Research Project

Installation manual

## Content

[Necessities 3](#_Necessities)

[Source code 3](#_Source_code)

[Add extra data (optional) 3](#_Add_extra_data)

[Create weights 9](#_Create_weights)

## Necessities

* A laptop/pc with a GPU and python & pytorch installed (GPU not necessary but recommended)
* A drawing tablet (makes life easier) or pen and paper
* Drawing program (illustrator, photoshop, paint, …)

## Source code

You can find the source code in the repository linked below. The latest version will always be on the main branch, but each version has its own branch.

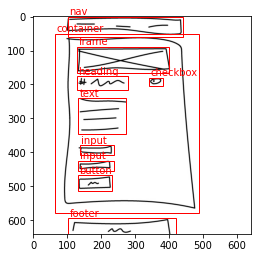
<https://github.com/SoensJakob/Research-Project>

## Add extra data (optional)

In the latest version there are 110 samples in the data folder. 85 training samples, 15 validation samples and 10 test samples in de testset map.

I made these samples with a drawing tablet in illustrator, added bounding boxes using CVAT and created the dataset using roboflow.  
When creating new data. Make sure the wireframes are in a photo format such as .jpg or .png. I used a 640 x 640 canvas to draw, but you can use a different format. Thought results and model speed might be different.

When drawing wireframes make sure to use the right convention. The model will have an easier time recognizing the different elements and the accuracy will go up.



### Conventies

* Navigation (nav)
* Container
* Frame
* Heading
* Text
* Input
* Button
* Footer
* Checkbox

Diagram

Description automatically generated

I’ve added nine elements, but it is very easy to add more. Just make sure the elements are distinguishable from each other.   
For more information on this TeleportHQ had a similar idea with there vision API. In the blogpost linked below they go into detail on how important the different conventions are.

<https://teleporthq.io/blog/new-vision-api>

### Bounding boxes

When you’ve made new data, you also have to add bounding boxes to this. So the model knows where all the elements are. There are many tools for this, but I used [CVAT](https://cvat.org) for the bounding boxes and [Roboflow](https://roboflow.com/) to create the dataset.

Wanneer je nieuwe data hebt aangemaakt moet je hier ook bounding boxes aan toevoegen om aan te tonen waar de verschillende elementen staan. Ik heb dit gedaan via [CVAT](https://cvat.org) en [Roboflow](https://roboflow.com/).

**CVAT**

1. Create a new task
2. Add labels in the constructor
3. Upload the dataset and click “submit”
4. Go to tasks and click “job”
5. Now you can start creating bounding boxes
   1. Use shortcut “N” to create a new bounding box
   2. When you’ve drawn a bounding box around the element give it the right label
   3. Repeat for every element in the dataset
6. Wanneer alle elementen een bounding box hebben kan je de dataset downloaden op de task pagina. Kies hier “Pascal VOC”

Graphical user interface, application

Description automatically generatedGraphical user interface, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Diagram

Description automatically generated

A screenshot of a computer

Description automatically generated

**Roboflow**

1. Create a new project
2. Upload the images from CVAT
3. Click “Finish Uploading” and choose train, valid and test split
4. Remove all preprocessing steps but check if the labels haven’t been renamed. You can change this by adding a modify class step then click generate
5. Export the dataset and put it in the data folder

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, application

Description automatically generated

## Create weights

I used [Detecto](https://detecto.readthedocs.io/en/latest/index.html) to create an object detection model. It’s very simple and easy to set up.   
It uses a [Faster R-CNN ResNet-50 FPN](https://pytorch.org/docs/stable/torchvision/models.html#object-detection-instance-segmentation-and-person-keypoint-detection) pre-trained pytorch model.

For more information look at the API documentation

<https://detecto.readthedocs.io/en/latest/api/index.html>

Install the requirements. (Make sure you’re in the correct directory)

pip install -r requirements.txt

To create weights, you just need to execute the generate\_weights.py file. This will take about 15 minutes depending on how strong your computer is. Without an enabled GPU this will take a lot longer.

Text, letter

Description automatically generated

**Note**

If you added new data with different elements you will need to change a line of code.

On line 27 there is a list of strings with the name of each element. Change this to whatever elements you have.

Text

Description automatically generated