**How to run YOLOv7 on NVIDIA DGX**

* **Log in to Kubeflow** at [http://192.168.12.1:31380](http://192.168.12.1:31380/) using your credentials on the FIREFOX web browser
* **Start a new notebook server**
  + PyTorch Image
  + 8 CPU, 16 Ram
  + 1 NVIDIA 20 GB GPU
  + No persistent storage
* **Start a new terminal in Kubeflow, then execute (all subsequent commands need to be executed in the terminal)**
  + *cd /workspace/*
* **Clone the YOLOV7 repository from GitHub**
  + *git clone* [*https://github.com/WongKinYiu/yolov7.git*](https://github.com/WongKinYiu/yolov7.git)
  + change the directory to this cloned repository
    - *cd yolov7/*
* **Install the necessary packages from the requirements.txt file**
  + *pip3 install -r requirements.txt*
* **Install additional support libraries required by OpenCV**
  + *apt update*
  + *apt install ffmpeg libsm6 libxext6 -y*
* **To test the inference capabilities of Yolov7, run the command**
  + *python3 detect.py --weights yolov7.pt --conf 0.25 --img-size 640 --source inference/images/horses.jpg*
    - **[OPTIONAL]** By default, YOLOv7 tries to run on GPU. If you want to run it specifically on CPU, use the flag “--device cpu” as follows:
      * *python3 detect.py --weights yolov7.pt --conf 0.25 --img-size 640 --source --device cpu inference/images/horses.jpg*
  + The test images/videos are present in *yolov7/inference* directory
  + The inferred images/videos are present in *yolov7/runs/detect* directory



Fig: inference/images/horses.jpg

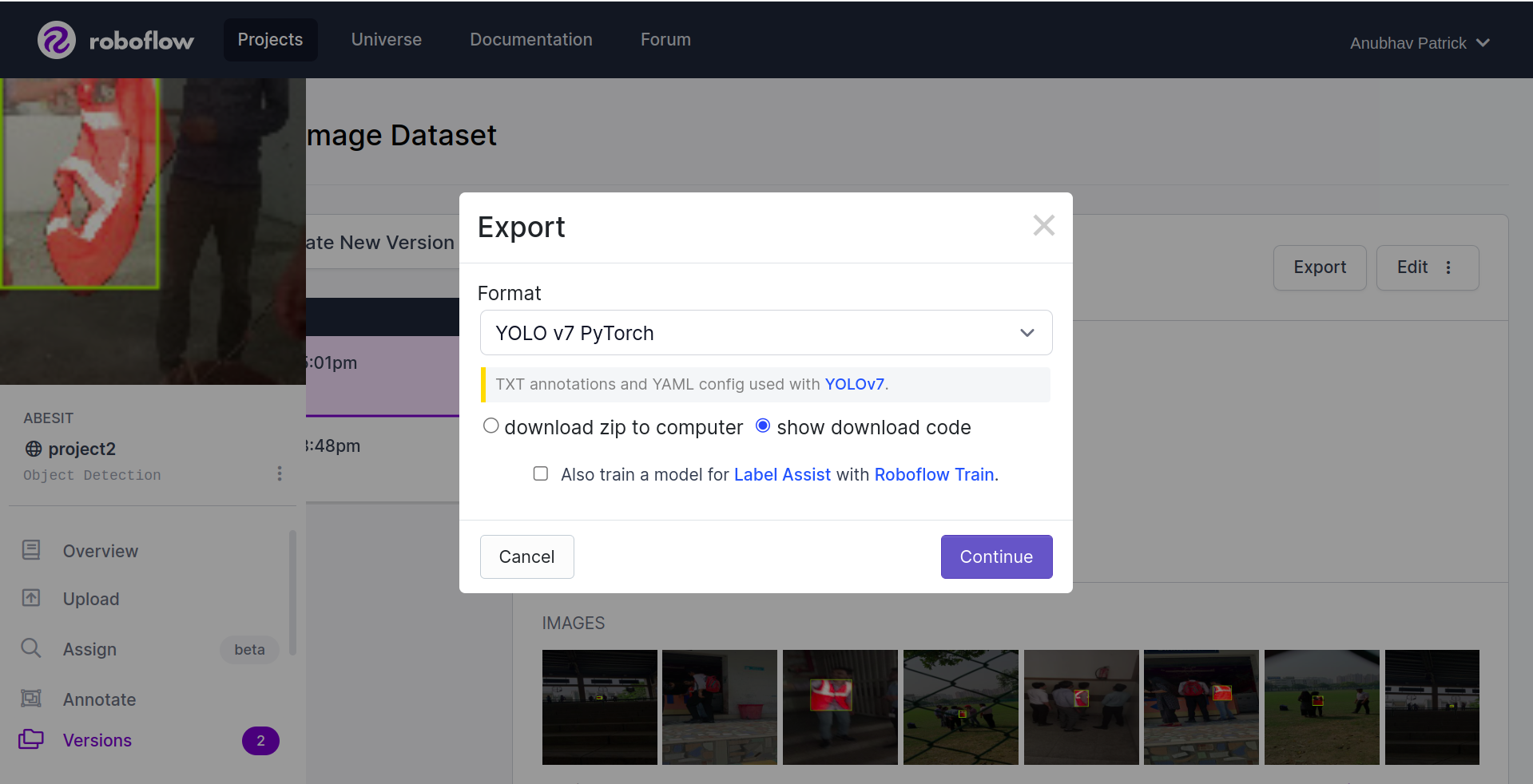


Fig: runs/detect/exp/horses.jpg

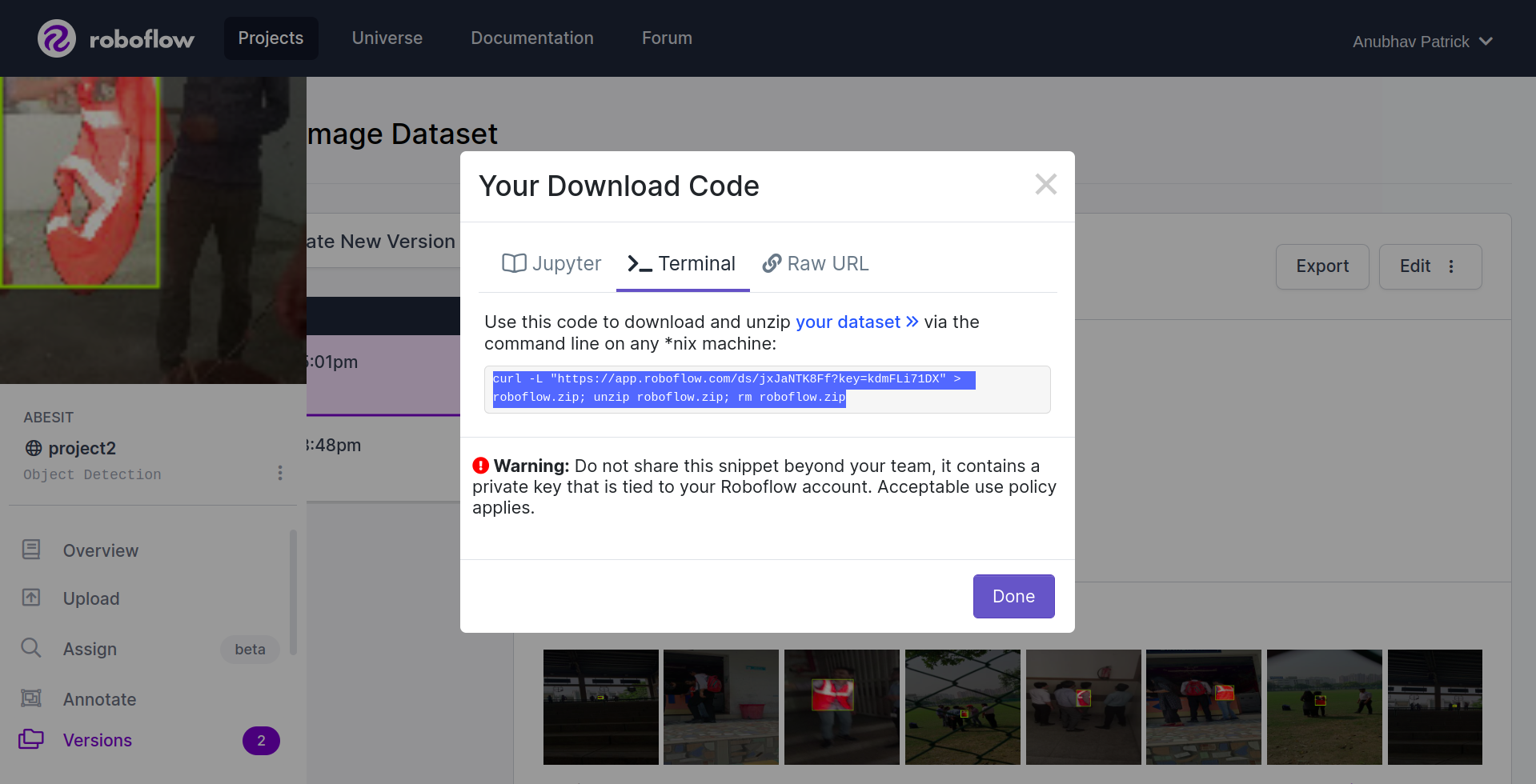
**How to train YOLOv7 on NVIDIA DGX**

***Note*** *- all the code needs to be executed in the Terminal in yolov7/ directory*

* **Export annotated dataset from Roboflow** 
  + Goto Roboflow -> Generate Dataset
  + In the preprocessing section, select the *Resize* parameter to 640\*640 then select *Generate*
  + Goto *Version* then select the dataset and select *Export*
  + The format is *YOLO v7 PyTorch*
  + Select *show download code* -> *Continue*



* + Copy the code under the *Terminal* tab



* + **Note**: Temporarily, you can use the below code for downloading the annotated dataset from my account into DGX

*curl -L "https://app.roboflow.com/ds/jxJaNTK8Ff?key=kdmFLi71DX" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip*

* **Download the annotated dataset into DGX -** Paste the download code in the Terminal of the Kubeflow notebook

*mkdir custom\_dataset*

*cd custom\_dataset/*

*curl -L "https://app.roboflow.com/ds/jxJaNTK8Ff?key=kdmFLi71DX" > roboflow.zip*

*unzip roboflow.zip; rm roboflow.zip*

*cd ..*

* **Update Custom Dataset configuration**
  + Open *data.yaml* file by browsing to *yolov7/custom\_dataset/* directory in Kubeflow File Browser
  + Update train, val, and test fields as given below:

*train: /workspace/yolov7/custom\_dataset/train/images*

*val: /workspace/yolov7/custom\_dataset/valid/images*

*test: /workspace/yolov7/custom\_dataset/test/images*

* + **Note**: Do not make any other changes to the file
  + Rename the file *data.yaml* to *custom\_data.yaml*
  + Copy *custom\_data.yaml* into *yolov7/data/* directory

*cp /workspace/yolov7/custom\_dataset/custom\_data.yaml /workspace/yolov7/data/*

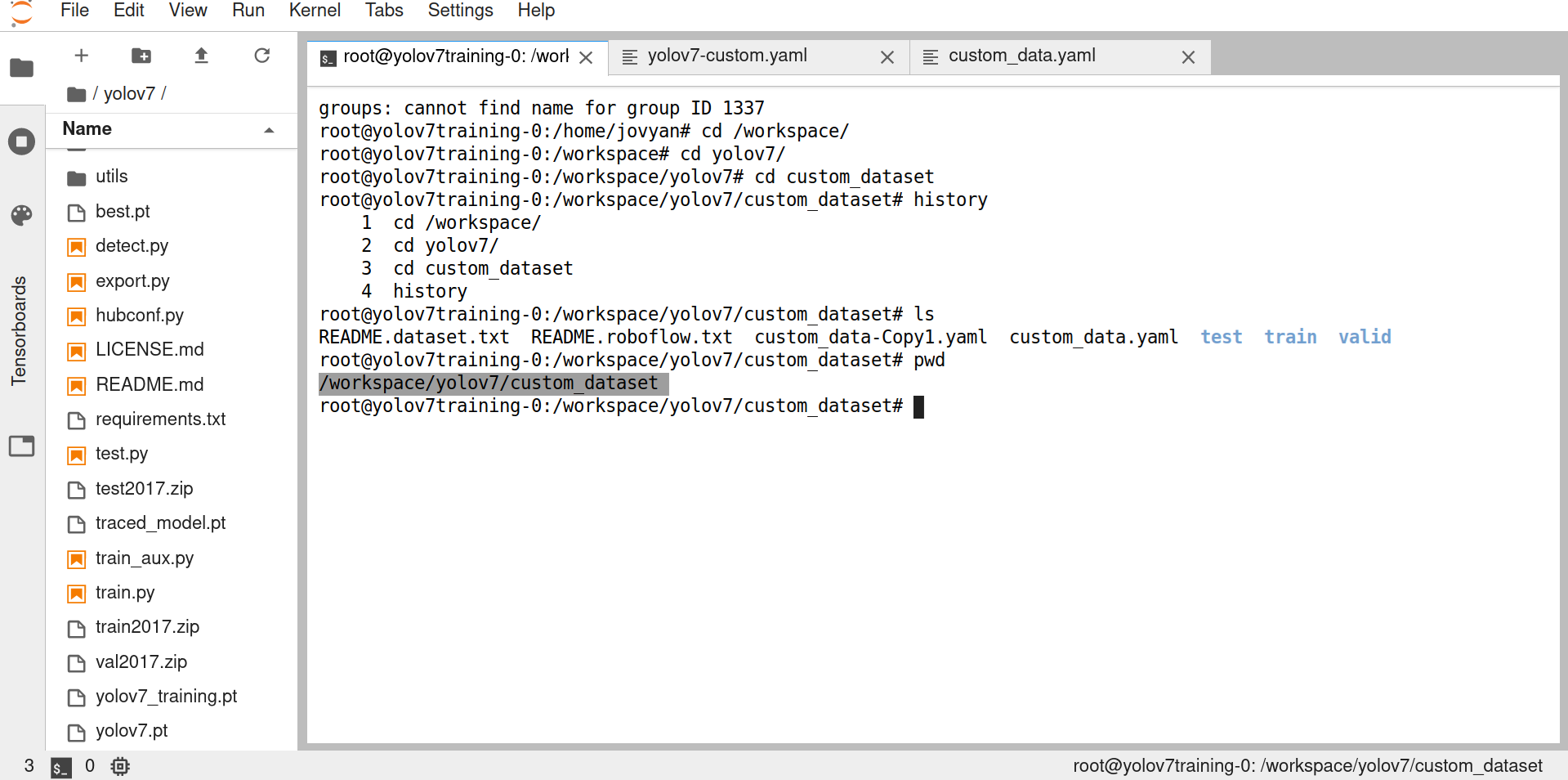
* **Update Yolov7 configuration**
  + Open *yolov7.yaml* file by browsing to *yolov7/cfg/training/* directory in Kubeflow File Browser
  + Update the *nc* field (number of classes)

*nc: 1*

* **Note**: Do not make any other changes to the file
* Rename the file *yolov7.yaml* to *yolov7\_custom.yaml*
* **Download pre-trained model weights into your local machine either from the official GitHub repository of Yolov7 or using the following link:**

<https://drive.google.com/drive/folders/1FJ3gHmOvpeZ82OqQ-ACklQbJqDjFc9VN?usp=sharing>

* The name of the file is - *yolov7\_training.pt*
* **Upload the *yolov7\_training.pt* file to your Kubeflow’s *yolov7/* directory**



* **Start the training process**

*python train.py --workers 8 --device 0 --batch-size 16 --data data/custom\_data.yaml --img 320 320 --cfg cfg/training/yolov7\_custom.yaml --weights yolov7\_training.pt --name yolov7\_custom --hyp data/hyp.scratch.p5.yaml*

* We can use *--device cpu* if we are training the model on CPU
* **Explore the results of training**
  + The trained models are present in *yolov7/runs/train/* directory
  + The directory *yolov7/runs/train/yolov7\_custom/weights/* contains the weights of the model at different times in the training process
    - The file *best.pt* contains the best weights of the trained model

**TASKS**

* **Do inference on custom data**
  + **Copy the trained model** (best.pt) from *yolov7/runs/train/yolov7\_custom/weights/* to *yolov7/* directory
    - *cp yolov7/runs/train/yolov7\_custom/weights/best.pt yolov7/*
  + **Test model on your custom data**
    - Download test images:

<https://drive.google.com/drive/folders/1MdDnRSamFoVC5qxzUhj-y62rlWNWmR19?usp=sharing>

* + - Upload test images to *yolov7/inference/images/* directory using Kubeflow’s file browser
    - Run the following script in Terminal

*python3 detect.py --weights best.pt --conf 0.25 --img-size 640 --source inference/images/img1.jpg - -epochs<number of epochs>*

* View the results in *yolov7/runs/detect* directory
* **Do the performance of your custom model**
  + Run the following script in your Terminal

*python test.py --data data/custom\_data.yaml --img 640 --batch 16 --conf 0.001 --iou 0.65 --device 0 --weights best.pt --name yolov7\_custom\_test*

* + The results of the performance evaluation are stored in yolov7/runs/test/yolov7\_custom\_test/ directory
  + Interpret the results

**VERY IMPORTANT:** You always need to delete your Kubeflow notebook as soon as you finish your task since it blocks GPU to be utilized by other students!!!

If you fail to do so then your account will be deactivated

