

Label Studio

During this work you'll use Label Studio, a labeling tool, to annotate some data.

Installation

The best practice is to [setup a new environment using conda](#) for each of your projects, this will avoid conflict between the libraries you installed for project A and the ones you're installing for project B.

Setup the environment for Label Studio with the following commands in your terminal:

```
conda create --name LabelStudio python=3.10
conda activate LabelStudio
pip install label-studio
```

The first line will create the environment named LabelStudio and install Python 3.10

The second line switches you from the current environment (probably "base") to the LabelStudio environment.

The last line installs label-studio with a version superior or equal to 1.9.0

Being in the right environment, you can launch Label Studio with the following command in the terminal:

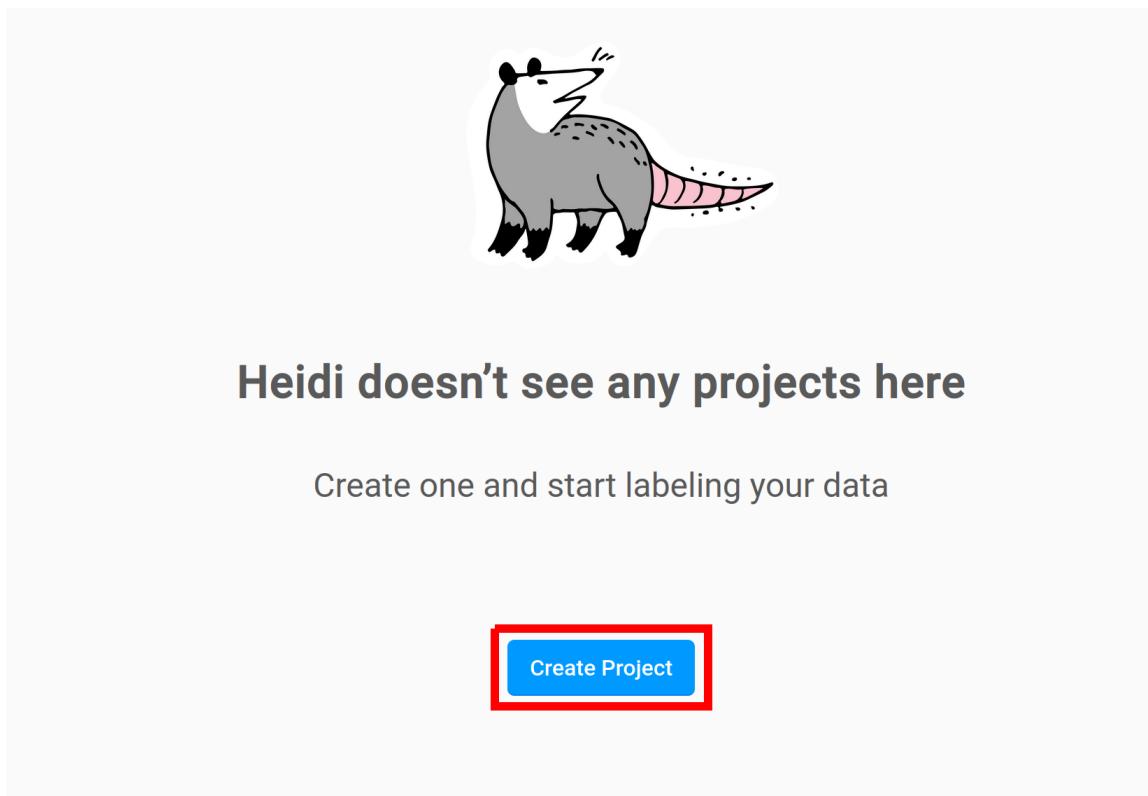
```
label-studio
```

Then in your browser navigate to <http://localhost:8080/> if it's not done automatically and create your account.

Usage

We'll now see how to annotate data using Label Studio

1. Start by creating a new project:



Create Project

Project Name Data Import Labeling Setup Delete Save

Project Name
Project in Data Science

Description
Create mask for the Project in Data Science course

Workspace + Enterprise
Select an option

Simplify project management by organizing projects into workspaces. [Learn more](#)

2. Go to the Data Import section and add your images with the Upload Files

Create Project Project Name Data Import Labeling Setup Delete Save

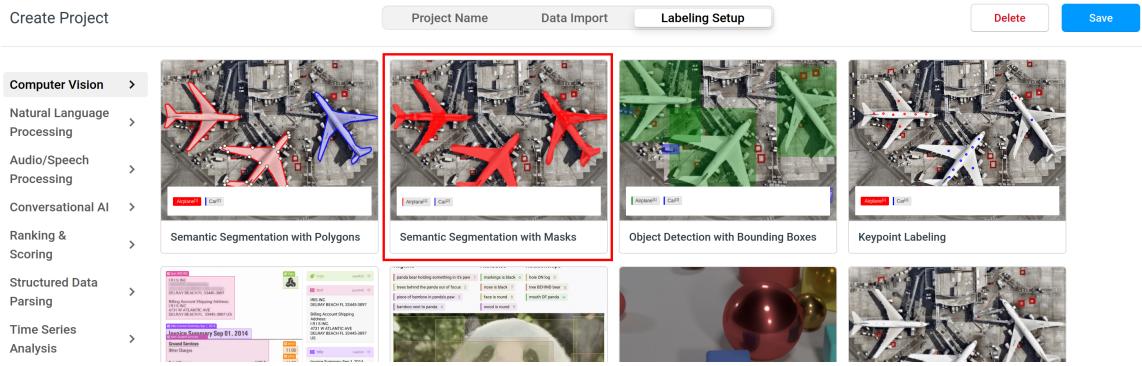
Dataset URL Add URL Upload Files

Drag & drop files here or click to browse

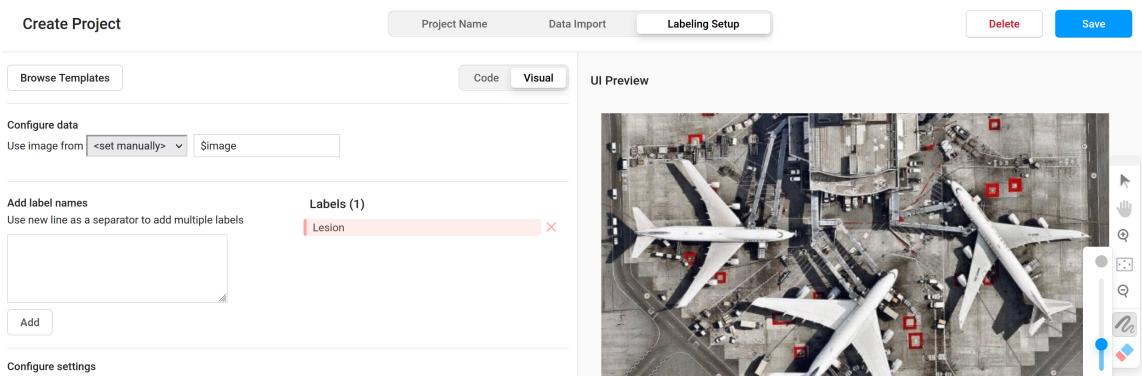


Text: txt
Audio: wav, mp3, flac, m4a, ogg
Video: mpeg4/H.264 webp, webm*
Images: jpg, jpeg, png, gif, bmp, svg, webp
HTML: html, htm, xml

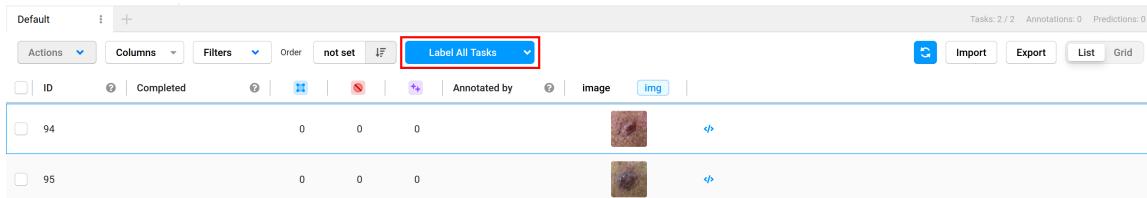
3. Go to the Labeling Setup section and select Semantic Segmentation with Mask



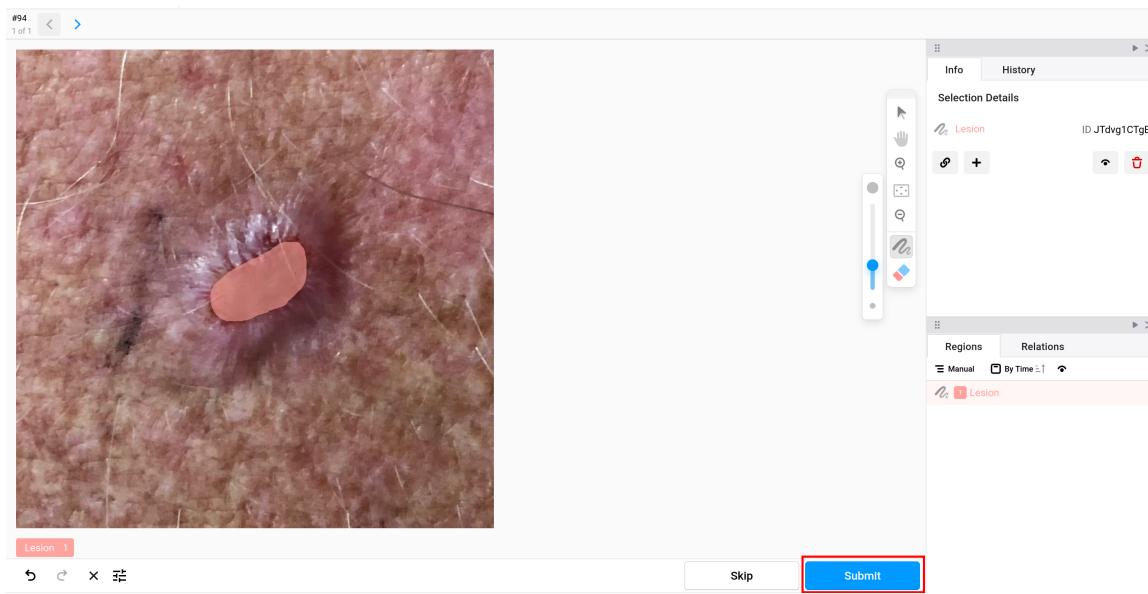
4. Change the Labels to match your task and click save. Your project is now setup.



5. Click "Label All Tasks"



6. For each image select the class and brush the zone of interest with your cursor then click submit



Export

On the homepage of your project, go into export and select JSON. Using the Python script `convert_json_to_image.py` you'll be able to convert this JSON more easily than if you export to PNG.

The script `convert_json_to_image.py` will transform the exported JSON into real mask images with the following command: (See script for more details)

```
python convert_json_to_image.py PATH_TO_JSON PATH_TO_OUTPUT_M
```

The script `merge_masks.py` will create a CSV based on the masks from several users to get the area of each lesion. It can be used with the following command: (See script for more details)

```
python merge_masks.py PATH_MASK_FOLDERS
```

Your task

Now that you know how Label Studio works:

1. Annotate some data from the dataset
2. Convert the JSON to masks with `convert_json_to_image.py`
3. Share your masks with another student
4. Use `merge_masks.py` to create a CSV with the area for each mask

Annotation with numerical values

For this, we won't use a template as before but in your labeling setup, go to the Code section and add the following code:



The screenshot shows the Label Studio interface. On the left, a sidebar lists options: General, Labeling Interface (which is selected), Instructions, Machine Learning, Cloud Storage, Webhooks, and Danger Zone. In the center, there's a "Browse Templates" section with a "Code" tab selected, displaying the following XML code:

```
<View>
  <Image name="image" value="$image"/>
  <Text name="txt_feature1" value="Feature 1"/>
  <Number name="feature1" toName="txt_feature1" min="0" defaultValue="0"/>
  <Text name="txt_feature2" value="Feature 2"/>
  <Number name="feature2" toName="txt_feature2" min="0" defaultValue="0"/>
</View>
```

To the right, the "UI Preview" section shows an aerial view of an airport tarmac with four large aircraft. Red bounding boxes are drawn around each aircraft. Below the preview are two input fields labeled "Feature 1" and "Feature 2", each containing the value "0".

You can then export as a CSV or JSON.