SIYI

UniAI Studio
User Manual

SIYI Technology (Shenzhen) Co., Ltd.

SIYI.biz/en

Thank you for choosing a product from SIYI Technology.

UniAI Studio is a professional platform for model development and deployment, intended for users with basic technical knowledge. Please read the user guide carefully before use to ensure proper operation.SIYI Technology is not responsible for data loss, system errors, or other consequences caused by misuse or improper operation.This commercial software is licensed only for lawful industrial and research use. Military or illegal applications are strictly prohibited. Reverse engineering, modification, copying, or distribution without written consent from SIYI Technology is forbidden.

This manual addresses most common usage questions. For additional assistance, please visit the official SIYI Technology website at www.siyi.biz, call our support hotline at 400-838-2918, or email our engineering support team at support@siyi.biz for product inquiries or technical support.

SIYI

CONTENT

Contact Us: **SIYI Official Website** *(https://SIYI.biz/en)*

SIYI User Group - Facebook	
Facebook	
Linkedln	
YouTube	

SIYI

User Manual Update Log

Version	Date	Updates
1.0	2025.05	Original version
1.1	2025.06	Added the list of supported device models

1 Product Overview

UniAI Studio is a software platform that enables users to train, evaluate, quantize, and deploy AI models with ease. By leveraging a user-friendly GUI, users can complete the full workflow without needing to manage underlying technical complexities. The platform comes pre-installed with a Python environment and essential deep learning frameworks. To run UniAI Studio, the host machine only requires a Linux operating system and NVIDIA GPU drivers.

1.1 Product Summary

Purpose: Supports AI model training, evaluation, quantization, and deployment on the platform.

Core Functions: Model training, model evaluation, model validation, and model export.

Supported Platforms: Host machines running Linux OS with NVIDIA GPUs.

1.2 User Guide Structure

Introduction: Covers software installation and uninstallation (environment setup), module overviews, and usage instructions.

Intended Audience: Users with a working knowledge of Linux systems

and AI-related technologies.

2 Installation and Configuration

2.1 System Requirements

The system requires a CPU with 6 cores or more, at least 16 GB of physical memory, 16 GB or more of swap memory, and at least 100 GB of storage space. The operating system must be Ubuntu 22.04 or higher. The GPU must have at least 8 GB of VRAM and a compute capability of 7.5 or higher. GPU computing power reference links, refer to:

https://developer.nvidia.cn/cuda-gpus#compute

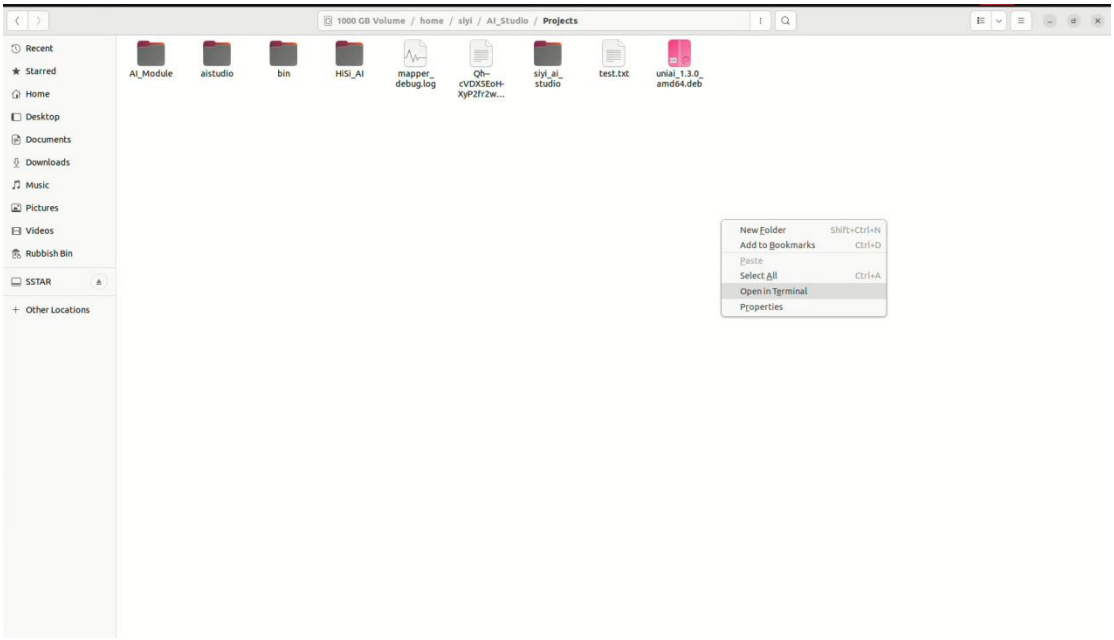
The GPU driver must be CUDA version 12.8 or later. Reference:

https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html#id7

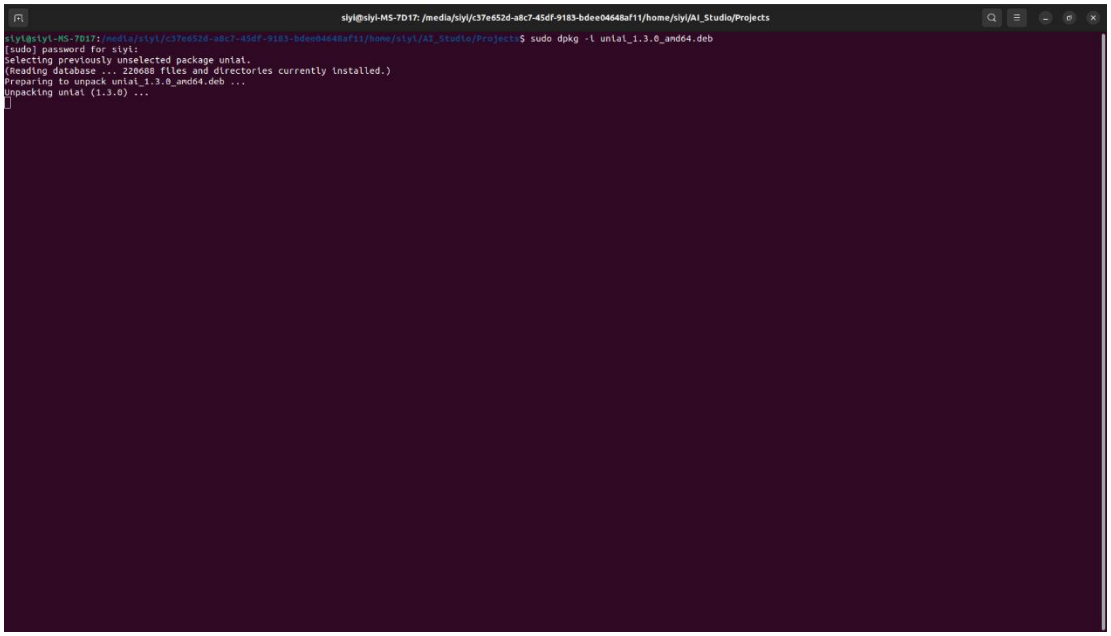
Driver download link: https://www.nvidia.cn/drivers/unix/

2.2 Installation Steps

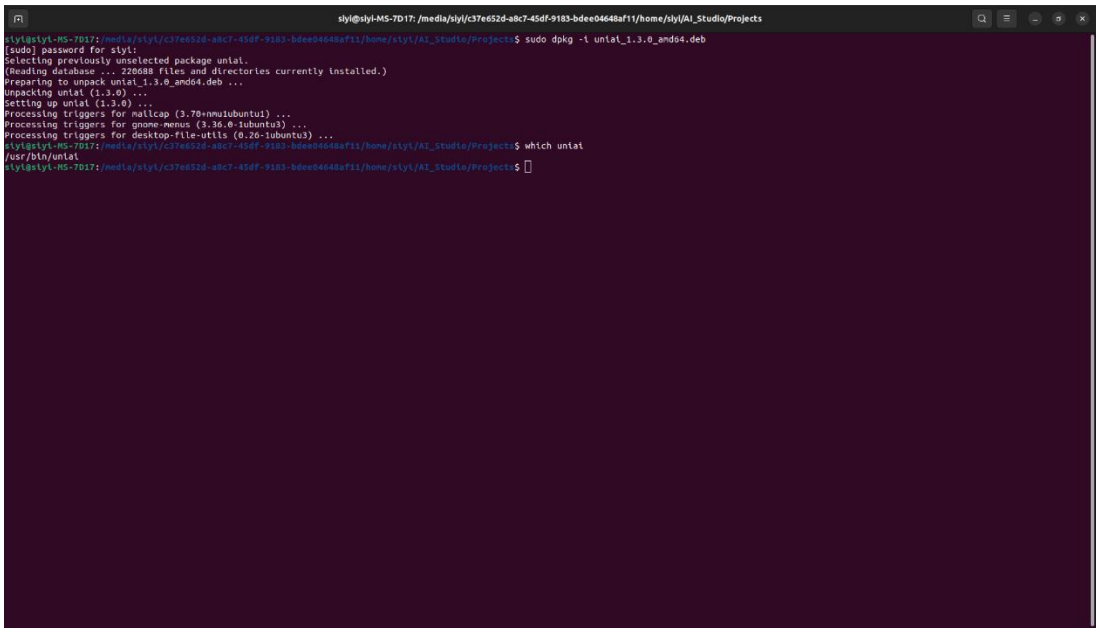
Installation: Open a terminal in the installation package directory and enter the command to proceed with installation.



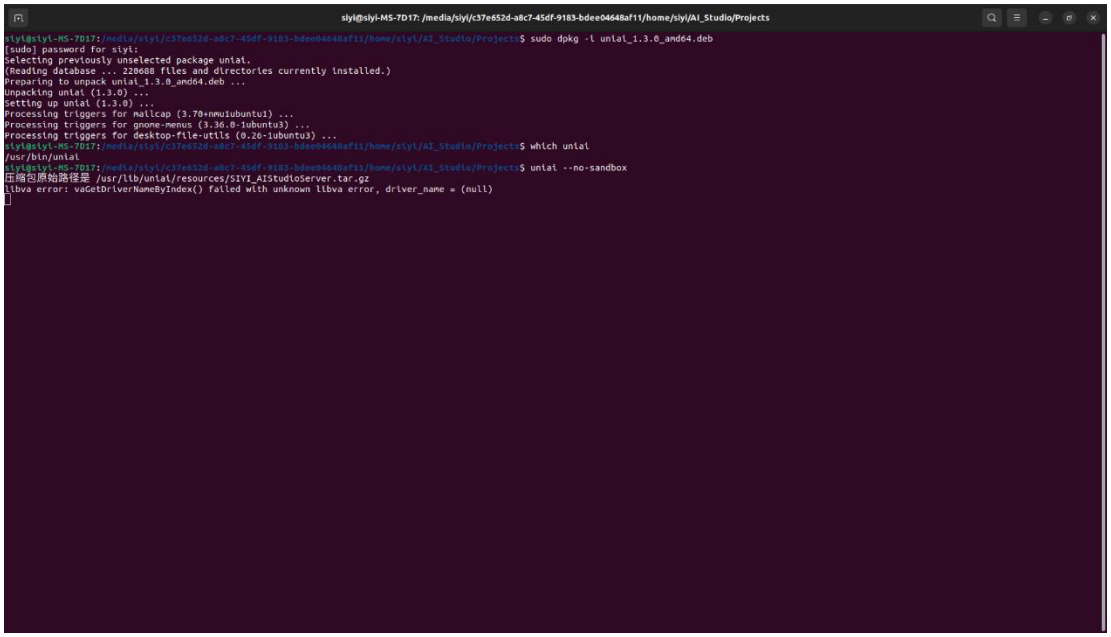
sudo dpkg -i uniai_1.3.0_amd 64.deb



which uniai

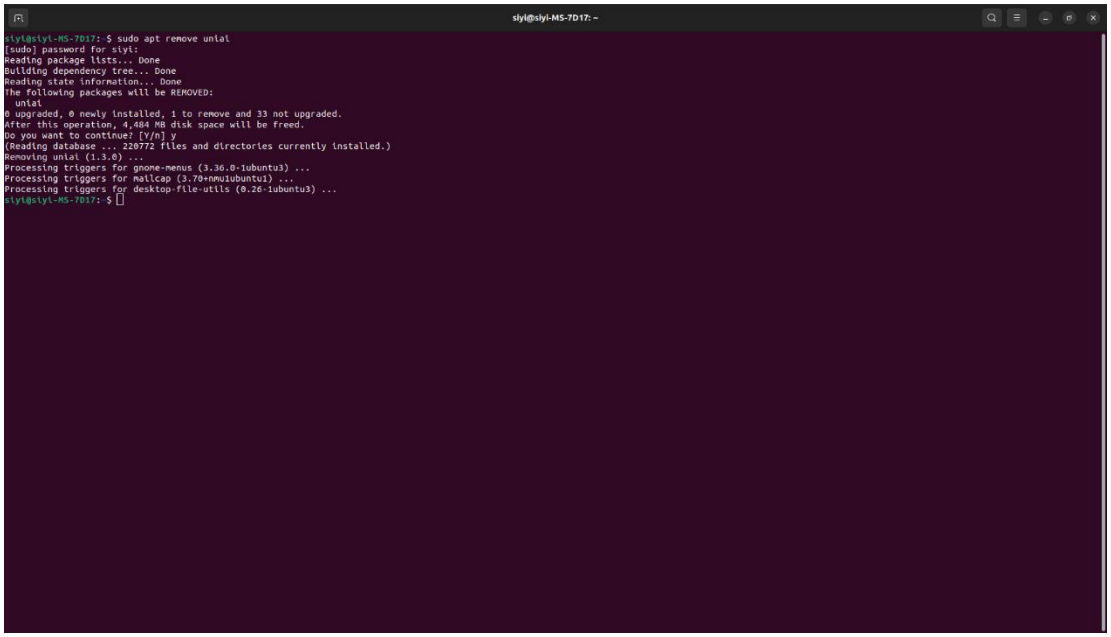


uniai --no-sandbox

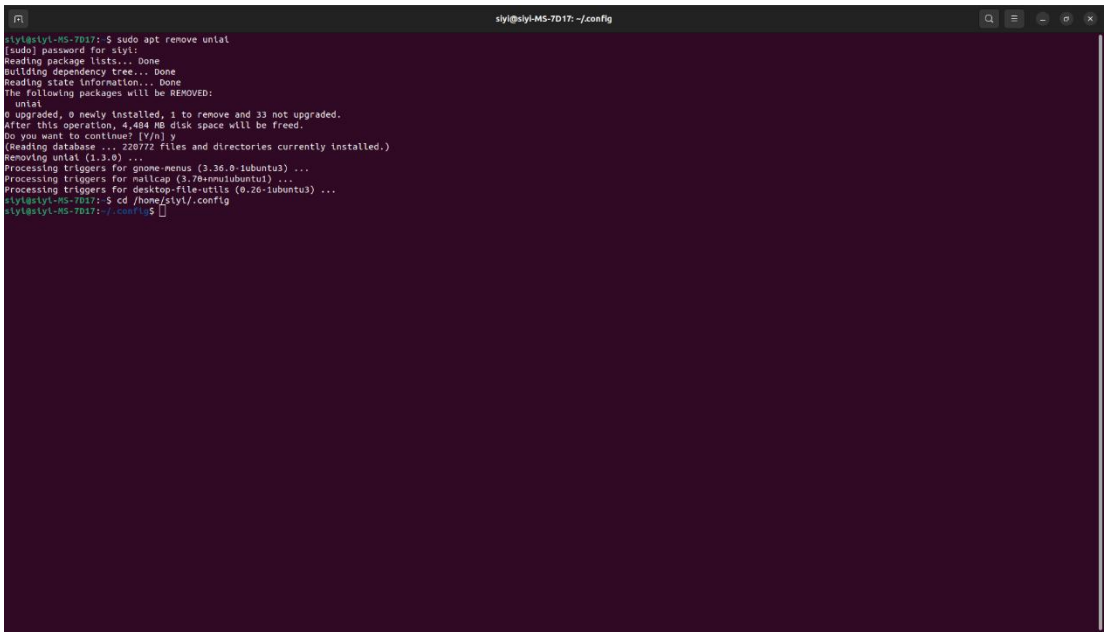


Uninstallation

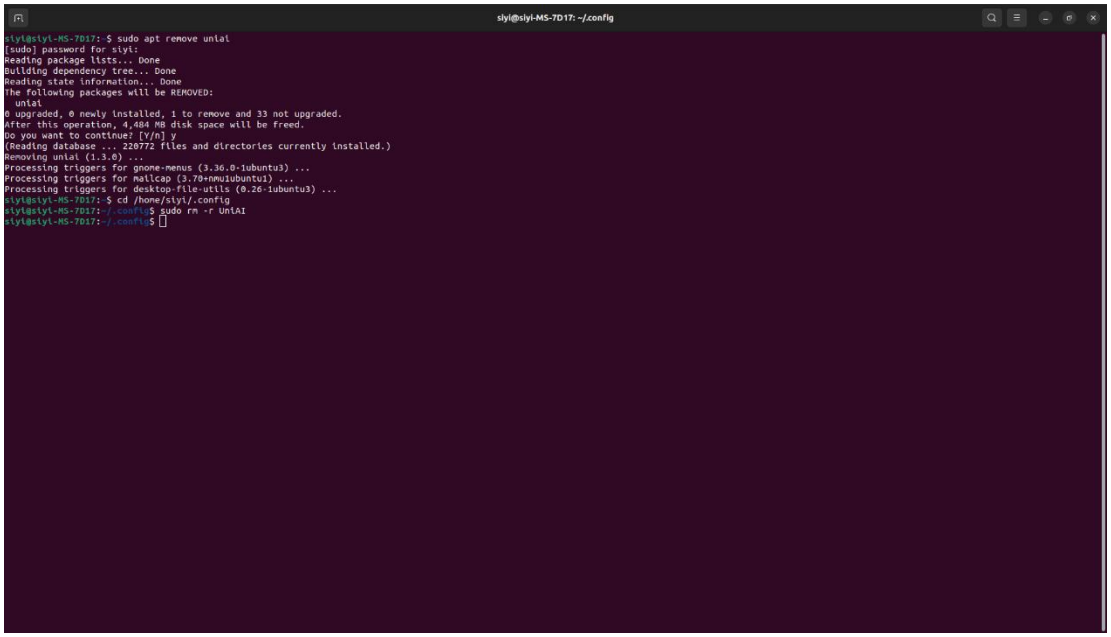
sudo apt remove uniai



cd ~/.config



sudo rm -r UniAI



2.3 Supported Device Models

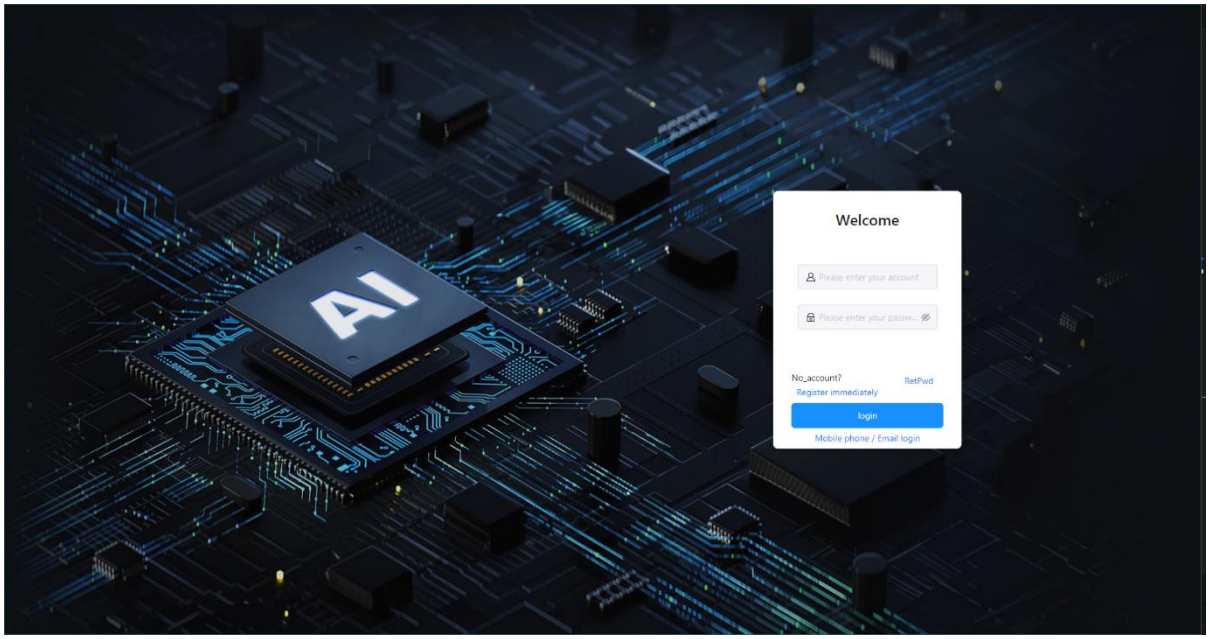
Only the SIYI AI Tracking Module Ⅱ is supported now.

3 Quick Start

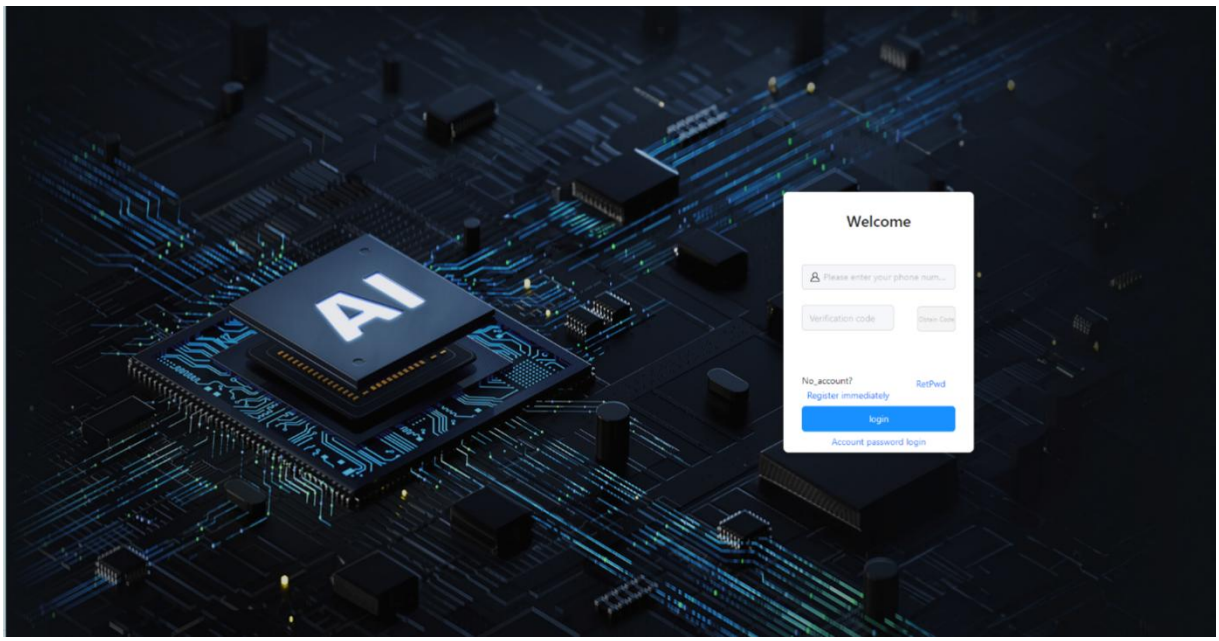
3.1 User Login

3.1.1 Login with Username and Password

Launch the application. The default screen will display the username and password login interface.



3.1.2Login with Phone Number or Email

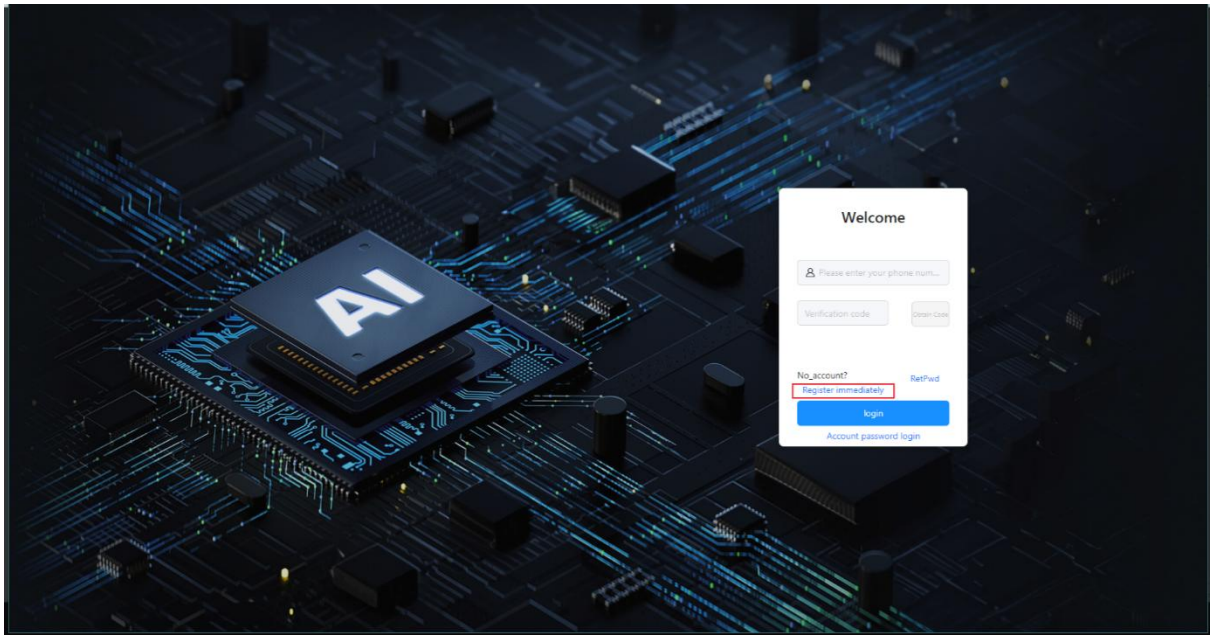


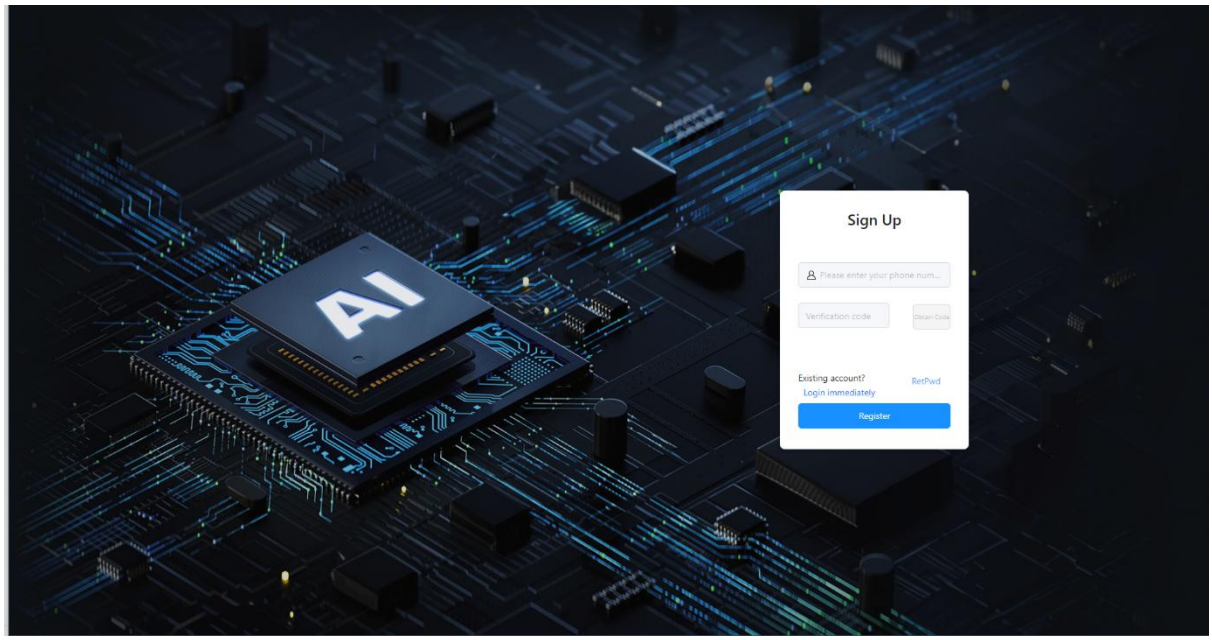
Click on "Login with Mobile/Email" to switch to the mobile number or email login interface.

3.1.3 Account Registration

If you don’t have an account, click “Register Now,” enter your mobile

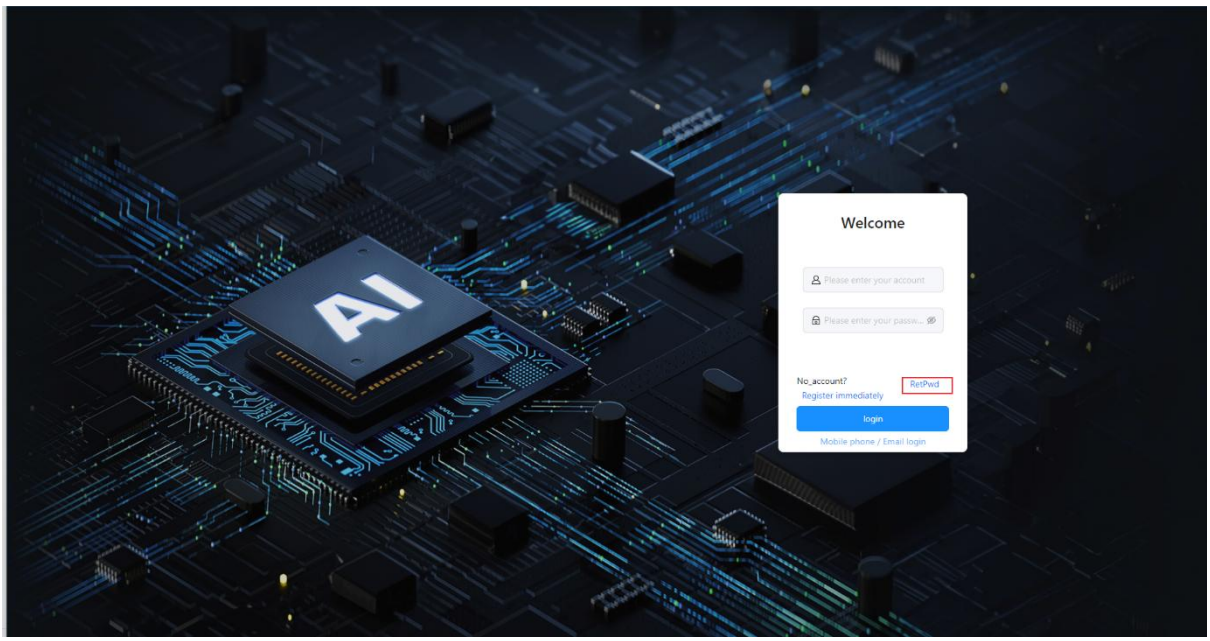
number or email, click to get the verification code, and follow the prompts to complete the registration. Once registered, enter your username and password to log in.

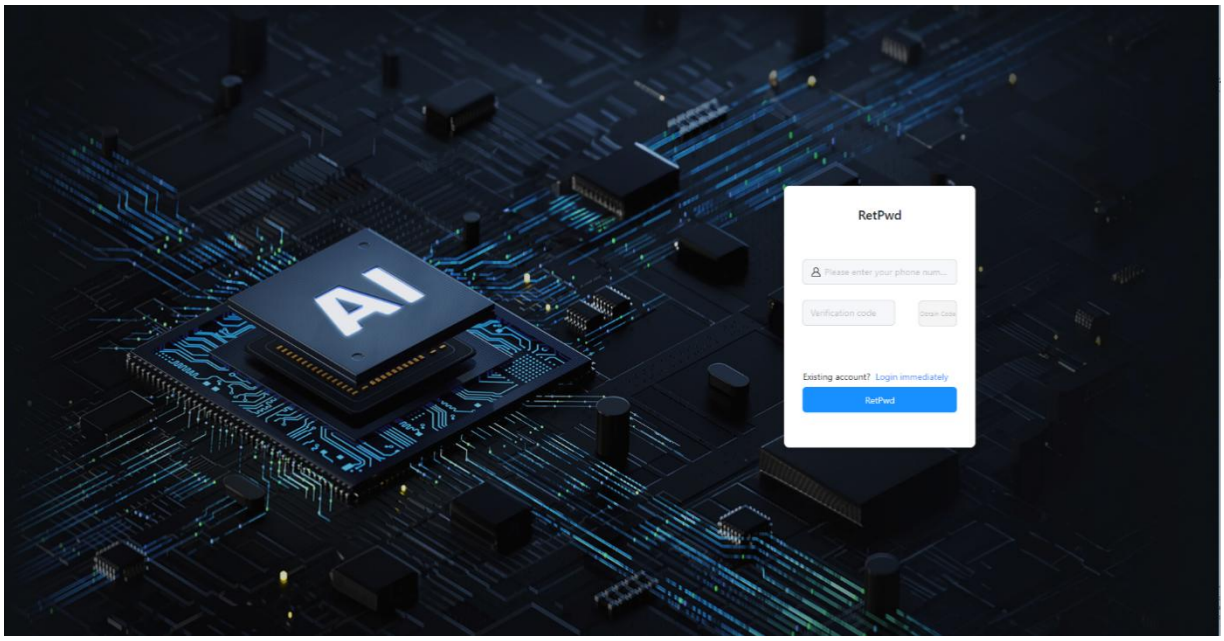




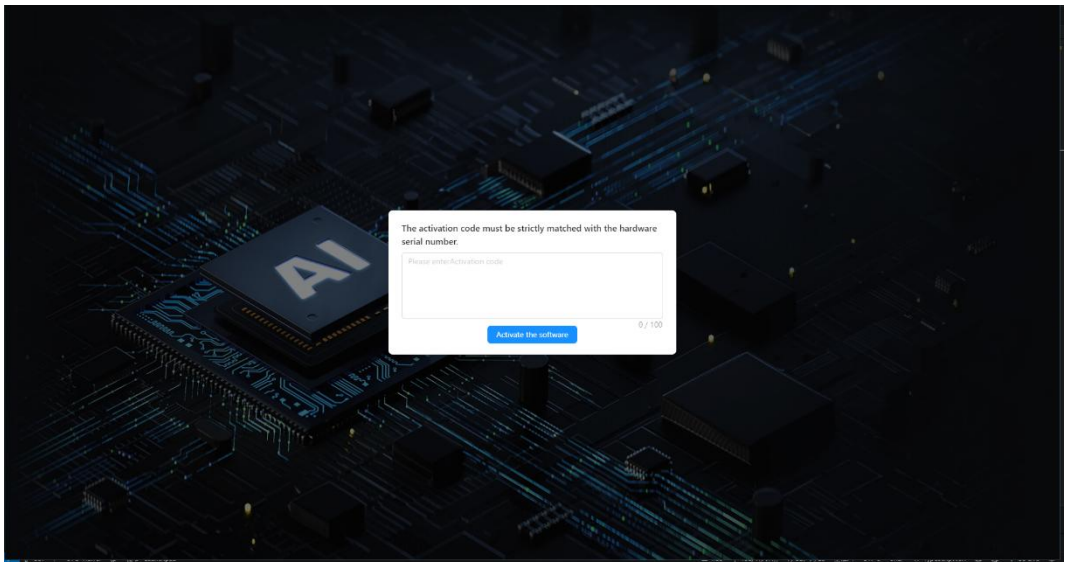
3.1.4 Password Recovery

Click “Forgot Password” and follow the prompts to enter a new password.



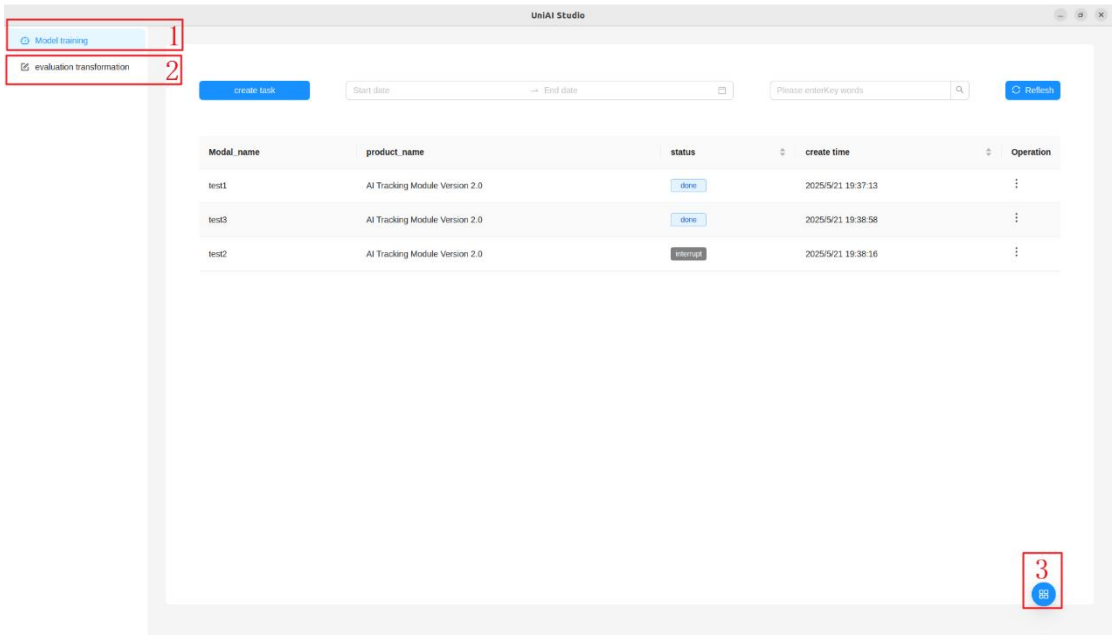


3.1.5 Activation Code



For first-time use, please enter the activation code. The software can only be used normally after the activation code is successfully verified.

3.2 Main Interface Overview



1.Model Training: Includes task editing and creation, task start and pause, task monitoring, task status viewing, and task filtering functions.

2.Model Evaluation and Conversion: Includes model evaluation, model validation, and model export functions.

3.Personalized Settings: Includes day/night mode switching and language switching between Chinese and English.

4 Core Function Usage Guide

This section introduces the functionalities and usage methods of model training, model evaluation, model validation, and model export.

4.1 Model Training

Definition: Model training involves using labeled data to train a machine learning or deep learning model, enabling it to learn features and patterns from the data and ultimately generate a function (model) capable of making predictions or classifications.

Key Processes:

Data Input: Feed the preprocessed training dataset (training set) into the model.

Parameter Optimization: Adjust model parameters using algorithms such as backpropagation and gradient descent to minimize prediction error (loss function).

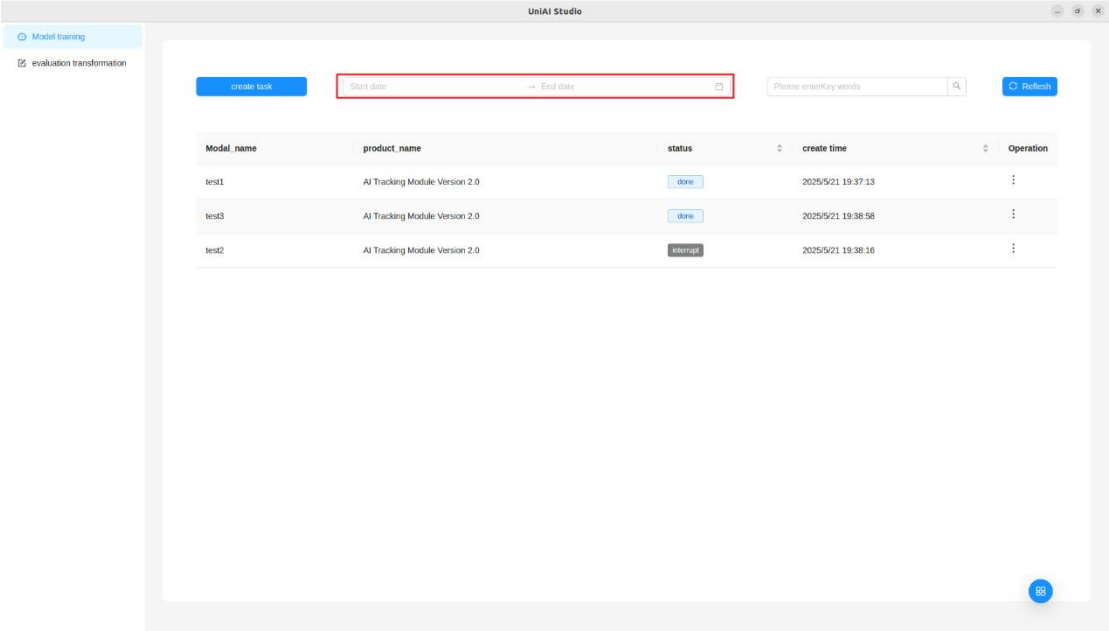
Iterative Updates: Improve the model's ability to fit the data over multiple training cycles (epochs).

Example: When training an image classification model, features are extracted via convolutional layers, classification is performed by fully connected layers, and weights are adjusted using an optimizer (such as Adam).

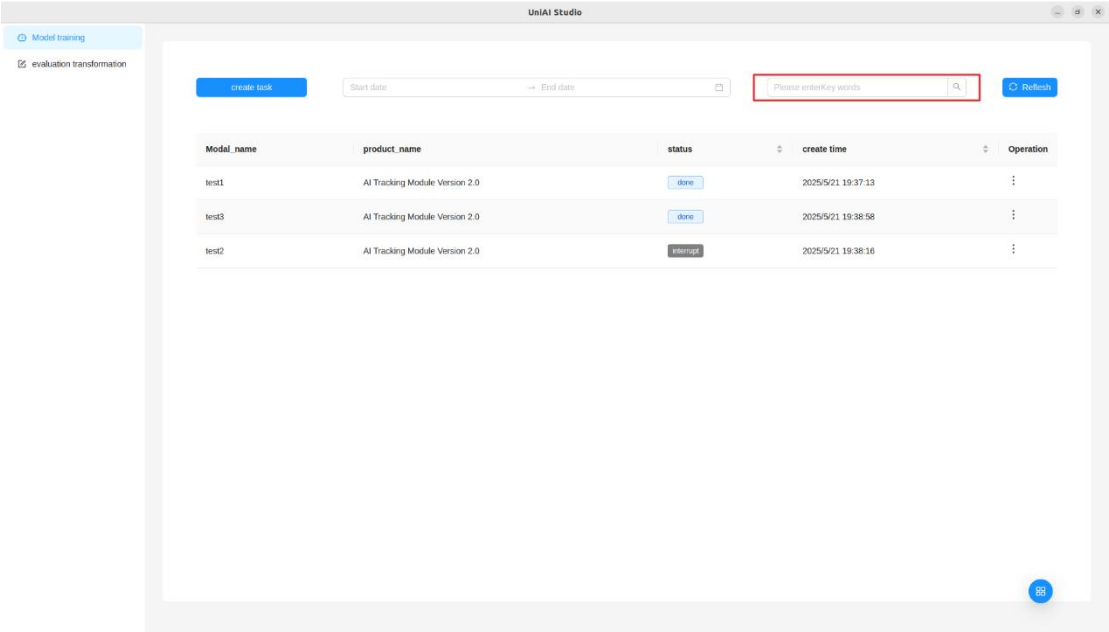
4.1.1 Model Training

4.1.1.1 Model Training List

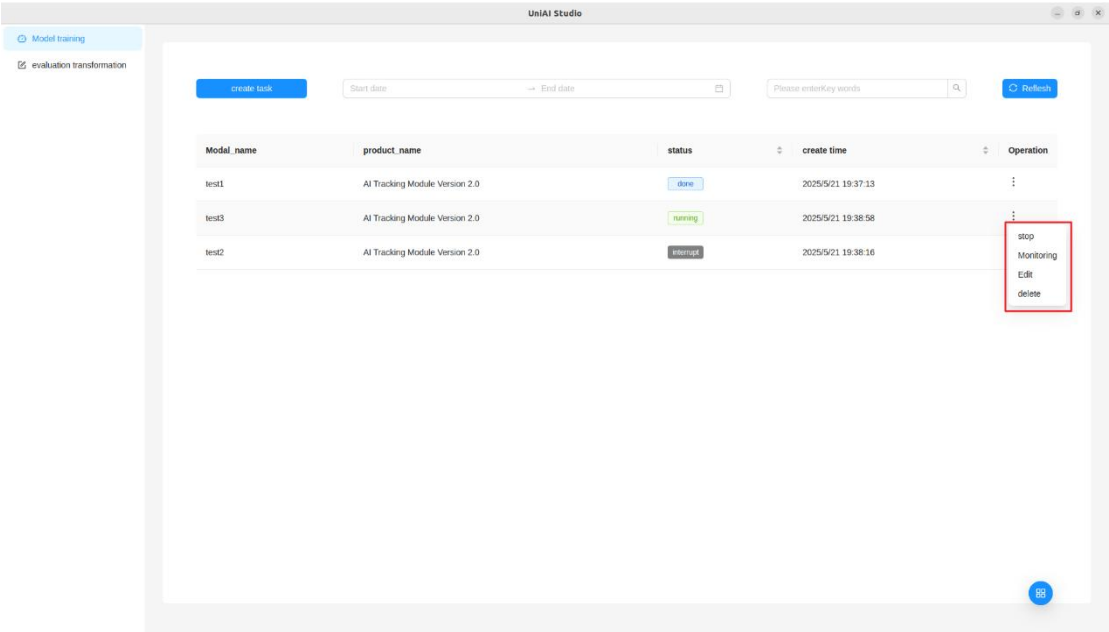
1.Task Filtering (Time Filter): Filters tasks within the selected time range.



2.Task Filtering (Name Filter): Filters tasks that contain the specified model name.

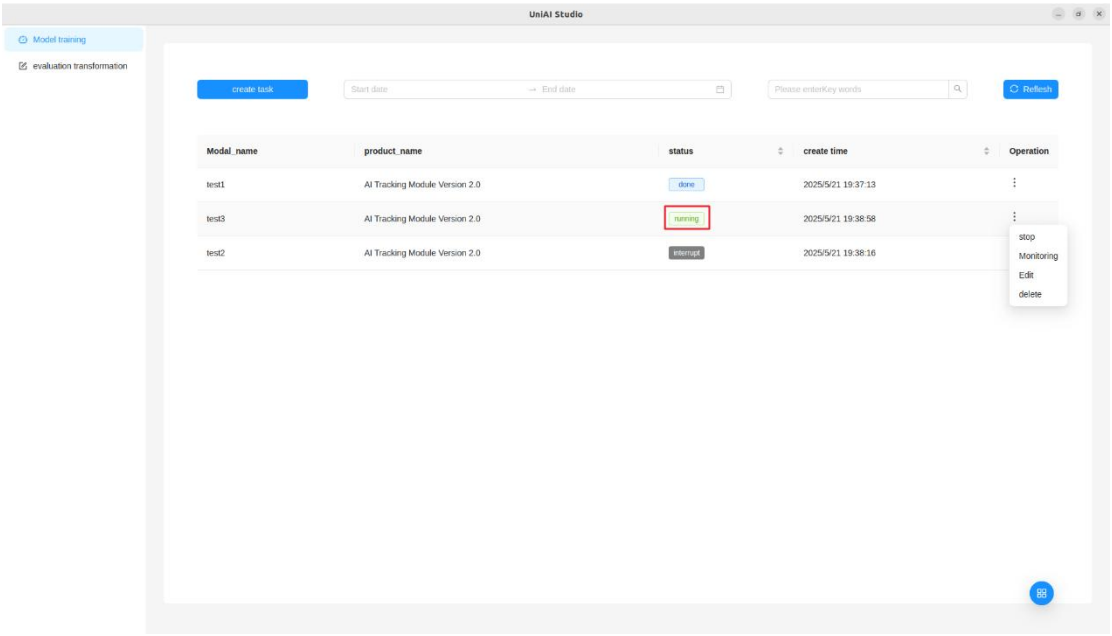


3.Task Operations: You can perform operations such as start/pause, monitor, edit, and delete on individual tasks.

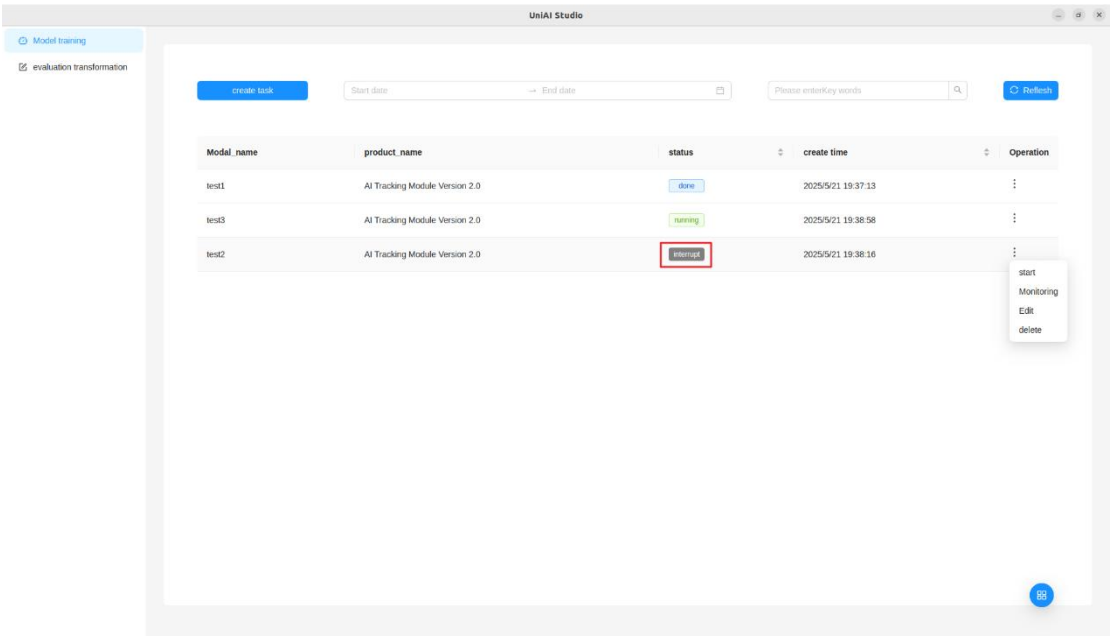


4.Start/Pause: Start or pause the selected task:

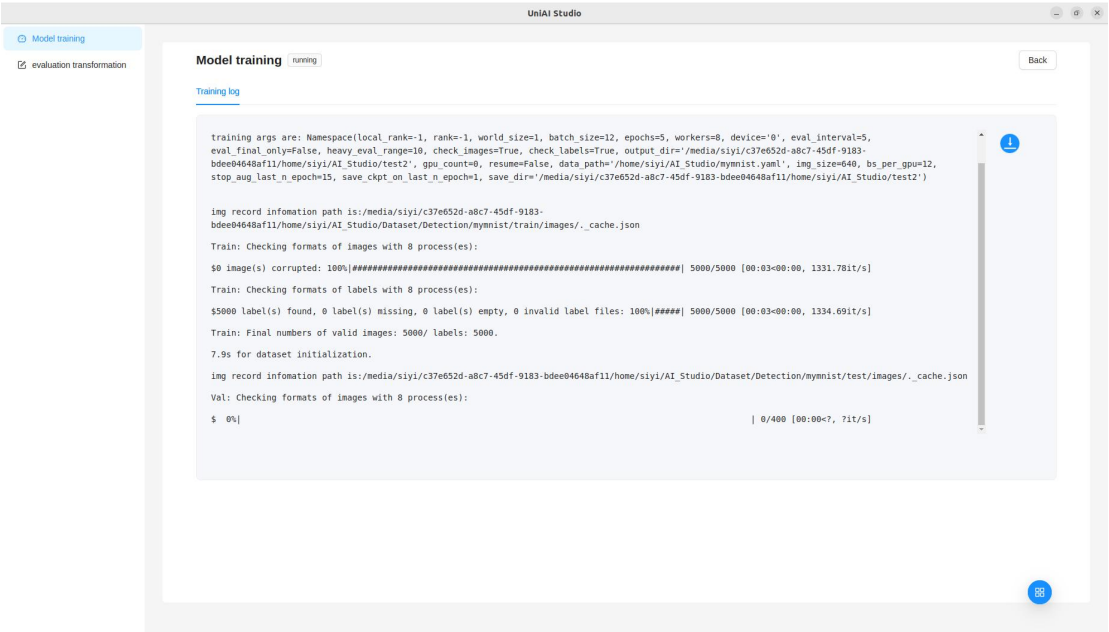
1) Click "Start" to change the task status to "running", indicating the task is in progress.



2）Click "Pause" to change the task status to "paused", temporarily halting the task execution.

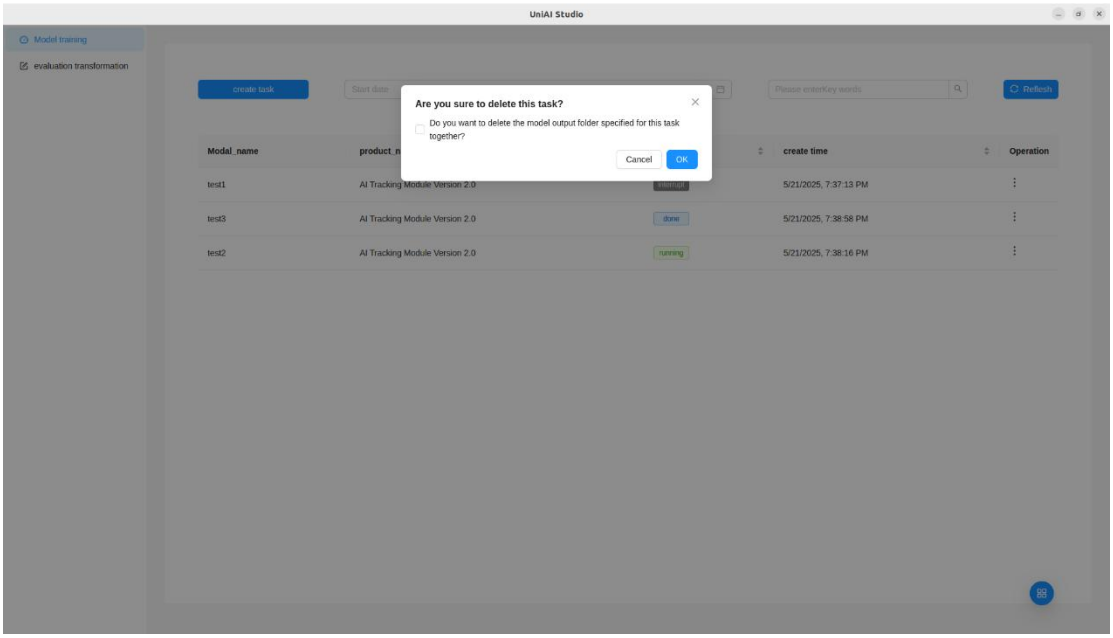


5.Monitor: View the running logs of the current task.

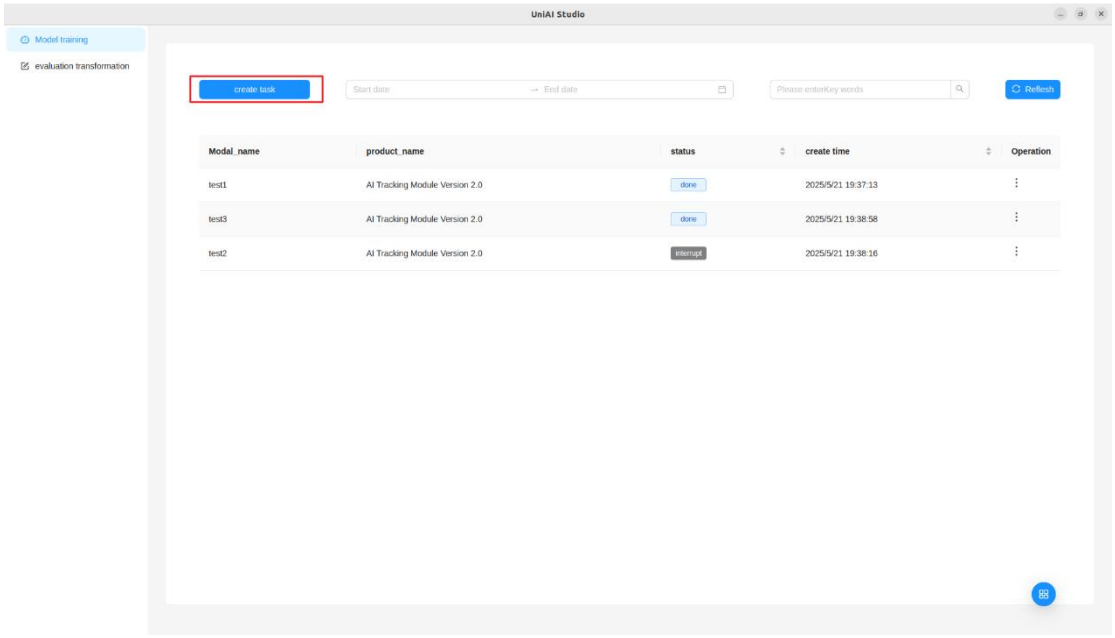


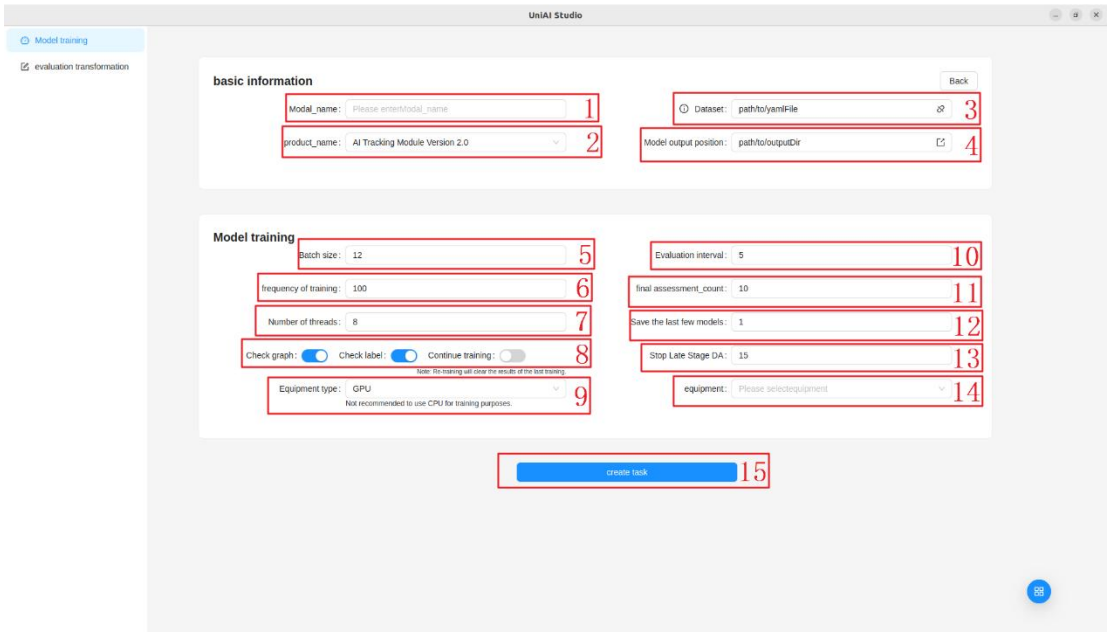
6.Edit: After a task is created, it can be edited again.

7.Delete: Delete the selected task. If "Also delete the model output folder associated with this task" is checked, the training results will be deleted as well.



4.1.1.2 Task Settings





1.Model Name: The name assigned to the training task.

2.Product Name: The product to which the trained model will be applied after training is completed.

3.Dataset: The source of the data.

Definition: A dataset is a structured collection of data samples used in artificial intelligence (AI) and machine learning (ML) for training, validation, and testing of models. It consists of organized, labeled, or processed data samples, serving as the "raw material" for model learning and determining the model's effectiveness in completing tasks.

File Format Specifications:

```
# dirTree
# path
# ├── val
# │   ├── images
# │   │   ├── 0001.jpg
# │   │   └── ...
# │   └── labels
# │       ├── 0001.txt
# │       └── ...
# └── train
#     ├── images
#     │   ├── 0002.jpg
#     │   └── ...
#     └── labels
#         ├── 0002.txt
#         └── ...
train: /path/train/images/
val: /path/val/images/

# number of classes
nc: 10

# class names
names: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
```

NOTE：

1) The dataset uses the YOLO dataset format. The directory structure is shown in dirTree, where image files and their corresponding label files share the same filename.

2) Label files are in txt format with the following content structure:

<object-class-index> <x> <y> <width> <height>

Example: 0 0.412500 0.318981 0.358333 0.636111

Explanation: <object-class-index>: The label index of the target object.

x，y：The normalized coordinates of the object's center relative to the image width (W) and height (H), i.e., x/W and y/H.

3) The train field specifies the location of training dataset images, and val specifies the location of validation dataset images. Label files will be automatically located.

4) nc indicates the number of classes in the dataset.

5) names specifies the names of the classes in the dataset, where the index corresponds to the target label index in the label files.

4.Model output location: The directory where the trained model will be saved after training is completed.

5.Batch size: The number of samples input into the model for each iteration to update parameters. It is a key hyperparameter in training that directly affects training speed, memory usage, convergence, and generalization ability.

6.Number of epochs: The number of times the model iterates through the entire dataset. This directly influences the extent of learning—too few epochs may cause underfitting (insufficient learning of data patterns),

while too many can lead to overfitting (excessively memorizing training details).

7.Number of threads: Refers to the count of concurrent execution units used to accelerate data preprocessing, parallel computing, or distributed training. Setting an appropriate number of threads can significantly improve training efficiency but should be optimized according to hardware resources (CPU/GPU) and task type.

8.Check images, check labels, continue training: Enabling “Check images” and “Check labels” will, upon starting the task, first verify that images and labels are complete and correspond correctly. When a training task is completed, enabling “Continue training” during task re-editing allows restarting the task without deleting the original files.

9.Device type: Select whether to use CPU or GPU for training.

NOTE：

When the training task is large, choosing CPU computation may cause the computer to freeze.

10.Model evaluation interval: In machine learning and deep learning model training, the evaluation interval refers to how often the model’s performance is assessed on the validation or test set, either every certain number of training steps or epochs. The core purpose is to monitor the

model's generalization ability during training, prevent overfitting or underfitting, and provide a basis for hyperparameter tuning and early stopping strategies.

11.Number of final evaluations: This refers to evaluating the model several times near the end of training, which serves as a strategy to determine whether the model has stably converged.

12.Save the last several models: In deep learning and machine learning training, this strategy saves the model weights or checkpoints from the most recent training stages to preserve multiple versions of the model.

13.Stop data augmentation for the last few iterations: This optimization strategy disables data augmentation in the final training stages, fine-tuning the model on original data to improve convergence stability and final performance.

14.Device: Select the GPU(s) involved in computation.

15.Create training task

4.2 Model Evaluation and Conversion

4.2.1 Model Evaluation

Definition: Verifying the model's performance on an independent test dataset to measure its generalization ability (whether it is overfitting or underfitting).

Core methods:

1）Metric calculation: Choose evaluation metrics based on the task type:

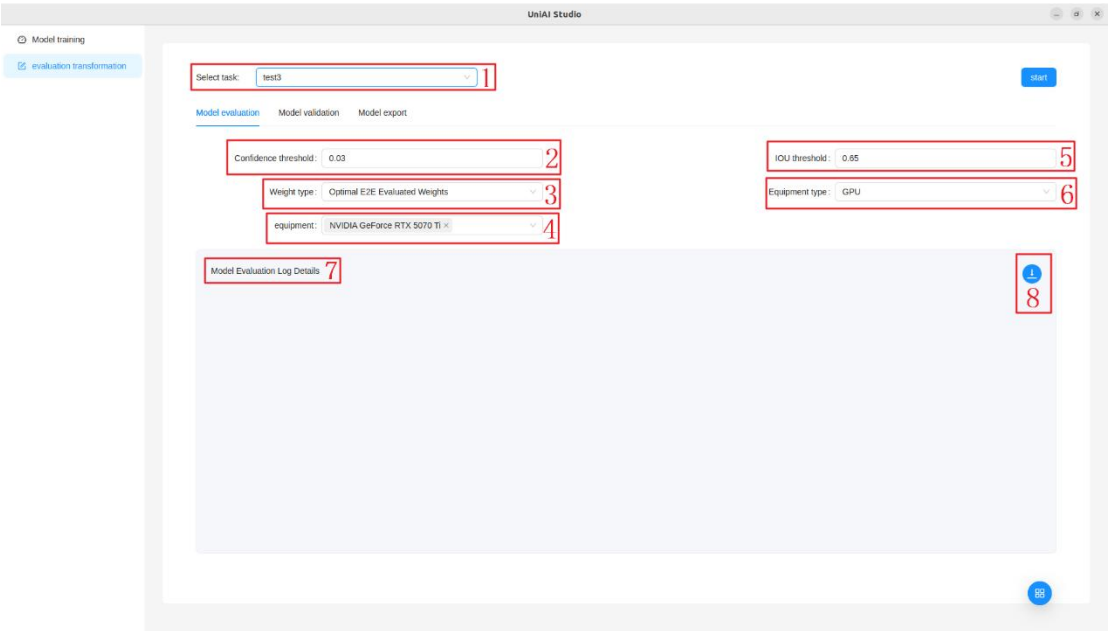
2 ） For classification tasks: Accuracy, Precision, Recall, F1-Score, AUC-ROC curve.

3）For regression tasks: Mean Squared Error (MSE), Mean Absolute Error (MAE), R^2 coefficient.

4 ） Cross-validation: Use K-Fold cross-validation to reduce the randomness caused by data splitting.

Purpose: To determine whether the model meets the expected performance and to guide subsequent optimization directions (such as hyperparameter tuning).

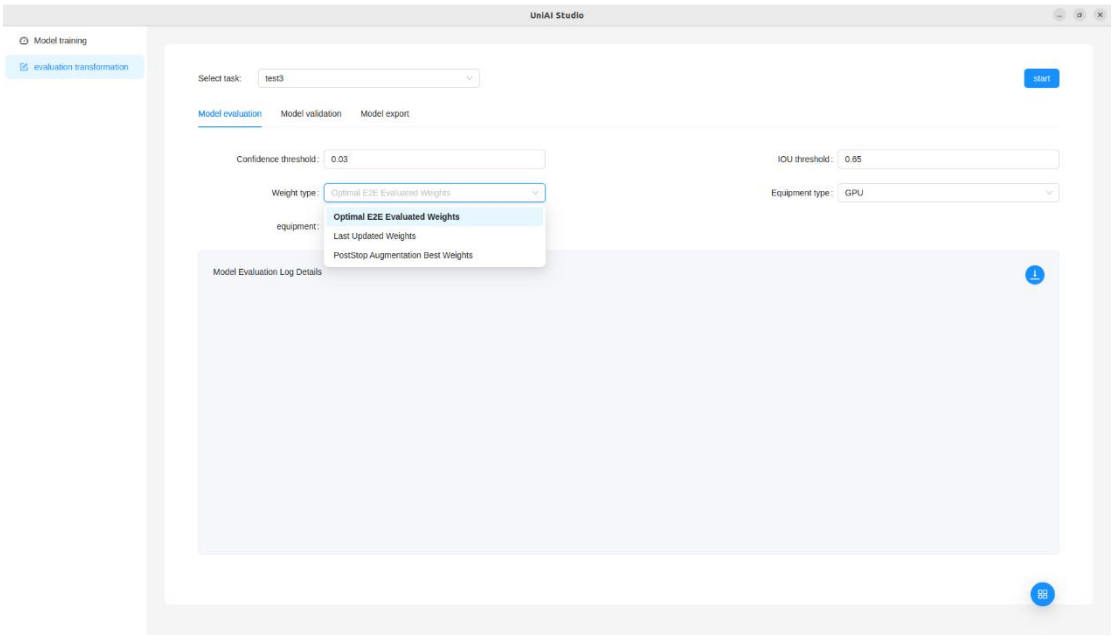
Model evaluation task settings



1.Select Task: Choose a completed task for evaluation.

2.Confidence Threshold: The confidence threshold is a probability cutoff set during model prediction to determine whether a sample belongs to a certain class.

3.Weight Type: Includes the best weights from the entire evaluation process, the weights from the last update, and the best weights after stopping augmentation.

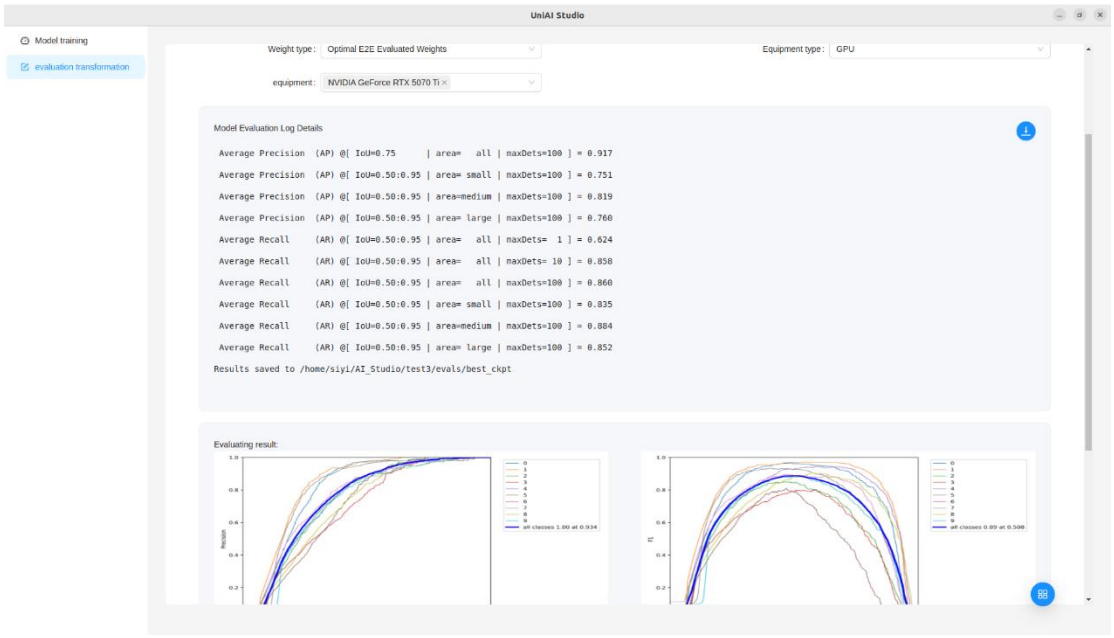


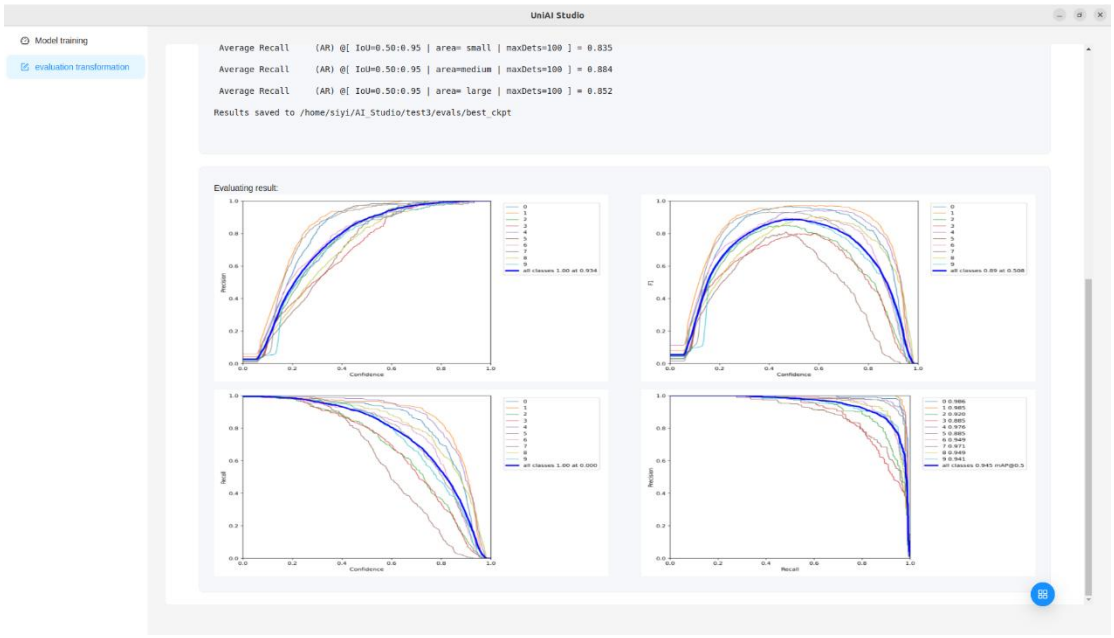
4.Device: Select the GPU to be used for computation.

5.IOU Threshold: Intersection over Union (IoU) measures the overlap between the predicted bounding box and the ground truth box.

6.Device Type: Choose whether to use CPU or GPU for training.

7.Model evaluation log:





8. Return to latest: Click this button to scroll the log back to the most recent status.

4.2.2 Model Validation

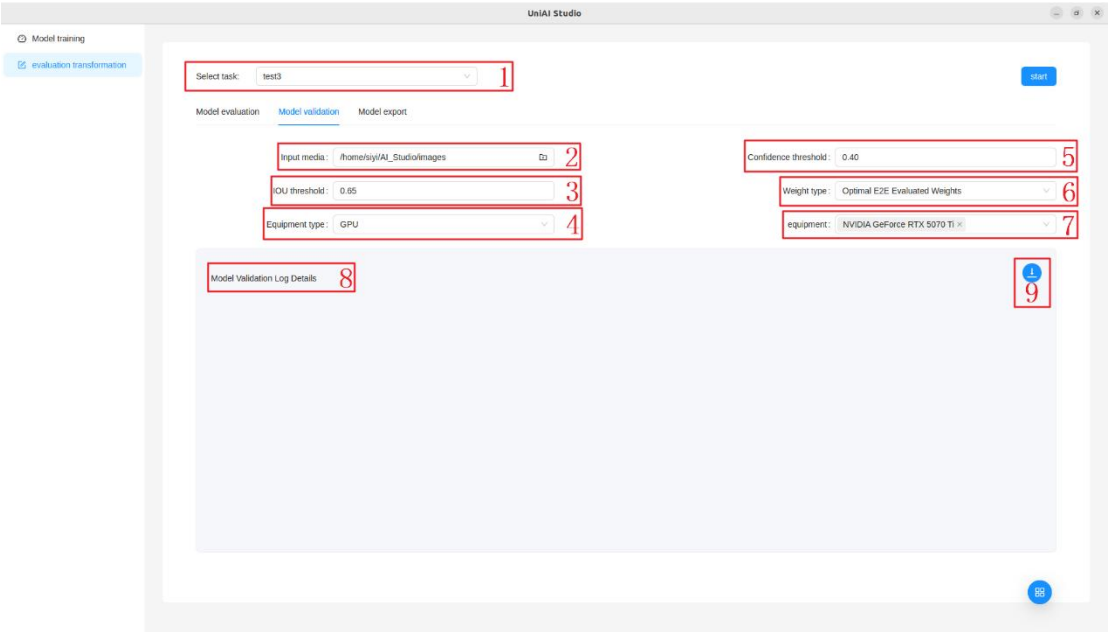
Meaning: Broadly considered a part of evaluation, but with more

emphasis on verifying the model's business logic and security checks to ensure the model meets practical application requirements.

Validation content:

Data distribution consistency: Verify whether the distribution of training data matches that of production environment data (to avoid data drift).

Model validation task settings:



1.Select task: Choose a completed task for evaluation.

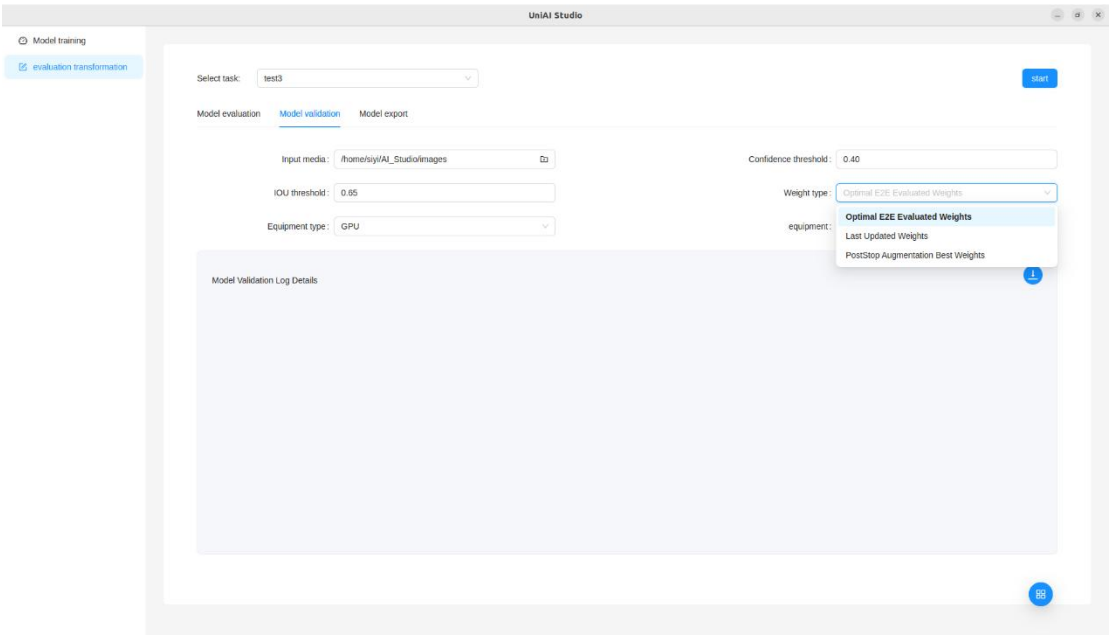
2.Input media: Images used for recognition, which should not be the same as those in the training dataset.

3.IOU threshold: IOU (Intersection over Union) measures the overlap between the predicted bounding box and the ground truth box.

4.Device type: Choose whether to use CPU or GPU for training.

5.Confidence threshold: The probability threshold set during model prediction to determine whether a sample belongs to a certain class.

6.Weight type:



1) Best Weights During Full Evaluation

- Definition: Model weights saved when the validation set metrics (e.g., accuracy, loss) reach their best during the entire training process (across all epochs).
- Saving Timing: Updated whenever validation performance exceeds previous best.
- Characteristics:
    - Advantages: Ensures model is at global optimal state, avoids late-stage overfitting effects.

    - Disadvantages: May miss potential improvements from fine-tuning at the end of training (especially if early stopping is triggered too soon).
- Applicable Scenarios:
    - When the validation and test set distributions are highly consistent.
    - When training is highly volatile, and best intermediate results need to be preserved.

2) Last Updated Weights

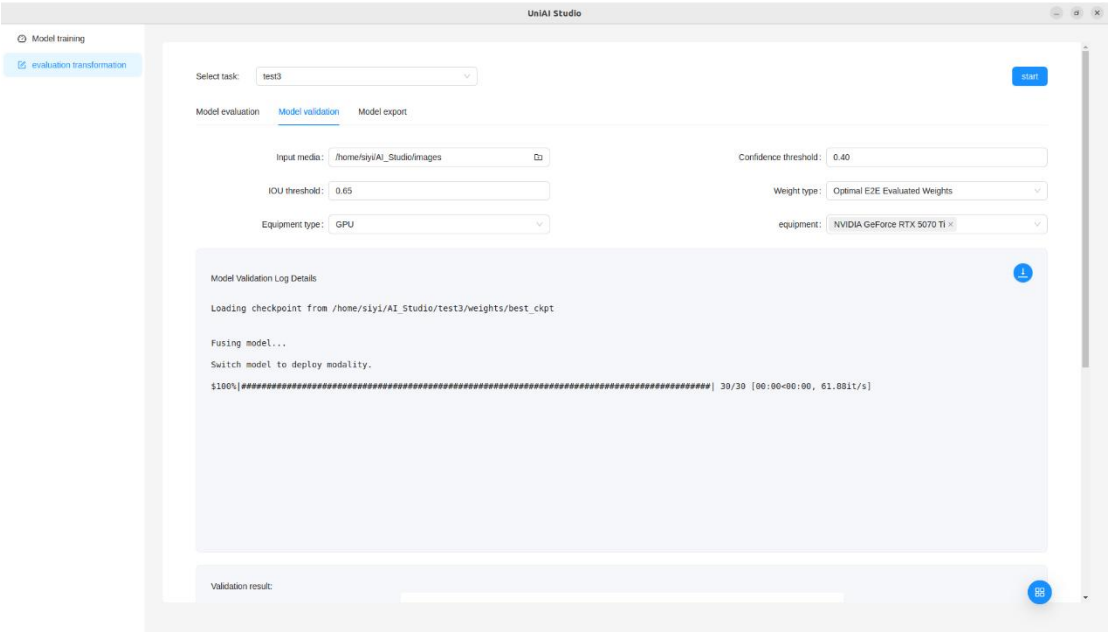
- Definition: Model weights from the final epoch or step when training ends.
- Saving Timing: At the natural or forced termination of training.
- Characteristics:
    - Advantages: Retains final-stage parameters, which may include fine adjustments to training data.
    - Disadvantages: If overfitting occurred, performance on the test set may degrade significantly.
- Applicable Scenarios:
    - When training is stable without signs of overfitting (e.g., loss curve continues to decline smoothly).
    - When the goal is to analyze the model's final convergence state.

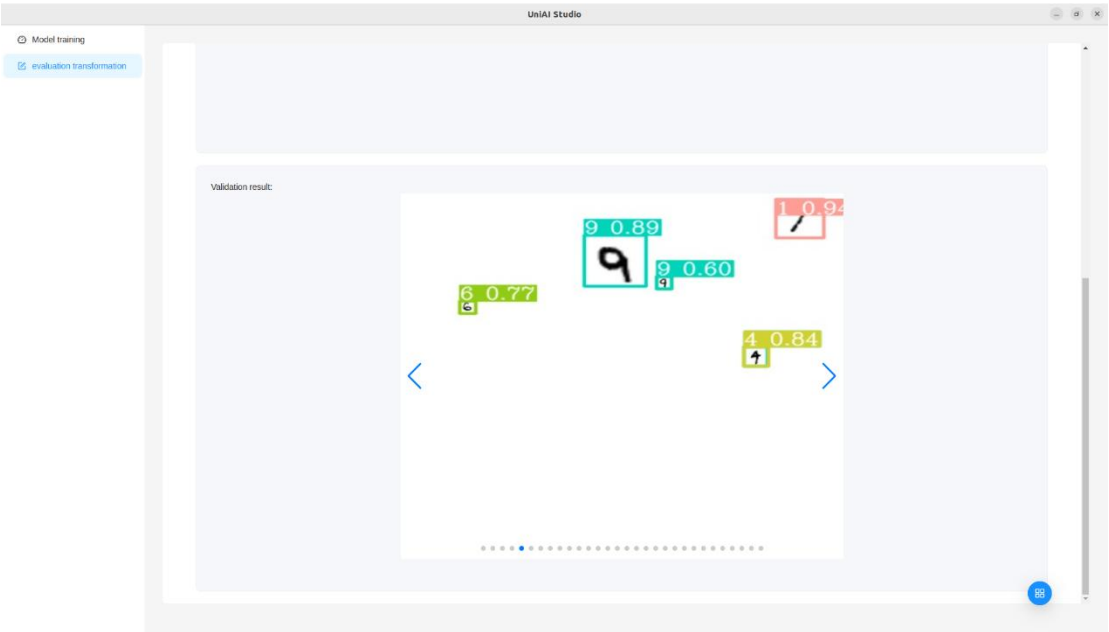
3) Best Weights After Stopping Data Augmentation

- Definition: Weights that yield the best validation performance after data augmentation has been turned off during the final stage of training.
- Saving Timing: During a designated phase after augmentation is disabled (e.g., last N epochs).
- Characteristics:
  - Advantages: Fine-tuning on "clean" data without augmentation, better reflecting real-world inference.
  - Disadvantages: Effectiveness depends on timing of stopping augmentation—too early or too late may reduce performance.
- Applicable Scenarios:
  - When data augmentation is strong but not used during inference (e.g., industrial deployment).
  - When stable feature extraction with reduced noise is needed in later training.

7.Device: Select the GPU(s) to participate in computation.

8.Model Validation Log Type:



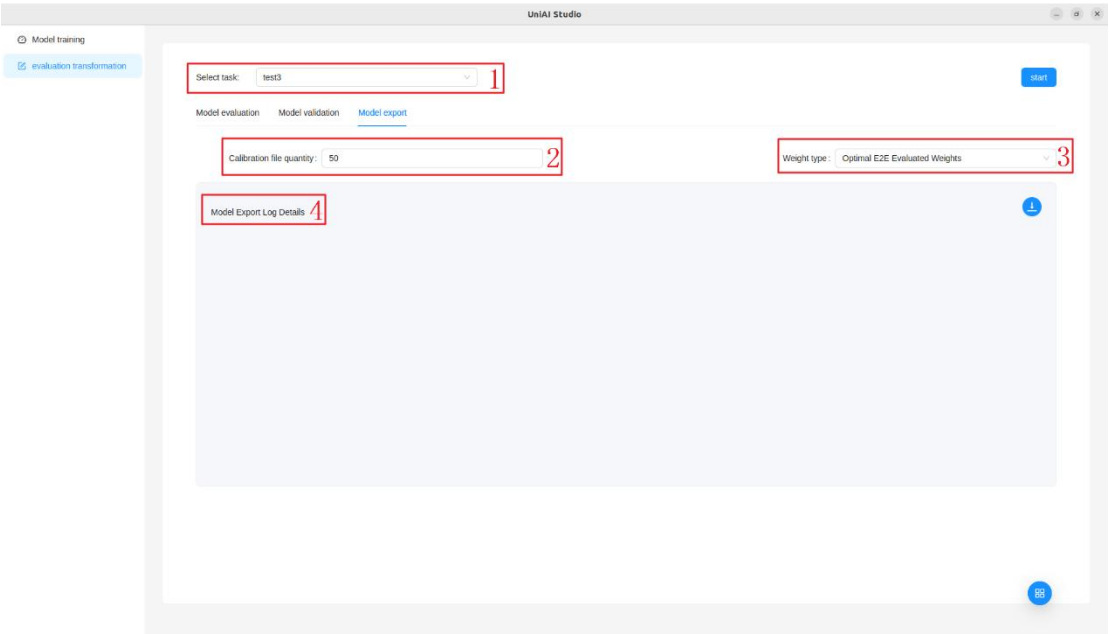


9、Back to Latest: Click this button to scroll the log back to the latest status.

4.2.3 Model Exporting

Meaning： Model exporting refers to converting a trained model into a format that can be deployed in a production environment.

Model Export Task Settings



1、Select Task: Choose a completed task for evaluation.

2 、 Number of Calibration Files: Used for model quantization and calibration.

3、Weight Type:

3）Best weights during the entire training process

- Definition: The model weights saved when the validation metrics (e.g., accuracy, loss) reached the best value throughout all epochs.
- Save Timing: Updated whenever the validation performance

exceeds the previous best.

- Characteristics
    - Pros: Ensures the model is in a globally optimal state, avoiding potential overfitting in later training stages.
    - Cons: May miss potential performance gains from late-stage fine-tuning, especially if early stopping is triggered prematurely.
- Applicable Scenarios:
    - When the validation and test data distributions are highly consistent.
    - When training is unstable and it's important to preserve the best intermediate result.

4) Last Updated Weights

- Definition: The model weights saved at the final epoch or step when training completes or is forcefully stopped.
- Saving Condition: Triggered upon the normal or manual termination of the training process.
- Characteristics:
    - Pros: Captures the most recent parameter updates, which may reflect fine-tuning on the training data.
    - Cons: If overfitting has occurred, generalization performance on unseen data may be compromised.

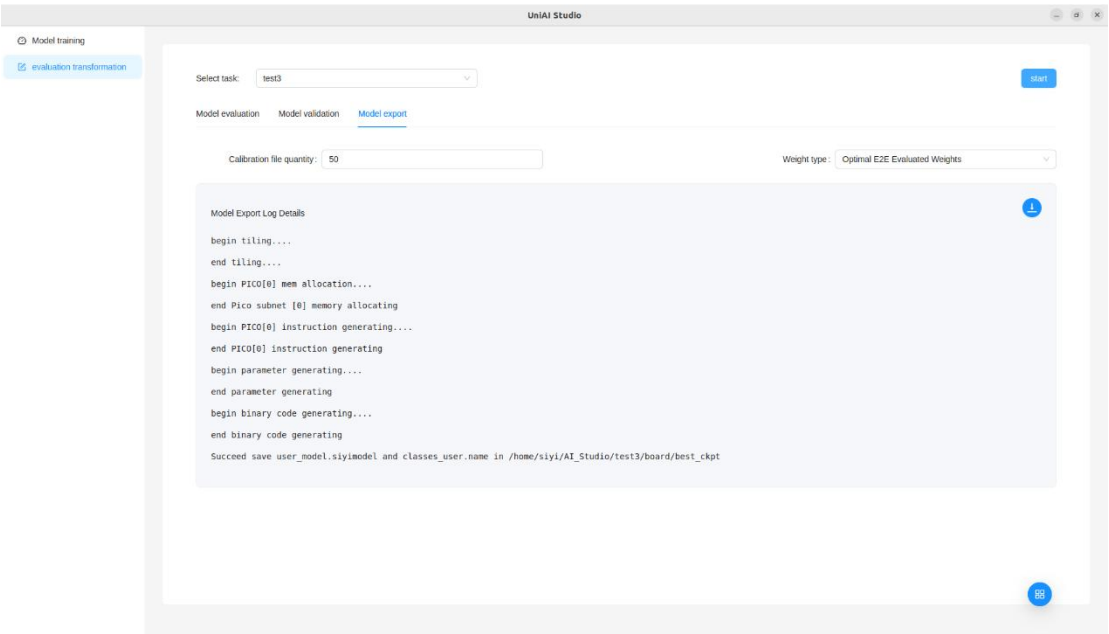
- Applicable Scenarios
    - When training is stable with no significant signs of overfitting (e.g., consistently decreasing loss).
    - For research purposes to analyze the model's final convergence state.

5) Best Weights After Data Augmentation Stopped

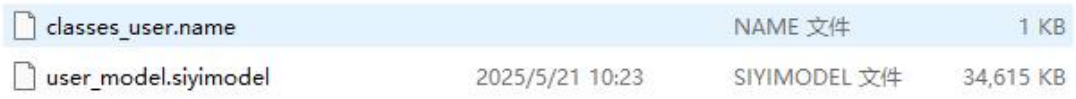
- Definition: Model weights saved when validation performance is best after data augmentation is disabled in the final training stages.
- Saving Condition: During a specified period (e.g., last N epochs) after turning off data augmentation.
- Characteristics:
    - Pros:
    - • Fine-tunes the model on "clean" data, better reflecting real-world inference conditions.
    - • Helps stabilize feature extraction by removing augmentation-induced noise.
    - Cons:
    - • Effectiveness depends on when augmentation is disabled — stopping too early or too late may reduce benefits.
- Applicable Scenarios
    - When data augmentation is strong during training but not used during inference (e.g., industrial deployment).

- When a stable and clean representation is needed in the final model stages.

4、Export Log



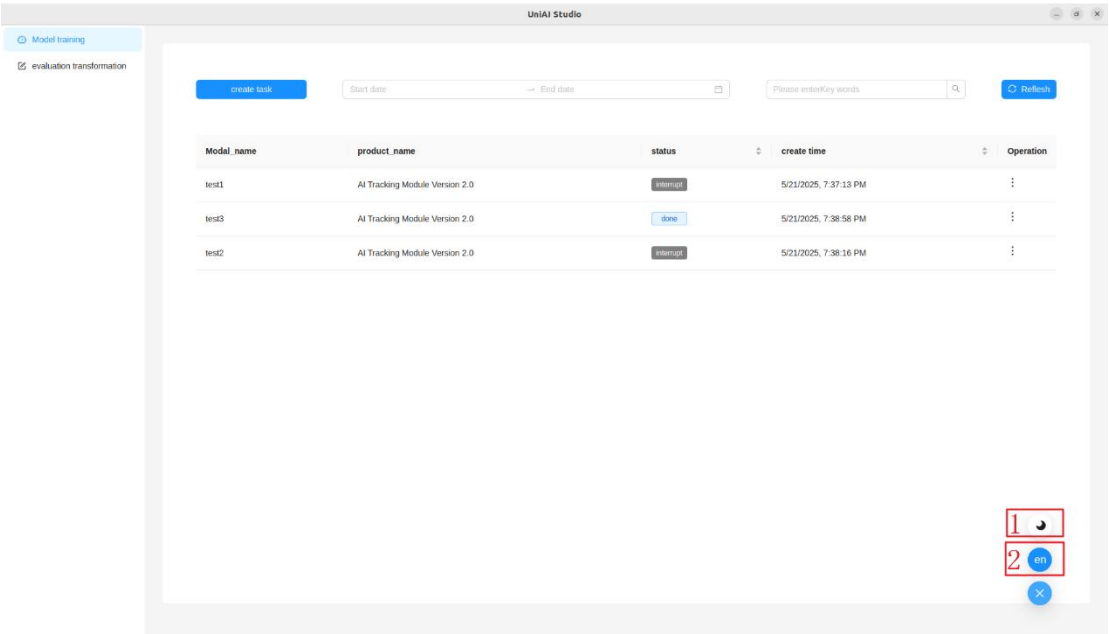
Export logs according to the model. You can find the two generated files in the export folder. Import them into the root directory of the device, and then you can use them.



5 Settings and Preferences

1.Day/Night Mode: Click this button to switch between day mode and night mode.

2.Chinese/English Switch: Click this button to switch between Chinese and English modes.



6 Troubleshooting

6.1 Common Errors

Unable to train: If the dataset is on another drive, please check whether the drive is mounted.

Unable to validate: Insufficient number of training epochs.

NOTE

If you have already checked the above steps but still cannot locate the problem, please contact your dealer immediately or reach out directly to Siying Technology for troubleshooting and resolution.

6.2 Contact Support Team

Email / Online Customer Service / Community, etc.

Please visit Siying Technology https://www.siyi.biz/index.php?id=support for the latest updates.

7 Appendix

7.1 Glossary of Terms

Confidence Threshold

1. Basic Definition

The confidence threshold is the probability cutoff set during model prediction to determine whether a sample belongs to a certain class.

Core function: To filter out low-confidence predictions, balancing the model output's precision and recall.

2. Impact of Threshold Adjustment

Threshold Change	Precision	Recall	Applicable Scenario
Increase Threshold	↑ (Stricter positive filtering)	↓ (Misses some positives)	Reduce false positives
Decrease Threshold	↓ (Looser positive filtering)	↑ (Captures more positives)	Avoid false negatives

3.Applicable Scenarios

Used in classification tasks, especially for screening uncertain samples in binary or multi-class classification.

By reasonably setting the confidence threshold, developers can flexibly control the model's sensitivity to better fit actual business needs.

Intersection over Union Threshold

1. Basic Definition

IOU (Intersection over Union) measures the overlap between the predicted bounding box and the ground truth bounding box. The calculation formula is:

$$\text{IOU} = \frac{\text{Predicted Box} \cap \text{Ground Truth Box}}{\text{Predicted Box} \cup \text{Ground Truth Box}}$$

IOU Threshold: The critical value used to determine whether a predicted bounding box is correct. If the IOU between the predicted box and the ground truth box is greater than or equal to the threshold, the prediction is considered a True Positive (TP); otherwise, it is considered a False Positive (FP).

2. mpact of Threshold Adjustment

Threshold Change	Detection Tendency	Applicable Scenarios
Increase Threshold	Reduces false positives (FP↓), but increases missed detections (FN↑)	When high precision is required
Decrease Threshold	Captures more targets (FN↓), but increases false positives (FP↑)	When high recall is required

By reasonably setting the IoU threshold, developers can precisely control the balance between localization accuracy and recall ability of the model, thereby adapting to the needs of different business scenarios.

8 After-Sales and Warranty

Please visit the SIYI Technology support page at Service and Support - SIYI Technology | Empowering and Building an Intelligent Robot Ecology for the latest after-sales and warranty information.