howest
hogeschool

Research Project

User manual

Content

Necessities

Check out the Installation manual for the source code and how to set up the model.

- You need a laptop/pc with python installed
- Some way to draw a wireframe (drawing tablet, pen and paper, …)

Wireframe to code

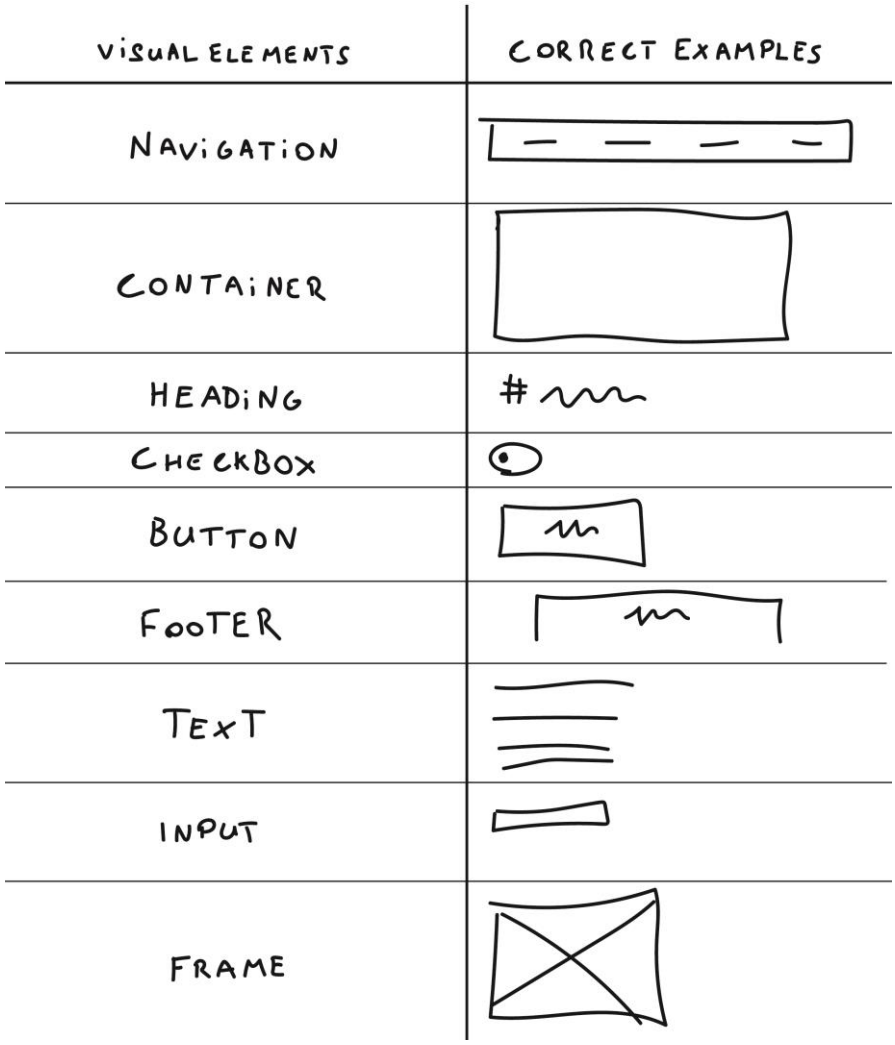
As an example, I will draw a wireframe using my drawing tablet and illustrator. I've tested out drawing on paper and it works as good, but it's quite tedious because you must scan the image to your computer.

If you need more information on the library, I used to predict these images check out Detecto.

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

Conventions

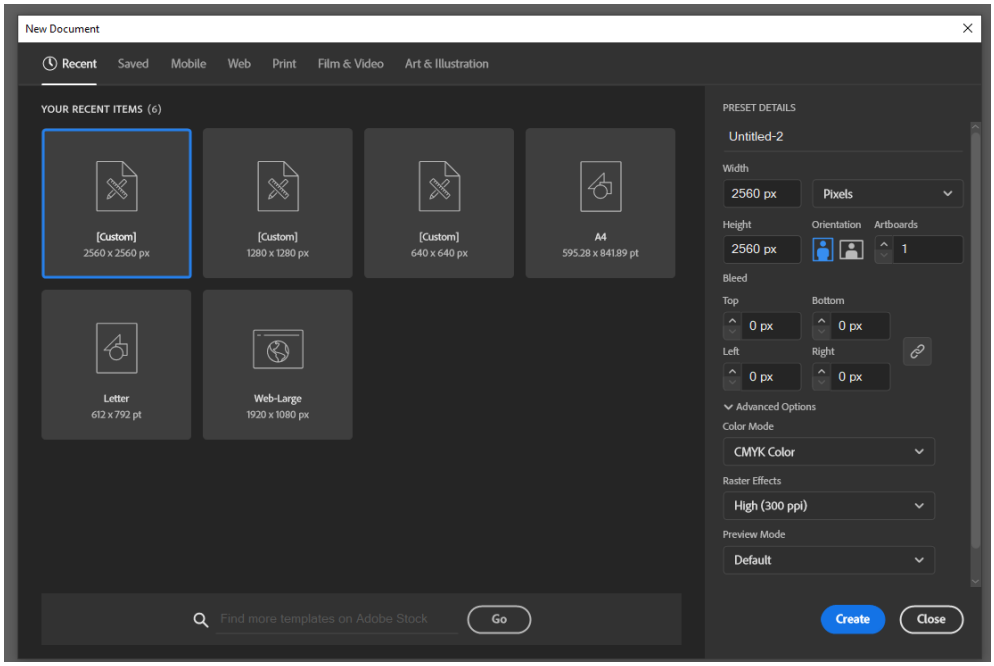
I used certain conventions when creating the dataset. This way the model has an easier time detecting elements. These are the conventions I used. When drawing your own wireframe make sure that you use similar styles.



Resolution

The images used in training were 640x640px, this is the minimum resolution to use. The bigger you go the better the model will predict the right element. When using a higher resolution make sure you also increase the stroke.

For this example, I will use 2560x2560px (640 * 4).



Drawing a wireframe

When drawing elements make sure you don't draw them too sloppy and that there is enough space between the elements else the prediction will mix them up.
Then save the image as a .png or.jpg

Wireframe to code

To convert the created image to html/css code you just need to run the python script predict.py with the correct parameters.

-i --input	Input image we want our model to convert.
-n --name	The output folder name

Example

```
C:\Users\Jakob\Documents\Repositories\Research-Project>cd v2

C:\Users\Jakob\Documents\Repositories\Research-Project\v2>python predict.py -i wireframe-1.jpg -n wireframe-code
C:\Users\Jakob\AppData\Local\Programs\Python\Python39\lib\site-packages\torch\functional.py:445: UserWarning: torch.m
eshgrid: in an upcoming release, it will be required to pass the indexing argument. (Triggered internally at  ..\aten
\src\ATen\native\TensorShape.cpp:2157.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
True

C:\Users\Jakob\Documents\Repositories\Research-Project\v2>
```

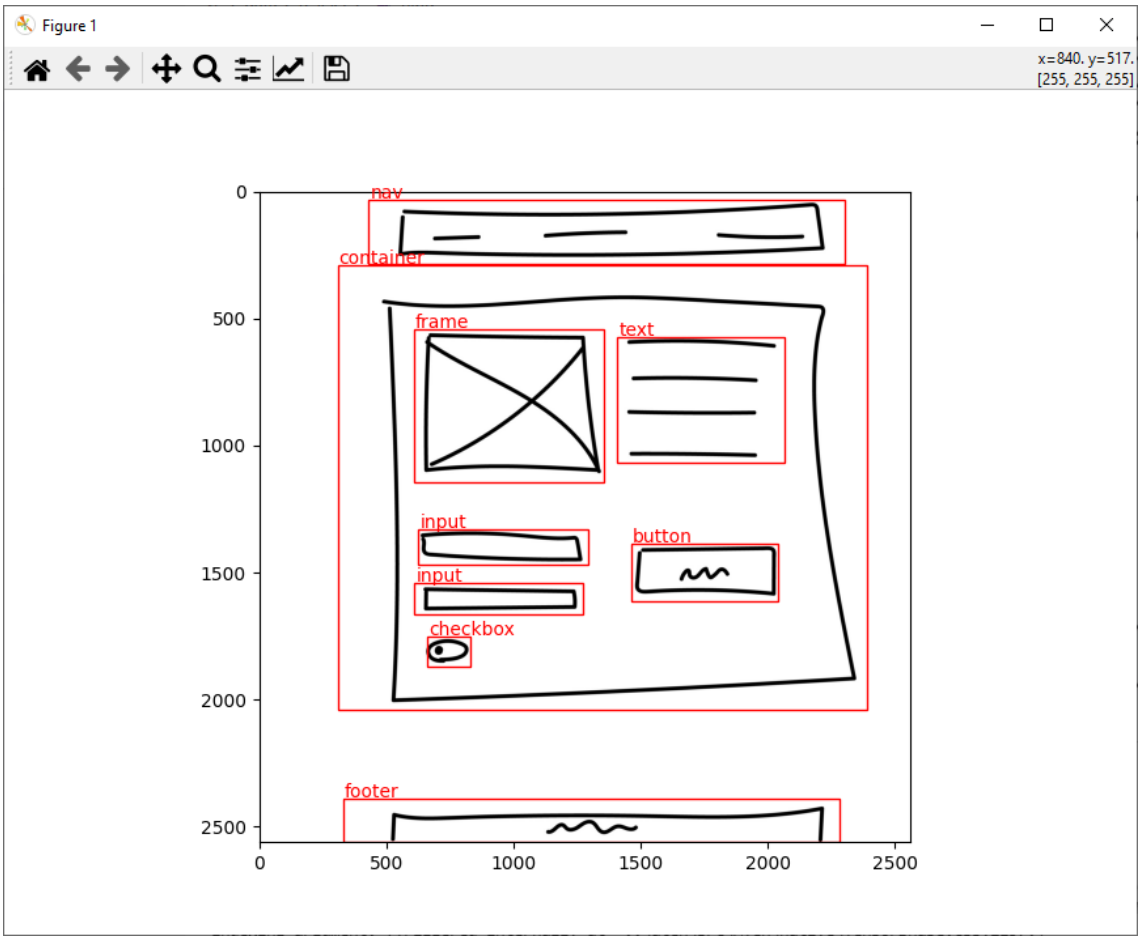
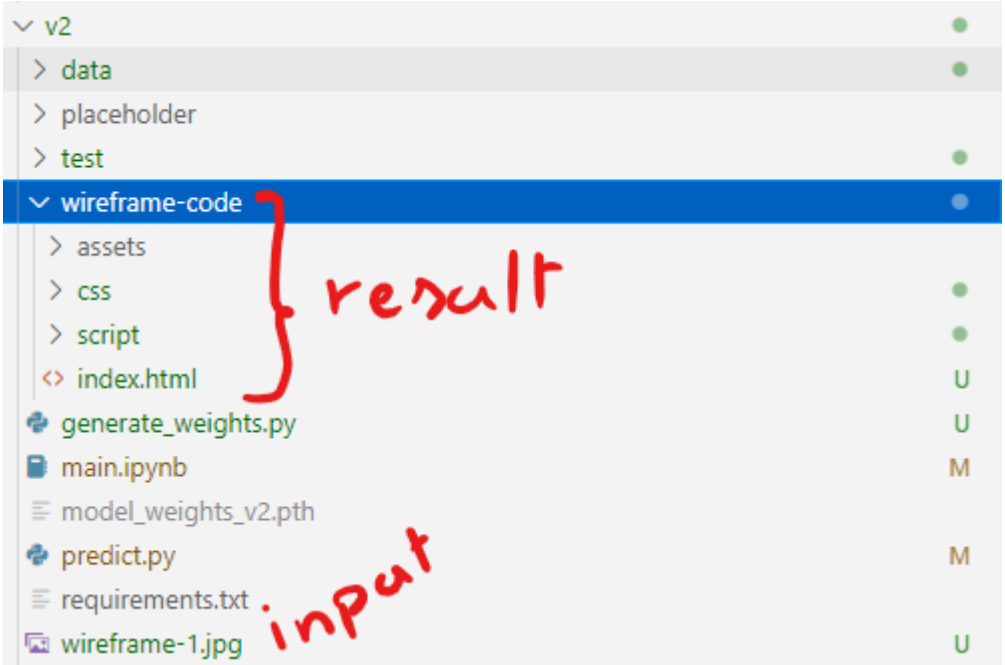
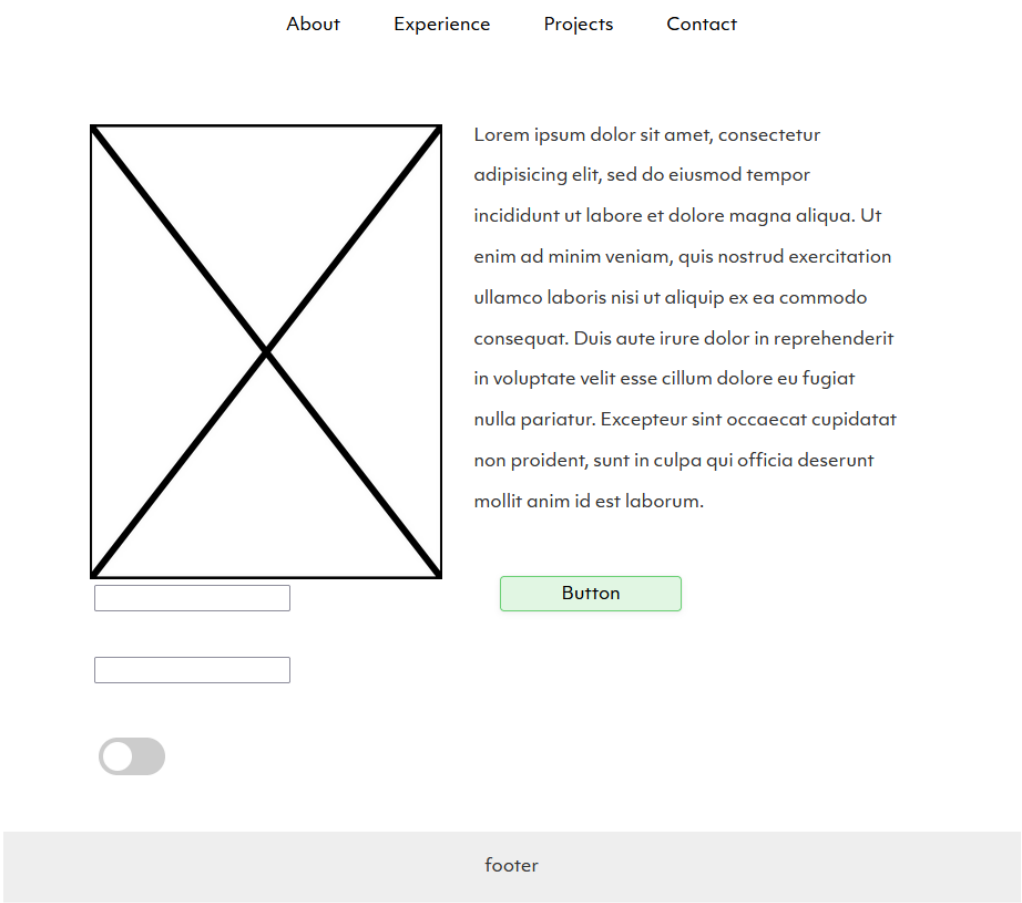


Figure 1
x=840. y=517.
[255, 255, 255]
nav
container
frame
text
input
input
button
checkbox
footer
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500


v2
data
placeholder
test
wireframe-code
assets
css
script
index.html
U
generate_weights.py
U
main.ipynb
M
model_weights_v2.pth
predict.py
M
requirements.txt
wireframe-1.jpg
U
result
input



Extra dataset (optional)

In the installation manual there was an option to add your own data with different elements. Then you created a new model with different labels these labels need to be changed in de predict python file as well.

On line 223 when we load in our weights. We add a list of labels change these to the new labels you created when adding bounding boxes to your dataset.

```
216 if __name__ == "__main__":
217     parser = argparse.ArgumentParser()
218     parser.add_argument("-i", "--input", help = "give input file to make a prediction")
219     parser.add_argument("-n", "--name", help = "give name for result folder")
220     args = parser.parse_args()
221
222     # Load in model
223     model = core.Model.load("model_weights_v2.pth", ["nav","frame","heading","text","checkbox","input","button","container","footer"])
224
225     # Get predcition for given image
226     filtered_boxes, filtered_labels = predict(model, args.input)
227
228     # Convert the prediction to
229     jsonList = prediction_to_json(filtered_boxes, filtered_labels, args.input)
230
231     image = PIL.Image.open(args.input)
232
233     width, height = image.size
234
235     json_to_html(jsonList, width, height, args.name)
236
237
```

howest
hogeschool

howest.be