

GIGABYTE™

AI TOP Utility ATOM

Version 4.2.0

User Manual

1. Installation and preparation	P.1
1-1. AI TOP Utility Software installation	P.1
2. Terms and functions description	P.1~4
2-1. Dashboard	P.1~2
2-2. RAG	P.2
2-3. Inference	P.3
2-4. Machine Learning	P.3~4
2-5. Settings	P.4
3. Operation tutorial	P.4~18
3-1. RAG	P.4~7
3-2. Inference	P.8~13
3-3. Machine Learning	P.14~18
4. Supported Models	P.19~21
4-1. Supported list of LLM/LMM models	P.19
4-2. Supported list of embedding models (RAG)	P.19
4-3. Download Model	P.20
4-4. Download LLM/LMM models from Hugging Face	P.20~21

1. Installation and Preparation

1-1. AI TOP Utility Software installation

Please visit [this link](#) to download the installation package. Download the version with the suffix "ATOM".

After downloading, copy the file to your ATOM home directory and extract the .zip file.

Double-click the .deb file and click **Install**. When the authentication window appears, enter your system password to continue. After a short while, the **Install** button will change to **Open**. Click it to launch a terminal window and the installation process will begin.

Please monitor the installation closely, as you will need to enter your system password when prompted.

After the installation is complete, create two new folders named model and data in your home directory.

This step should be performed only once, immediately after installation. Do not repeat it during future updates to avoid overwriting or deleting your existing models, data, or checkpoints.

After downloading, copy the file to your ATOM's home directory and extract the .zip file.

2. Terms and functions description

2-1. Dashboard

Current GPU / CPU load	The current (%) usage of GPU/CPU. Indicates the current percentage of GPU and CPU processing power utilization. This is often monitored to ensure that resources are being efficiently utilized and to diagnose performance bottlenecks.
DRAM usage	Display the current utilization of the random-access memory.
NVMe SSD usage	Display the current utilization of the solid-state drive.
GPU Core Temperature (°C)	The real-time temperature of the GPU's primary processing die.
DRAM Loading state (%)	The loading state (% of total memory) of System DRAM during the training process. Shows the percentage of DRAM being utilized. This metric helps in understanding if there are memory constraints that could be impacting the performance of the training process.
CPU Loading state (%)	The average loading state (% of total memory) of all CPU cores in the training process. The percentage of CPU capacity currently in use. Monitoring this helps in understanding the CPU's role and load during the training process, particularly for tasks not offloaded to the GPU.

2-2. RAG

Document Folder	The folder contains reference files.
Model	The LLM model for Q&A with RAG.
Text Embedding Model	The text embedding model, e.g., all-MiniLM-L6-v2.
Audio Embedding Model	The audio embedding model, e.g., larger_clap_general.
Image Embedding Model	The image embedding model, e.g., CLIP-ViT-B-32-laion2B-s34B-b79K.
Video Embedding Model	The video embedding model, e.g, CLIP-ViT-B-32-laion2B-s34B-b79K.
System Prompt (optional)	A set of instructions and guidelines is provided to the LLM to control its behavior, tone, and the overall context of its responses.
Temperature	A parameter that controls the randomness and creativity of the model's output. It adjusts the probability distribution of the words the model considers when generating a response. Ranging from 0 to 1, higher values yield more diverse, creative responses
Maximum New Tokens	The upper limit on the number of tokens the model can generate in a single response.

2-3. Inference

Inference Type	Select the type of inference: (1)Text to Text, (2)Text to Image, (3)Text to Video, (4)Image Text to Text.
Backbone Model	Select the LLM model with GGUF or safetensors format.
System prompt (optional)	A set of instructions and guidelines is provided to the LLM to control its behavior, tone, and the overall context of its responses.
Maximum tokens	The maximum tokens for the sentence length of the query and answer.
Temperature	A parameter that controls the randomness and creativity of the model's output. It adjusts the probability distribution of the words the model considers when generating a response. Ranging from 0 to 1, higher values yield more diverse, creative responses
Top p	A parameter that controls the randomness and creativity of the model's output. It determines a probability threshold, and only the most likely tokens that collectively sum to that probability are considered. Ranging from 0 to 1, higher values yield more diverse, creative responses

GPU Offload (GGUF)	Set up how many model layers mount to VRAM run by the GPU .
CPU Threads (GGUF)	Set up how many CPU threads to run the model layers, and mount them to DRAM
Fine-tuning type (safetensors)	Select the type of fine-tune for this model. If none, set it to Full.
Adaptar model (safetensors)	Select the adapter model. When your model is fine-tuned by LoRa .
Width	The width of the image generated by the text-to-image model.
Height	The height of the image generated by the text-to-image model.
FPS	The frames per second of the video generated by the text-to-video model.
Length of the Video (seconds)	The duration of the generated video in seconds.
Save path	The directory where the image or video will be saved.

2-4. Machine Learning

(Project) Folder	The directory where the machine learning project is located. All ML projects will be stored in this folder.
Task Type	The type of machine learning task is the project. Currently supported types include: image classification , object detection , image segmentation , and OCR (Optical Character Recognition).
(Project) Name	The name of the project. The name can be modified later, but cannot be duplicated.
Python Version	The Python version used in the project. Currently, Python 3.10 is supported.
Go to Annotate	Annotation tools suitable for the selected task type will be provided to assist users in labeling their own datasets.

2-5. Settings

License	End user license agreement (EULA) in English, Simplified Chinese, and Traditional Chinese versions.
Version	The current version of the software.

3. Operation tutorial:

This is a step-by-step instruction for using the AI TOP Utility to infer pretrained LLMs and perform machine learning tasks.

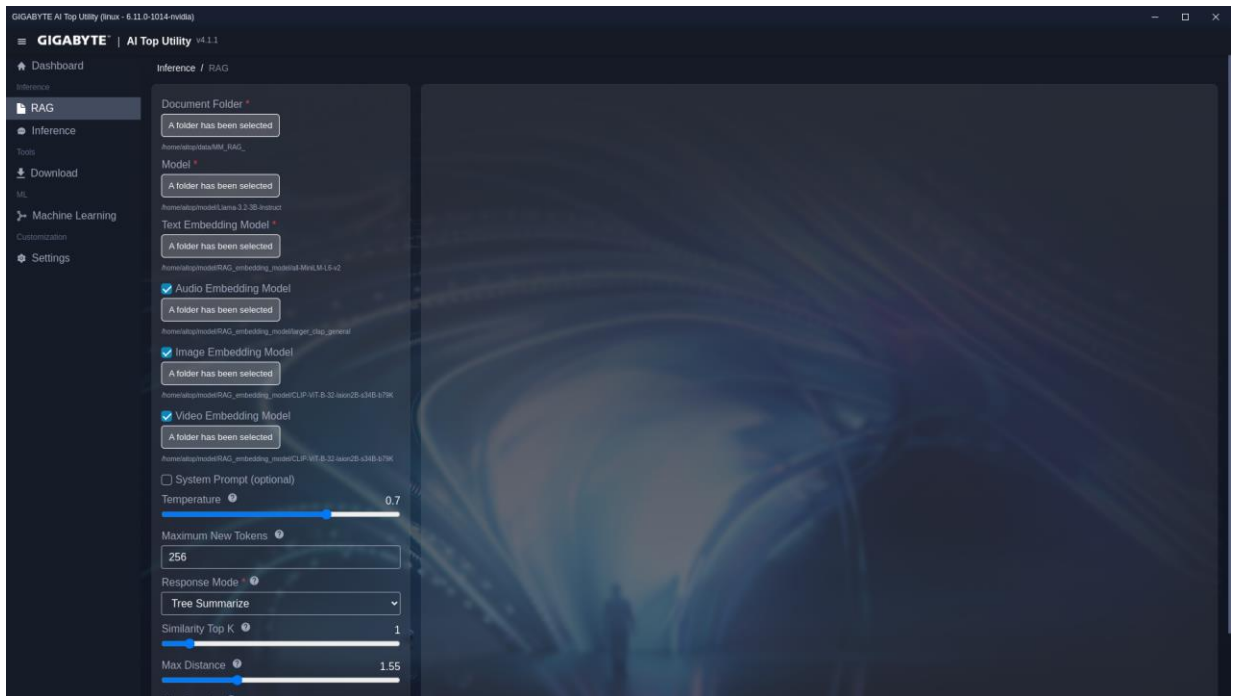
3-1. RAG

Here's a step-by-step explanation of how Retrieval-Augmented Generation (RAG) works with a vector database and a Large Language Model (LLM):

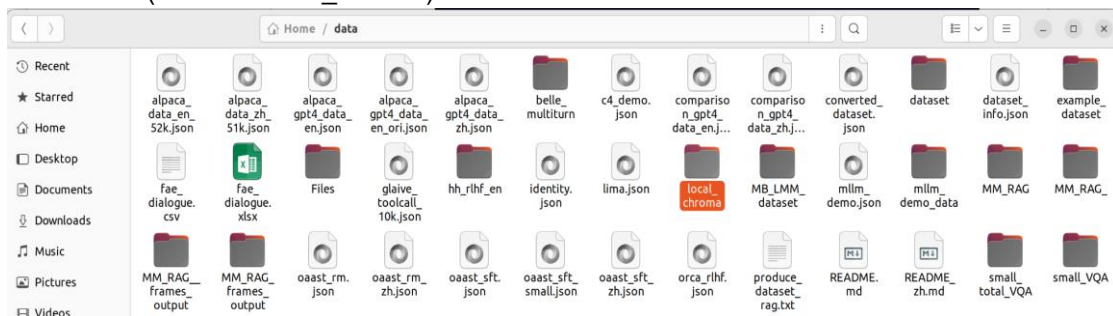
- Document Encoding: Documents are first processed and encoded into high-dimensional vectors using techniques such as embeddings. These vectors represent the semantic content of the documents and are stored in a vector database.
- Query Encoding: When a query is received, it is also converted into a vector using the same encoding method used for the documents.
- Retrieval: The query vector is compared against the vectors in the vector database. This comparison is typically done using similarity metrics like cosine similarity or dot product. The database retrieves the most similar documents or pieces of information based on the query vector.
- Contextual Integration: The retrieved documents are then used to provide context to the LLM. This means the LLM gets access to relevant information that it needs to generate a response.
- Generation: The LLM processes the query along with the context from the retrieved documents. It generates a response that incorporates the relevant information to ensure accuracy and relevance.
- Output: The final output is a response generated by the LLM that is informed by both the query and the relevant documents retrieved from the vector database.

Let's start with the Multi-Modal RAG feature:

- (1) Click the "RAG" tab

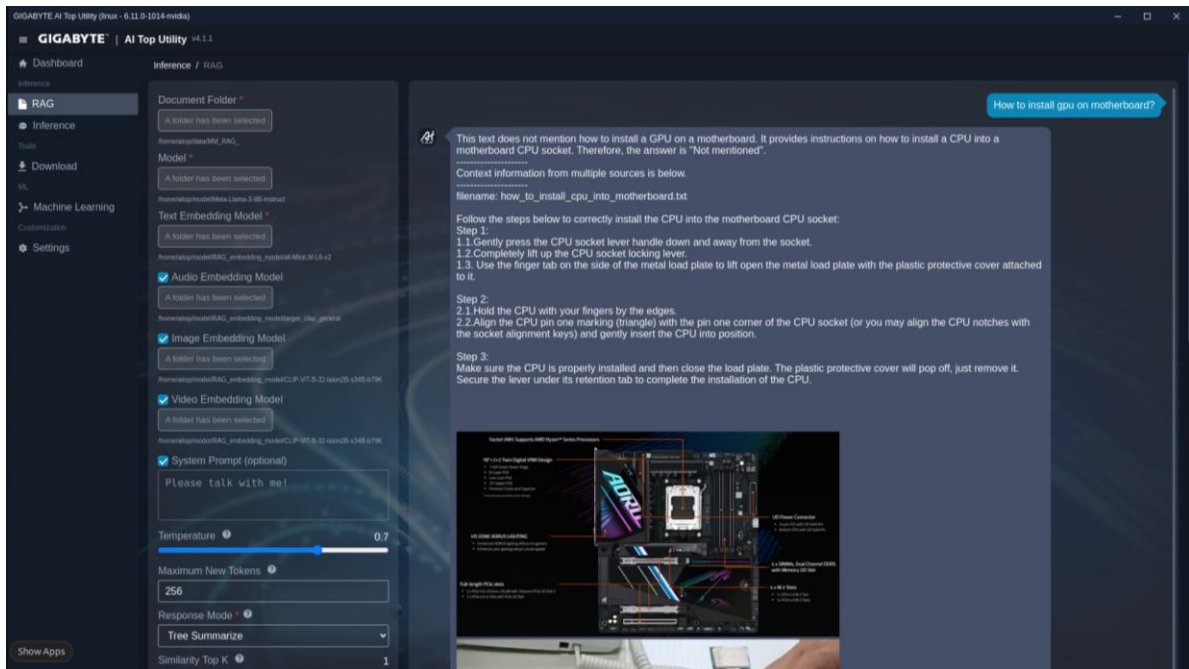


- (2) In the "Document Folder," select the directory containing the files you want to add to ChromaDB. We support the following file types: Images (.jpg), Audio (.wav), Video (.mp4), and Documents (.txt, .json, .csv, .pdf, .docx, .xlsx).
- (3) In the "Model," select the LLM model you want to perform RAG.
- (4) In the "Text Embedding Model" section, select the text embedding model. It will convert the text documents into embedding vectors and save them in the ChromaDB text collection.
- (5) In the "Audio Embedding Model" section, select an audio embedding model. Once selected, the model will return the audio that is most similar to your input.
- (6) In the "Image Embedding Model" section, select an image embedding model. Once selected, the model will return the image that is most similar to your input.
- (7) In the "Video Embedding Model" section, select a video embedding model. Once selected, the model will return the video that is most similar to your input.
- (8) Then, the Multi-Modal RAG will convert these files into embedding vectors and save them into ChromaDB (at ~/data/local_chroma).

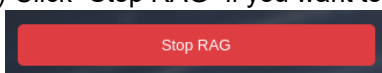


- (9) In the "System prompt", set up the prompt for your inference chat, if desired. This is an optional setting.

- (10) Set the **Response Mode**, **Similarity Top K**, **Max Distance**, and **Max Results** (if applicable) to control the generation process.
- (11) Click "Start RAG" to load the model. Start asking a question in the dialog box.



- (12) Click "Stop RAG" if you want to stop asking for documents.

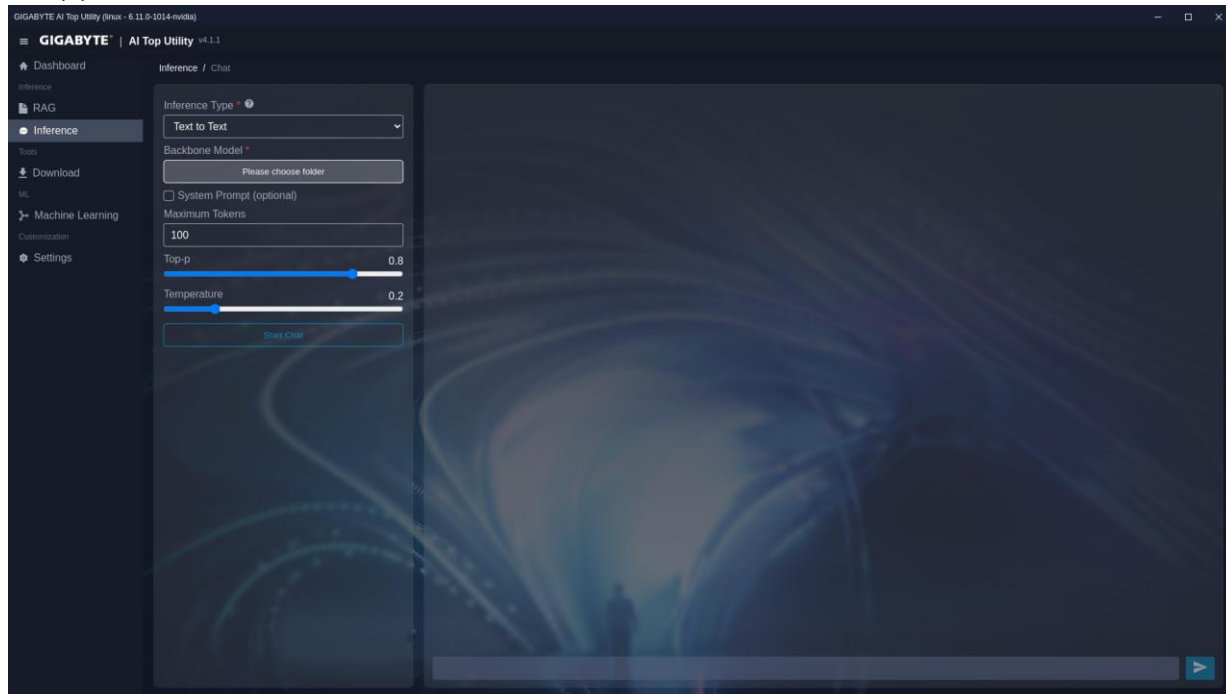


To add new data to the RAG database, stop the RAG service first, place the new files into the designated directory, and then restart the service.

3-2. Inference

Our built-in inference tool lets you run models in either **Safetensors** or **GGUF** format. You can download models from the **Download** tab. We support various inference types, including Text to Text, Text to Image, Text to Video, and Image Text to Text.

(1) Click the **"Inference"** tab.



(2) Choose **"Inference Type"**.

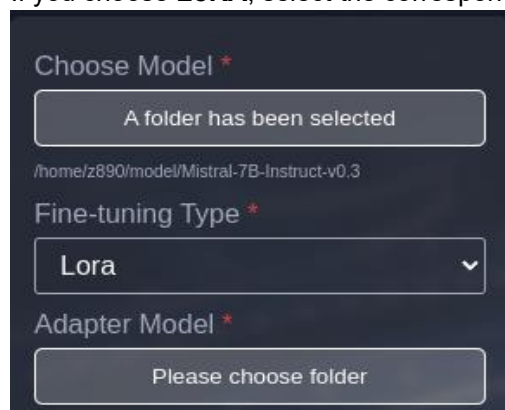
❖ Suppose we choose **Text to Text**.

(3) In the **"Backbone Model"** section, please select one of the text-to-text models, e.g., **"Llama-3.1-8B-Instruct"**

(4) Configure model options

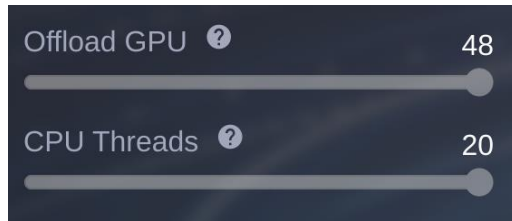
a. If you're using the **safetensors** format model.

- Pick the fine-tuning type: **Full**, **Freeze**, or **LoRA**. If you have not fine-tuned it, choose **Full**.
- If you choose **LoRA**, select the corresponding adapter in **Adapter Model**.

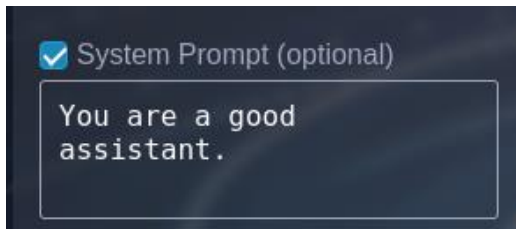


b. If you're using the **GGUF** format model.

- In **Offload GPU**, specify how many layers to keep in **VRAM (GPU)**.
- Set the number of **CPU threads** for the layers that stay in **DRAM**.



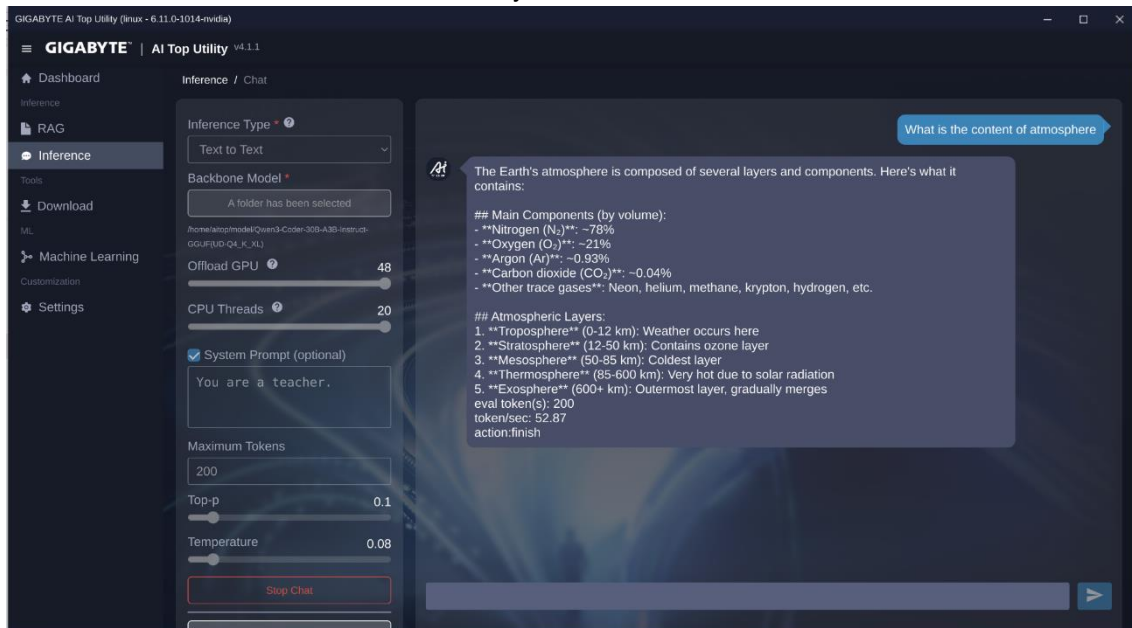
(5) Tick **System Prompt** and enter a prompt if you need one (Optional).



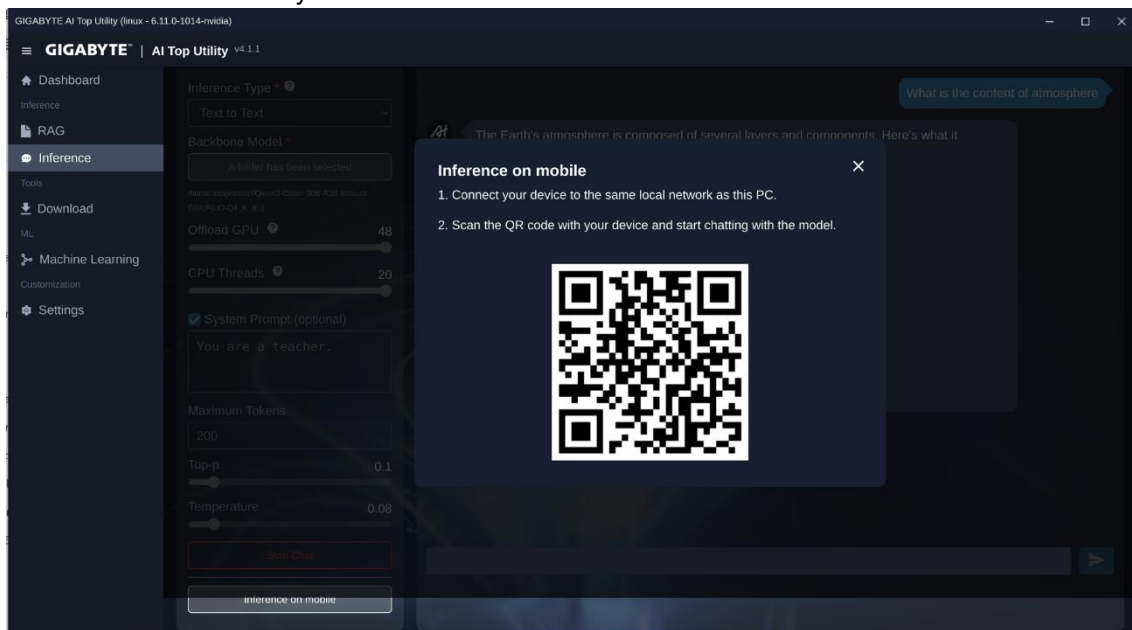
(6) Adjust **Maximum Tokens**, **Temperature**, and **Top-p** settings to control generation.



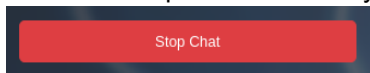
- (7) Click "Start Chat" to load the model. Then you can chat with the model.



- (8) To run inference on another device, click "Inference on Mobile" and make sure the device is on the **same local network** as your PC.

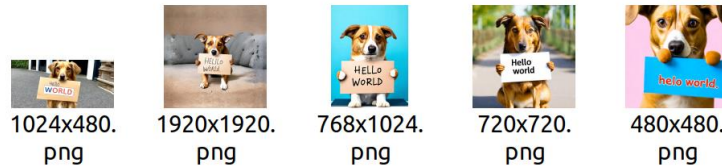


- (9) Click the "Stop Chat" button if you want to stop the inference.

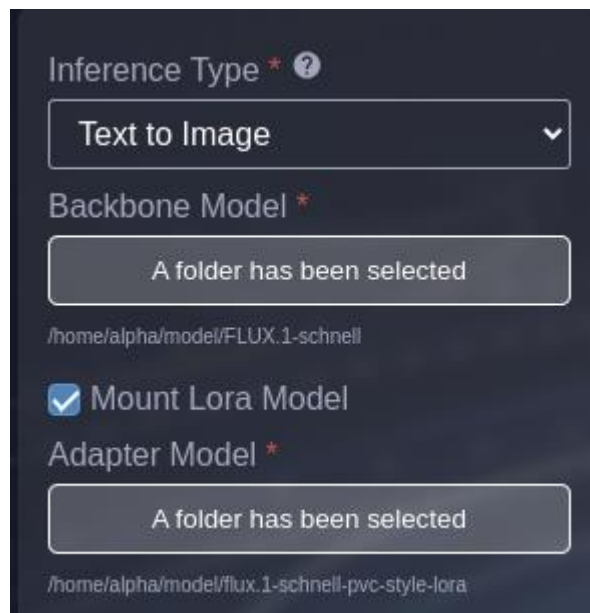


Suppose we choose **Text to Image**.

- (1) In the "**Backbone Model**" section, please select one of the text-to-image models, e.g., "**stable-diffusion-3.5-large**".
- (2) You can set the width and height attributes of the image and input any content for the image you wish to generate. However, the model may impose certain restrictions. Please refer to the model's Hugging Face page for more details. Usually, the height and width must be multiples of 16. You may try these sizes of image to begin with: 1024x480, 1920x1920, 768x1024, 720x720, 480x480.

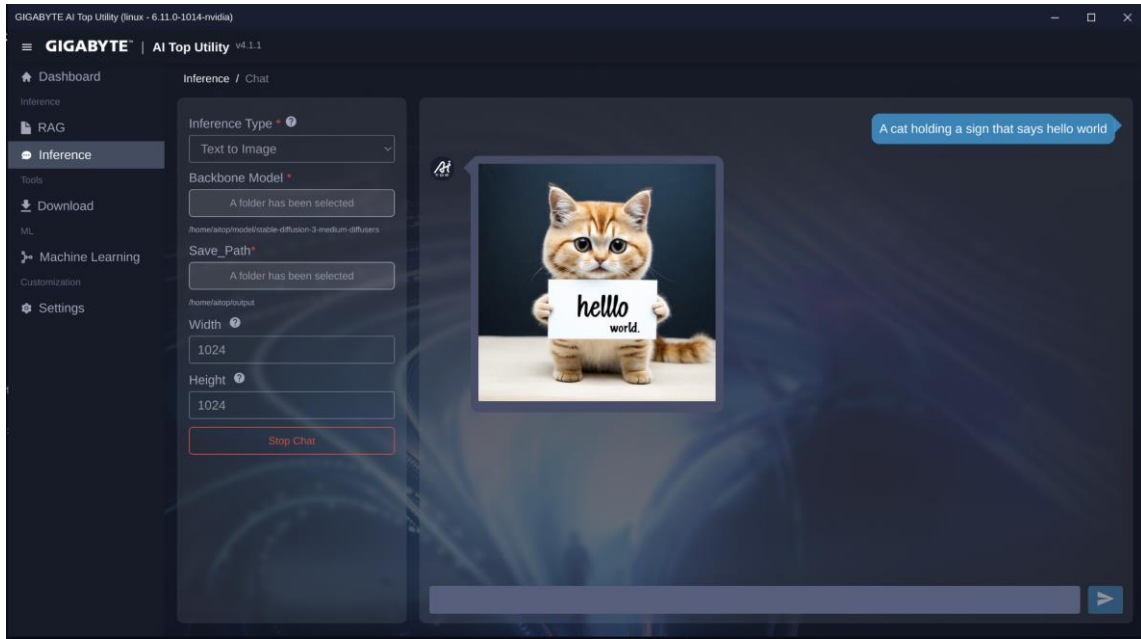


- (3) To mount a LoRA model, click "Mount LoRA Model" and select the desired LoRA file.

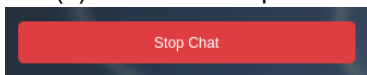


Make sure the LoRA model you select is compatible with your chosen backbone model.

- (4) Click "Start Chat" to load the model. Then, you can input any prompt for the image you wish to generate. Refer to the model's page for best prompting practice.

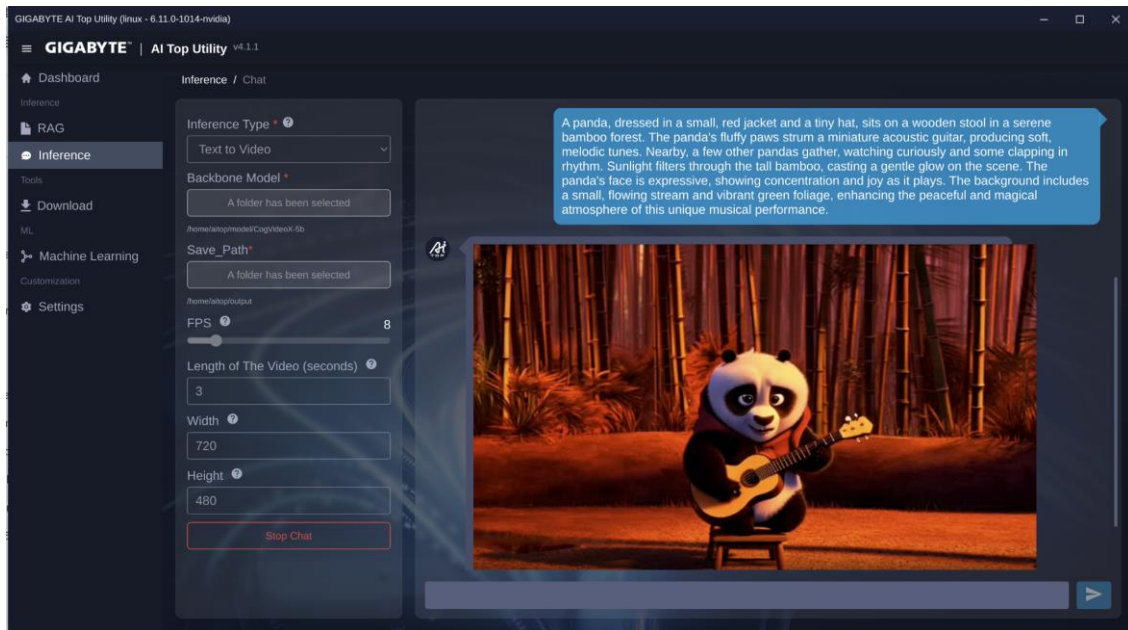


(5) Click the "Stop Chat" button if you want to stop the inference.

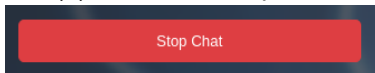


❖ Suppose we choose **Text-to-video**.

- (1) In the **"Backbone Model"** section, please select one of the text-to-video models, e.g., **"CogVideoX-5b"**
- (2) You can set the width and height attributes of the video. However, the model may impose certain restrictions. For example, CogVideoX-5b is limited to a height of 480 and a width of 720. Please refer to the model's Hugging Face page for more details.
- (3) You can set the attribute of FPS and the Length of The Video (second). Again, the model may impose certain restrictions. Please refer to the model's Hugging Face page for more details.
- (4) Click "Start Chat" to load the model. Then, you can input any prompt for the video you wish to generate. Refer to the model's page for best prompting practice. If the FPS attribute is set to 8 and the video length is 3 seconds, it will take approximately 2 to 3 minutes to process.



(5) Click the "Stop Chat" button if you want to stop the inference.

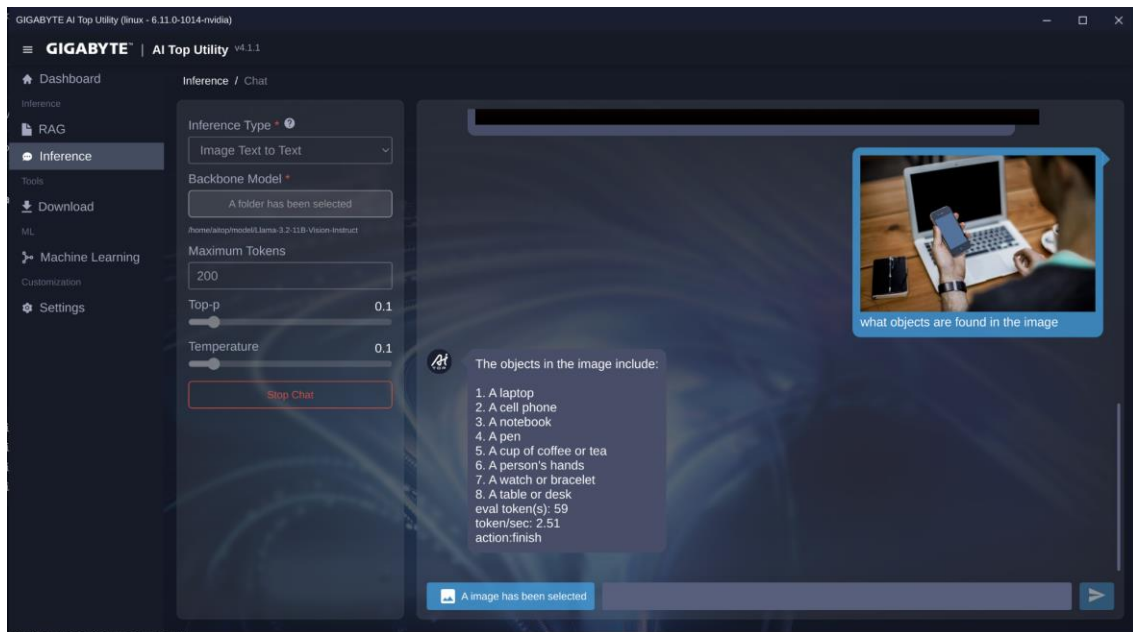


❖ Suppose we choose **Image Text to Text**.

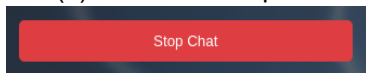
- (1) In the "**Backbone Model**" section, please choose one of the image-text-to-text models, such as "**Llama-3.2-11B-Vision-Instruct**"
- (2) Adjust **Maximum Tokens**, **Temperature**, and **Top-p** settings to control generation.



- (3) Click "Start Chat" to load the model. Then, you can choose the image you want the LLM to analyze and input your question.

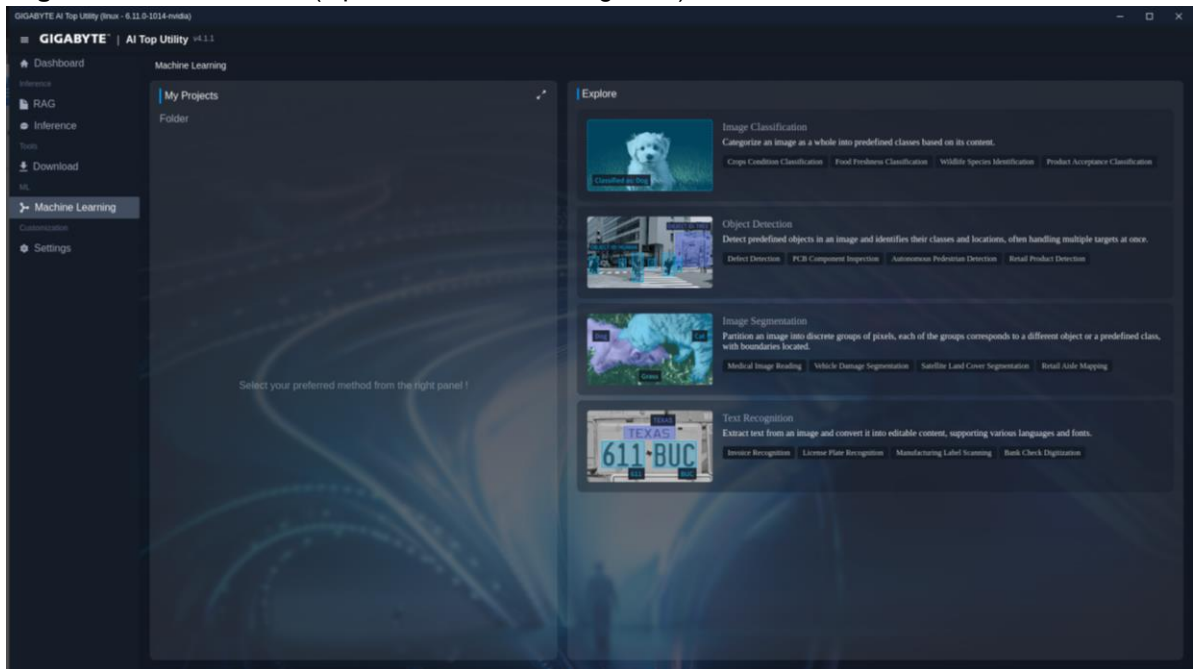


(4) Click the "Stop Chat" button if you want to stop the inference.

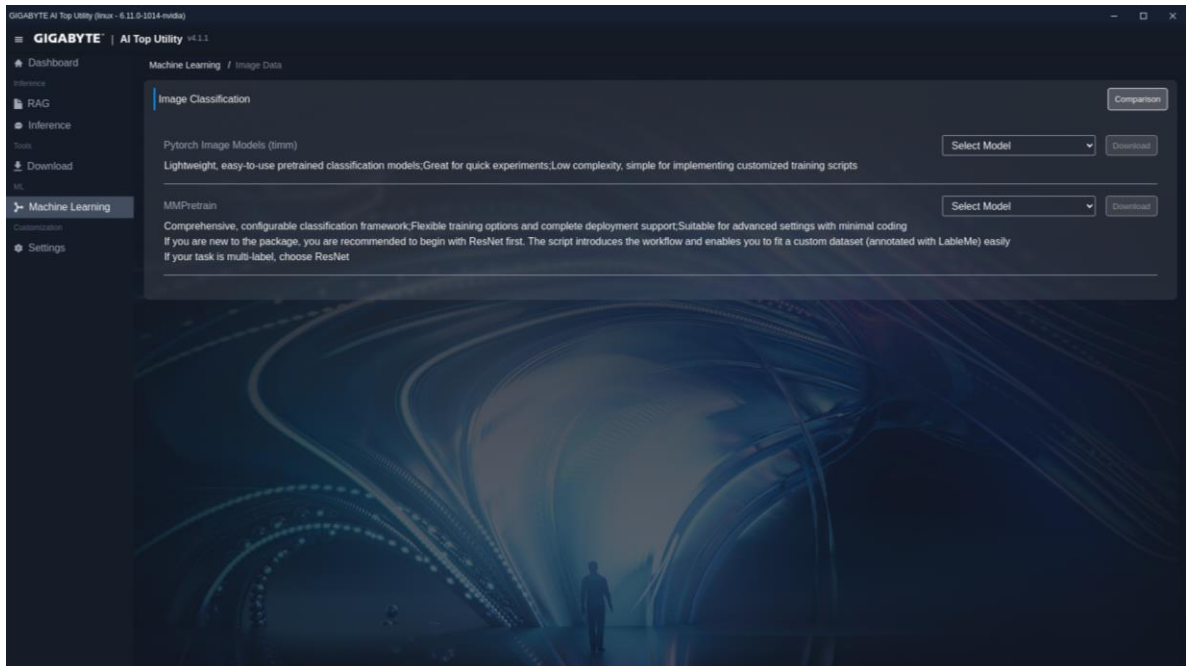


3-3. Machine Learning

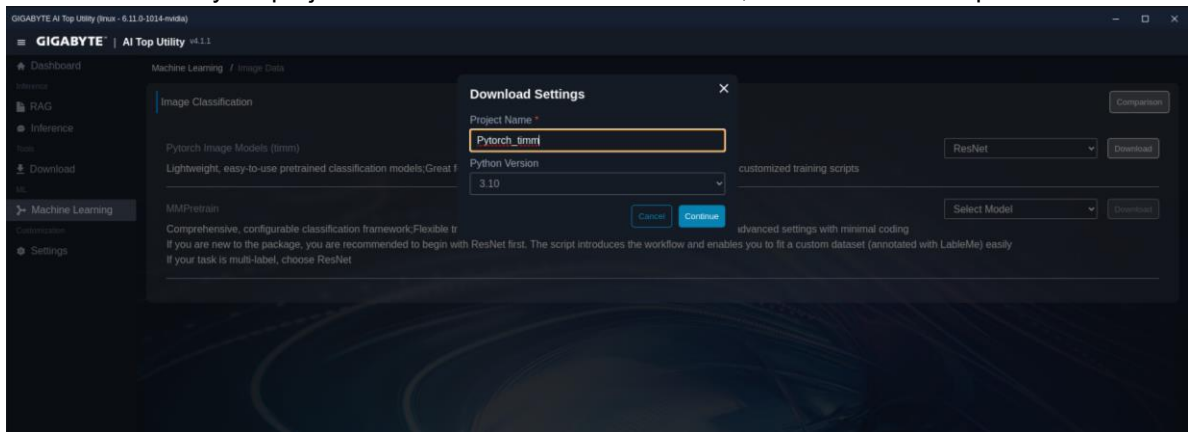
- (1) Click the **"Machine Learning"** tab. Choose the machine learning task type you want to work on. The currently supported types include: **image classification**, **object detection**, **image segmentation**, and **OCR** (Optical Character Recognition).



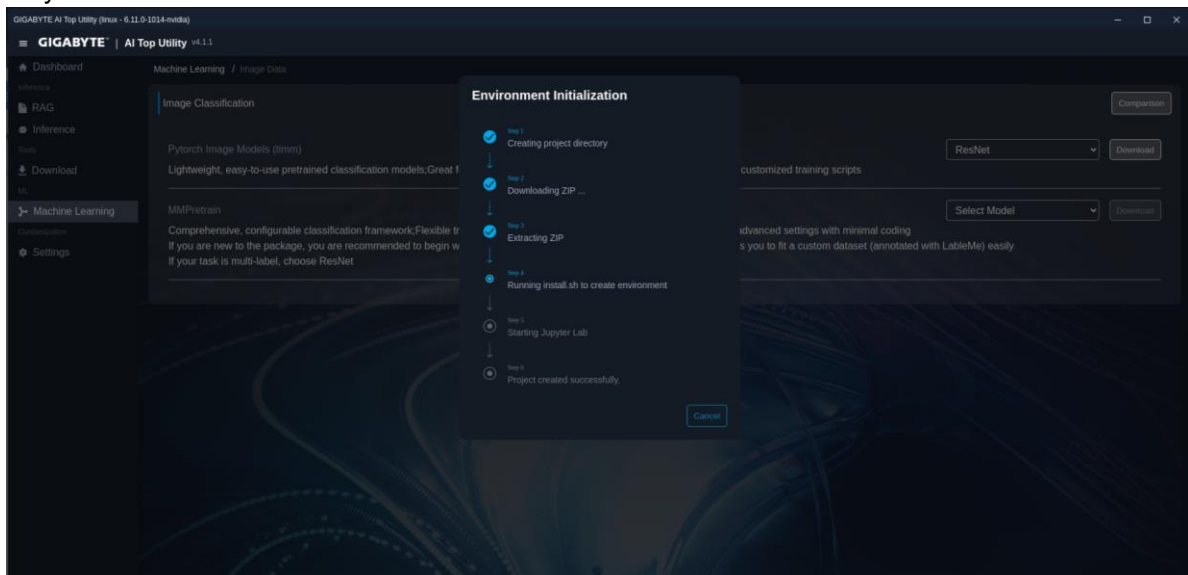
- (2) Select the package and model you want to use. If you are unsure which one to choose, click the "Comparison" button for guidance.



- (4) Enter a name for your project. The name can be modified later, but it cannot be duplicated.

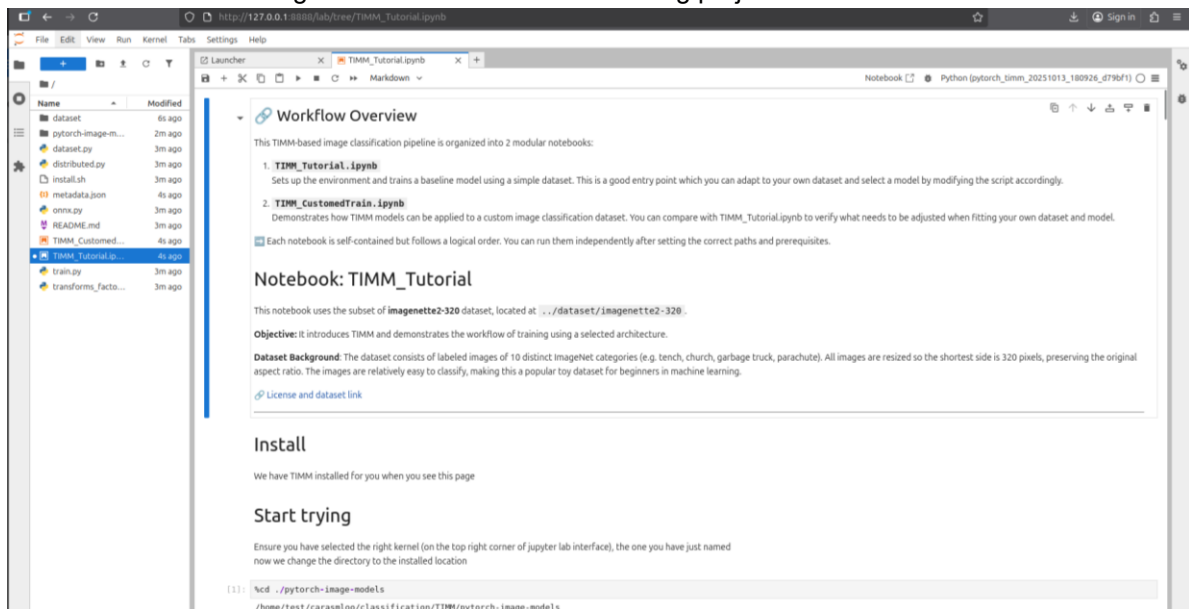


- (5) The project download and installation will begin. The process may take up to 60 minutes, depending on your internet connection.

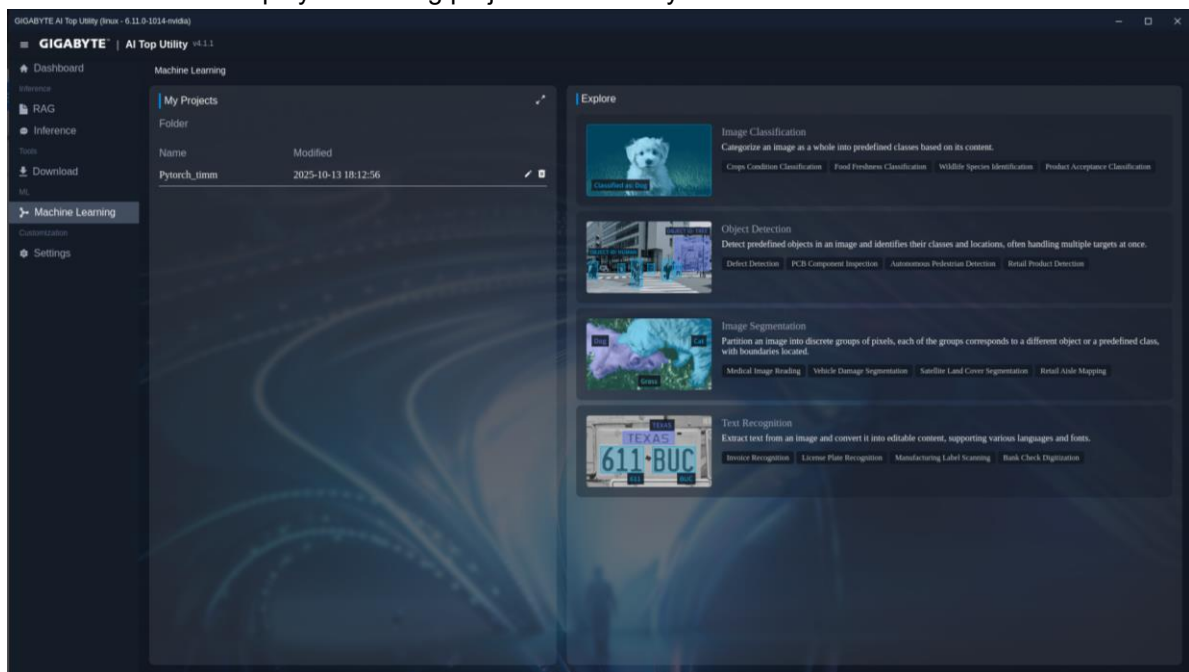


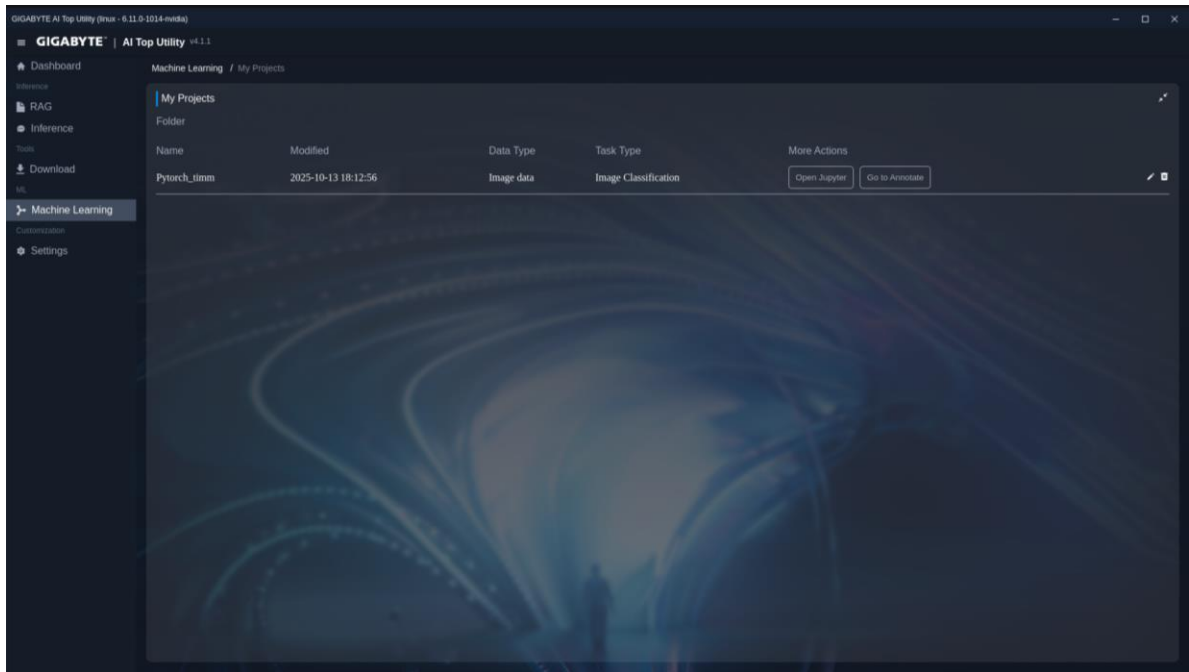
- (6) Once it is completed, a Jupyter Notebook will pop up automatically. Follow the instructions within the

notebook to start learning how to build a machine learning project.



(7) Click the button to display all existing projects saved on your device.





- (8) If you want to label your image dataset, click “Go to Annotate”. This will take you to the LabelMe or Labellmg interface, where you can make annotations.

4. Supported Models

4-1. Supported list of LLM/ LMM models

Task	Model	Model size
Text to Text	Qwen3-Coder-30B-A3B-Instruct-GGUF	17.65 GB
	Qwen3-30B-A3B-Instruct-2507-GGUF	17.65 GB
	Llama-4-Scout-17B-16E-Instruct-GGUF	61.95 GB
	GLM-4.5-Air-GGUF	72.95 GB
	Phi-3.5-MoE-instruct-GGUF	25.25 GB
	gpt-oss-20b-GGUF	11.85 GB
	gpt-oss-120b-GGUF	62.50 GB
	(NEED ACCESS) Llama-3.2-1B-Instruct	4.90 GB
	(NEED ACCESS) Llama-3.2-3B-Instruct	12.85 GB
	(NEED ACCESS) Meta-Llama-3.1-8B-Instruct	32.05 GB
Text to Image	(NEED ACCESS) stable-diffusion-3.5-large-turbo	71.55 GB
	(NEED ACCESS) stable-diffusion-3.5-large	71.55 GB
	(NEED ACCESS) FLUX.1-dev	57.85 GB
	(NEED ACCESS) FLUX.1-schnell	57.75 GB
	Qwen-Image	57.65 GB
Text to Video	CogVideoX-2b	13.75 GB
	CogVideoX-5b	21.45 GB
	Wan2.1-T2V-1.3B-Diffusers	28.85 GB

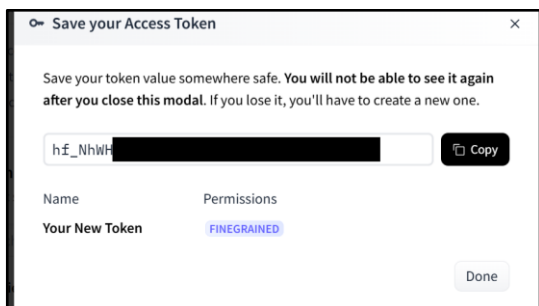
Image Text to Text	(NEED ACCESS) Llama-3.2-11B-Vision-Instruct	42.55 GB
	(NEED ACCESS) gemma-3-4b-it	8.59 GB
	(NEED ACCESS) gemma-3-12b-it	24.35 GB
	(NEED ACCESS) gemma-3-27b-it	54.85 GB
	Qwen2.5-VL-7B-Instruct	16.55 GB

4-2. Supported list of embedding models

Task	Model	Model size
Text	all-MiniLM-L6-v2	0.93 GB
Audio	larger_clap_general	0.73 GB
Image	CLIP-ViT-B-32-laion2B-s34B-b79K	2.37 GB
Video	CLIP-ViT-B-32-laion2B-s34B-b79K	2.37 GB

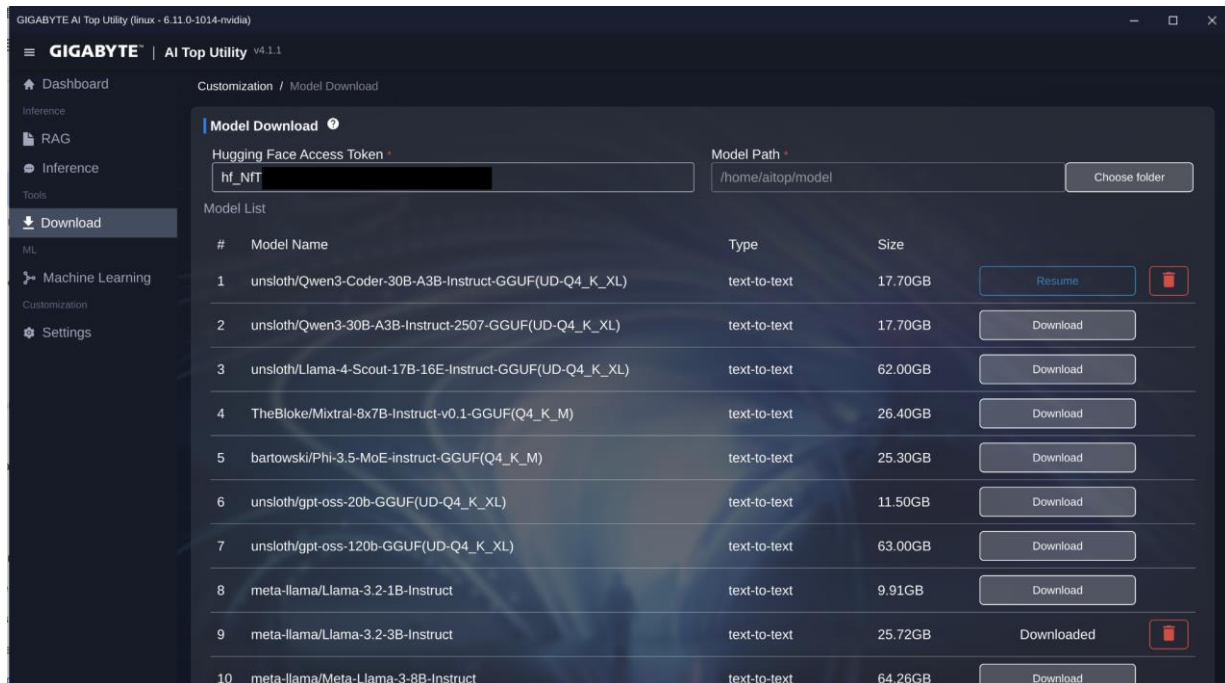
4-3. Download Model

The “model” folder in the Home directory is for locating your LLM backbone models. Before downloading, you need to create a Hugging Face account and generate a User Access Token to download LLM backbone models. You can refer to the [User access tokens](#) page for guidance on creating one. Be sure to store your Access Token securely so you do not lose it.



The software provides a convenient feature to help users download LLM/LMM models from Hugging Face without needing to run command-line syntax.

Note: Some LLM models (e.g, Llama family models) require users to agree to share contact information to access models in order to download from Hugging Face, so you need to visit the model's page to submit a request and grant approval.

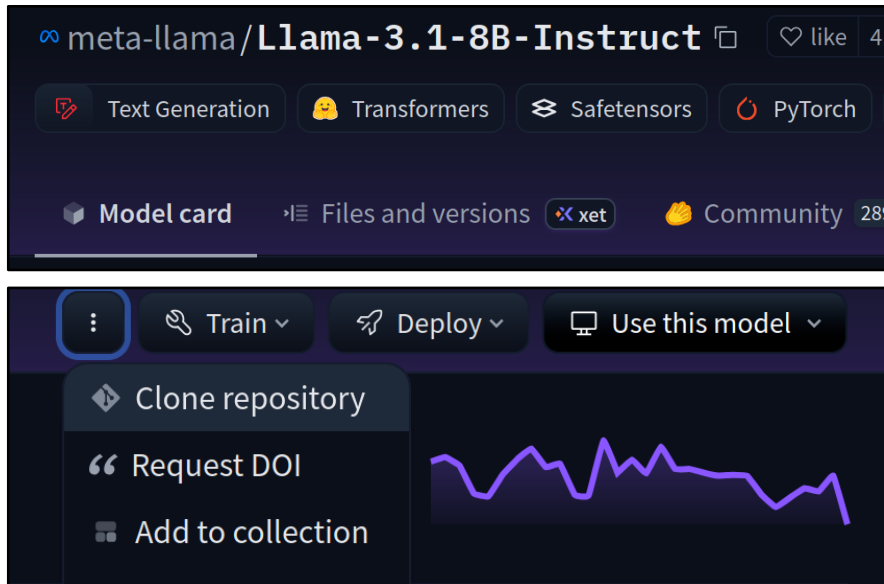


4-4. Download LLM/LMM models from Hugging Face

- The LLM backbone models can be downloaded from the Hugging Face website at <https://huggingface.co/> and should be saved in the “model” folder. In the “model” folder, right-click any space in the file explorer and select “Open in Terminal.” Then, run the commands below to install the required packages.

```
sudo apt install git-lfs
git lfs install
pip install huggingface_hub
```

- On the Hugging Face webpage, select a model — for example, Llama-3.1-8B-Instruct. Click the three-dot button on the right, then choose Clone repository. This will take you to the model’s repository link.



- Go back to the terminal. Then, run the commands below to download the model. You will be asked to enter your Hugging Face user name and password (Access Token).

```
git clone https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
```

Note: Some LLM models (e.g, Llama family models) require users to agree to share contact information to access models in order to download from Hugging Face, so you need to visit the model's page to submit a request and grant approval. Also, it is normal to be requested to enter your Hugging Face account and Access Token multiple times when downloading LLM or LMM models.