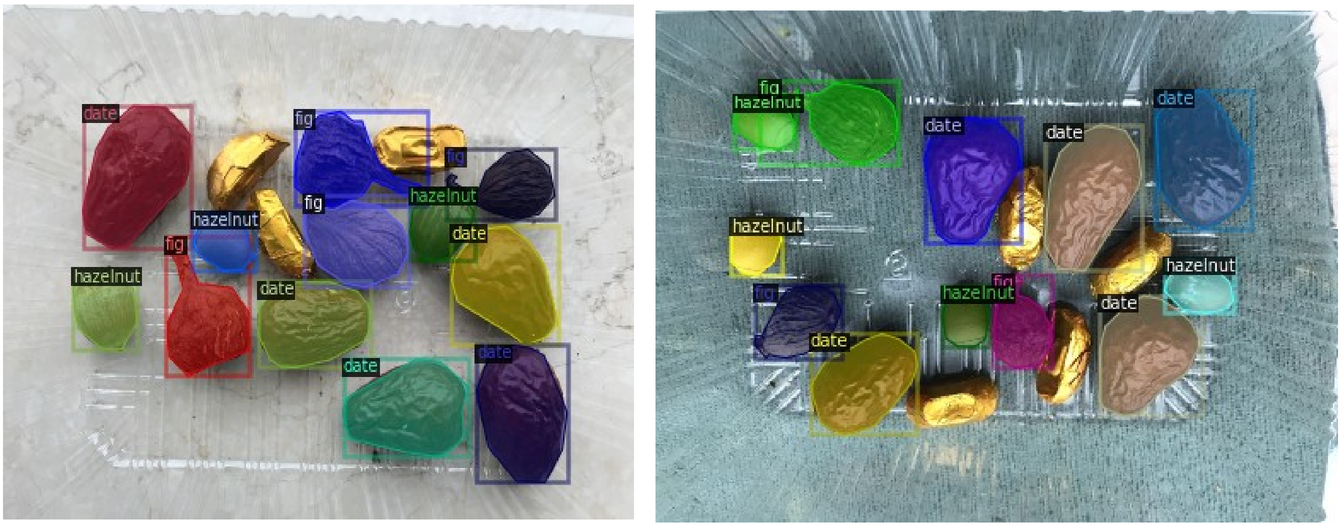
Training a D2 Model on a Custom Dataset

Introduction: This lab is concern with exploring Detectron2 in-deptly by preparing a custom data set and training on it. As done in Lab1, we need to have all the necessary requirement met before anything else can be done. For this lab, the data set consist of 3 classes 3 classes: ***date***, ***fig***, and ***hazelnut***, and which is annotated with instance masks. The data contains an image directory, a train.json file containing the train annotations in COCO format, a val.json file that contains the val annotations in COCO format.

Methodology: (Part A) involves the data format, preparation and registration. To register the instances, we supply the json file, the images and then give a name for the instances. (***nuts_train and nuts_val***). After that is done, we need to visualize to verify that the data loading is correct, this is achieved with the help of the ***MetadataCatalog***. The next step (Part B) was Model Initialization and Training Schedule, the Mask R-CNN model, with a ResNet50 FPN backbone was initialized using 2 different schemes, ***COCO dataset*** and ***ImageNet weights***, the training for both models were 300 iterations, a start learning rate of 0.02, 2 images per batch, and 128 regions per batch, afterwards we Visualize the training curves for both models in tensor-board. Finally, (Part C) was all about the Inference and evaluation of the Trained Model, we visualize predictions of both trained models, on the images of the ***nuts_val,*** and then evaluate the performance of both models using ***AP metric*** implemented in COCO API.

Implementations Results and Interpretations.

1. **visualizations of the training annotations from Part A.**



The above figures are 2 randomly selected samples in the training set. The visualization was done using the MetadataCatalog. From the figure, we observed that each of the figues were annotated.

2. Comparing the training curves visualized by tensorboard for both *COCOinit* and *Ininit*

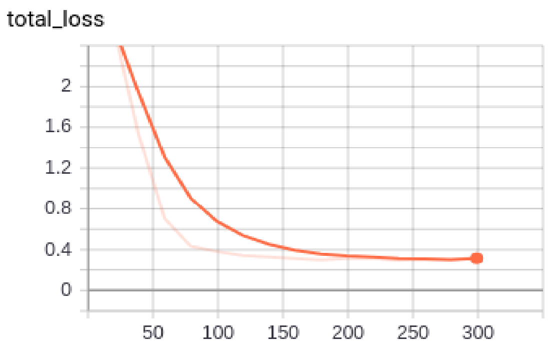


Figure 1: COCO dataset training loss

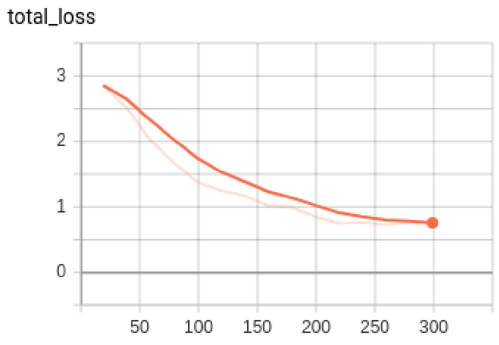


Figure 2: ImageNet training loss

The total training loss for the COCO data set was 0.311 while that of ImageNet was 0.712, also, it is obvious that the training loss of COCO data set is lower and also converging faster than ImageNet.

3. Comparing the visualize predictions on the val set for both `models.`

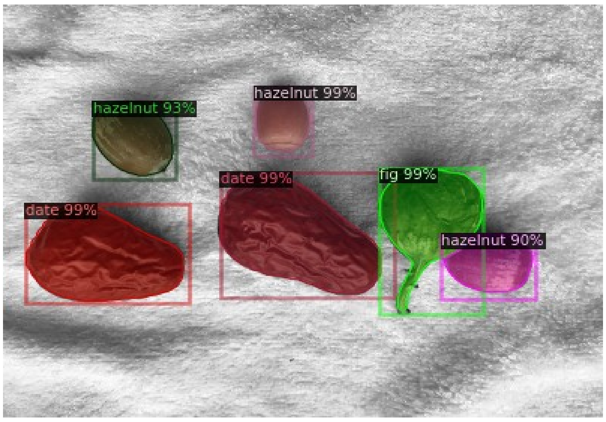


Figure 3: COCO Model prediction

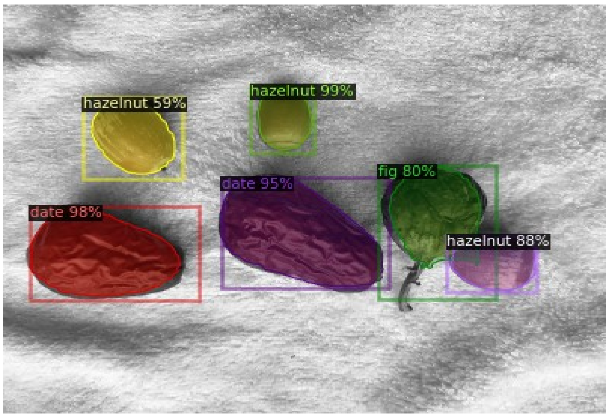


Figure 4: ImageNet Model prediction

Observation: It is observed that the prediction score from the COCO model is much more higher than that of the ImageNet model, for instance, the COCO model predicted hazelnut with 93% score while that of ImageNet was 59% on that same hazelnut.

Table 1: Shows the evaluation results of both models.

Models	**Results**	**AP**	**AP-date**	**AP-fig**	**AP-haz**	**AP50**	**AP75**	**APl**	**APm**	**APs**
COCO	bbox	81.04	83.19	80.40	79.53	100.00	93.73	87.03	75.30	nan
	segm	93.10	97.20	90.99	91.12	100.00	100.00	95.28	89.99	nan
Image Net	bbox	67.26	75.27	58.65	67.85	100.00	83.15	61.44	65.88	nan
	segm	79.85	85.14	75.06	79.35	100.00	97.53	77.49	79.72	nan

From the table above, we can deduce that the COCO models performed better than it's counterpart, this is observed in the AP's for both the bbox ans segmentation. This is because the COCO dataset is comprised of 2.5 million labeled instances in 382, 000 images, while ImageNet dataset has 100,000 images across 200 classes. Hence, the COCO data learns more.