Course 4, Yolo

# Section 1 :

* Install Python if not already installed.
* Install Zip Extractor if not available.
* Open the following Google Drive Link and install both the zip file and onnx model: <https://drive.google.com/drive/u/2/folders/1TcmrgjyI3C__m-wDgy403xcN2BX1zqyc>
* Extract the zip file, preferably in “D-Drive” else in “users/documents”.

(Mine is C:\Users\ruler\Documents\yolov7-python)

* Open the extracted zip file named yolov7-python in cmd.

Either by typing cmd in the top file location bar, or by using

cd C:\Users\ruler\Documents\yolov7-python

* Once in cmd install requirements, by typing following in cmd

pip install -r requirements.txt

* Transfer the onnx model file previously downloaded to the user/documents/yolo7-python folder where all the extracted content is.
* Finally, you can run the detection.py program using yolo specifically the onnx model. To do this type the following in cmd with the correct directory :

python detection.py --weights yolov7.onnx --source data\videos\road.mp4 (for video)

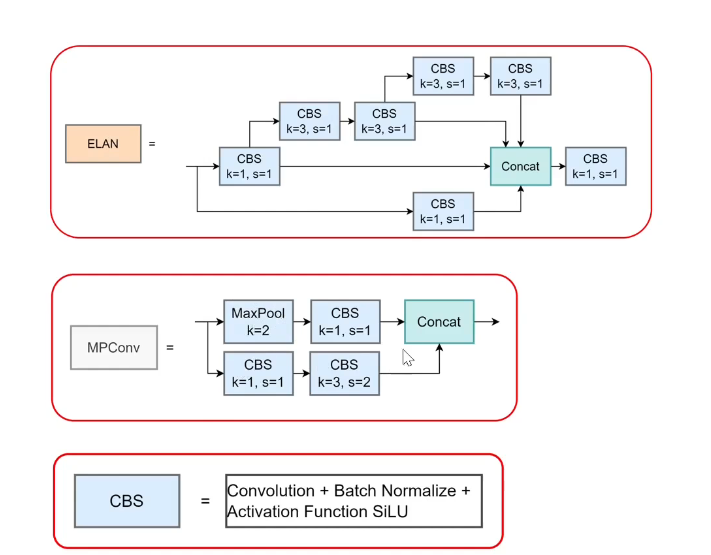
python detection.py --weights yolov7.onnx --source data\images\horses.jpg (for image)

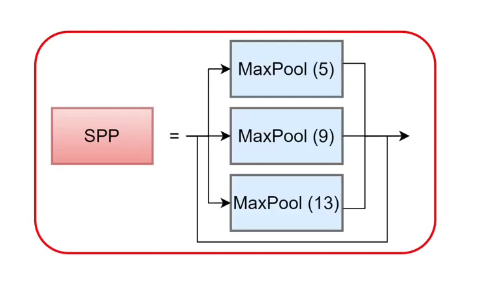
python detection.py --weights yolov7.onnx –source 0 (for webcam)

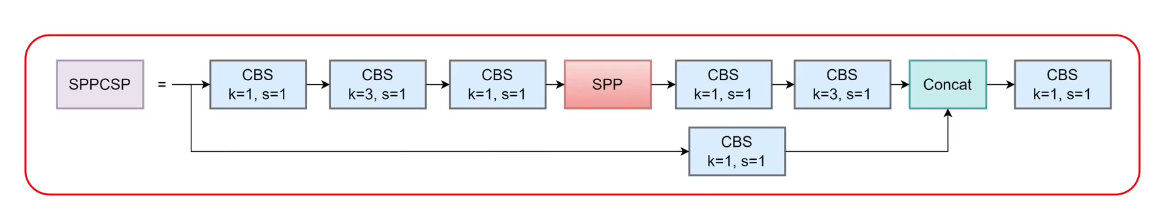
# Section 2 :

* AI/ML/NeuralNetwork/DL venn diagram
* Neural Network components input, hidden, and output layer, bias, activation function and types of activation functions including Mish and SiLU, perceptron. Types of Neural Networks. (Do from PDL notes.)
* ResNet was introduced to solve the vanishing gradients problem, It works on the principle of adding the prior result to the subsequent result.
* How CNN works (Do from PDL notes.)
* YOLO7 Architecture :
  1. Components :
     + CBS = Convolution + Batch Normalization + SiLU (A.F)
     + MPConv = MaxPool + CBS + Concat
     + ELAN = CBS + Concat + CBS
     + SPP = MaxPool \* 3
     + SPPCSP = CBS\*3 + SPP + CBS\*2 + /CBS/ + Concat + CBS
     + RepConv = (Conv + Batch Norm)\*2 + Batch Normalization + ADD //Train

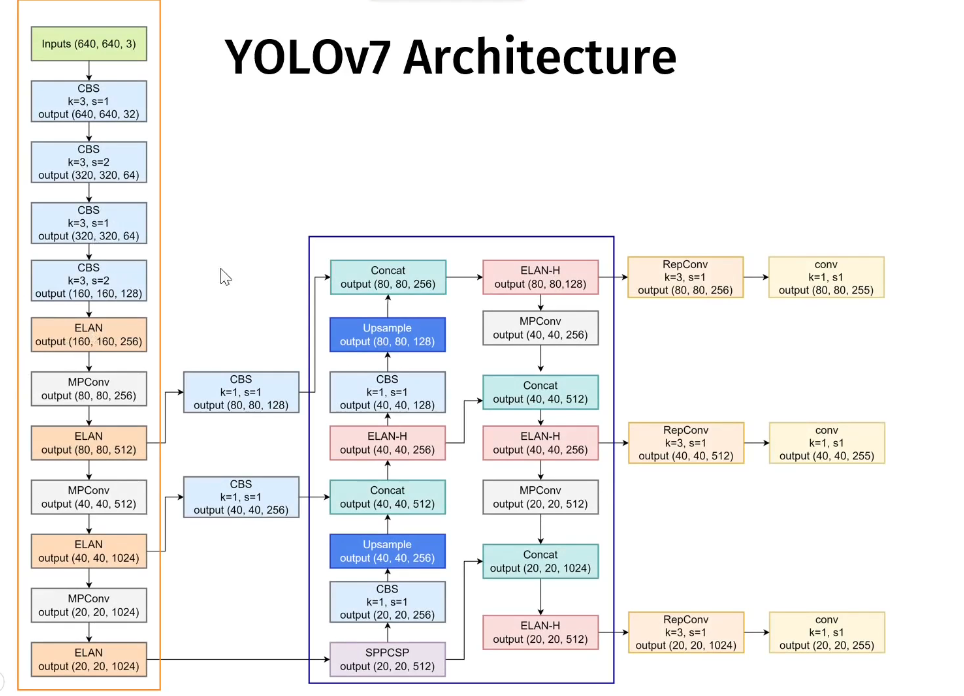
= (CBS + Batch Normalization) // Deploy







* 1. Architecture Diagram :



# Section 3 :

* Anaconda Installation, click on “x”, then sign in, then click on download anaconda distribution, then double click on the downloaded installer, then click next until finish. To check enter anaconda in search to see if anaconda navigator and prompt is available.
* Git Installation, click on “https://git-scm.com/downloads”, then click on “download for windows”, then click on “Git for windows/x64 setup”, open the downloaded git installer by double clicking on it, finally choose next until finish.

To check If downloaded type “git —version” in cmd.

# Section 4 :

* Dataset, The dataset must be labelled and relevant to the problem statement for training a yolo model. Kaggle can be used for selecting a dataset however for commercial use license must be checked. We download face mask dataset from Kaggle, and extract it into the same folder as where the yolo model is for uniformity.
* Yolo Dataset format, The dataset must have annotated labels with the format

<object-class> <x> <y> <width> <height> //object class = label, x and y = center of bounding box, width and height = Bounding Box width and height.

* To create an environment in anaconda, open anaconda navigator, then click on environments, then click on create an environment named “labelimg” and select python 3.9 to finish.
* How to annotate, We make use of annotating tools specifically ‘Labelimg’ which can be downloaded using ‘pip install labelimg’ in anaconda prompt. To do this:

1. Open anaconda prompt.
2. Type ‘activate labelimg’ //labelimg = environment name.
3. Type ‘pip install labelimg’.
4. Create a text file called ‘classes’ with the class labels (in this case Mask, No Mask, Bad Mask) and move it in the images folder i.e the Image dataset folder.
5. Open anaconda prompt again in the labelimg environment as mentioned above.
6. Change Directory to the directory where all the files are stored, In my case used

cd "C:\Users\ruler\Documents"

1. Open the annotating tool, by using labelimg images\classes.txt
2. Now open the images from the folder if not opened by default.
3. Select the createrectbox button and drag the mouse to highlight the mouse and enter the class label if not retrieved from classes.txt .
4. Change the annotation format to YOLO by clicking on button below save until it shows YOLO. (Stores in txt)
5. Save the file.
6. Now Repeat for all.

* Dataset Splitting Standard for all, Training and Testing and/or Validation.

# Section 5 :

* Install GPU Driver from Nvidia’s website. To check type “nvidia-smi”.
* To install CUDA, Go on “<https://developer.nvidia.com/cuda-toolkit-archive>”, Then Select The CUDA toolkit version which is shown when we ran nvidia-smi, Then select Windows, Then select x64, Then select 11, and then finally select exe(local).

Once downloaded click next until finish, and to check if the download is successfully run “nvcc --version” on cmd.

* To download CUDNN, Go on “https://developer.nvidia.com/rdp/cudnn-archive” after login into NVIDIA, Then select the CUDNN version compatible with the CUDA version (mine was “[Download cuDNN v8.9.7 (December 5th, 2023), for CUDA 12.x](https://developer.nvidia.com/rdp/cudnn-archive#a-collapse897-120)”), Then click on Local Installer for Windows (Zip). After installation move the ZIP file into D directory (In my case Documents) and extract it there, Open the extracted folder, open the bin folder, copy all files there and go to C directory program files, NVIDIA GPU Computing Toolkit CUDA, The cuda version, bin and paste it there. Do the same for all the files In include folder of extracted folder to the include folder in C directory. Do the same for all files in lib folder from extracted folder to x64/lib.

# Section 6 :

* YOLO download for CPU Mode, Create an environment in anaconda called

“yolov7-cpu-env”(python 3.10), Then activate yolov7-cpu-env in anaconda prompt, Then change directory to the directory where all the files are (in my case documents,

cd Documents), Then clone the yolo repo from github by using “git clone <https://github.com/pHidayatullah/yolov7.git>”, Then change the name of the yolov7 model to yolov7-cpu by using

“ren yolov7 yolov7-cpu”, Then change the directory to yolov7-cpu using “cd yolov7-cpu”, Then download requirements using “pip install -r requirements.txt”, Then download a yolo model for weights by going on the above mentioned github repo and downloading YOLOv7 model, Then move it to yolov7-cpu folder. Finally, to test the model run

“python detect.py --weights yolov7.pt --source inference\images\bus.jpg” in anaconda prompt in the cpu-env. (If CV2 error then pip install opencv-python, and also pytorch as given in GPU mode, If more then python version was incorrectly chosen),You can view the results in the directory mentioned at the end of the execution in cmd.

* YOLO download for GPU Mode, Create an environment in anaconda called

“yolov7-gpu-env”, Then activate yolov7-gpu-env in anaconda prompt, Then change directory to the directory where all the files are (in my case documents i.e

cd Documents), Then clone the yolo repo from github by using “git clone https://github.com/pHidayatullah/yolov7.git”, Then change the name of the yolov7 model to yolov7-gpu by using

“ren yolov7 yolov7-gpu”, Then change the directory to yolov7-gpu using “cd yolov7-gpu”, Then install pytorch by going to “https://pytorch.org/” selecting options and copying the link provided into anaconda prompt i.e “pip3 install torch torchvision torchaudio --index-url <https://download.pytorch.org/whl/cu118>”.

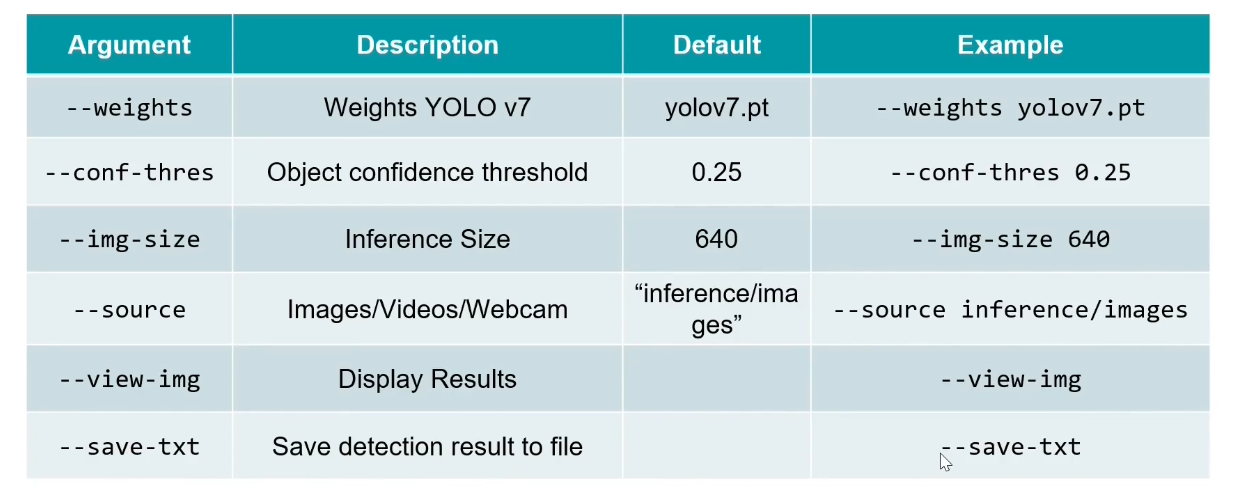
Then download requirements using “pip install -r requirements\_gpu.txt”, Then download a yolo model for weights by going on the above mentioned github repo and downloading YOLOv7 model, Then move it to yolov7-gpu folder.

Finally, to test the model run “python detect.py --weights yolov7.pt --source inference\images\horses.jpg” in anaconda prompt in the gpu-env. You can view the results in the directory mentioned at the end of the execution in cmd.

# Section 7 :

* The below are some parameters that we can choose while running the yolo model i.e

In python detect.py….(You can view the results at the directory returned at the end)



* For Object Detection in Image,

1. Activate yolov7-gpu-env
2. Change directory cd yolov7-gpu-env
3. Python detect.py –weight yolov7.pt --conf-thres 0.5 --img-size 640 --source inference\images --view-img --save.txt

* For Object Detection in Image,

1. Activate yolov7-gpu-env
2. Change directory cd yolov7-gpu-env
3. Python detect.py --weight yolov7.pt --conf-thres 0.5 --img-size 640 --source inference\images --view-img

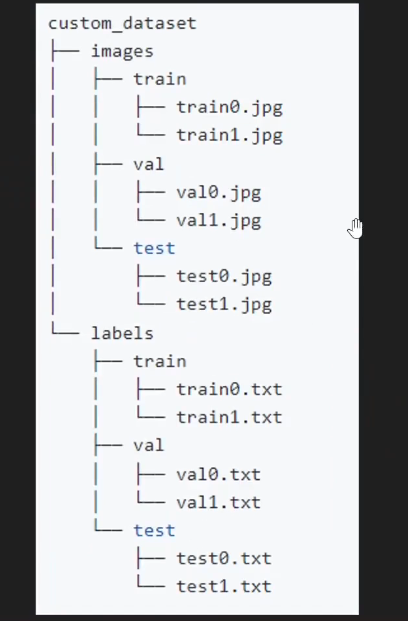
* For Object Detection in Image,

1. Activate yolov7-gpu-env
2. Change directory cd yolov7-gpu-env
3. Python detect.py --weight yolov7.py --conf-thres 0.5 --img-size 640 --source 0

# Section 7 :

* Data Splitting,

1. Download the dataset and move it to data folder of yolov7-gpu model.
2. Ensure that the data split structure follows the yolov7 folder structure as mentioned below



1. The data splitting code is provided in the data folder of yolov7-gpu. The parameters are as follows :
   1. Train, default = 80
   2. Validation, default = 10
   3. Test, default = 10
   4. Folder, specifies where the dataset is stored before splitting.
   5. Dest, specifies where the split results will be saved.

To split the code execute the data split code,

1. Activate environment
2. Cd Documents
3. Cd yolov7-gpu
4. Cd data
5. python split\_dataset.py --folder face\_mask --train 80 --validation 10 --test 10 --dest face\_mask\_dataset

This creates two folders just like the above mentioned yolo folder structure.(labels may contain empty txt files)

1. How to make configuration file,
   1. Copy the coco.yaml file as in data folder in yolov7-gpu again in the data folder and rename it to face\_mask.yaml.
   2. Now open the face\_mask.yaml file and remove everything except train, test, val, nc and names. Now change the path for each train, val, test to data/face\_mask\_dataset/images/train,

data/face\_mask\_dataset/images/val,

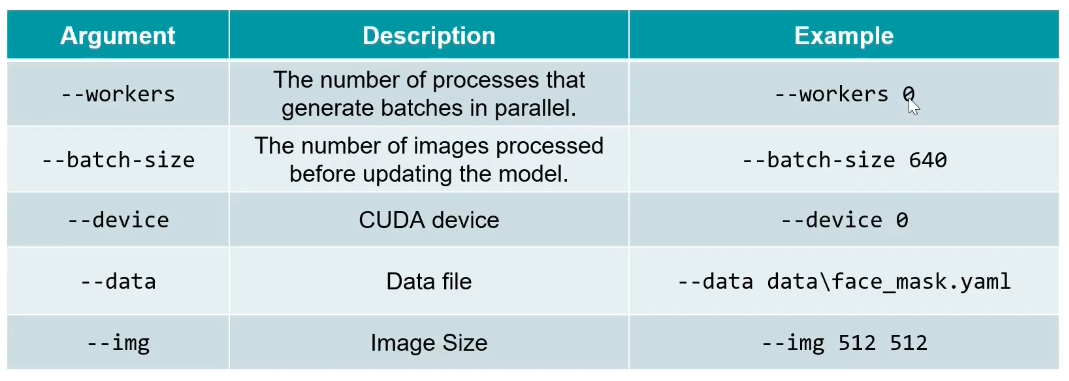
data/face\_mask\_dataset/images/test (respectively)

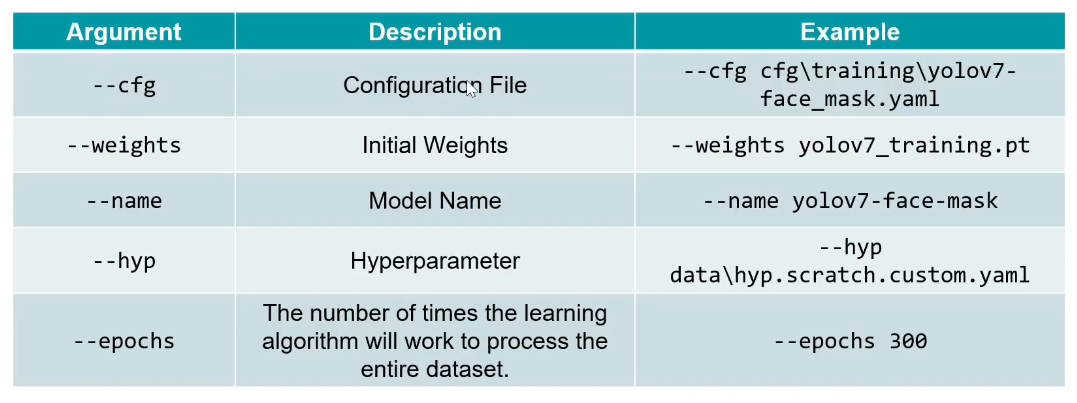
nc = 3

names = [“Mask”, “No Mask”, “Bad Mask”]

* 1. Copy the yolov7.yaml file as in cfg\training folder in yolov7-gpu again in the cfg\training folder and rename it to yolov7\_face\_mask.yaml.
  2. Now open the yolov7\_face\_mask.yaml file and change nc = 3

1. Training the model,
   1. Leverage transfer learning by downloading yolov7\_training.pt file from <https://github.com/pHidayatullah/yolov7?tab=readme-ov-file>, Then moving the file to yolov7-gpu folder.
   2. The following arguments are used while training,





* 1. Launch anaconda prompt, activate the environment, navigate to yolov7-gpu folder, and execute the command :

python train.py --workers 0 --batch-size 4 --device 0 --data data\face\_mask.yaml --img 640 640 --cfg cfg\training\yolov7\_face\_mask.yaml --weights yolov7\_training.pt --name yolov7-face-mask --hyp data\hyp.scratch.custom.yaml --epochs 300 //Make changes to train.py if errors are in

* 1. You can pause the training using CTRL C, and can resume the training by using the command :

python train.py --workers 0 --batch-size 4 --device 0 --data data\face\_mask.yaml --img 640 640 --cfg cfg\training\yolov7-face\_mask.yaml --weights runs\train\yolov7-face-mask\weights\last.pt --name yolov7-face-mask --hyp data\hyp.scratch.custom.yaml --epochs 300 --resume

* 1. To view the training result, navigate to yolov7-gpu\runs\train\yolov7-face-mask\weights, The best is the weights with the highest accuracy/MLP values, Last is the weights of the last epoch.
  2. To view the accuracy graph we make use of tensorboard, using the command :

tensorboard --logdir runs\train

Then going to the retuned local address on the web browser.

* 1. To test, we move the image/video to be tested into the inference folder of yolov7-gpu. Now execute the command :

Python detect.py –weights runs\train\yolov7-face-mask\weights\best.pt –source inference\face-mask.png //for image

Python detect.py –weights runs\train\yolov7-face-mask\weights\best.pt –source inference\face-mask-video.mp4 –view-img//for video

Python detect.py –weights runs\train\yolov7-face-mask\weights\best.pt –source 0 //for webcam

* 1. To view the results navigate to the directory returned after the execution of above commands.

# Section 8 :

* The primary way to test the accuracy of your yolo model is to view the Precision, Recall, and MAP. This can be viewed by executing the command :

1. python test.py --weights runs\train\yolov7-face-mask\weights\best.pt --batch-size 2 --device 0 --data\face\_mask.yaml --img 640 --conf-thres 0.01 --iou 0.5 --name yolov7-face-mask-val --task val //for validation
2. python test.py --weights runs\train\yolov7-face-mask\weights\best.pt --batch-size 2 --device 0 --data\face\_mask.yaml --img 640 --conf-thres 0.01 --iou 0.5 --name yolov7-face-mask-tst --task test //for testing

* The results look like :

