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 pylas library:

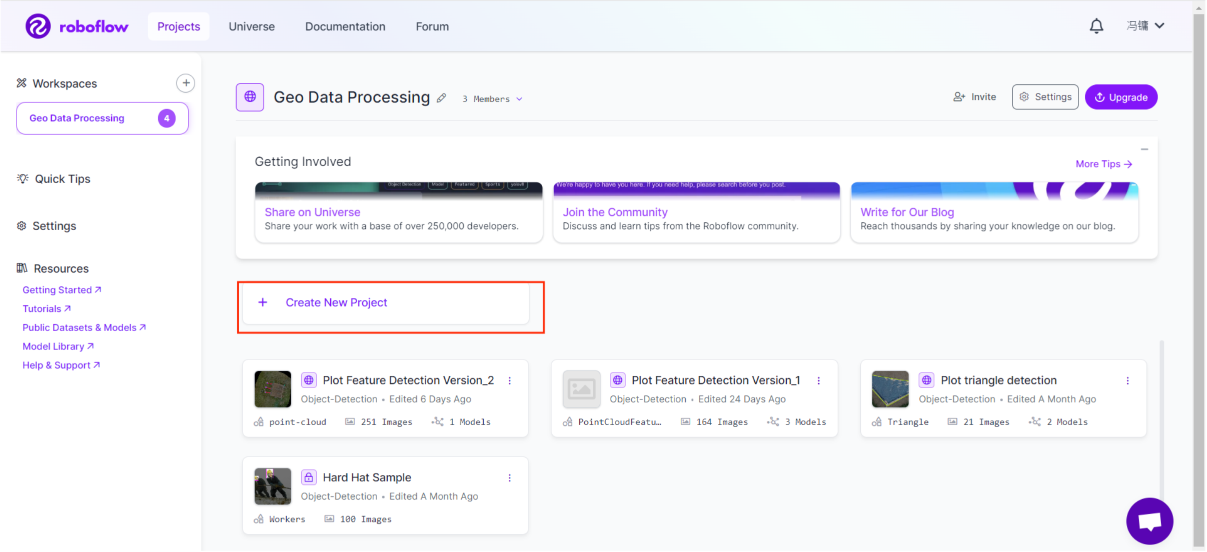
1. `pip install pylas`

If you want to build a stronger or more customized model, you can use the 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.



Share on Universe
Share your work with a base of over 250,000 developers.

Join the Community
Discuss and learn tips from the Roboflow community.

Write for Our Blog
Reach thousands by sharing your knowledge on our blog.

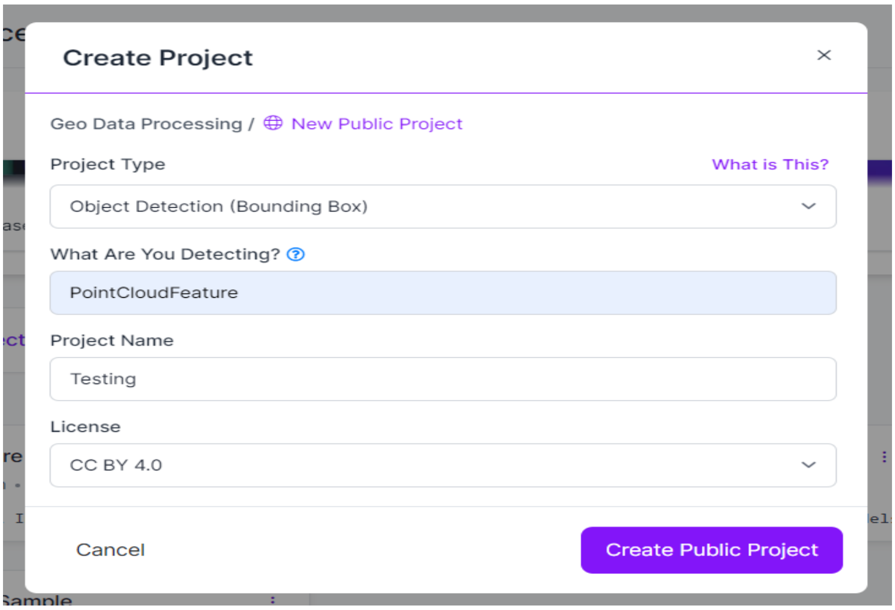
Plot Feature Detection Version_2
Object-Detection • Edited 6 Days Ago
point-cloud 251 Images 1 Models

Plot Feature Detection Version_1
Object-Detection • Edited 24 Days Ago
PointCloudFeatu… 164 Images 3 Models

Plot triangle detection
Object-Detection • Edited A Month Ago
Triangle 21 Images 2 Models

Hard Hat Sample
Object-Detection • Edited A Month Ago
Workers 100 Images

3.Input information about your project. After that, click the ‘Create Public Project’ button.



Geo Data Processing / New Public Project

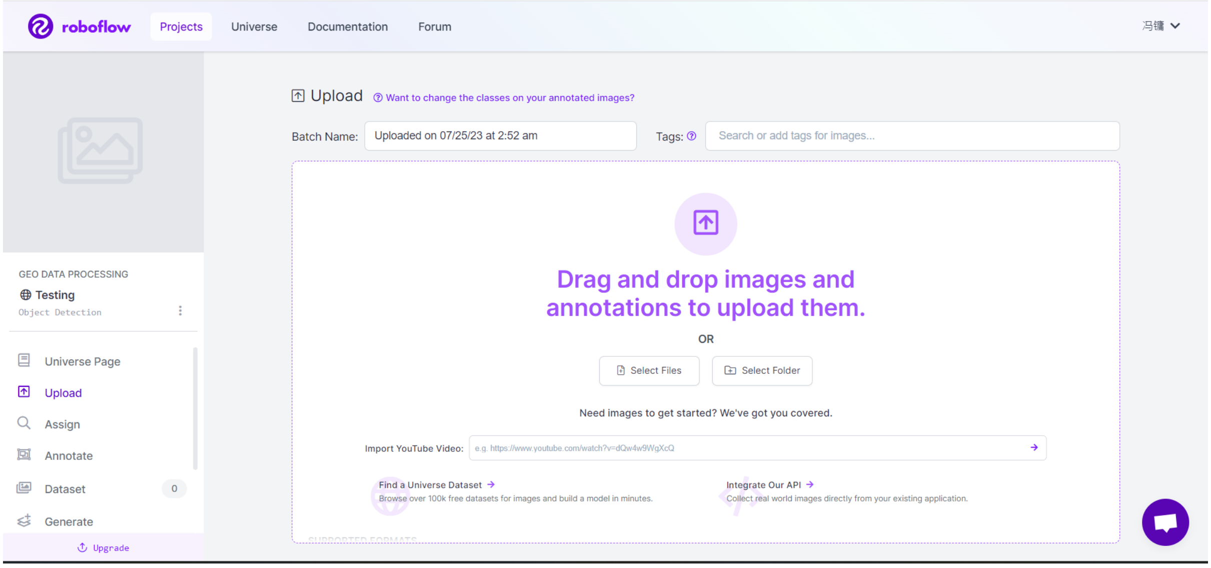
Project Type: Object Detection (Bounding Box)

What Are You Detecting? PointCloudFeature

Project Name: Testing

License: CC BY 4.0

4. Input your dataset.



Want to change the classes on your annotated images?

Batch Name: Uploaded on 07/25/23 at 2:52 am

Tags: Search or add tags for images...

Drag and drop images and annotations to upload them.

OR

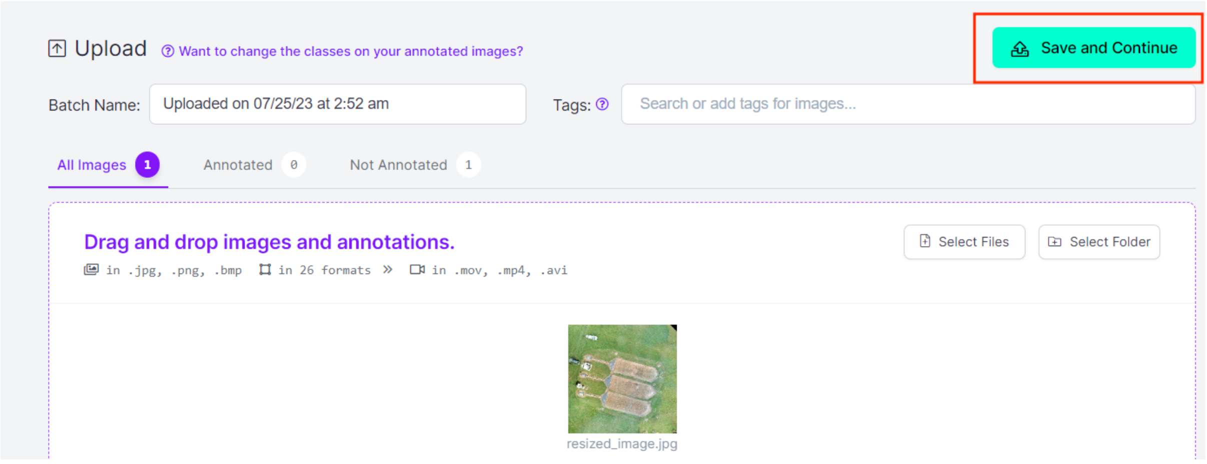
Need images to get started? We've got you covered.

Import YouTube Video: e.g. https://www.youtube.com/watch?v=dQw4w9WgXcQ

Find a Universe Dataset →
Browse over 100k free datasets for images and build a model in minutes.

Integrate Our API →
Collect real world images directly from your existing application.

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



Want to change the classes on your annotated images?

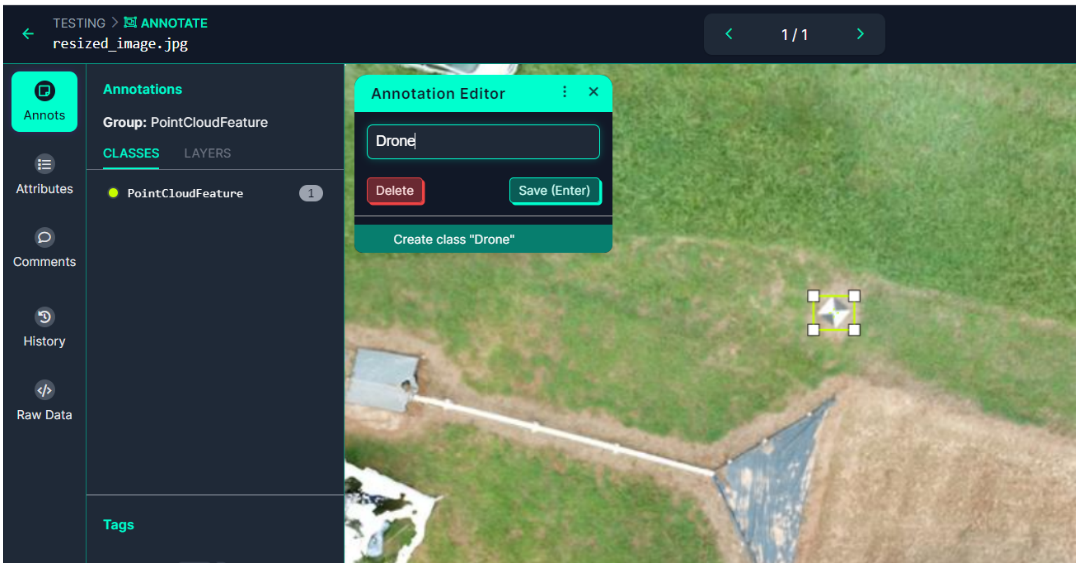
Batch Name: Uploaded on 07/25/23 at 2:52 am

Tags: Search or add tags for images...

Drag and drop images and annotations.

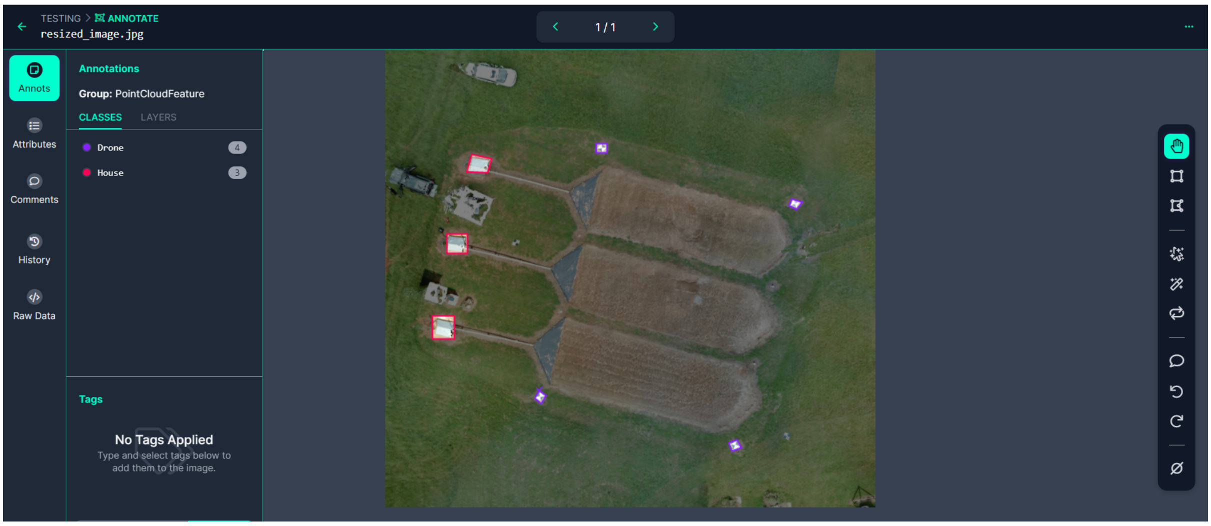
in .jpg, .png, .bmp in 26 formats » in .mov, .mp4, .avi

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



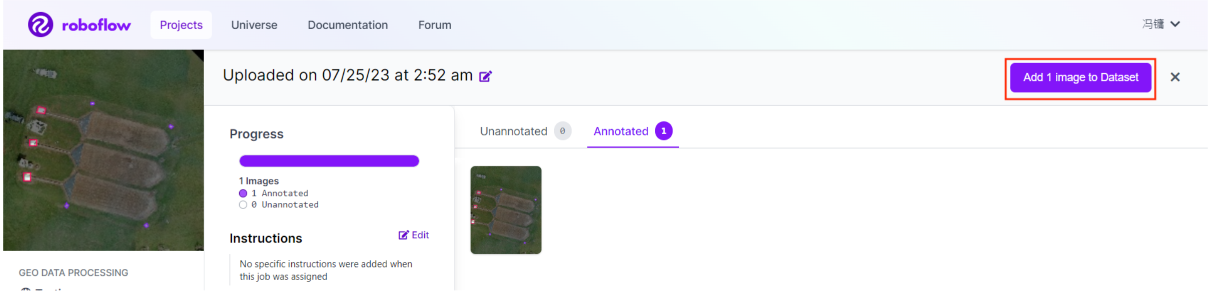
Drone

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



No Tags Applied
Type and select tags below to add them to the image.

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

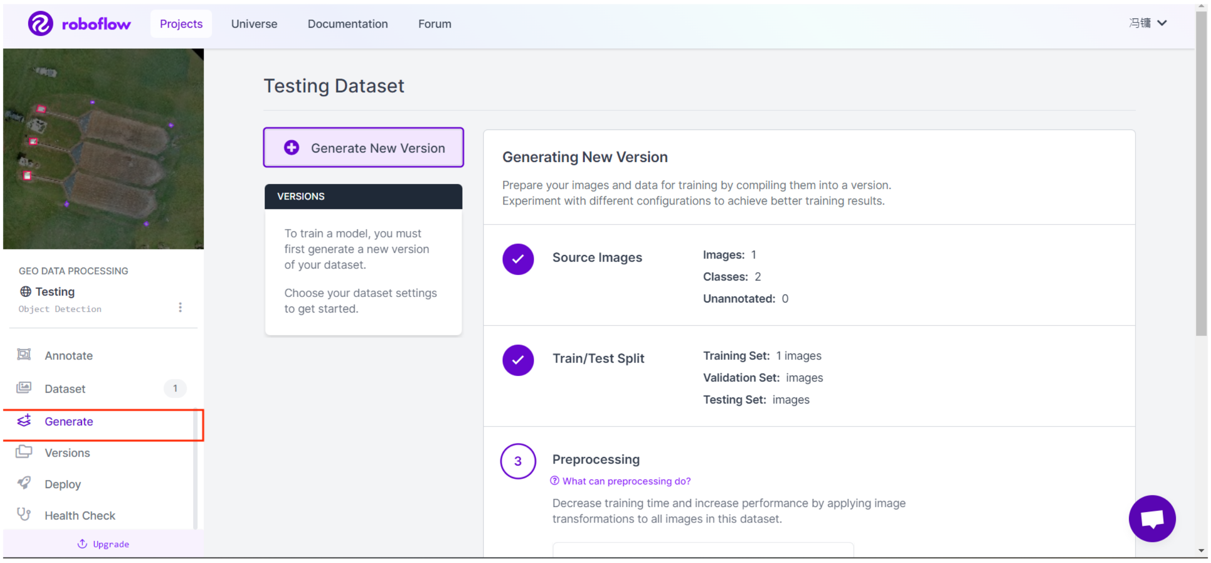


Uploaded on 07/25/23 at 2:52 am

1 Images
● 1 Annotated
○ 0 Unannotated

No specific instructions were added when this job was assigned

9. Generate your dataset.



roboflow Projects Universe Documentation Forum 冯镭

Testing Dataset

GEO DATA PROCESSING

Testing

Object Detection

Annotate

Dataset 1

Generate

Versions

Deploy

Health Check

Upgrade

Generate New Version

VERSIONS

To train a model, you must first generate a new version of your dataset.

Choose your dataset settings to get started.

Generating New Version

Prepare your images and data for training by compiling them into a version. Experiment with different configurations to achieve better training results.

✓ Source Images

Images: 1
Classes: 2
Unannotated: 0

✓ Train/Test Split

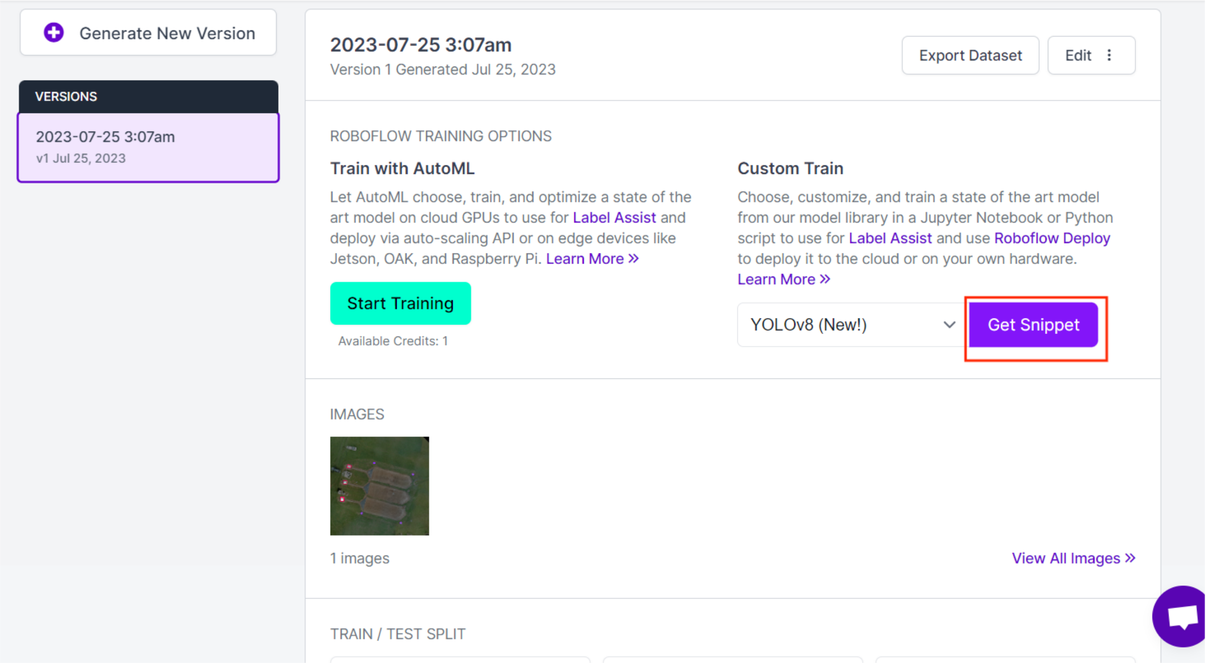
Training Set: 1 images
Validation Set: images
Testing Set: images

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 the dataset, in order to train it using google colab.



Generate New Version

VERSIONS

2023-07-25 3:07am
v1 Jul 25, 2023

2023-07-25 3:07am
Version 1 Generated Jul 25, 2023

Export Dataset Edit

ROBOFLOW 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 »

Start Training

Available Credits: 1

Custom Train

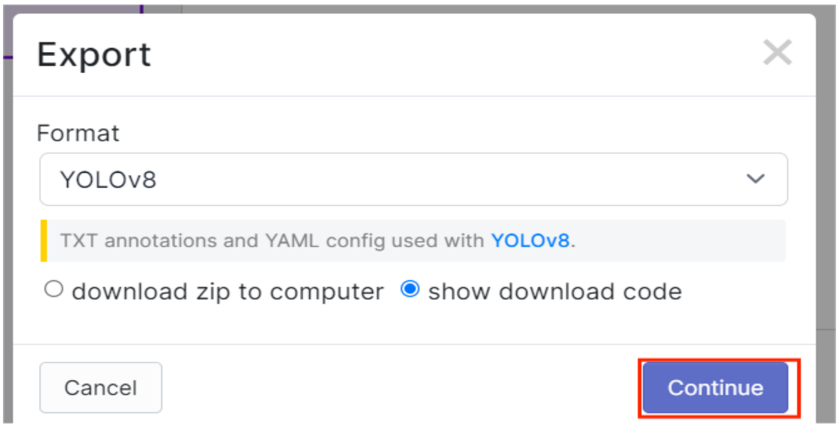
Choose, customize, and train a state of the art model from our model library in a Jupyter Notebook or Python script to use for Label Assist and use Roboflow Deploy to deploy it to the cloud or on your own hardware. Learn More »

YOLOv8 (New!) Get Snippet

IMAGES

1 images View All Images »

TRAIN / TEST SPLIT



Export

Format

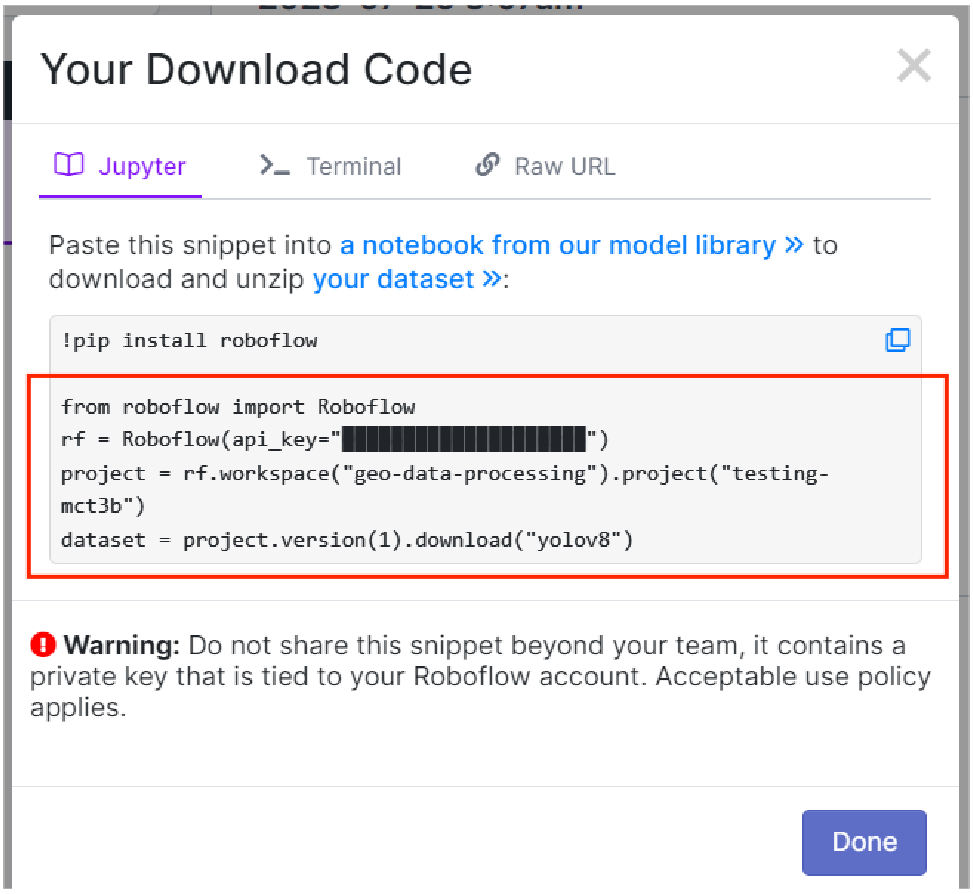
YOLOv8

TXT annotations and YAML config used with YOLOv8.

○ download zip to computer ● show download code

Cancel Continue

Copy the code inside the red rectangle



Paste this snippet into a notebook from our model library » to download and unzip your dataset »:

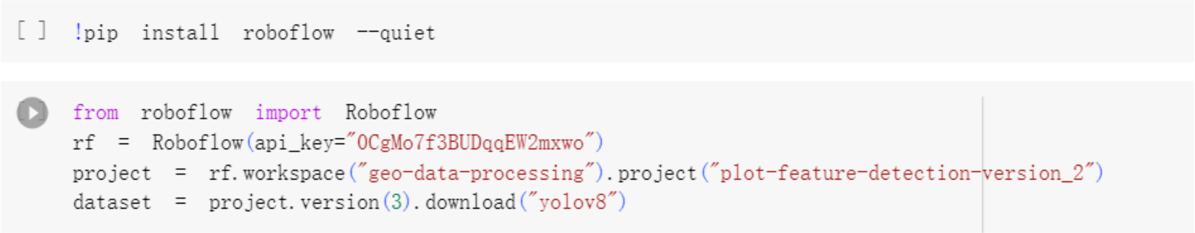
```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="[redacted]")
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.

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.

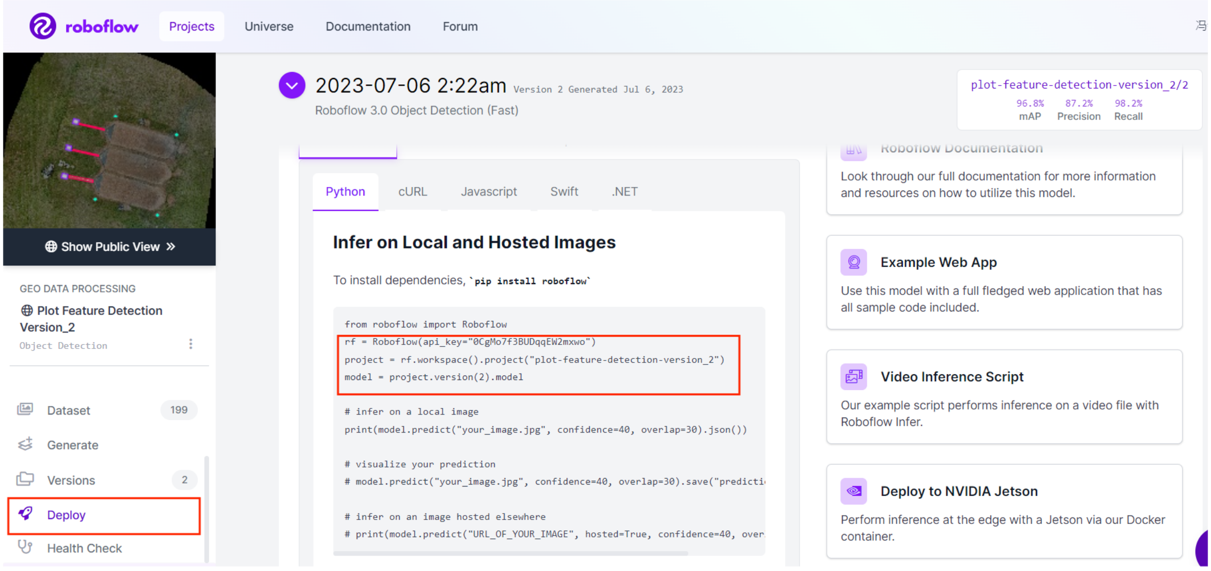


```
[ ] !pip install roboflow --quiet
```

```
from roboflow import Roboflow
rf = Roboflow(api_key="0CgMo7f3BUDqqEW2mxwo")
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.



2023-07-06 2:22am Version 2 Generated Jul 6, 2023

Roboflow 3.0 Object Detection (Fast)

plot-feature-detection-version_2/2

mAP	Precision	Recall
96.8%	87.2%	98.2%

GEO DATA PROCESSING

Plot Feature Detection Version_2

Object Detection

Infer on Local and Hosted Images

To install dependencies, `pip install roboflow`

```
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("predicti

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

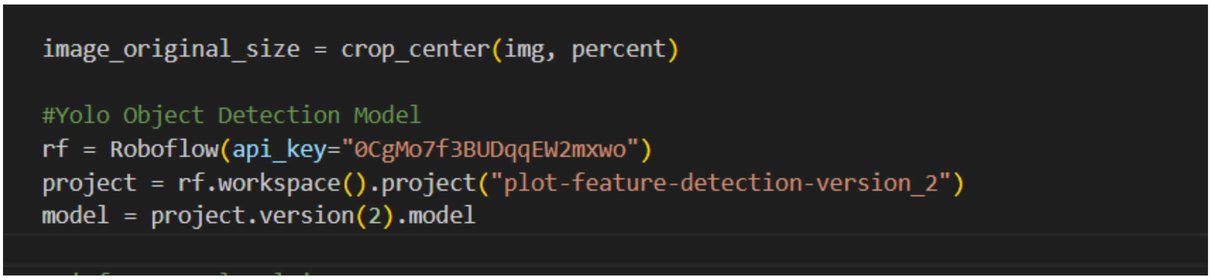
Look through our full documentation for more information and resources on how to utilize this model.

Use this model with a full fledged web application that has all sample code included.

Our example script performs inference on a video file with Roboflow Infer.

Perform inference at the edge with a Jetson via our Docker container.

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
```