Project Report On:

Custom Object Detection using Yolo-Nas



Guide:

Dr. J. V. Satyanarayana

Designation: Sc ‘G’

Directorate: DEAIS

By:

Name: JHARINATH ILAPARTI

University: Graphic Era Deemed to be University

**Methodology**

First we custom train a Yolo-Nas model, using the yolo\_nas\_l architecture and the coco dataset to use as a base for the model we are going to custom train.

We establish a pipeline code which can be used to custom train the yolo-nas on any kind of data based on our requirements.

* **Defining the libraries requires**

from super\_gradients.training.dataloaders.dataloaders import coco\_detection\_yolo\_format\_train, \  
 coco\_detection\_yolo\_format\_val  
from super\_gradients.training import models  
from super\_gradients.training.losses import PPYoloELoss  
from super\_gradients.training.metrics import DetectionMetrics\_050  
from super\_gradients.training.models.detection\_models.pp\_yolo\_e import PPYoloEPostPredictionCallback  
from super\_gradients.training import Trainer  
import torch  
import random  
from super\_gradients.training import dataloaders  
import supervision as sv  
from tqdm.notebook import tqdm

* **Defining the path from where data is to be taken**

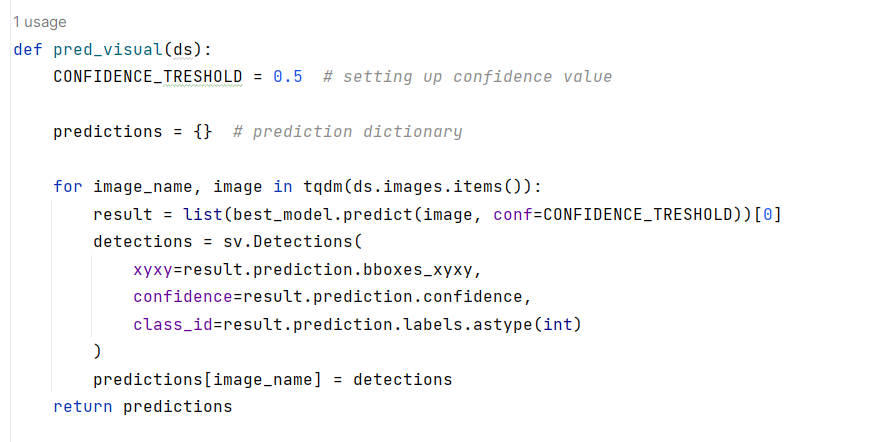
dataset\_params = {  
 'data\_dir': 'D:\\YOLO-NAS\\archive',  
 'train\_img\_dir': 'D:\\YOLO-NAS\\archive\\train\\images',  
 'train\_label\_dir': 'D:\\YOLO-NAS\\archive\\train\\labels',  
 'val\_img\_dir': 'D:\\YOLO-NAS\\archive\\valid\\images',  
 'val\_label\_dir': 'D:\\YOLO-NAS\\archive\\valid\\labels',  
 'test\_img\_dir': 'D:\\YOLO-NAS\\archive\\test\\images',  
 'test\_label\_dir': 'D:\\YOLO-NAS\\archive\\test\\labels',  
 'classes': ['sunglass', 'hat', 'jacket', 'shirt', 'pants', 'shorts', 'skirt', 'dress', 'bag', 'shoe']  
}

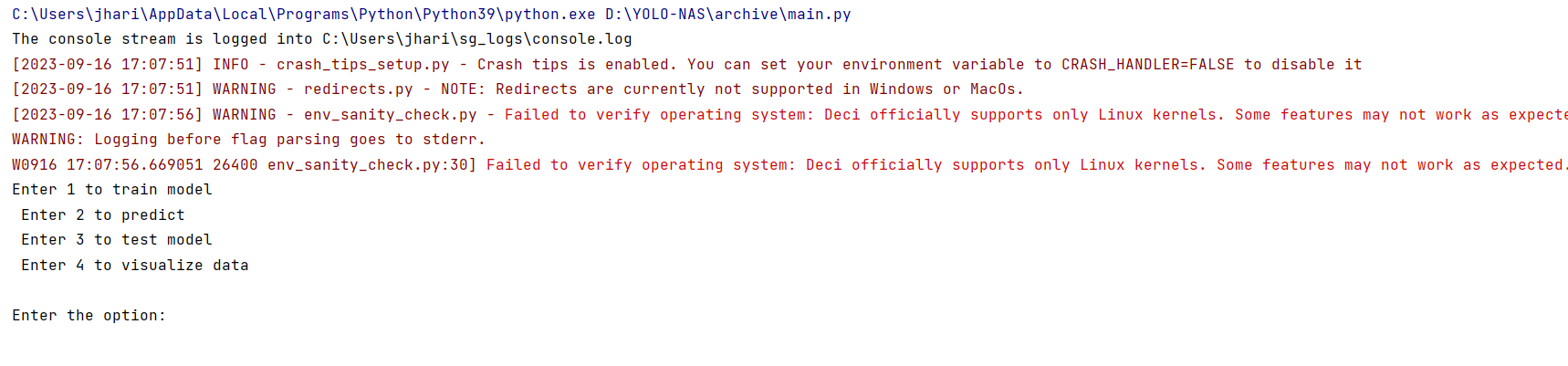
* **We establish various data loaders so that data can be fed to our yolo-nas model**
* **We define the parameters along which the data is to be trained**

****

* **We write the code to load the trainer and train our model and test it**

****

* **We define the code to visualize our predictions in comparison to actual data**
* **We can start our model by running the code and selecting the operation we want to perform**

****