# MODLBOX

## USER GUIDE

# Table of Contents

# Introduction

Welcome to the User Guide for MODLBOX!

## About MODLBOX

MODLBOX is an application to speed up the Machine learning/deep learning development lifecycle. It aims to provide an intuitive user interface and is helpful for people with limited coding experience.

## Purpose

This User Guide is designed to provide comprehensive instructions and guidance on using MODLBOX effectively. Whether you need assistance with installation, configuration, or understanding specific features, you'll find the information you need here.

## How to Use this Guide

This guide is organized into sections to help you quickly find the necessary information. You can navigate through the guide using the table of contents to locate specific topics.

## Let’s Get Started

We're excited to have you on board and ready to help you unlock the full potential of MODLBOX. Let's dive in and get started!

# Installation of Application

Step 1: Click on the link provided.

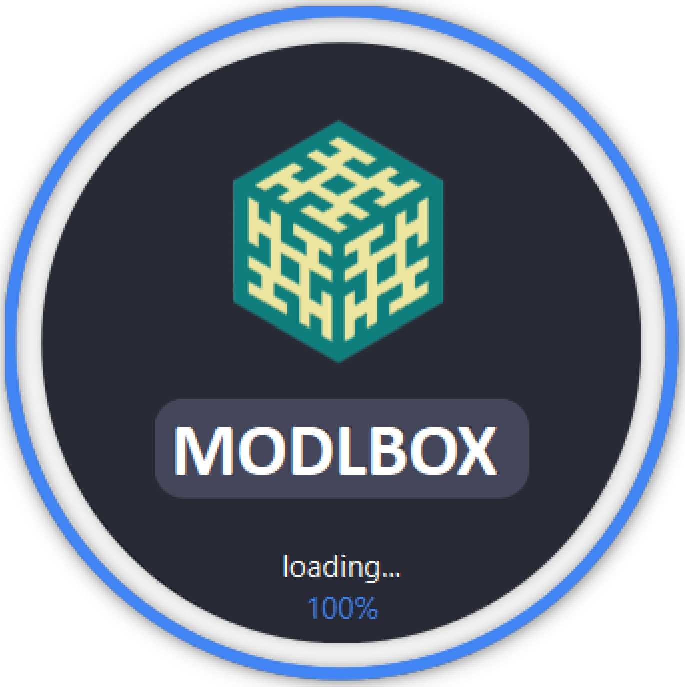
Step 2: Download the zip file to your local machine.

Step 3: Extract the files to the location of your choice.

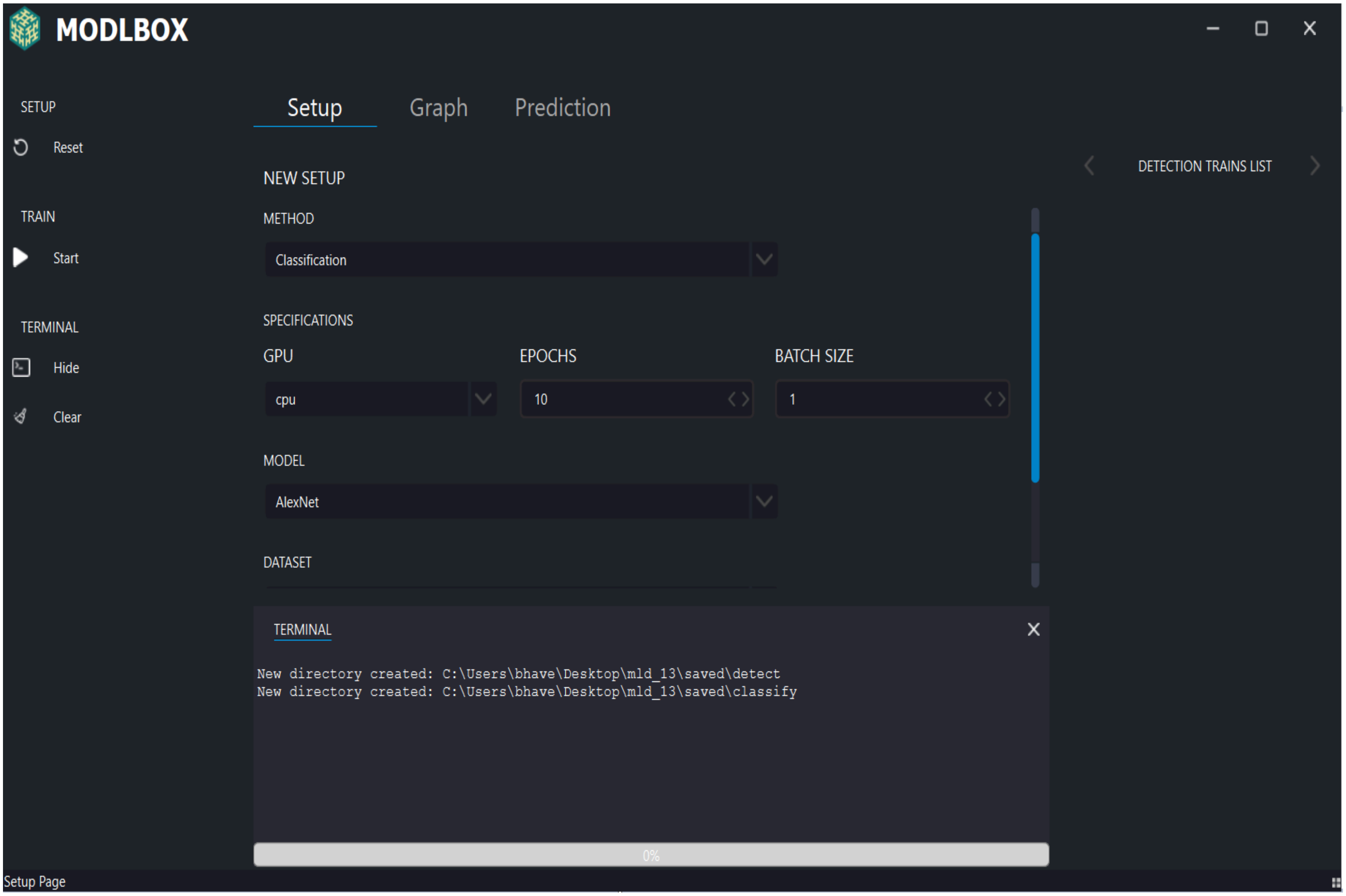
Step 4: Read the README file to check if the requirements are met.

Step 5: Click on the MODLBOX.exe file to begin running the application.

Step 6: If the above steps are followed correctly, you will see the console and the following loading screen:



After the loading screen completion, you will be greeted with the main screen of our application:



METHOD: Classification

SPECIFICATIONS

GPU: cpu

EPOCHS: 10

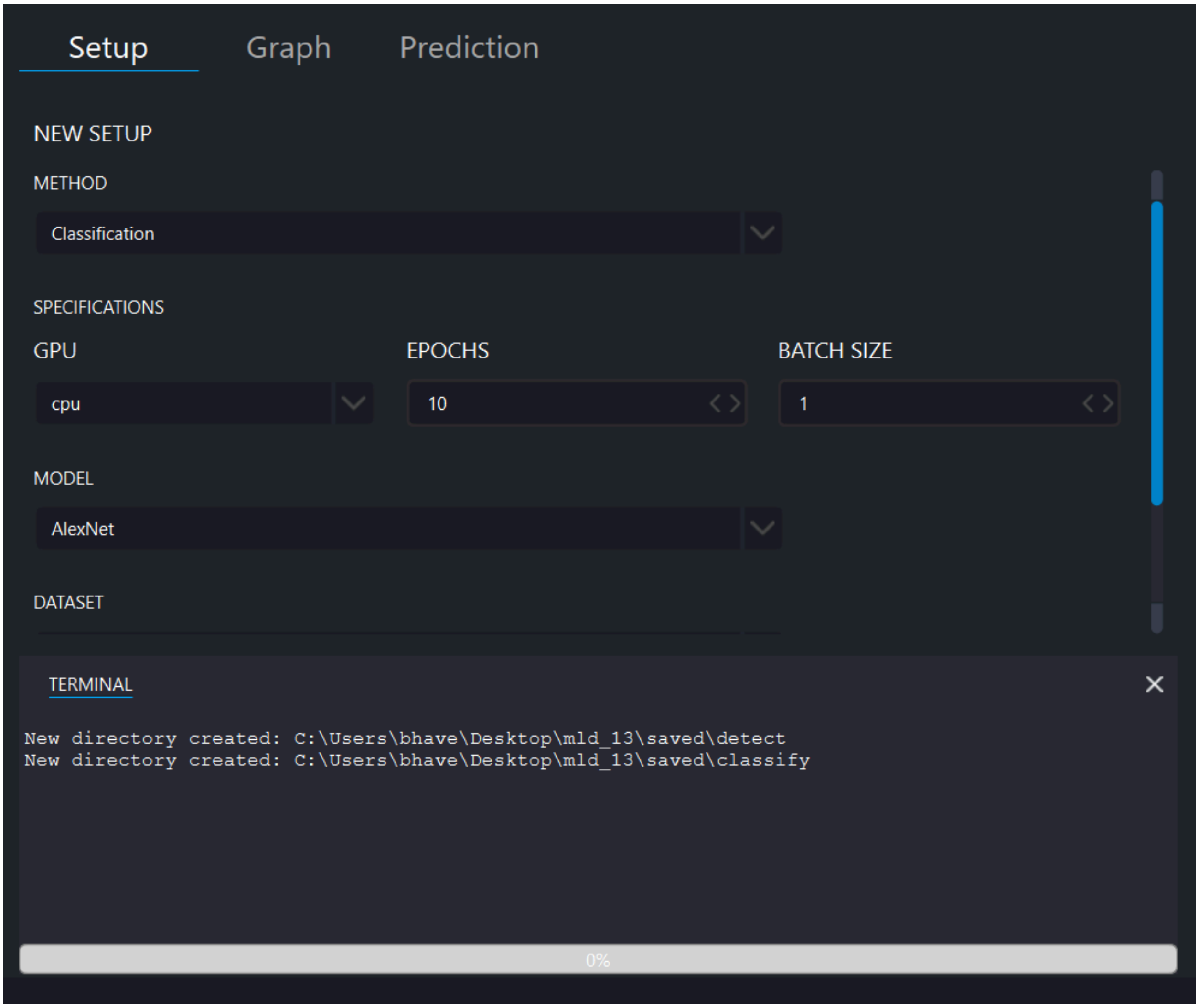
BATCH SIZE: 1

MODEL: AlexNet

DATASET

```
New directory created: C:\Users\bhave\Desktop\mld_13\saved\detect
New directory created: C:\Users\bhave\Desktop\mld_13\saved\classify
```

# Setup Page



NEW SETUP

METHOD: Classification

SPECIFICATIONS

GPU: cpu

EPOCHS: 10

BATCH SIZE: 1

MODEL: AlexNet

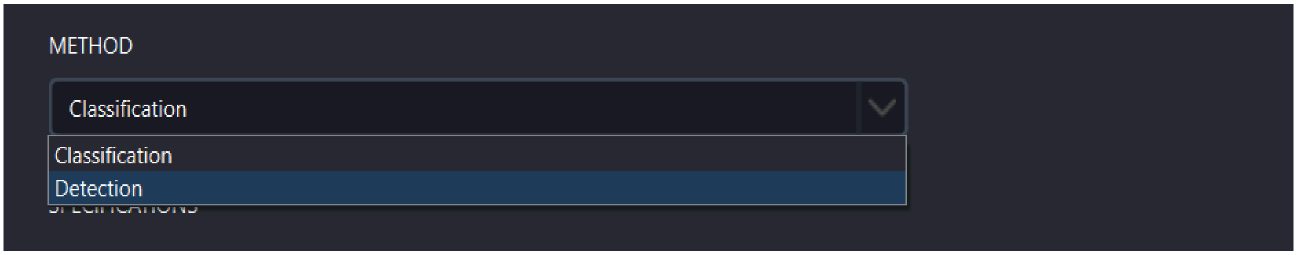
DATASET

```
New directory created: C:\Users\bhave\Desktop\mld_13\saved\detect
New directory created: C:\Users\bhave\Desktop\mld_13\saved\classify
```

This is the initial UI presented to the user when the application is opened. This is where the user can select different parameters before starting the training/running process.

Next are the step-by-step instructions explaining the parameters, options available and other information pertaining to it.

## Step 1: Method



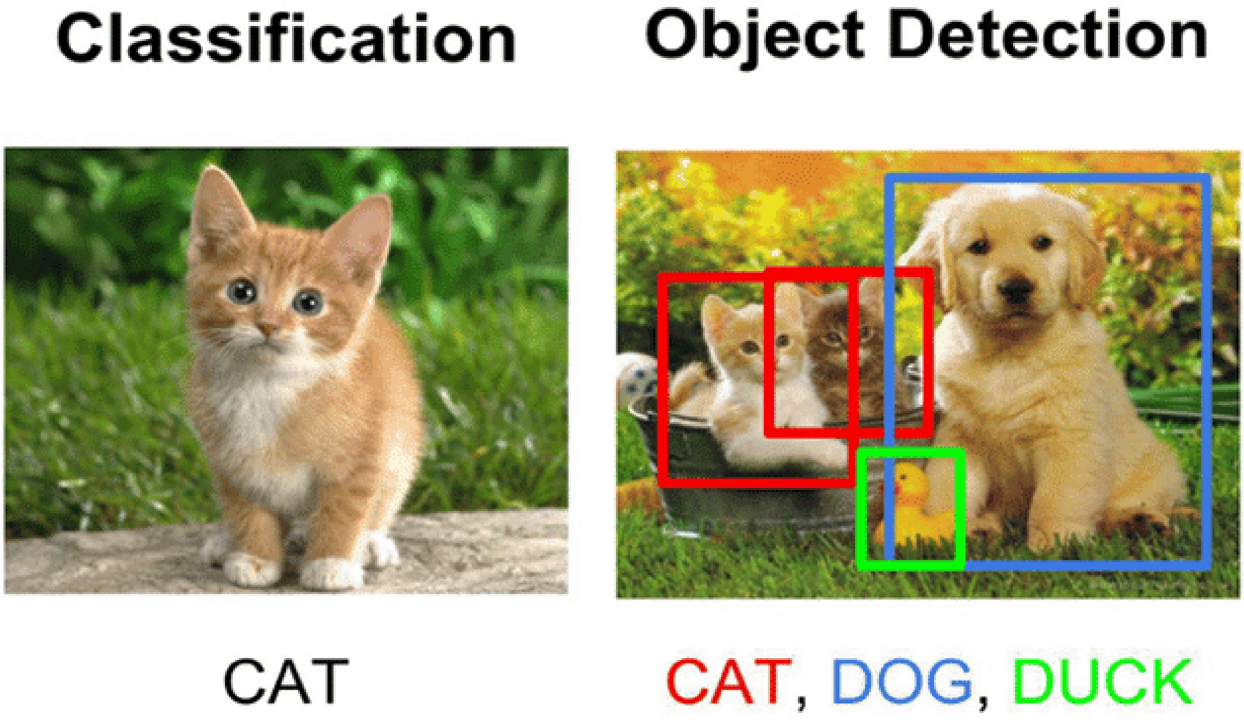
METHOD

Classification

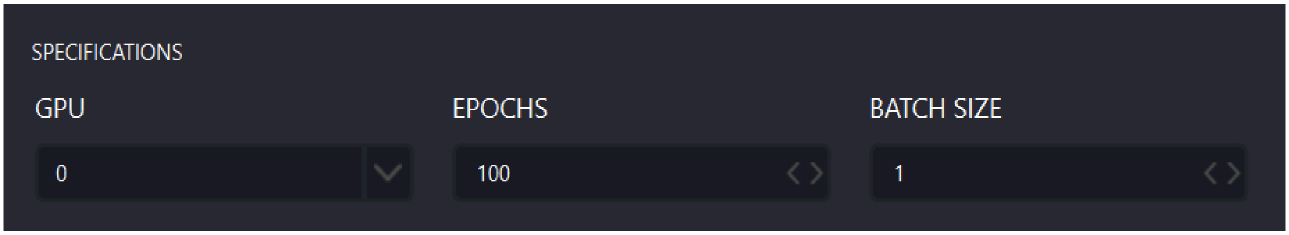
- The user must select the first parameter between Classification and Detection from a drop-down menu.
- To open the drop-down menu, the user should click the downward arrow at the end of the box and select the method.
- The default value for this parameter is **Classification.**

**Purpose**

**Classification:** involves assigning a single label or category to an entire image or object.
**Detection:** identifies and localizes multiple objects within an image by drawing bounding boxes around them.



## Step 2: Specifications



SPECIFICATIONS

GPU
0

EPOCHS
100

BATCH SIZE
1

Under Specifications, the user is presented with three parameters.

### GPU (Graphics Processing Unit)

- The user can select the number of GPUs using the dropdown menu.
- To open the dropdown menu, the user should click on the downward arrow at the end of the box and select the number of GPUs.
- The default value is **0** which will use CPU for the computation.

Note- If the only option available is “cpu”, it’s because torch.cuda library is unable to fetch the number of GPUs available using device_count on your system.

**Purpose**

- It allows users to utilize GPUs for computation during training.
- GPUs are well-suited for parallel processing tasks, which can significantly decrease training times compared to using only the CPU.
- The workload can be distributed across multiple devices by selecting the number of GPUs for faster training.

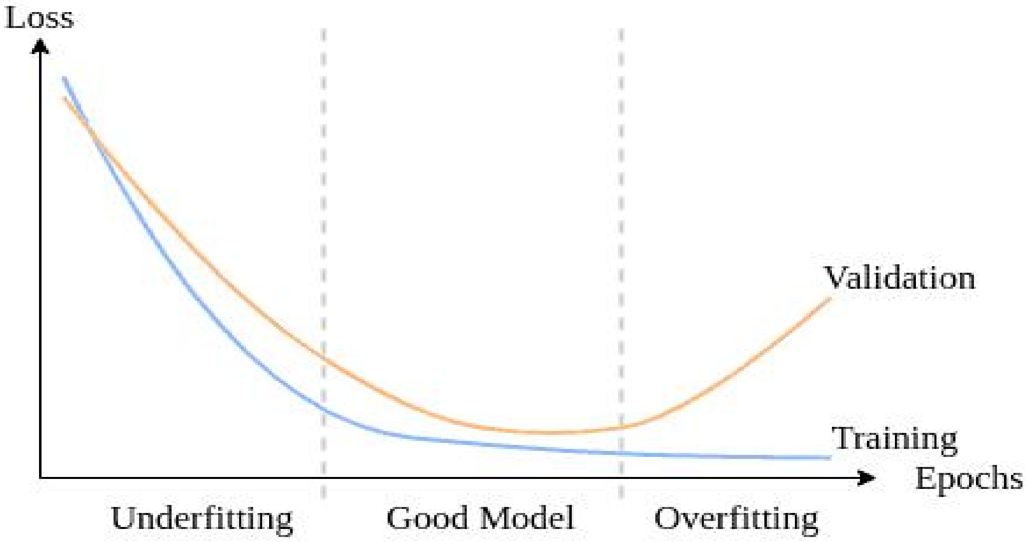
### Epoch

- The user can increase or decrease the value using the right or left arrow present at the end of the box or input the value directly.
- This parameter's minimum and maximum values are 1 and 10000, respectively.
- The default value is **10**.

**Purpose**

- Increasing the number of epochs can improve model performance.

- Used to specify the number of times the entire dataset should be used during training.
- Using too many or too few epochs can lead to overfitting or underfitting.



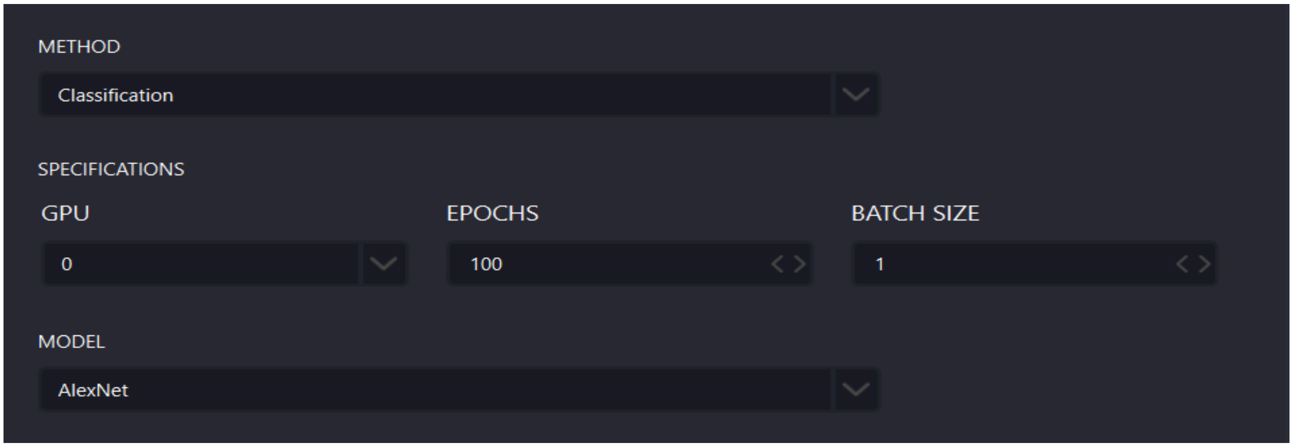
## Batch Size

- The user can increase or decrease the value using the right or left arrow present at the end of the box or input the value directly.
- This parameter's minimum and maximum values are 1 and 100, respectively.
- The default value is **1**.

**Purpose**

- It refers to the number of training examples used in one iteration(Epoch) of model training.
- Large batch sizes can lead to faster training times but may result in lower accuracy.
- Smaller batch sizes can provide better accuracy but can be computationally expensive and time-consuming.

## Step 3: Model



METHOD

Classification

SPECIFICATIONS

GPU: 0

EPOCHS: 100

BATCH SIZE: 1

MODEL

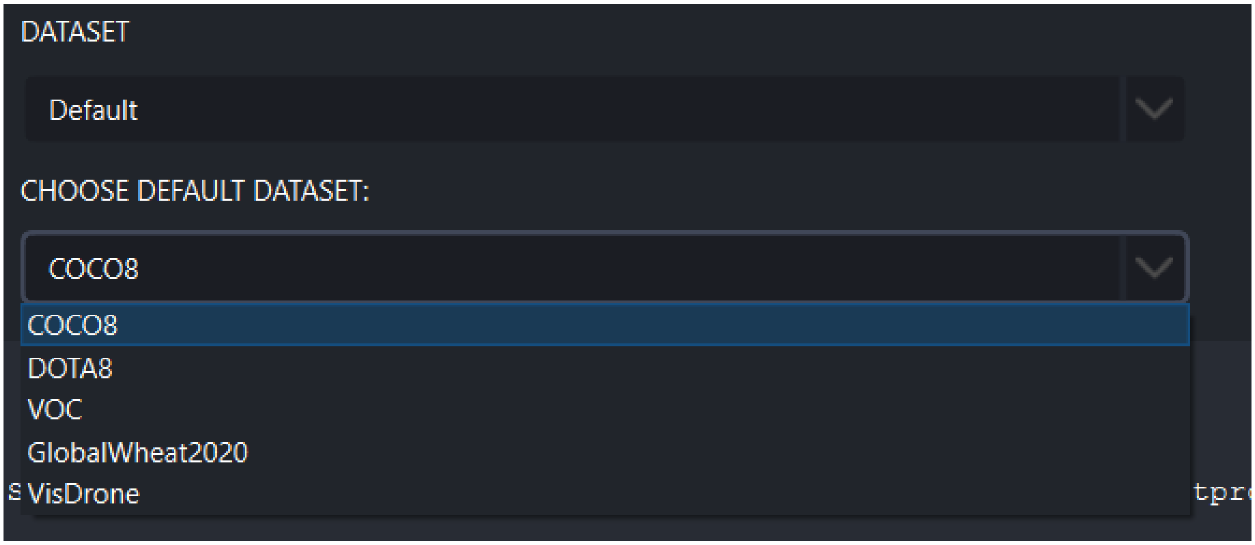
AlexNet

- The model will change depending on the choice made in step 1.
- It is a dropdown menu to select from the list of models. To open the dropdown menu, click on the down arrow at the end of the box and choose the model to train/run the dataset on.

**Purpose**

Learn patterns and relationships from data to make predictions, classifications, or decisions on new, unseen data.

## Step 4: Choose Dataset



DATASET

Default

CHOOSE DEFAULT DATASET:

COCO8

- The user can choose one of the datasets from the available dropdown menu (different for classification and detection).
- To open the drop-down menu, click on the arrow at the end of the box and select the dataset by clicking on it.

**Purpose**

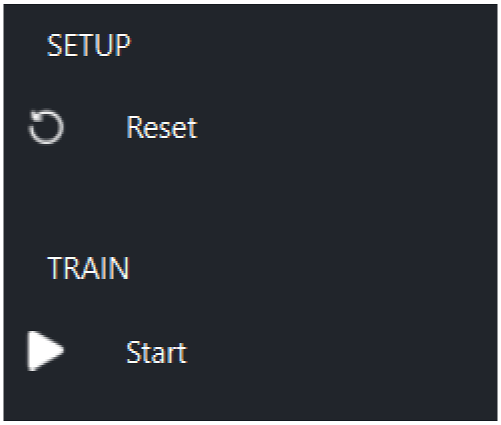
Choosing the dataset will determine on which the model will be trained on. It also determines the quality of model training and the subsequent predictions.

## Step 5: Starting Running/Training

After Completing all of the steps mentioned above, you can click on the **“START”** button in the top left corner of the MODLBOX app. Once the **“START”** button is clicked, the user will be directed to the **“Graph”** page.

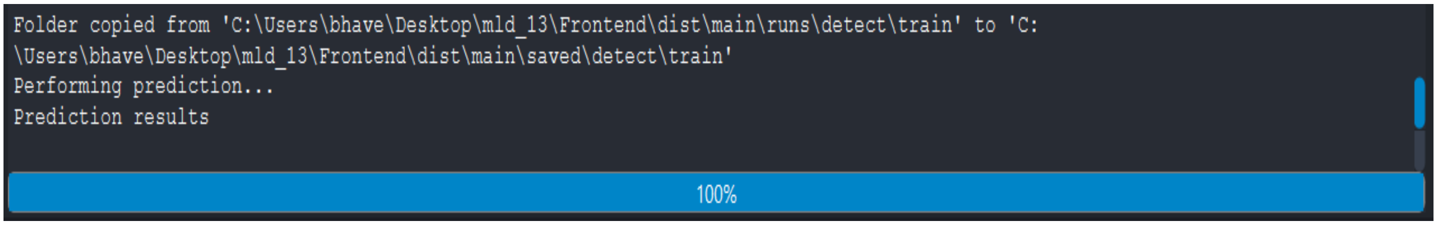
If at any point you are not satisfied with any option selected, you can click on the **“RESET”** button provided above the **“START”** button to reset all the values to default ones.

Note- Once the training is started, the user has to wait for the training to be completed.



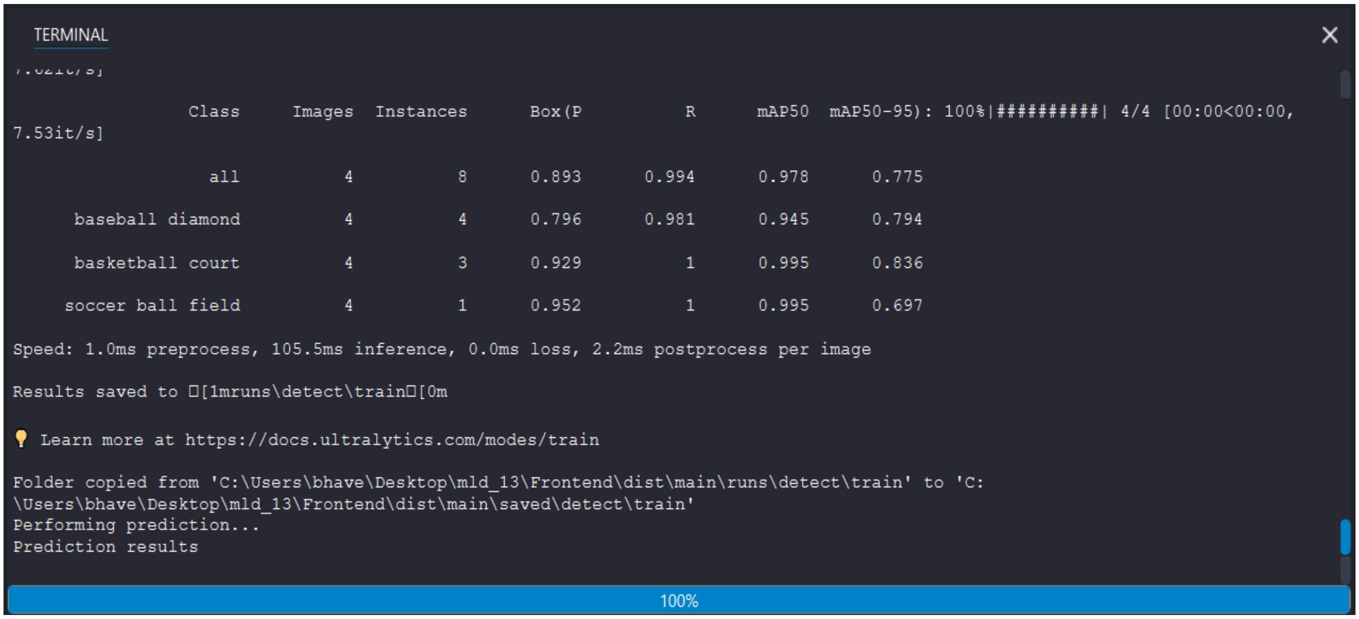
## Progress Bar

The progress bar is shown at the bottom of the terminal. The progress bar will get updated with each epoch.
For Example: If training with 100 epochs, the progress bar will get updated by 1% for each epoch.



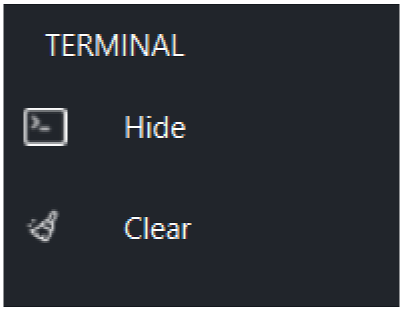
```
Folder copied from 'C:\Users\bhave\Desktop\mld_13\Frontend\dist\main\runs\detect\train' to 'C:
\Users\bhave\Desktop\mld_13\Frontend\dist\main\saved\detect\train'
Performing prediction...
Prediction results
```

## Terminal



```
7.62it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|##########| 4/4 [00:00<00:00,
7.53it/s]
                   all          4          8      0.893      0.994      0.978      0.775
      baseball diamond          4          4      0.796      0.981      0.945      0.794
      basketball court          4          3      0.929          1      0.995      0.836
     soccer ball field          4          1      0.952          1      0.995      0.697
Speed: 1.0ms preprocess, 105.5ms inference, 0.0ms loss, 2.2ms postprocess per image
Results saved to □[1mruns\detect\train□[0m
💡 Learn more at https://docs.ultralytics.com/modes/train
Folder copied from 'C:\Users\bhave\Desktop\mld_13\Frontend\dist\main\runs\detect\train' to 'C:
\Users\bhave\Desktop\mld_13\Frontend\dist\main\saved\detect\train'
Performing prediction...
Prediction results
```

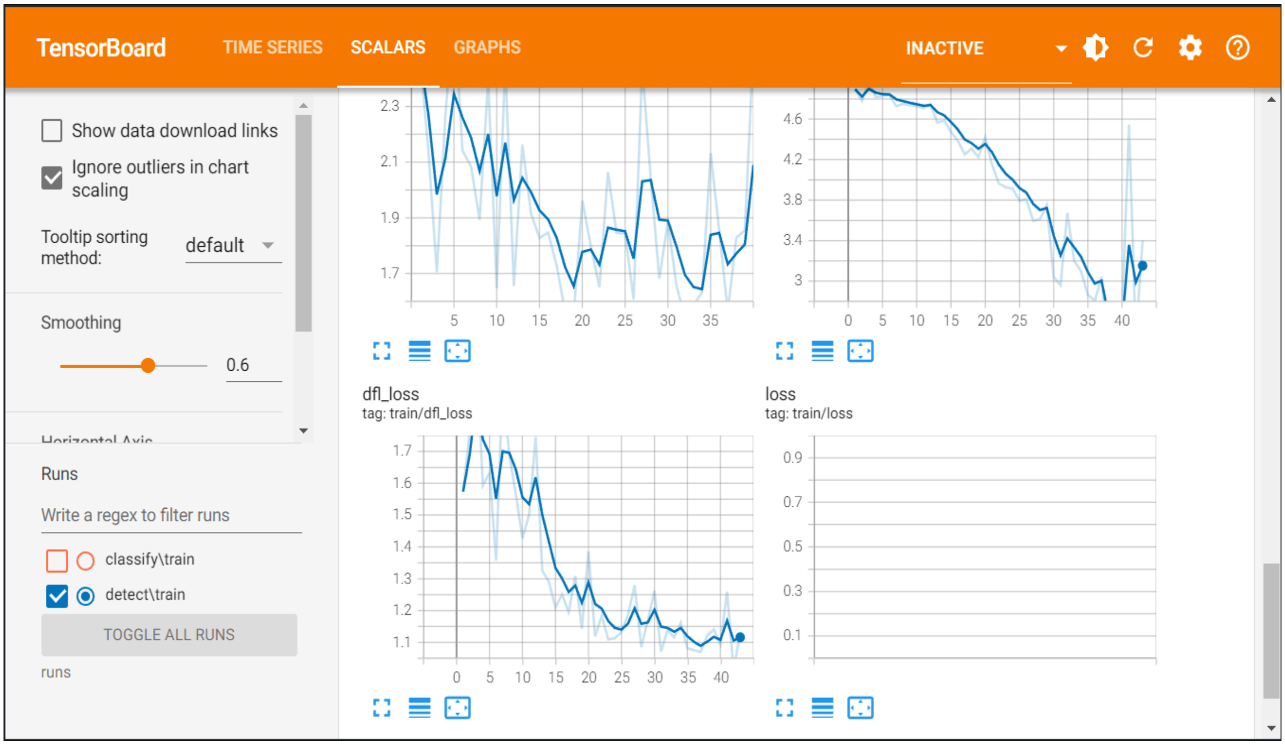
- The terminal Window will display all the data related to epochs the model is performing on the dataset.
- Once the training/running of the model is completed. The location of the results folder will be shown in the terminal window.
- The terminal window is shown on all three pages (Setup, Graph, Prediction)
- User can resize the terminal window according to their preference. User can also scroll up/down the terminal window.
- There are also options to Hide/Clear the terminal. These options are on the left side below the start button.



# Graph Page

Once the user sees the first epoch value in the Terminal window, then the user will be able to see Graph values.

Note- The graph window will only pop up once the user clicks on the start button.



□ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

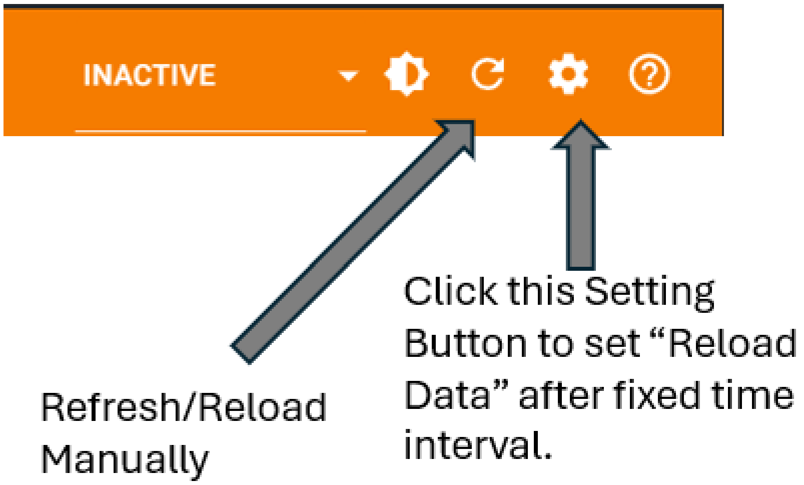
Runs

□ ○ classify\train

☑ ● detect\train

You might not be able to see the graphs right away. So follow the below steps:

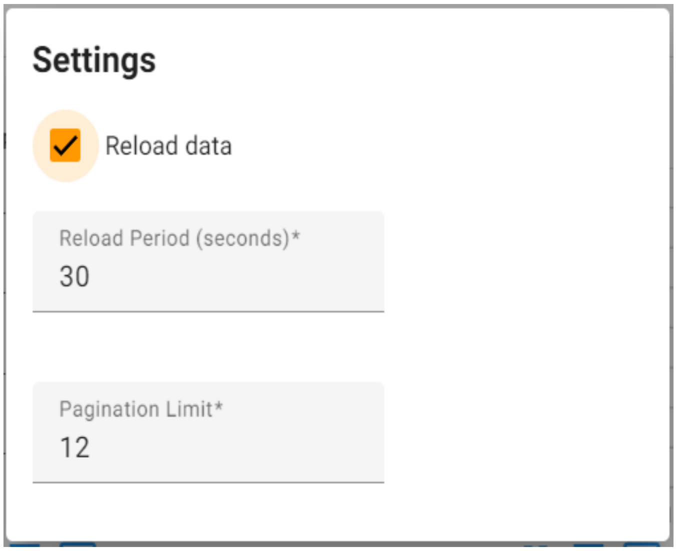
## Reload Data



Refresh/Reload Manually

Click this Setting Button to set “Reload Data” after fixed time interval.

After Clicking “Settings” Button. There will be a pop up, where user can check the reload data box. User can also set the Reload Period and Pagination Limit. The Page will refresh after the Period specified.



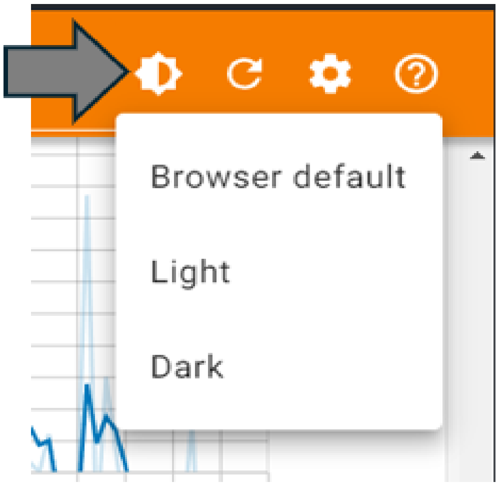
Settings

☑ Reload data

Reload Period (seconds)*
30

Pagination Limit*
12

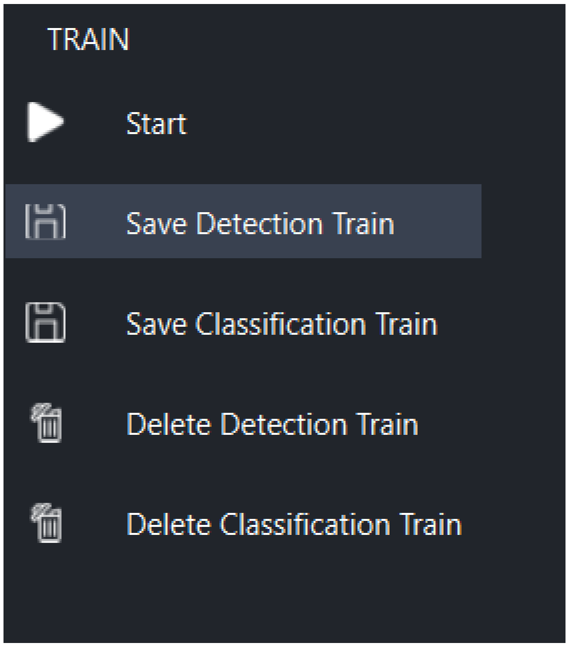
## Graph Theme



Users can change the theme of the Graph Page by clicking on the icon specified in the image above. The Graph page theme will be set to the User’s Browser default but can be changed to Light or Dark according to the user’s choice.

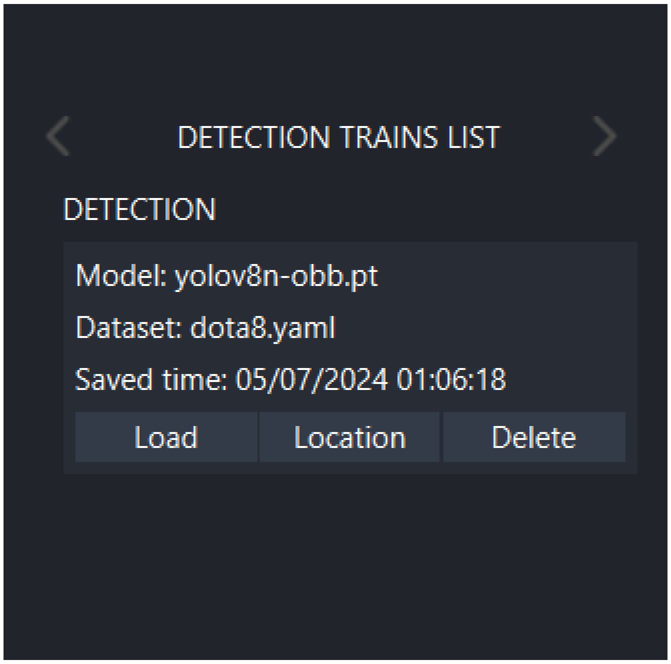
# Training Completion

After the training is completed, the User will see an option on the left side of the screen below the "**Start**" button to save or delete the classification or detection train, depending upon which is trained.



In this image, first detection model was trained, followed by a classification model.

**If the user decides to save the model, it will be displayed on the right side of the app.**



DETECTION TRAINS LIST

DETECTION

Model: yolov8n-obb.pt

Dataset: dota8.yaml

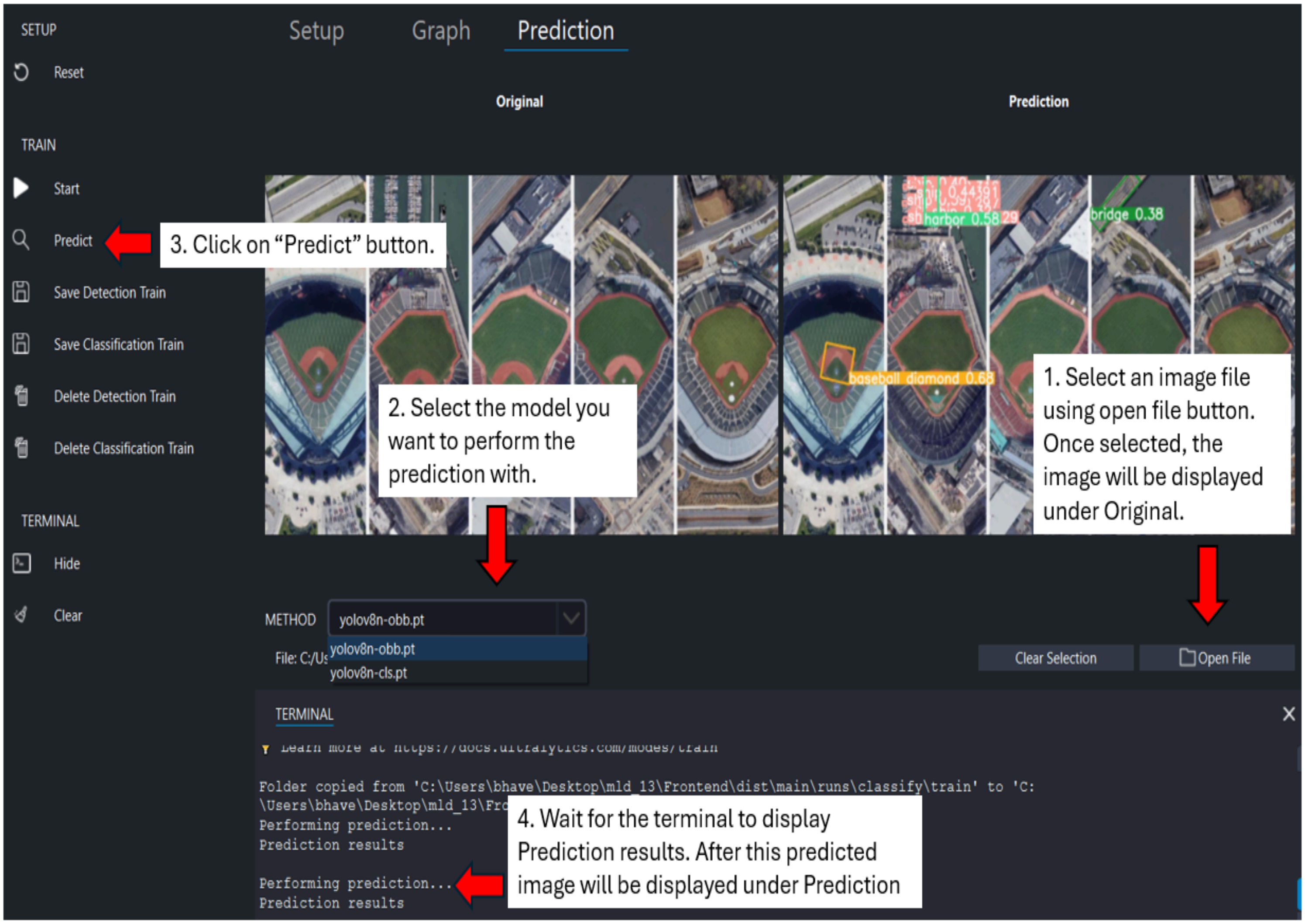
Saved time: 05/07/2024 01:06:18

- The user can go to the Classification Trains List by clicking on the right arrow next to the Detection Trains List.
- Under the list, the user can see all the models saved.
- For each model saved, it will show the model name, the dataset the model was trained on, and the time when the model was saved.
- The user further has three options:
  - Load: To load the model for prediction, discussed in the prediction page section.
  - Location: Open the location of the saved model.
  - Delete: Delete the model from the saved folder.

# Prediction Page

After the training is completed, the user can predict by going to the prediction page.

If the user clicks the load button on the detection/classification train list. It will show up under the method. It will only show one detection and one classification model and overwrites if you load another detection/classification model.



1. Select an image file using open file button. Once selected, the image will be displayed under Original.

2. Select the model you want to perform the prediction with.

3. Click on "Predict" button.

4. Wait for the terminal to display Prediction results. After this predicted image will be displayed under Prediction

```
Learn more at https://docs.ultralytics.com/modes/train

Folder copied from 'C:\Users\bhave\Desktop\mld_13\Frontend\dist\main\runs\classify\train' to 'C:
\Users\bhave\Desktop\mld_13\Fro
Performing prediction...
Prediction results

Performing prediction...
Prediction results
```

# Contact Information

Thank you for using MODLBOX. If you have any questions, concerns, or feedback, please don't hesitate to contact us. We're here to help!

Bao Nguyen
Email: smarky@iastate.edu

Bhavesh Dewan
Email: bd0106@iastate.edu

Chad Monmany
Email: cmonmany@iastate.edu

Jay Tiwari
Email: jtiwari@iastate.edu

Joao Victor Lira
Email: jlira@iastate.edu

**Response Time: Please allow up to 5 business days for a response.**