

# Documentation

Steps to install WSL:

1. Open PowerShell (or Windows Command Prompt) and enter: `wsl --install`
2. Once the process of installing your Linux distribution with WSL is complete, open the distribution (Ubuntu by default) using the Start menu. You will be asked to create a User Name and Password for your Linux distribution.

Steps to install pip on wsl:

1. `sudo apt upgrade`
2. `sudo apt install python3`
3. `sudo apt install python3-pip`

Steps to connect Visual Studio Code to Wsl Terminal:

1. Open Visual Studio Code, install the Remote - WSL extension: Open Visual Studio Code, go to the Extensions view by clicking on the square icon on the left sidebar, and search for "Remote - WSL". Install the "Remote - WSL" extension provided by Microsoft.
2. Then back to Wsl terminal, go to the directory you want to edit your code file. Input command line : `"code ."`. Then the visual studio code should pop up immediately.

Steps to install roboflow server:

1. `pip install roboflow --quiet`

Steps to install numpy:

1. `pip install numpy`

Steps to install tkinter:

1. `sudo apt-get install python3-tk`

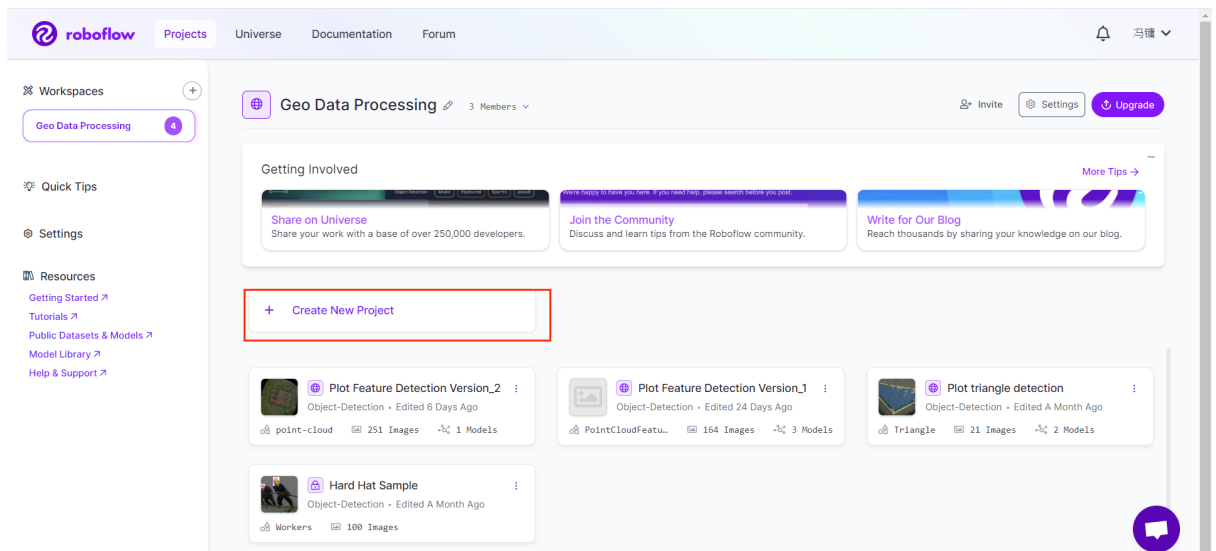
Steps to install tkinter:

1. `pip install pyasl`

**If you want to build a stronger or more customized model, you can use below steps to rebuild the program object detection model:**

Training Model:

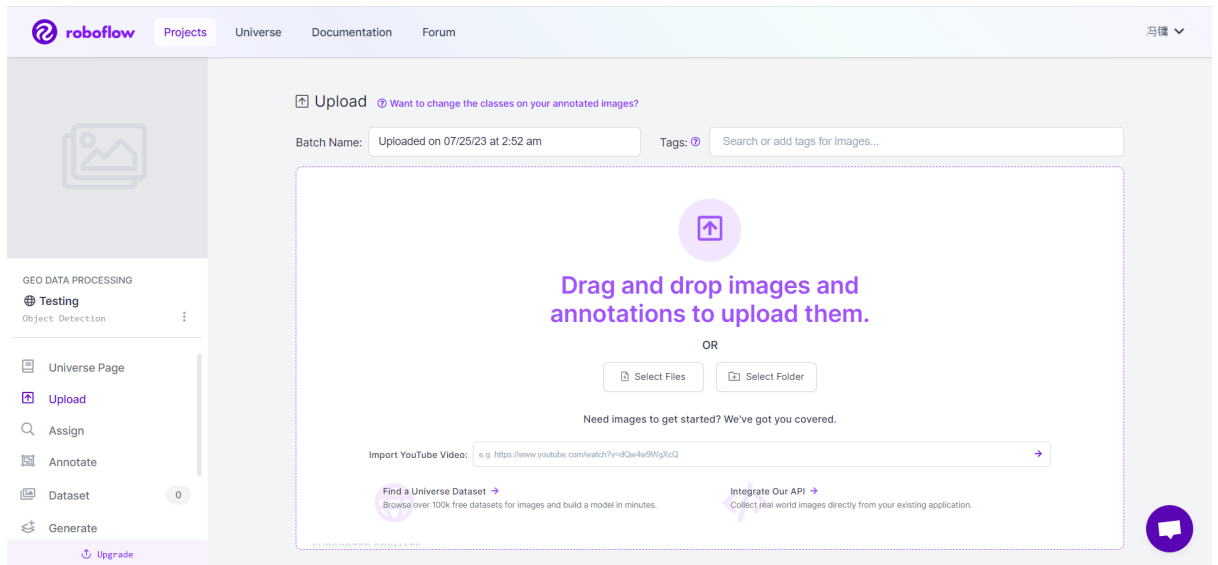
1. Google Roboflow and create your personal account. (<https://roboflow.com/>)
2. Click the Create New Project.



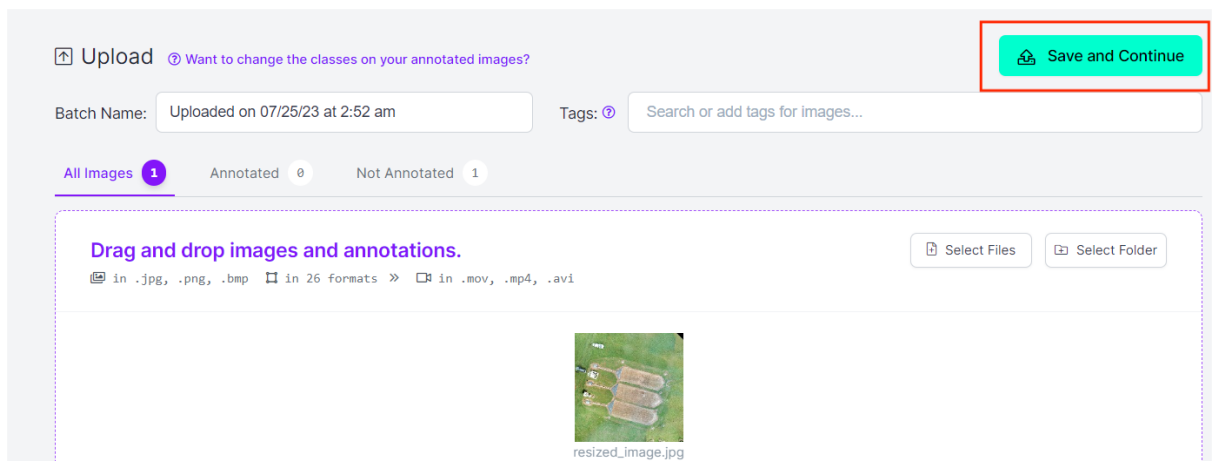
3. Input information about your project. After that click the 'Create Public Project' button.

A screenshot of the 'Create Project' modal form. The title is 'Create Project' with a close button. Below the title, it shows the project path 'Geo Data Processing / New Public Project'. The form has four main sections: 'Project Type' with a dropdown menu showing 'Object Detection (Bounding Box)'; 'What Are You Detecting?' with a dropdown menu showing 'PointCloudFeature'; 'Project Name' with a text input field containing 'Testing'; and 'License' with a dropdown menu showing 'CC BY 4.0'. At the bottom, there are two buttons: 'Cancel' and 'Create Public Project'.

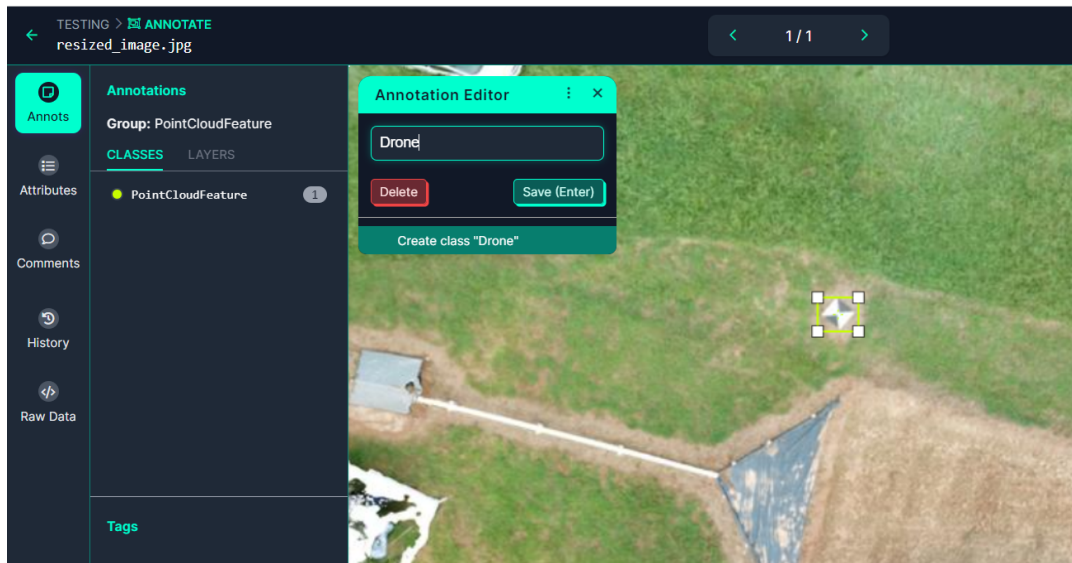
4. Input your dataset.



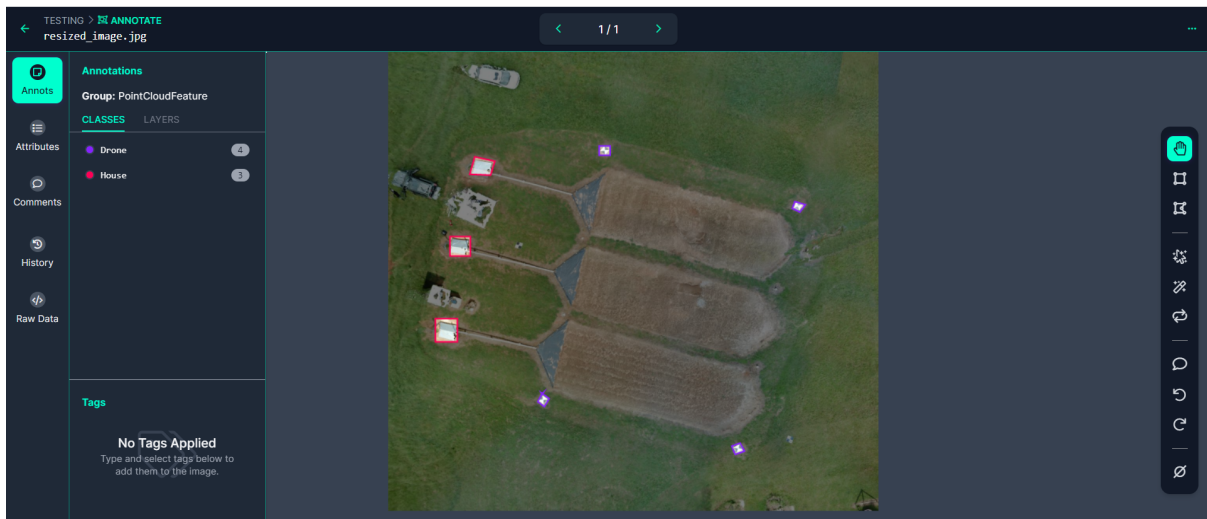
5. Click the 'Save and Continue' after the you have uploaded your image



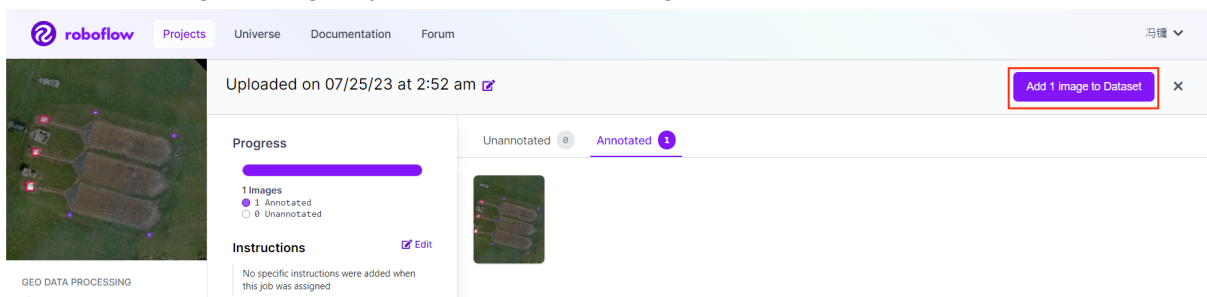
6. Draw bounding box to the object if you want the Model to automatically detect later.



7. After drawing the bounding box, the image will be like this:



8. After finishing all images, you can click 'Add Image to Dataset'



## 9. Generate your dataset.

The screenshot shows the Roboflow web interface for a 'Testing Dataset'. In the left sidebar, the 'Generate' button is highlighted with a red box. The main content area is titled 'Generating New Version' and includes instructions: 'Prepare your images and data for training by compiling them into a version. Experiment with different configurations to achieve better training results.' Below this, there are three steps: 1. Source Images (Images: 1, Classes: 2, Unannotated: 0), 2. Train/Test Split (Training Set: 1 images, Validation Set: images, Testing Set: images), and 3. Preprocessing (What can preprocessing do? Decrease training time and increase performance by applying image transformations to all images in this dataset).

## 10. Get the link of dataset, in order to train it using google colab.

The screenshot shows the Roboflow web interface for a specific dataset version: '2023-07-25 3:07am'. The 'Get Snippet' button is highlighted with a red box. The interface displays training options: 'Train with AutoML' (Let AutoML choose, train, and optimize a state of the art model on cloud GPUs to use for Label Assist and deploy via auto-scaling API or on edge devices like Jetson, OAK, and Raspberry Pi. Learn More ») and 'Custom Train' (Choose, customize, and train a state of the art model from our model library in a Jupyter Notebook or Python script to use for Label Assist and use Roboflow Deploy to deploy it to the cloud or on your own hardware. Learn More »). Below these, there is a 'Start Training' button and a dropdown menu showing 'YOLOv8 (New!)' with a 'Get Snippet' button. The 'IMAGES' section shows a list of images, with '1 images' displayed. A 'View All Images »' link is also present.

The screenshot shows an 'Export' dialog box. The 'Format' dropdown is set to 'YOLOv8'. Below it, there is a message: 'TXT annotations and YAML config used with YOLOv8.' There are two radio buttons: 'download zip to computer' and 'show download code', with 'show download code' selected. At the bottom, there are 'Cancel' and 'Continue' buttons. The 'Continue' button is highlighted with a red box.

Copy the code inside the red rectangle

### Your Download Code

Jupyter

Terminal

Raw URL

Paste this snippet into [a notebook from our model library >>](#) to download and unzip [your dataset >>](#):

```
!pip install roboflow
```

```
from roboflow import Roboflow
rf = Roboflow(api_key="████████████████████")
project = rf.workspace("geo-data-processing").project("testing-mct3b")
dataset = project.version(1).download("yolov8")
```

**Warning:** Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account. Acceptable use policy applies.

Done

11. Open the Google Colab and copy the code to your own Google Colab script (<https://colab.research.google.com/drive/1ghV3ore6JVpzQQ6a1C1BVWG48RXZLi63?usp=sharing>)

12. Replace this part with your own dataset.

```
[ ] !pip install roboflow --quiet
```

```
from roboflow import Roboflow
rf = Roboflow(api_key="OCgMo7f3BUDqQEW2mxwo")
project = rf.workspace("geo-data-processing").project("plot-feature-detection-version_2")
dataset = project.version(3).download("yolov8")
```

13. Run the whole Google Colab script.

14. After training, you shall see you API Key on the Roboflow website.

The screenshot shows the Roboflow web interface. On the left, a sidebar displays project details for 'Plot Feature Detection Version\_2', including a 'Deploy' button highlighted with a red box. The main content area shows the project's version history and a code editor for Python. The code editor displays a snippet for inferring on local and hosted images, with a red box highlighting the initialization of the Roboflow client and the selection of the project and model. The right sidebar contains links to documentation, an example web app, a video inference script, and deployment instructions for NVIDIA Jetson.

**Roboflow Interface Details:**

- Project Name:** Plot Feature Detection Version\_2
- Model:** Roboflow 3.0 Object Detection (Fast)
- Version:** 2
- Generated:** Jul 6, 2023
- Metrics:** mAP: 96.8%, Precision: 87.2%, Recall: 98.2%
- Code Snippet (Python):**

```
from roboflow import Roboflow
rf = Roboflow(api_key="0CgMo7f3BUDqqEW2mxwo")
project = rf.workspace().project("plot-feature-detection-version_2")
model = project.version(2).model

# Infer on a local image
print(model.predict("your_image.jpg", confidence=40, overlap=30).json())

# visualize your prediction
# model.predict("your_image.jpg", confidence=40, overlap=30).save("prediction.jpg")

# infer on an image hosted elsewhere
# print(model.predict("URL_OF_YOUR_IMAGE", hosted=True, confidence=40, overlap=30).json())
```

15. Copy the API back to the program and replace it with the original code in the Detection Model.

```
image_original_size = crop_center(img, percent)

#Yolo Object Detection Model
rf = Roboflow(api_key="0CgMo7f3BUDqqEW2mxwo")
project = rf.workspace().project("plot-feature-detection-version_2")
model = project.version(2).model
```