

# Training using online platforms

## Training using online platforms

### Introduction

1. Prepare training data
2. Training the model on the training platform
3. Use the code and model we downloaded
4. Effect Optimization

## Introduction

In this section, we introduce how to use the online cloud platform provided by CanMV to train models. We take the commonly used detection model as an example and let K230 recognize two items: my bracelet and screwdriver.

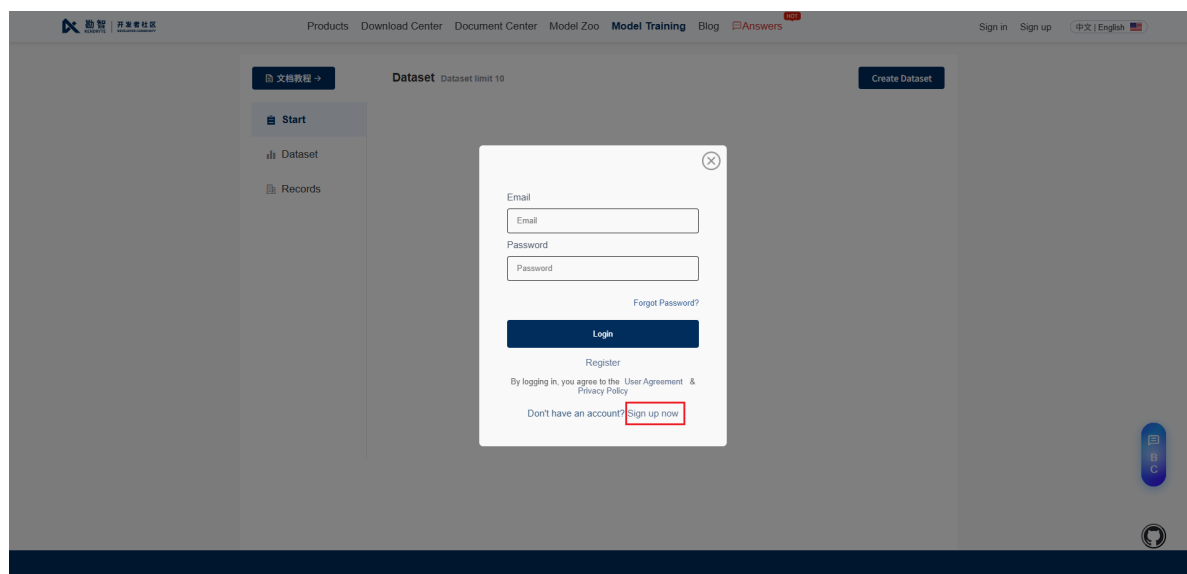
## 1. Prepare training data

We use the sample code in the [Basic Tutorial - Camera] section to take pictures. For specific operations, please refer to the tutorial in the [Basic Tutorial - Camera] section.

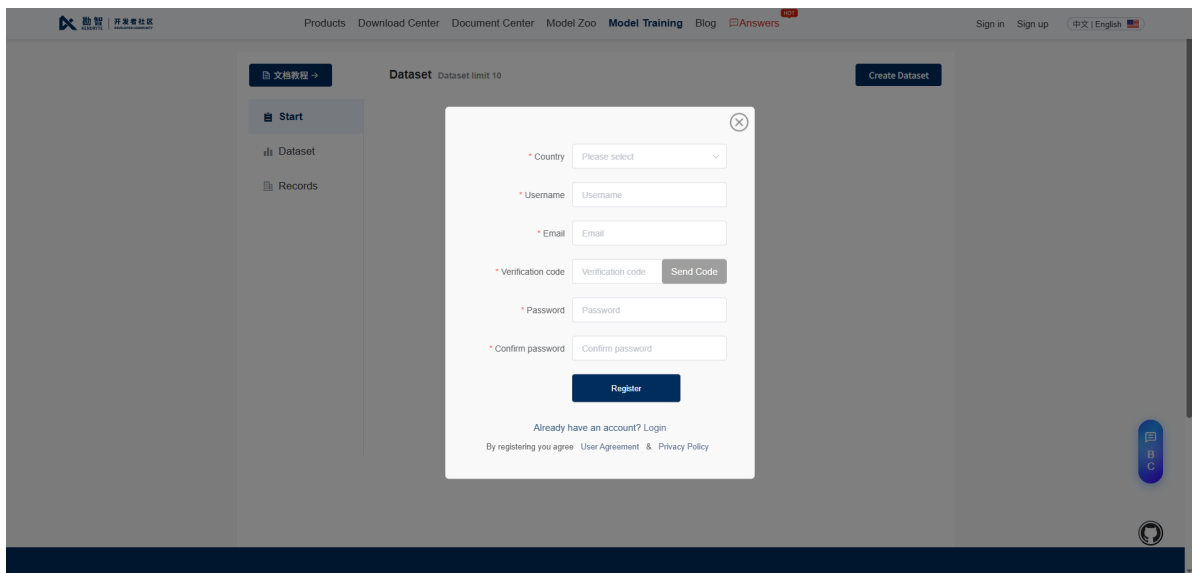
## 2. Training the model on the training platform

Online training platform address: <https://www.kendryte.com/en/training/dataset>

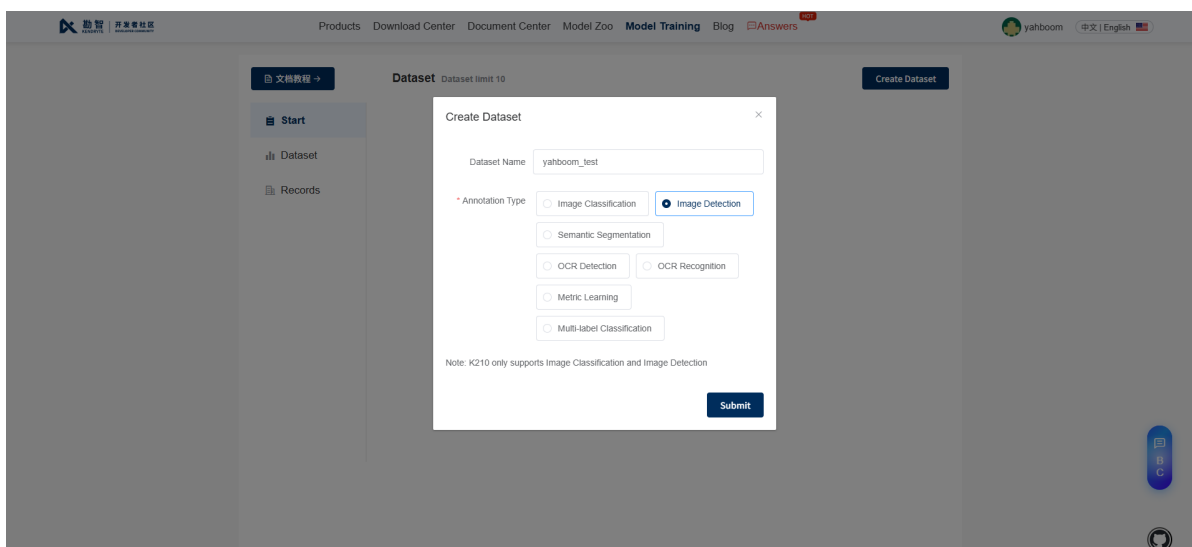
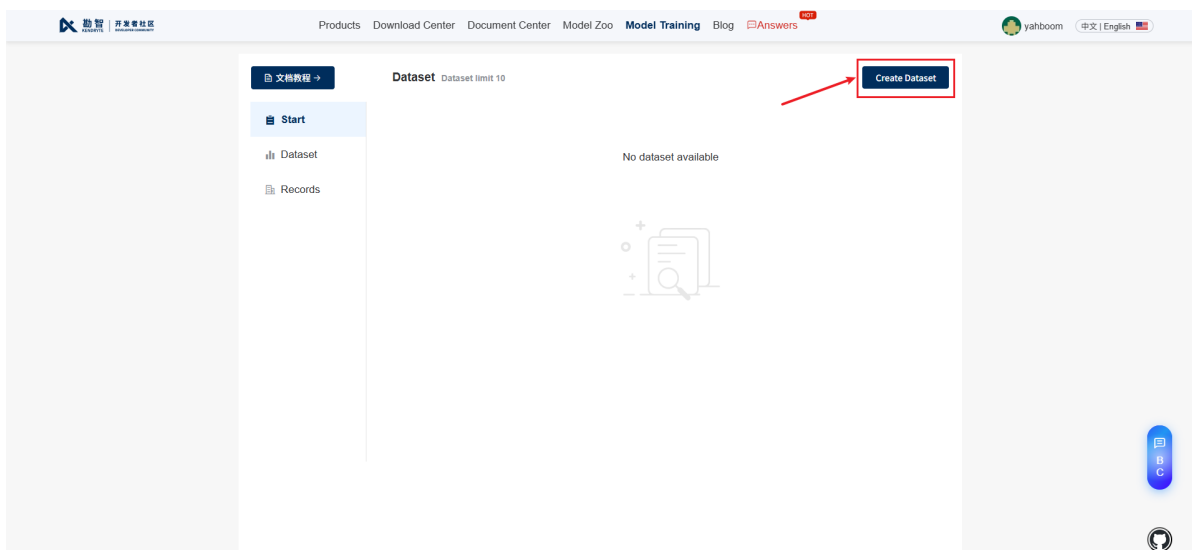
Click the Register button



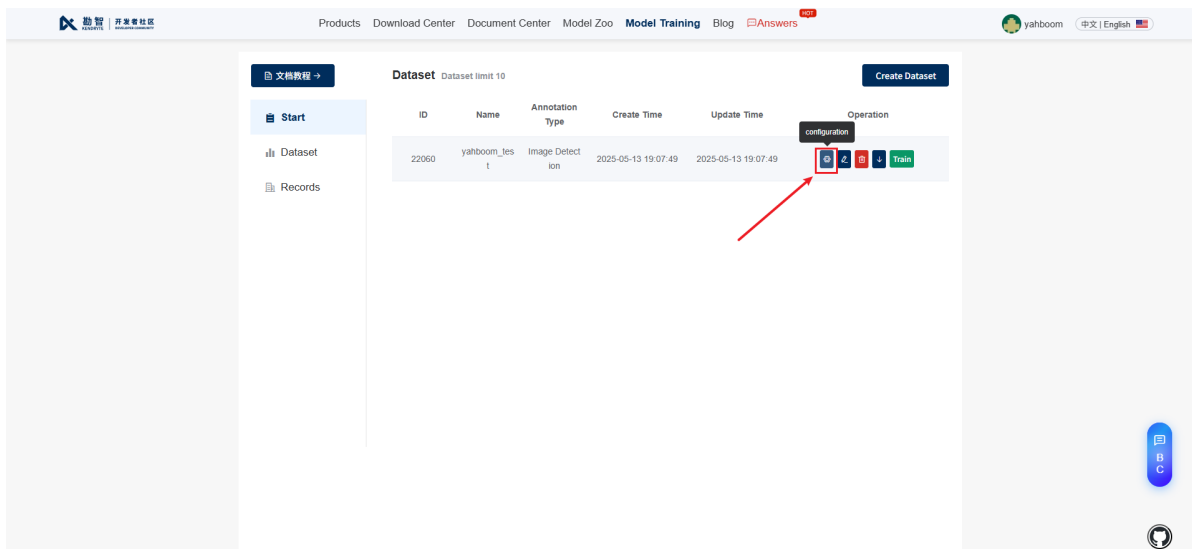
Fill in your personal information and click Register



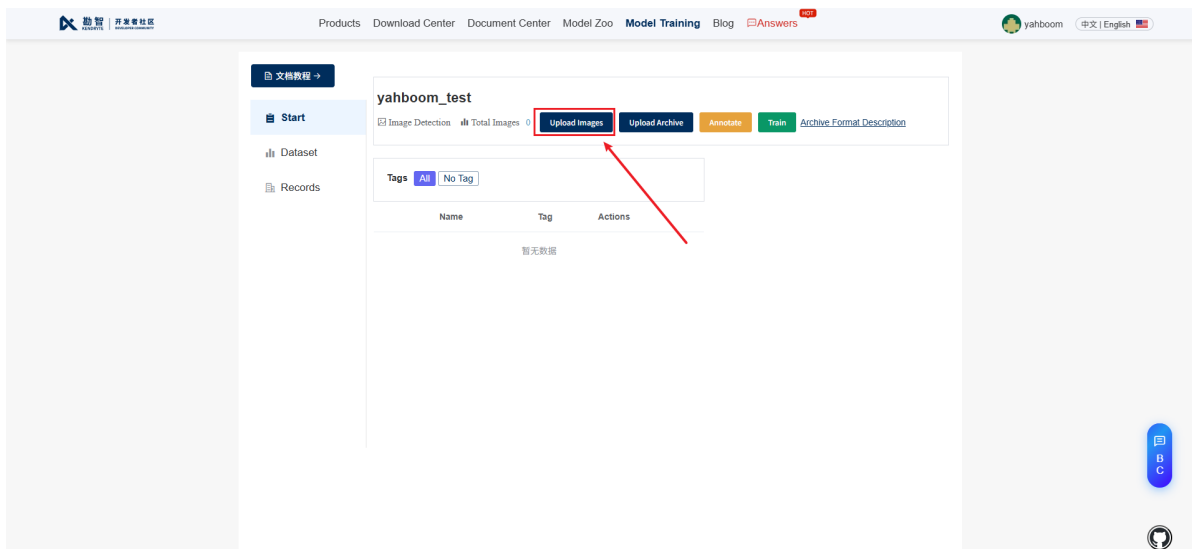
After successful registration, you will be automatically logged in. We click "Create training set"



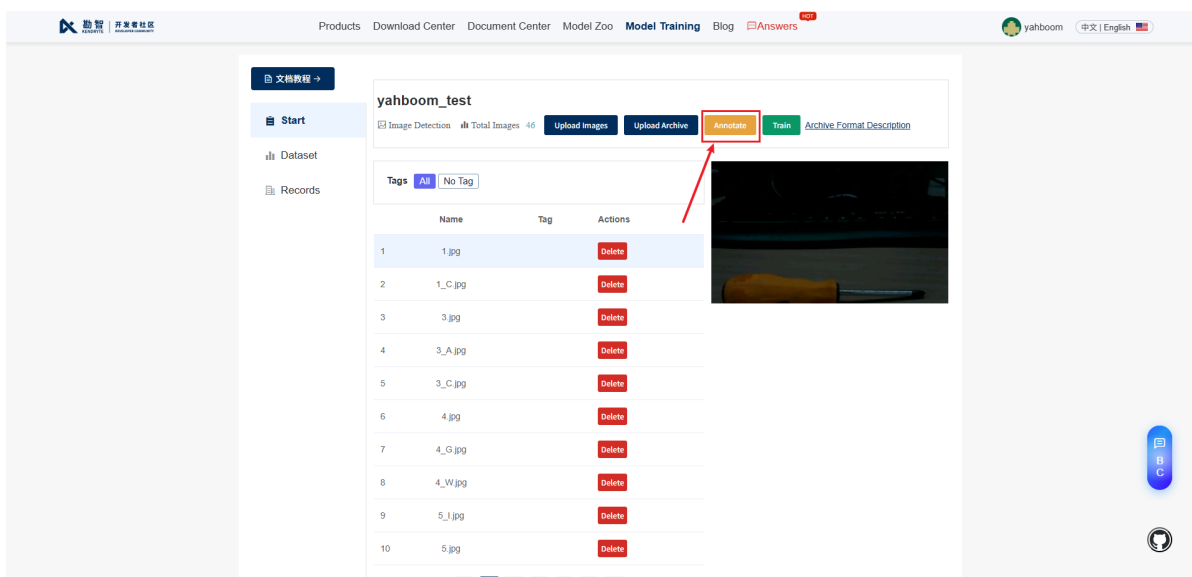
Click Edit



Click to upload image



After uploading the picture, we click the Annotate button



1. We first select the label box, enter a label name, and then press Enter.
2. Then we can use the mouse to mark the object to be detected on the picture.

文档教程 →

← 1 / 46


+ new tag

Start

Dataset

Records

Annotation steps: Select the tag and the cursor will show '+' on the image  
Deleting or modifying a callout step: Exit the annotation state and select a annotation box to delete or modify it



Exit the annotation state

Delete Ann

All (46) | Labeled (0) | Unlabeled (46)

1	1.jpg
2	1_C.jpg
3	3.jpg
4	3_A.jpg
5	3_C.jpg
6	4.jpg
7	4_G.jpg
8	4_W.jpg
9	5_I.jpg
10	5.jpg

< 1 2 3 4 5 >

3.

文档教程 →

← 1 / 46

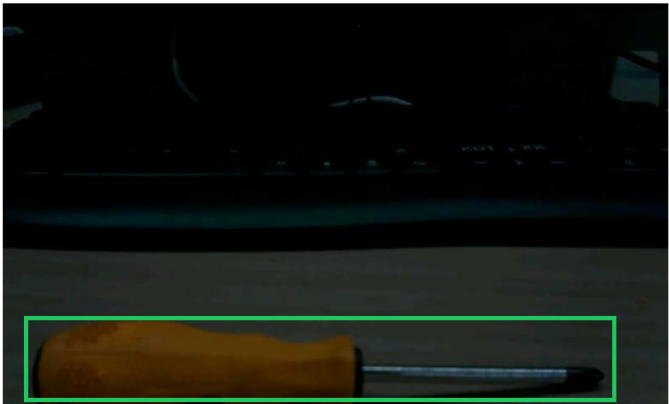
Screwdriver × + new tag

Start

Dataset

Records

Annotation steps: Select the tag and the cursor will show '+' on the image  
Deleting or modifying a callout step: Exit the annotation state and select a annotation box to delete or modify it



Exit the annotation state

Delete Ann

All (46) | Labeled (0) | Unlabeled (46)

1	1.jpg	Screwdriver
2	1_C.jpg	
3	3.jpg	
4	3_A.jpg	
5	3_C.jpg	
6	4.jpg	
7	4_G.jpg	
8	4_W.jpg	
9	5_I.jpg	
10	5.jpg	

< 1 2 3 4 5 >

文档教程 →

← 27 / 46

Screwdriver ×

Band

+ new tag

Exit the annotation state

Delete Annotation

All (46) | Labeled (46) | Unlabeled (0)

1	10.jpg	Screwdriver
2	11.jpg	Screwdriver
3	12.jpg	Screwdriver
4	6_X.jpg	Screwdriver
5	1.jpg	Band
6	2.jpg	Band
7	3.jpg	Band
8	6.jpg	Band
9	4.jpg	Band
10	10.jpg	Band


< 1 2 3 4 5 >

Start

Dataset

Records

Annotation steps: Select the tag and the cursor will show '+' on the image  
 Deleting or modifying a callout step: Exit the annotation state and select a annotation box to delete or modify it



After all the markings are completed, we click the return button

文档教程 →

← 46 / 46

Screwdriver ×

Band

+ new tag

Exit the annotation state

Delete Annotation

All (46) | Labeled (46) | Unlabeled (0)

1	17.jpg	Band
2	18.jpg	Band
3	21.jpg	Band
4	19.jpg	Band
5	20.jpg	Band
6	22.jpg	Band


< 1 2 3 4 5 >

Start

Dataset

Records

Annotation steps: Select the tag and the cursor will show '+' on the image  
 Deleting or modifying a callout step: Exit the annotation state and select a annotation box to delete or modify it



We click the training button

文档教程 →

Start

Dataset

Records

yahboom\_test

Image Detection Total Images 46

Upload Images

Upload Archive

Annotate

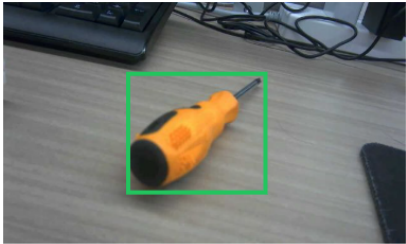
**Train**

Archive Format Description

Tags All No Tag Screwdriver Band

	Name	Tag	Actions
1	1.jpg	Screwdriver	Delete
2	1_C.jpg	Screwdriver	Delete
3	3.jpg	Screwdriver	Delete
4	3_A.jpg	Screwdriver	Delete
5	3_C.jpg	Screwdriver	Delete
6	4.jpg	Screwdriver	Delete
7	4_G.jpg	Screwdriver	Delete
8	4_W.jpg	Screwdriver	Delete
9	5_l.jpg	Screwdriver	Delete
10	5.jpg	Screwdriver	Delete

< 1 2 3 4 5 >



Fill in the content as shown in the picture and click OK

文档教程 →

Start

Dataset

Records

yahboom\_test

Image Detection Total Images 46

Upload Images

Upload Archive

Annotate

**Train**

Archive Format Description

Tags All No Tag Screwdriver Band

1	1.jpg	Screwdriver	Delete
2	1_C.jpg	Screwdriver	Delete
3	3.jpg	Screwdriver	Delete
4	3_A.jpg	Screwdriver	Delete
5	3_C.jpg	Screwdriver	Delete
6	4.jpg	Screwdriver	Delete
7	4_G.jpg	Screwdriver	Delete
8	4_W.jpg	Screwdriver	Delete
9	5_l.jpg	Screwdriver	Delete
10	5.jpg	Screwdriver	Delete

< 1 2 3 4 5 >

Create task

\* Task Name Test Bands and Screwdrivers

\* Platform ☐ k210 ☒ k230

nnCase Version 2.9.0  
Please select the version that matches the firmware

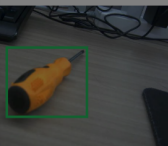
\* Iterations 240

\* Batch Size ☐ 8 ☒ 16 ☐ 24 ☐ 32

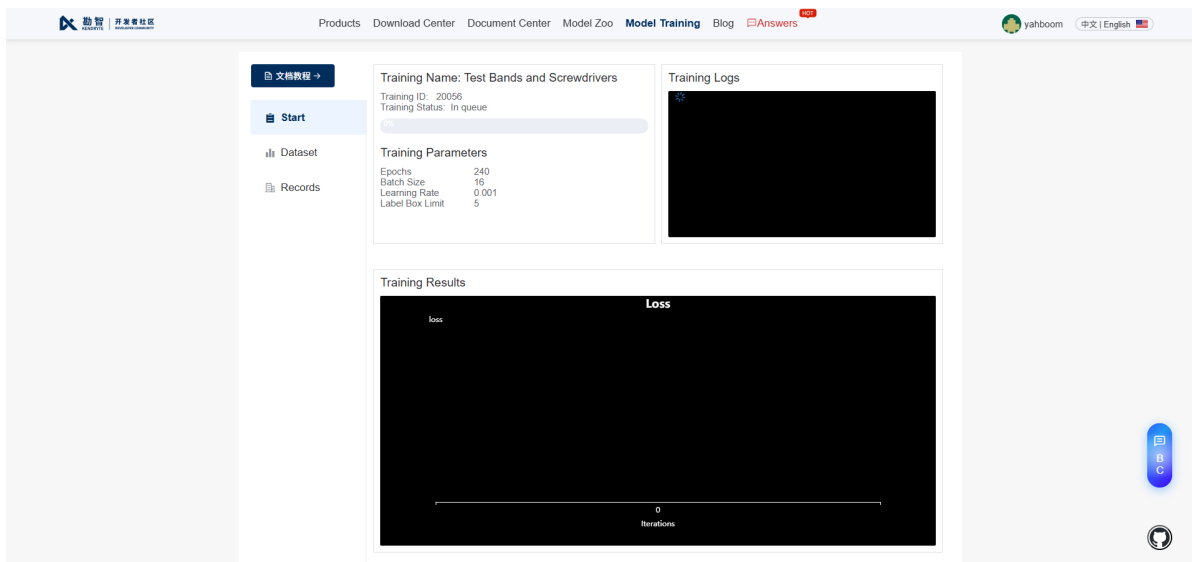
\* Learning Rate 0.001

\* Label Box Limit 5

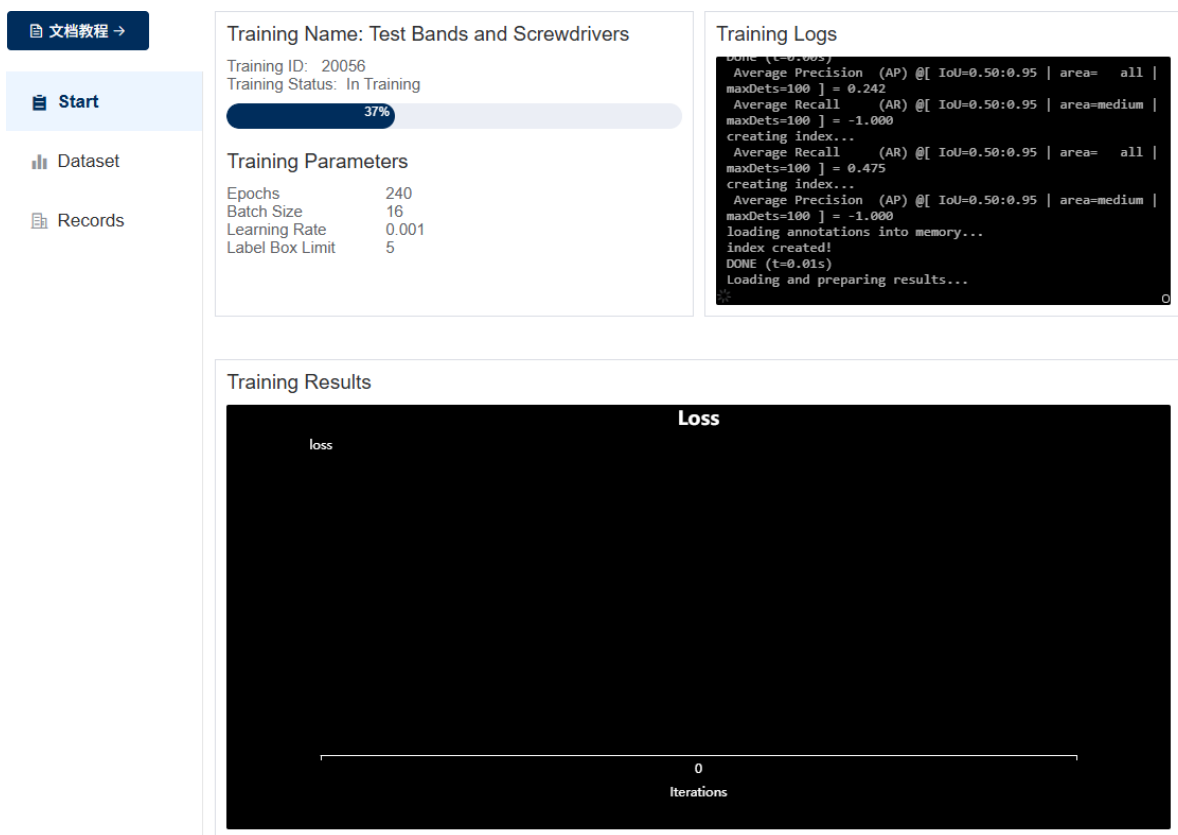
Confirm



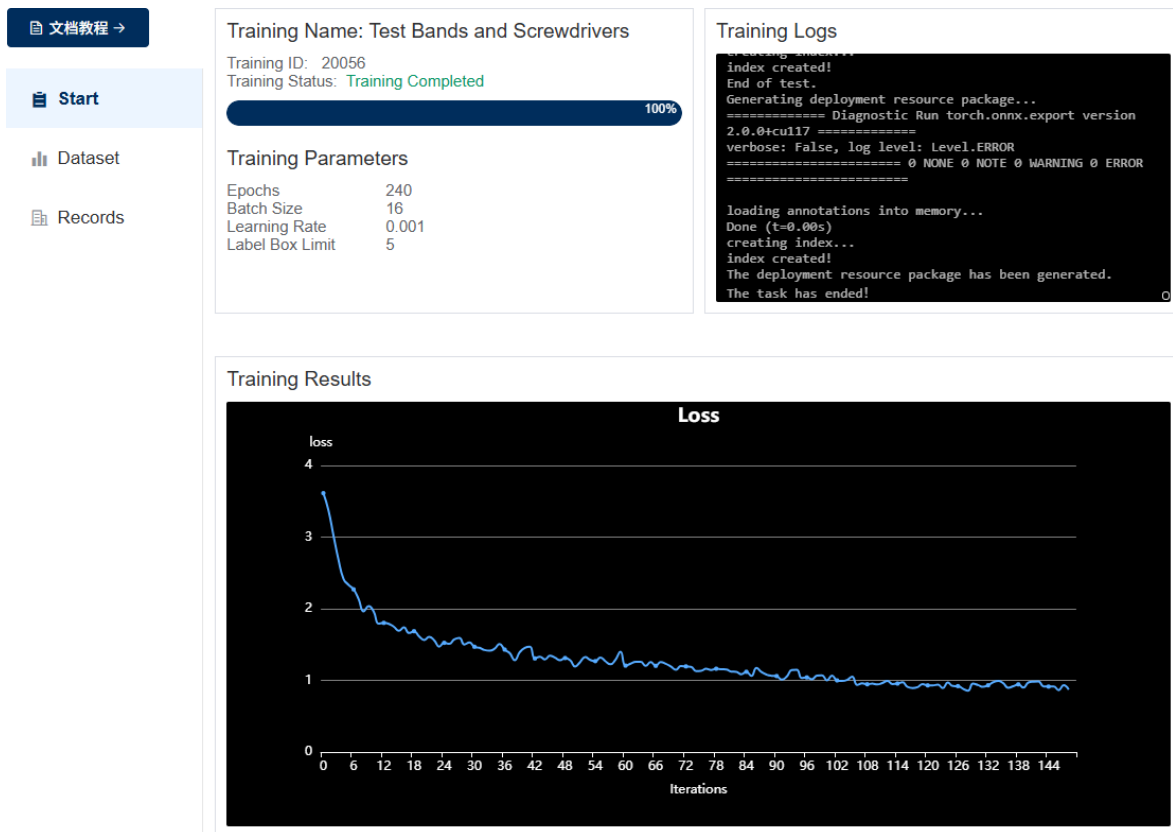
Waiting for training (need to queue if there are many people)



The training takes a long time and it is uncertain how long it will take. Closing the webpage or turning off the computer will not affect the training. You can do other things first.



After waiting for the training to end, the interface is as follows



We return to the place before starting training, find the training record, and click Download

文档教程 →

Start

Dataset

Records

Training Record

Tip: 1. After the training is completed, the model will be automatically sent to the registered email.  
2. Each user supports only one training task running at the same time.  
3. The maximum training duration for a single task is 24 hours. If it exceeds the time, the task will be terminated early according to the server resource situation.  
4. The training duration is strongly related to the number of files and training iterations of your dataset. If the training time is too long, consider reducing the corresponding quantity.

ID	Name	Dataset	Type	Chip	Status	Update Time	Operation
20056	Test Bands and Screwdrivers	yahboom_test	Image Detection	k230	✓ Completed	2025-05-13 1...	<div>DetailDownloadDelete</div>

### 3. Use the code and model we downloaded

The downloaded file is a compressed file with the following contents

Test Bands and Screwdrivers.zip - ZIP 压缩文件, 解包大小为 64,778,111 字节

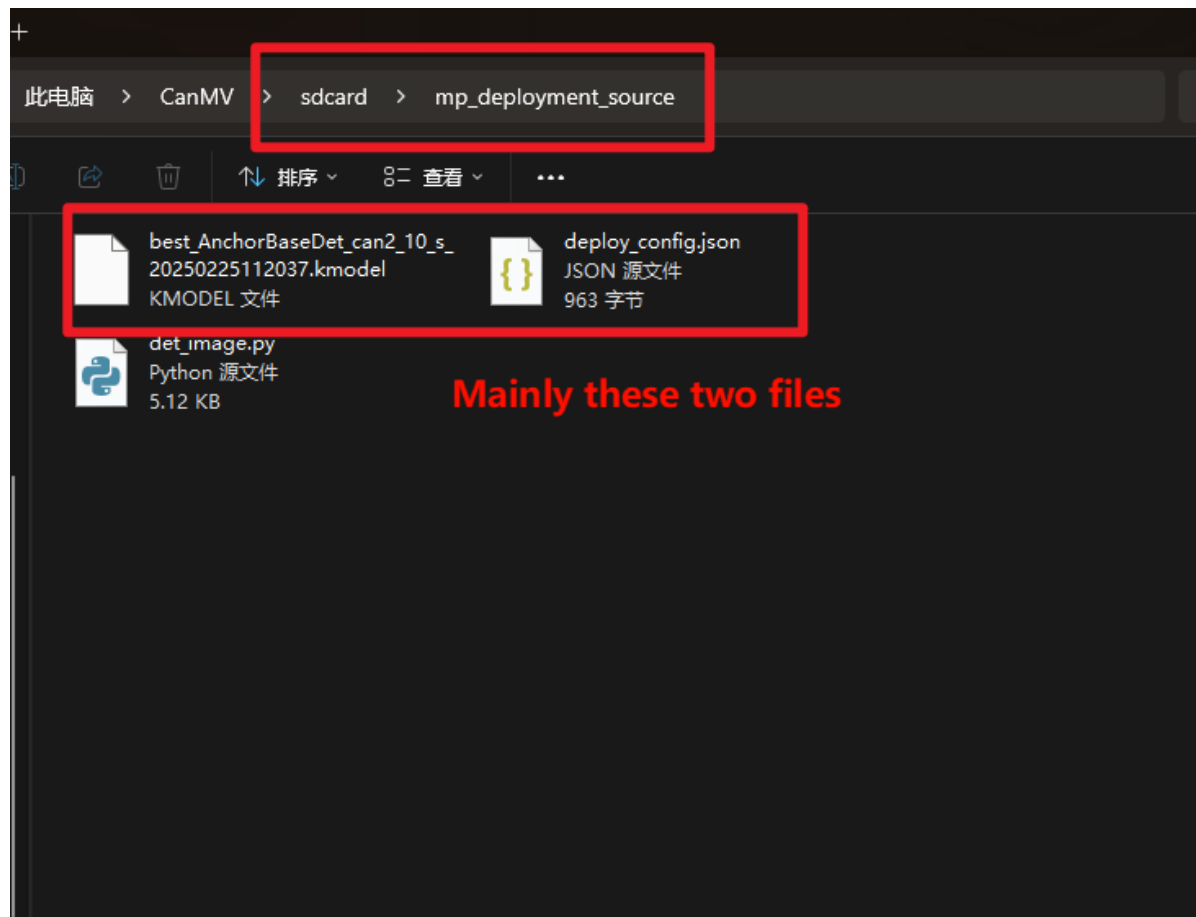
名称	大小	压缩后大小	类型	修改时间	CI
..			File folder		
det_results	126,479	126,479	File folder		
cpp_deployment_source.zip	58,384,570	58,384,570	WinRAR ZIP 压缩...	2/25/2025 7:38 ...	4C
mp_deployment_source.zip	6,257,196	6,257,196	WinRAR ZIP 压缩...	2/25/2025 7:38 ...	7F
README.md	9,866	9,866	Markdown File	2/25/2025 7:38 ...	3E

We can first look at the det\_results folder, which contains the results of the verification test. If the objects in the picture are correctly framed, then basically there is no big problem with the trained model.



Next, we unzip mp\_deployment\_source and create a mp\_deployment\_source folder in the /sdcard/ directory of K230, and put the unzipped contents into it.

(Mainly put the model file ending with .kmodel and the configuration file ending with .json in it. The remaining two py files are used for running and can be left out)



Then we open CanMV IDE, copy the code of det\_video.py and run it, and we can see that K230 can detect the objects we just marked.

Here we need to make some modifications to det\_video.py:

The original code is as follows:

```
display_mode="hdm"
if display_mode=="lcd":
    DISPLAY_WIDTH = ALIGN_UP(800, 16)
    DISPLAY_HEIGHT = 480
else:
    DISPLAY_WIDTH = ALIGN_UP(1920, 16)
    DISPLAY_HEIGHT = 1080

OUT_RGB888P_WIDTH = ALIGN_UP(1280, 16)
OUT_RGB888P_HEIGHT = 720
```

Modified to:

```
display_mode="lcd"  
if display_mode=="lcd":  
    DISPLAY_WIDTH = ALIGN_UP(640, 16)  
    DISPLAY_HEIGHT = 480  
else:  
    DISPLAY_WIDTH = ALIGN_UP(1920, 16)  
    DISPLAY_HEIGHT = 1080  
  
OUT_RGB888P_WIDTH = ALIGN_UP(1280, 16)  
OUT_RGB888P_HEIGHT = 720
```



## 4. Effect Optimization

At this time, we find that there are too many overlapping frames, so we can try to reduce the nms threshold in deploy\_config, so that there will be fewer duplicate frames.

```
{} deploy_config[1].json X
C: > Users > Administrator > AppData > Local > Microsoft > Windows > INetCache > IE > PZ9UWQH3 > {} deploy_config[1].json
1  {
2    "chip_type": "k230",
3    "inference_width": 640,
4    "inference_height": 640,
5    "confidence_threshold": 0.25,
6    "nms_threshold": 0.4,
7    "mncase_version": "2.9.0",
8    "model_type": "AnchorBaseDet",
9    "img_size": [
10     640,
11     640
12   ],
13   "anchors": [
14     [
15       191,
16       151,
17       255,
18       169,
19       239,
```

The default is 0.6, which causes duplicate boxes to appear.

