LAB2: Training a Detectron2 Model on a Custom Dataset Computer Vision 2

Mory Moussou Kouklibaly Traore

AMMI/AIMS April 2021

1 Introduction

The purpose of this lab2 is to train a D2 model on a custom datasets called **nuts**. One uses an existing model pretrained on Coco dataset for initialization and the second use a pre-trained Imagenet weights called ININit. The idea was here to to train those models with a number of iterations, visualize the train and prediction curves for the both. An other task was also evaluate the perfomance of the models using AP metric implemented in COCOAP.

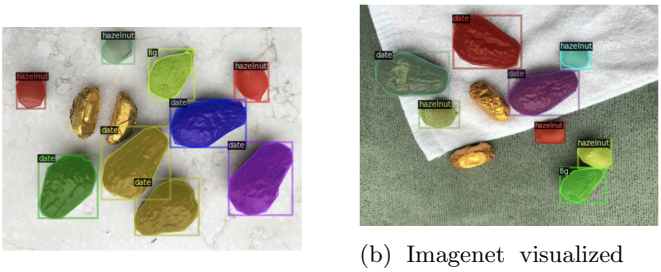
2 Part A: Download, Register Datasets, Visualization of randomly selected samples

2.1 Download, Register

In this part we downloaded and registered the datasets. For the registration, it is necessary because When we use a custom data set with Detectron2, we need to register it in COCO dataset. This will help D2 to have accesson the dataset. In this lab, we used a custom nuts dataset. It has 18 images and three categories of nuts: date, fig and hazelnut. We registred the entire training and validation using the COCO register functionality, which helps to save a dataset in the COCO annotation format.And The training data was registered as $\text{nuts}_t rainandtheval data as nuts_v al$.

2.2 Visualization of randomly selected samples

To verify the data loading is correct, let's visualize the annotations of randomly selected samples in the training set. This will allow to view their annotations.



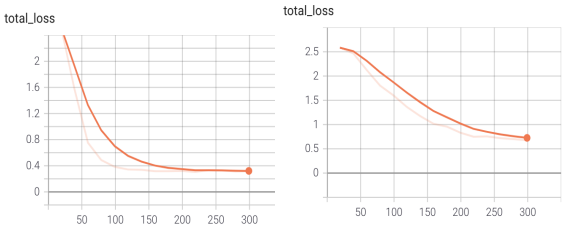
(a) CoCoinit visualized predictions (b) Imagenet visualized predictions

Figure 1: Segmentation and class name for two random samples from the training

3 Part B: Model Initialization and Training Schedule

In this part we have the visualization the curves of total training loss for the two model.

We notice that the curves for total loss of the both models are decreasing. Here we proceeded to training. We trained a Mask R-CNN model, with a ResNet-50-FPN backbone. the model was initialize with two initialization schemes. In (a) we have the COCOinit model. His initialization is done by using COCO dataset. And in (b) the ININit model was initialized by using the ImageNet pre-trained model weights. When we look the curve we see that COCOinit reaches a total loss less than 0.4 after 300 iterations, which is less than the final loss of the ININit model which is greater than to 0.6. So the loss for the Imagenet model that converges more slowly than the one for COCOinit model.



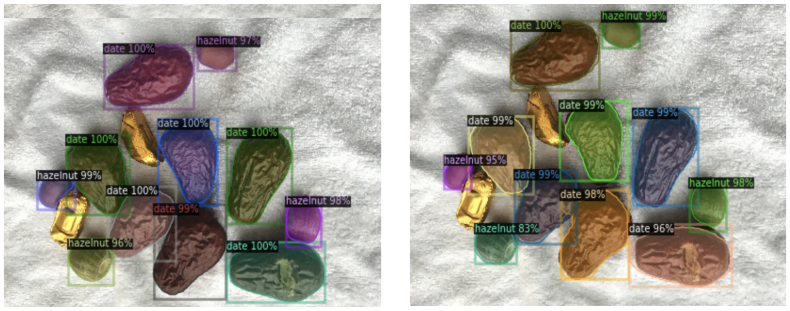
(a) CoCoinit (b) Imagenet

Figure 2: Curves of total training loss for the two model

4 Part C: Inference and Evaluation of the Trained Model

4.1 Visualizations

Here we Visualized predictions of both trained models, on the images of the val set.



(a) CoCoinit visualized predictions (b) Imagenet visualized predictions

Figure 3: Curves of total training loss for the two model

Here we realized that for COCOinit model all nuts were predicted with a higher probability than the ININit model. Also we had for the ININit model one nut's accuracy is 83. This is smaller than all accuracy for the model COCOinit in our case. So this showed by observations that COCOinit learned how to segment and predict them more accurately than ININit.

4.2 Evaluation of performance

Here we evaluated the performance of both models using AP metric implemented in COCO API.

$$AP\_Category\_Cocoinit = \begin{bmatrix} \textbf{Category} & date & fig & hazelnut \\ \textbf{Bbox AP} & 87.534 & 87.030 & 84.954 \\ \textbf{Segm AP} & 98.8 & 91.663 & 92.772 \end{bmatrix}$$

$$\text{AP\_Category\_ININit} = \begin{bmatrix} \textbf{Category} & date & fig & hazelnut \\ \textbf{Bbox AP} & 79.003 & 59.389 & 64.542 \\ \textbf{Segm AP} & 84.234 & 62.657 & 80.187 \end{bmatrix}$$

$$\text{AP\_Overall\_average\_precision} = \begin{bmatrix} \textbf{Models} & COCOinit & ININit \\ \textbf{Bbox AP} & 86.506 & 67.645 \\ \textbf{Segm AP} & 94.412 & 75.693 \end{bmatrix}$$

Here we see that ININit model has Average Precision smaller than COCOinit for the prediction of the bounding box and the mask. This is the same also for the overall average precision. This allow us to say that The performance of The the pre-trained model on COCO dataset is better than the performance the pre-trained model on ImageNet dataset.

5 Conclusion

In this lab2, we learned more in detectron2. We unsderstand how to prepare and train on a custom dataset of nuts, The comparison between the perfomance of the pre-trained model on ImageNet dataset and pre-trained model on COCO dataset.