NOTES 2: DATATHON COMPETITION

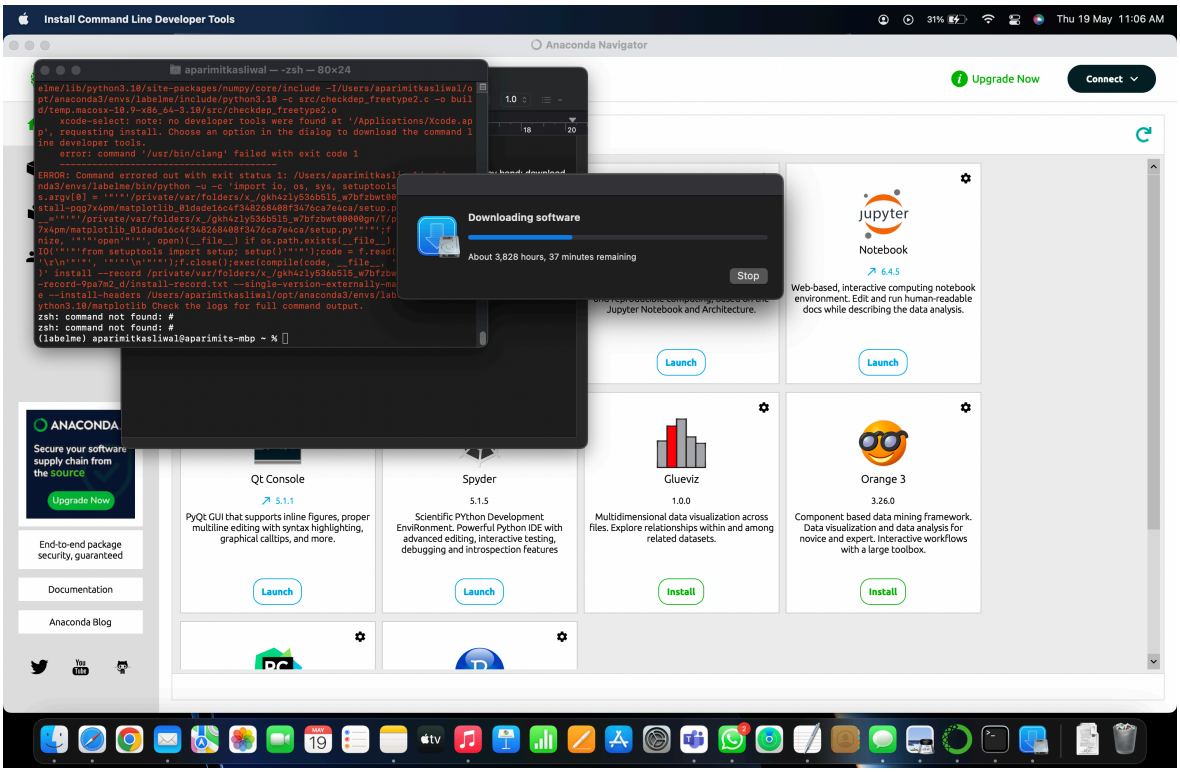
- CVAT: Computer Vision Annotation Tool

Sig up on CVAT; upload images that are to be labelled; identify / label by drawing boxes by hand; download in PASCAL VOC 1.1; now upload both images and annotations to ROBOFLOW; it will automatically relate the annotations to the corresponding images

Too tedious to be done for multiple images since the process is more or less manual. We have been specifically asked in the instruction sheet to kinda proceed with the “State-of-the-art” labelling processes, not sure if this would do the job for us

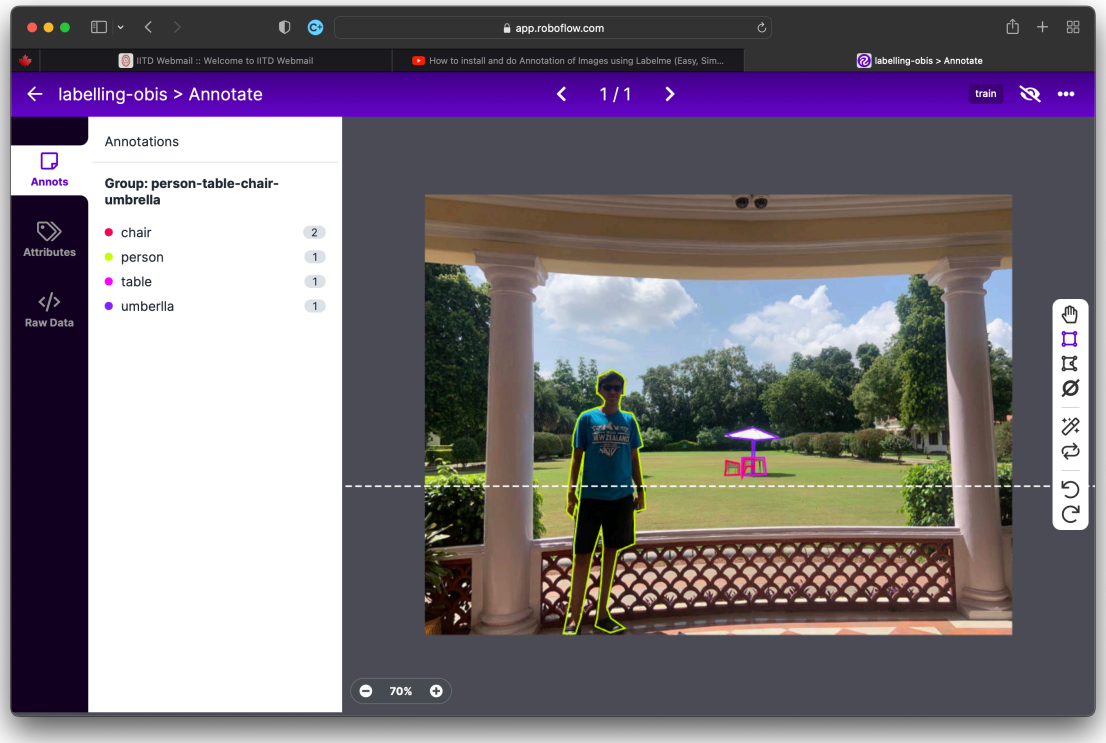
- LabelMe

(Initially looked!) Computationally expensive to install; refer to the screenshot showing an exacerbated timeline just for installation

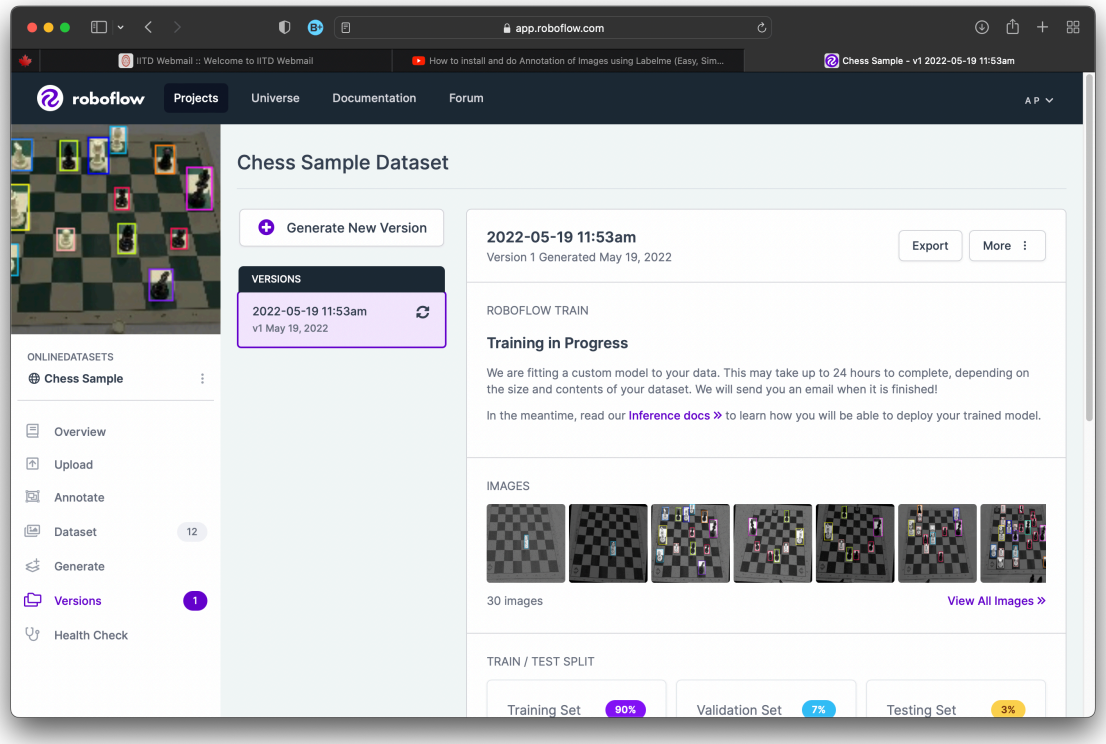


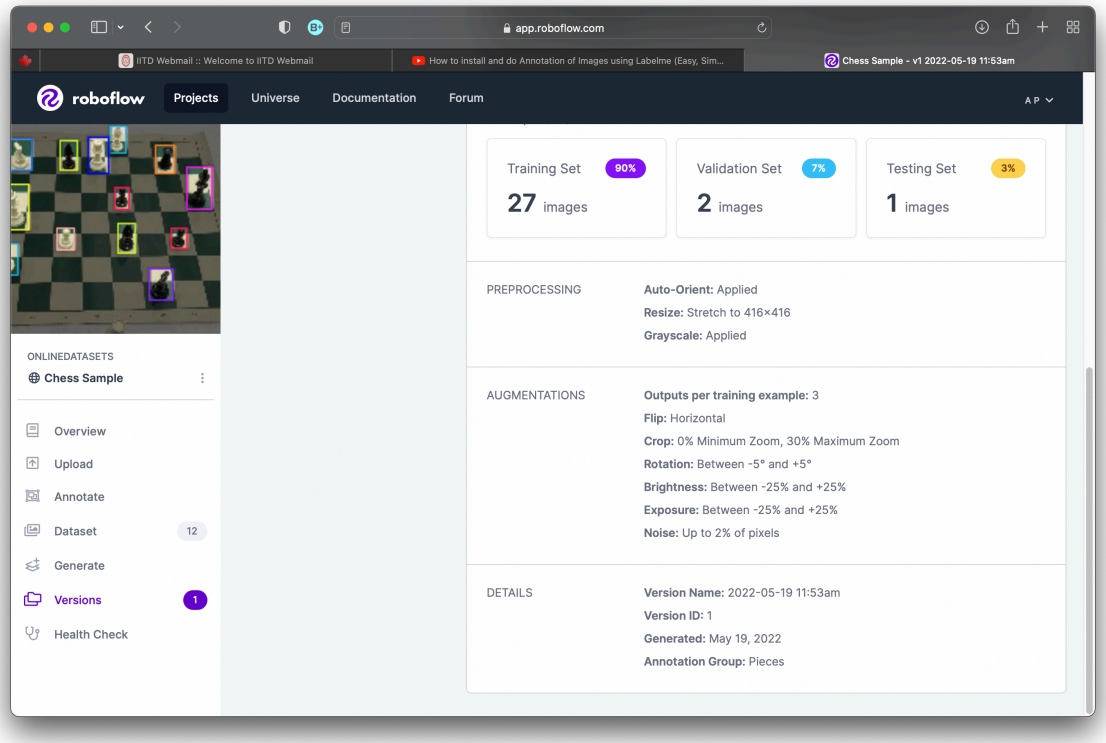
Finally, was able to install LabelMe; annotated an image with 4 objects (mentioned below); the annotated / labelled file was saved as a .json file which when clubbed with the same image using an appropriate software like ROBOFLOW will be able to generate the labelled image

Did the above for a particular image with annotated labels as: person, umbrella, chair, table; here’s what it looks like



Furthermore, datasets can also be labelled at the ROBOFLOW UI as well. An example I carried out was for an online available dataset of images for chess boards in which the end-goal was to perform object detection for “chess-pieces”.



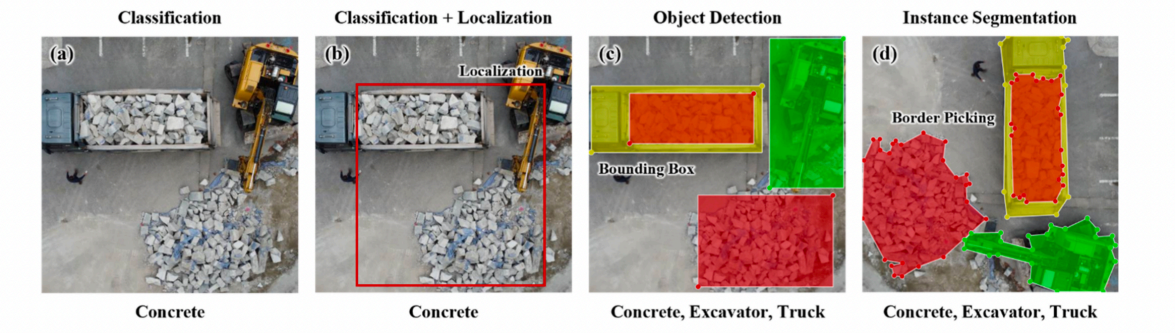


Auto-trained a model for the 12 images of chess-boards with different arrangement of pieces. Various pre-processing and efficiency-enhancement tools are included in the ROBOFLOW environment including customising test-train-validation split, augmentation techniques and quantifying the degree to which one needs them, re-sizing (basically down-sizing) the input images, grayscale, brightness, exposure and noise variation etc.

- The model achieved better than decent accuracy. Also the model was able to figure out the individual pieces when a completely fresh image of a chess board (taken from google images) was tested.

From Sunwoo et.al.:

- "Thus, in this study, we developed an AI model by transfer learning YOLACT that classifies construction waste with the goal of distinguishing construction waste accurately and increasing the recycling rate."
- mAP: parameter to quantify model performance a.k.a. Mean Average Precision
- They seem to have emphasised more on "labelling the raw data better" than merely "increasing the amount of training data to increase accuracy".
- Fact: FCN (Fully Convolutional Network) does not have image size limits
- Methodology followed: AI Model Selection, Data Pre-Processing, Data Labelling, Data Quality Verification, Learning-Model Quality Verification



- Their dataset comprised 599 images (512 x 512 pixels, within 100KB each): board, brick, concrete, mixed waste, lumber
- Model parameters: batch size - 16, epochs - 10,000
- They try to justify by giving two hypothesis for why the mAP for the un-professional guy comes out to be better than that for the professional one, which is kinda counter-intuitive. FIRST: labelling methodology of the un-professional guy was quite similar to that of the one followed by YOLACT on COCO dataset. SECOND: non-professionals tend to label a much larger area, thereby increasing the parts that overlapped with the predicted mask, which led to an overall surge in the mAP
- They bring up the FIRST hypothesis (mentioned above) in the discussion section of the paper; there were differences in the type of objects that were meant to be recognised with the models (were quite regularly / typically shaped for COCO, not so much with the construction waste)
- Augmentations applied: noise and hue
- Augmentations evaluated using: FID (Frechet Inception Distance)

CONCLUSIONS

- Open source labelling platforms are available and are handy, though not sure if we would want to label all images by hand
- Sunwoo et.al. show the degree of sophistication we can incorporate into our model, but before that, we need to get started on building standard ones and seeing which works best for us
- There are slight differences in the understanding of instance segmentation, semantic segmentation and object detection which must be figured out to confirm the methodology we want to focus on (it probably is semantic segmentation)