầu vào, chỉ khác nhau ở kích thước của thông tin đầu ra, cho nên nhóm quyết định gộp hai module vào một hàm để cho code được rõ ràng và gọn gàng hơn.

Sau khi đã có các module, thì muốn tạo được một model SSD, thì cần phải kết nối các module lại với nhau. Từ đó, nhóm tạo class SSD:



Mô hình này muốn sử dụng trong module train của pytorch thì cần phải hiện thực một số hàm cần thiết, trong đó có hàm forward. Hàm forward dùng để biểu diễn quá trình tính toán theo tuần tự của dữ liệu khi cho vào mô hình, đi qua các module và tính toán ra được dữ liệu quá.



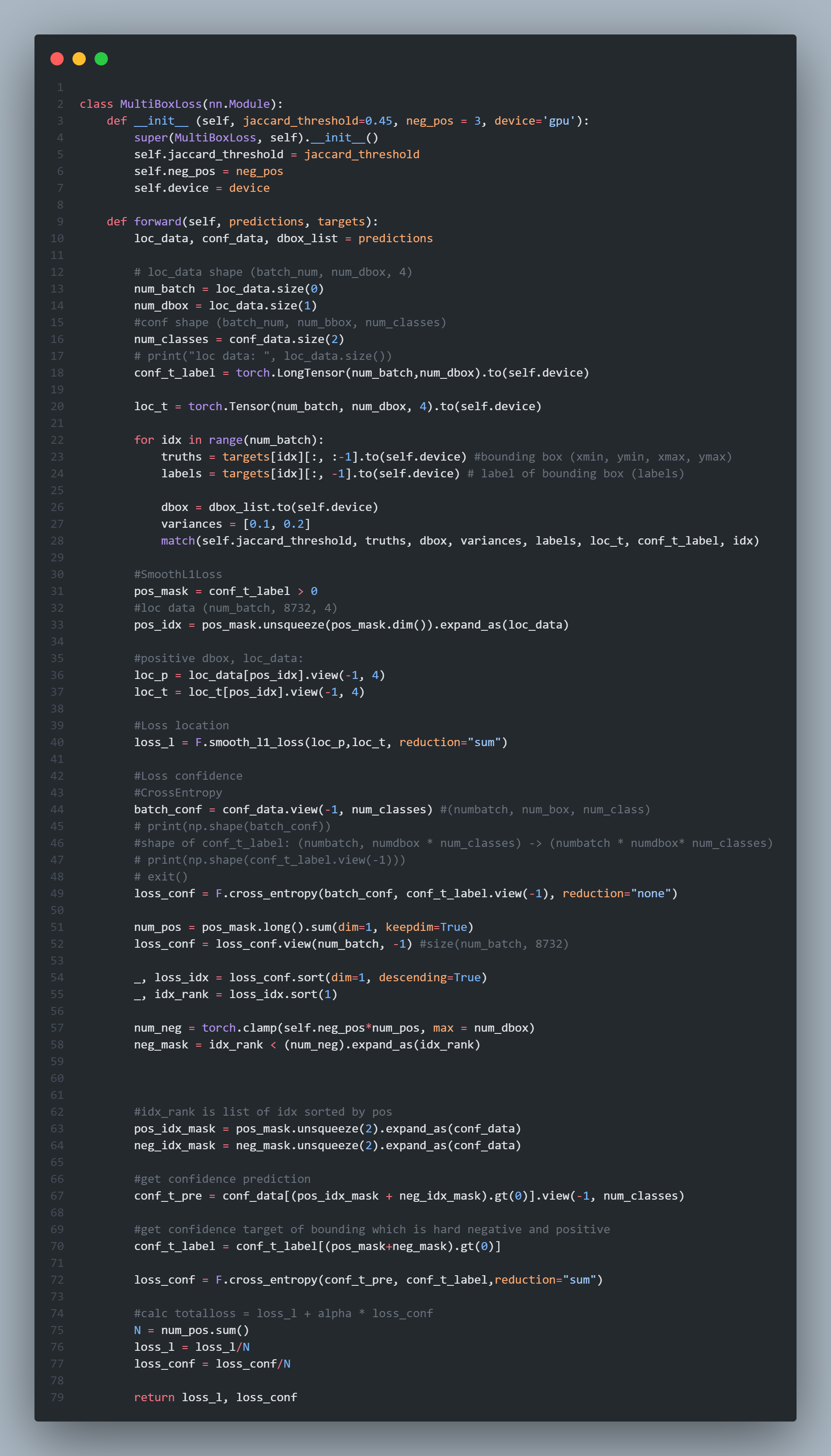
Bên cạnh đó còn cần các module cần thiết để giải mã các thông tin dự đoán được thành thông tin xác định để hiện thị thông tin cho người dùng. Bao gồm các hàm detect, decode\_bounding. Phần hiện thực cũng tương đối đơn giản, nên có thể tham khảo trong source code được đính kèm theo bài báo cáo này.

Ở source1, trước khi feature map được cho vào các module loc và conf, thì dữ liệu được chuẩn hoá qua một lớp L2Norm, nhằm làm cho dữ liệu được chuẩn hoá thành dạng chuẩn, độ chênh lệch của giá trị trong bộ dữ liệu nhỏ hơn, giúp cho các sự thay đổi trên bộ dữ liệu ảnh hưởng lớn đến kết quả dự đoán.



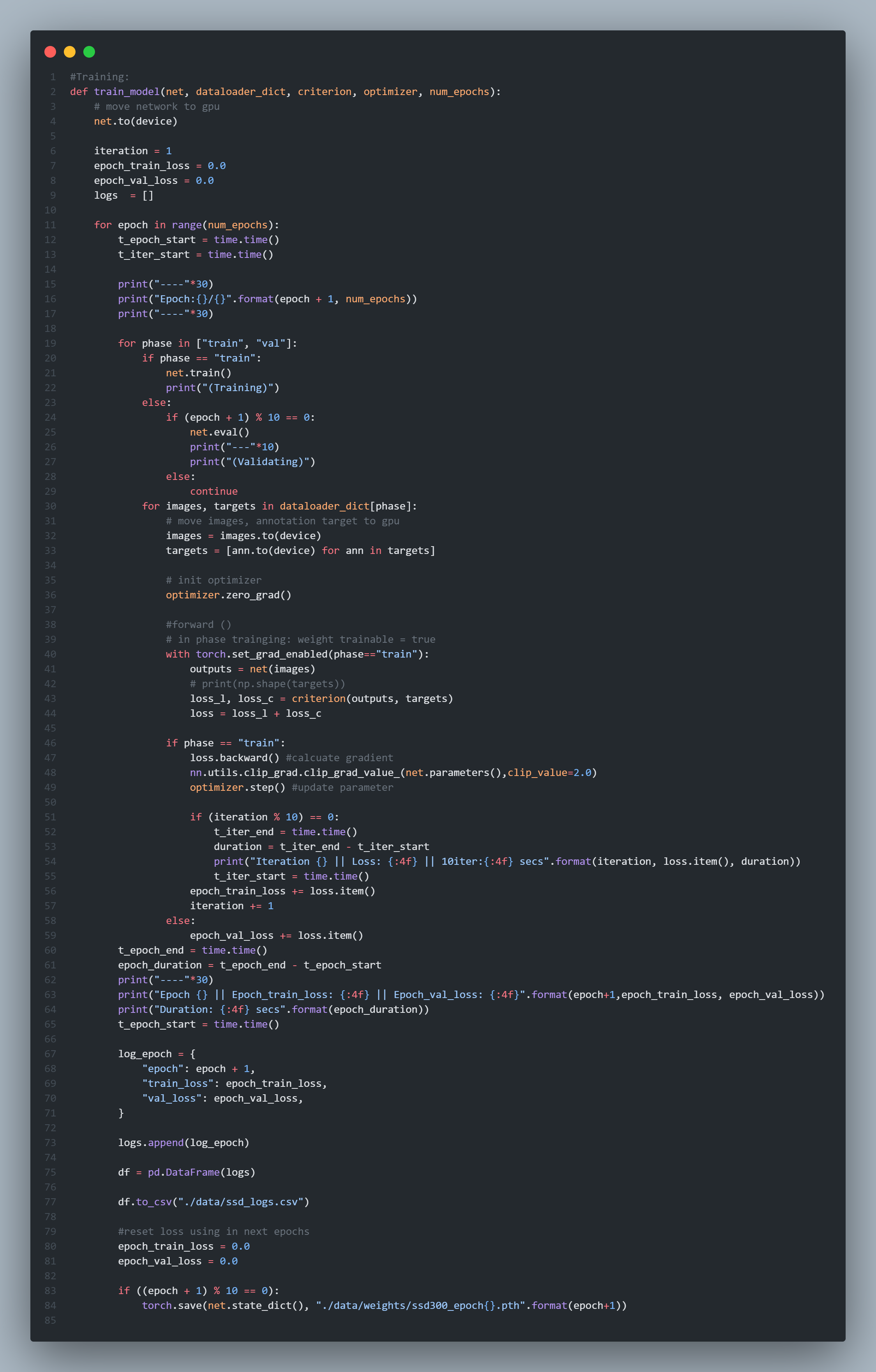
Hiện thực các hàm đánh giá, module huấn luyện mô hình.

Mô hình đưa vào huấn luyện cần có một hàm để đánh giá độ tốt của mô hình hiện, và tối ưu mô hình trong những lần huấn luyện tiếp theo. Việc đánh giá mức độ mất mát của mô hình được thực hiện thông qua module multiboxloss mà nhóm đã thực hiện.



Sau khi nhận vào kết quả của mô hình dự đoán, và kết quả đúng mà mô hình cần phải dự đoán, thì module sẽ tính toán và trả về hai lượng mất mát tương tự như phần kiến thức đã trình bày trước đó, chính là mất mát về việc xác định toạ đọ của bounding, và mất mát trong việc dự đoán loại vật thể. Hai giá trị này sau đó sẽ được cộng với nhau theo trọng số để được mất mát của mô hình.

Sau khi có đầy đủ các module, thì module cuối cùng cần hiện thực hính là module train:



Nhóm sử dụng cách train theo epoch với batchsize cụ thể, bằng cách tiếp cận này có thể giảm thiểu số lượng tính toán phải tính trong mỗi lần duyệt, tuy vậy muốn huấn luyện hết trong tập dữ liệu cần phải duyệt nhiều lần hơn. Điều đó giúp cho các máy có cấu hình thấp có thể đánh đối thời gian huấn luyện mà vẫn có thể có khả năng huấn luyện tập dữ liệu.

* Huấn luyện mô hình với tập dataset đã chuẩn bị.

Nhóm thực hiện huấn luyện một số lần với số lượng epoch khác nhau, sau một số lần thử nghiệm và sử lại thì nhóm nhận thấy việc huấn luyện với số lương epochs là 30 giúp cho mô hình học được tốt nhất, trong khi số lượng epoch lớn hơn làm cho mô hình bị overfitting.

Chính vì hạn chế trong tập dữ liệu mà nhóm thu thập được cho nên kết quả huấn luyện rất thiên vị về các lớp đa số trong tập dữ liệu.

* Kiểm thử mô hình, sử dụng mô hình cho việc tham chiếu.

Việc kiểm thử và sử dụng mô hình, nhóm hiện thực hai module tương ứng cho việc kiểm thử với ảnh và với live-video từ webcam. Phần hiện thực có thể tham khảo trong source code.

III. Tổng kết:

a. Điểm đạt được:

Sau một thời gian hiện thực và cải thiện mô hình, nhóm có một số đánh giá về quá trình hiện thực đồ án ở trên nhiều mặt.

Ở phía đội nhóm hiện thực:

* Thành viên trong nhóm, cụ thể là bản thân người hiện thực tiếp thu được rất nhiều kiến thức mới, đặc biệt là trong bài toán object detection.
* Học được khả năng giải quyết vấn đề, kiến trúc mô hình để cho quá trình thực hiện được nhất quán và suôn sẻ.

Về mặt kết quả đạt được:

* Mô hình nhận dạng được như dự đoán, có khả năng nhận dạng được một cách khá chính xác các vật thể thuộc các lớp đa số.
* Tốc độ nhận diện khá nhanh khi nhóm chọn được kiến trúc hợp lý, cân bằng được giữa khối lượng tính toán và độ chính xác đạt được.
* Có một vài module trong việc đánh giá mô hình như tính mAp, vẽ biểu đồ theo dõi quá trình train, nhóm vẫn chưa thực hiện được do giới hạn về nhân lực và thời gian.

b. Hạn chế, hướng khắc phục:

Ở phía nhóm hiện thực:

* Các thành viên trong nhóm không liên kết được với nhau, dẫn đến khối lượng công việc được phân chia không đồng đều.
* Tài nguyên sử dụng cho quá trình thực hiện còn hạn chế về nhiều mặt: sức mạnh phần cứng cho việc hiện thực và huấn luyện mô hình, dữ liệu có sẵn để huấn luyện mô hình.

Về mặt kết quả đạt được:

Hạn chế: Mô hình nhận diện được các trường hợp dễ, có độ khó thấp rất tốt, nhưng các trường hợp khó như là khuôn mặt khi nhận diện bị che khuất một số phần thì khó có khả năng nhận diện được, các trường hợp khó nhận diện như là đeo khẩu trang sai cách, khuông mặt trong đám đông xuất hiện mờ.

Suy đoán nguyên nhân: Nhóm nhận định, vấn đề đến từ việc đánh nhãn cho dữ liệu, và độ đa dạng của dữ liệu, khi mà dữ liệu nhóm tìm được và sử dụng huấn luyện bị mất cân bằng về các lớp đa số, dẫn đến kết quả dự đoán bị thiên vị về các lớp đa số đó nhằm giảm lượng mất mát trong quá trình huấn luyện.

Hướng giải quyết: Nhóm đề xuất một số hướng giải quyết cho vấn đề gặp phải bởi mô hình:

* Làm đa dạng dữ liệu, tăng số lượng dữ liệu của các lớp thiểu số, hoặc giảm số lượng của các lớp đa số mà vẫn đảm bảo các trường hợp của dữ liệu đủ đa dạng để giúp máy có thể nhận biệt các trường hợp thông thường.
* Thay đổi bộ dữ liệu, tìm kiếm bộ dữ liệu với độ phân phối giữa các nhóm đồng đều hơn.
* Thay đổi hàm đánh giá với mất mát cho từng nhóm riêng biệt, tăng lượng mất mát với các đối tượng được dự đoán thiểu số, nhằm làm cho mô hình không bỏ qua việc dự đoán sai các lớp thiểu số này.

IV. Lời kết:

Trải qua quá trình thực hiện đồ án, dù gặp nhiều khó khăn nhưng nhìn chung nhóm vẫn hoàn thành được tương đối mục tiêu đề ra đó chính là tìm hiểu được kiến thức cần thiết cho bài toán object detection và hiện được mô hình SSD để demo cho thầy. Em tin tưởng, trong tương lai, những trải nghiệm mà nhóm trải qua sẽ là những kinh nghiệm quý báu cho việc xử lý các vấn đề khác trong lĩnh vực thi giác máy tính.

Cuối cùng nhóm xin cám ơn thầy đã đồng hành cùng nhóm và giúp đỡ cho nhóm những lúc gặp khó khăn. Sự giúp đỡ của thầy chính là động lực để nhóm có thể hoàn thành được đồ án này.