Object detection using CNNs

Facemasks detection

Lilian Bour – 21906722

Pinned version of the colab notebook :
https://colab.research.google.com/drive/1c8QOUPuMUF36WfE-A6c6Mboxpkh-wCB9?usp=sharing

1. Presentation

In this project, the goal is to detect if people are wearing facemasks or not. We have 852 images, with 852 xml files containing the bounding boxes. The boxes are labeled as “With Mask”, “Without mask” or “Unsure”.

We’ll use a Faster R-CNN pre-trained on the MS-COCO dataset, with a new classifier head to fir our data.

2. Questions

Answer 1: The format used is (x,y) of top left and bottom right corners because we don't have any values about width and height. Moreover, we have the x and y values, they will be used to determine the 4 corners of the bounding box.

Answer 2: For example, we can see on the 5th image, that the person in the background is blurred because of the depth of field. As a result, the person has not been labeled as a person without mask. In image 7, the person in the background has been identified as a person without mask but it's very small, so it's difficult to say. For the 11th image, we can see on the top right corner, a green bounding box, but it doesn't seem to contain anything.
We can conclude that the model will probably struggle with people in the background because it is unclear, and even we as humans struggle to say if the person is wearing a mask or not. Furthermore, it may be difficult for the model to differentiate a poorly worn mask from a well-worn mask because most of the time, it is because we can see the nose, it would mean that the model will have to "recognize" the noses, which may be difficult if the person is not in the foreground for instance.

Answer 3: We have 852 images; this seems low compared to other datasets such as CIFAR10. We could either add a Fine-tuning step or use Data augmentation to overcome this issue. Here we'll use data augmentation with random horizontal flips and random rotations. Furthermore, the data will be split as follows :

- Train : 60%
- Validation : 20%
- Test : 20%

Answer 4: As we can see on the figure 1, there is a lot more data labeled as 'with masks' than 'without masks' or 'unsure'. It means that we may have a bias toward the first label because it's more prominent.

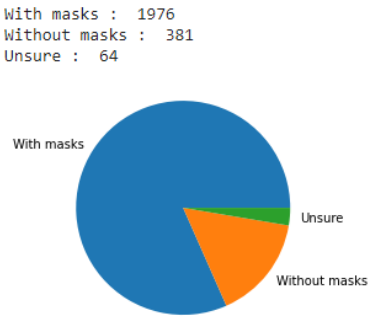

Figure 1 : label repartition

Answer 5: The learning rate is commonly set to 0.01 and the momentum at 0.9. The number of epochs is set to a great number. To select the best model, we will save only the model, if the validation

loss is lesser than the previous lesser validation loss.

We can see on the figure 2 that the best validation loss is obtained at five epochs, with a loss approximately equal to 0.14.

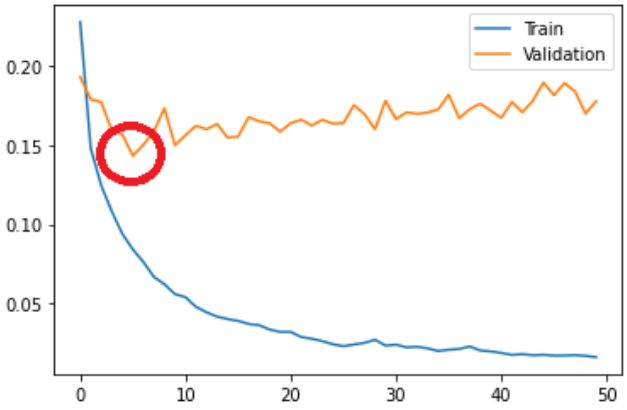

Figure 2 : losses variation

3. Discussion

a. Improvements

To select the best model, with the best number of epochs we set the number of epochs to 100. During the process, the model with the lesser validation loss will be saved and loaded for the training part. This ensures that we get automatically the best model.

We can use two different methods during the test to evaluate our predictions. This will determine the number of “good boxes”. The first method consists of checking if each predicted corner is inside an interval centered on the ground truth corner.

Is x inside [ground truth corner – threshold, ground truth corner + threshold] ?

The second method will check if the sum of the absolute value of the differences between each corner for the ground truth and the predicted data is lesser than a threshold.

Is ||ctxmin-cpxmin||+||ctymin-cpymin||+||ctxmax-cpxmax||+||ctymax-cpymax||<threshold ?

- ct : corner from ground truth
- cp : corner from predicted data

Furthermore, using a threshold does not seem to be the best method. Indeed, it is possible to assign a box to more than one class. If the threshold is set, for instance to 10 000, we will get more “good boxes” than predicted boxes. To avoid this, we could use lists and remove duplicates from one list in the first place, then duplicate between list should be removed, the one with the higher score should be kept.

However, this has not been done in the project, but this error has somewhat been controlled. We can roughly guess a correct threshold by looking at the real number of boxes and comparing it to the number of “good boxes” that we’ll get with a certain threshold.
For example (this test has been realized with only one epoch) :

- Threshold = 3, we get 648 good boxes for 820 real boxes, the difference is 172
- Threshold = 4, we get 850 good boxes for 820 real boxes, the difference is 30
- And we get more good boxes with a threshold equal to 5.

The threshold is then set to 4.

b. Results

We check the label repartition for the test set, and as we can see on the figure 3, it follows the same repartition as the train set.

```
With masks :  680
Without masks :  127
Unsure :  19
```

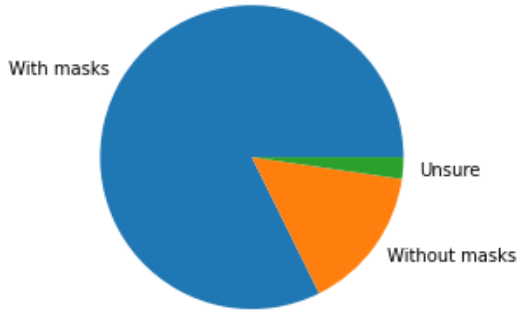

Figure 3 : test label repartition

The results are shown on the figure 4.

```
True boxes :  1363
'Good boxes' :  826
TEST LOSS :  tensor(0.3873)
CONFUSION MATRIX
WIMASK :  562                 16                  8
WOMASK :  14                  86                  11
UNSURE :  78                  58                  12
RECALL :  0.8593272171253823  0.5375              0.3870967741935484
PRECIS :  0.9590443686006825  0.7747747747747747  0.08108108108108109
F1-SCO :  0.9064516129032257  0.6346863468634686  0.1340782122905028
ACCURA :  0.7810650887573964
```

Figure 4 : results

Globally, the accuracy is equal to 78% which is pretty good. As tough, the results are very good for “With mask” as opposed to “Unsure”. We had a lot more data for the first category, hence it created a bias toward this category. Nevertheless, we still get good results for the category “Without mask” knowing there was still a lot more of “With mask” than “Without mask”.

To go further, we can see that 78 boxes have been classified as “With mask” instead of “Unsure” and 58 as “Without mask” instead of “Unsure. This means that the bias is nearly as much as important for the “Without mask” (39% are miss-classified) category as the “With mask” (52% miss-classified). Indeed, it is “only” a 13% difference. It isn’t much considering that the well-classified for “Unsure” only represent 8%.

To conclude, these results are pretty good, and could be used for real-world problems because the main task is to say if people are wearing a mask or not. Although, if we look at all the predicted boxes, it seems that sometimes, more boxes are predicted it could be interesting to display the predicted boxes and the true boxes on an image to compare them and try to see if we could improve this by adding layers or adding some pre-processing step.