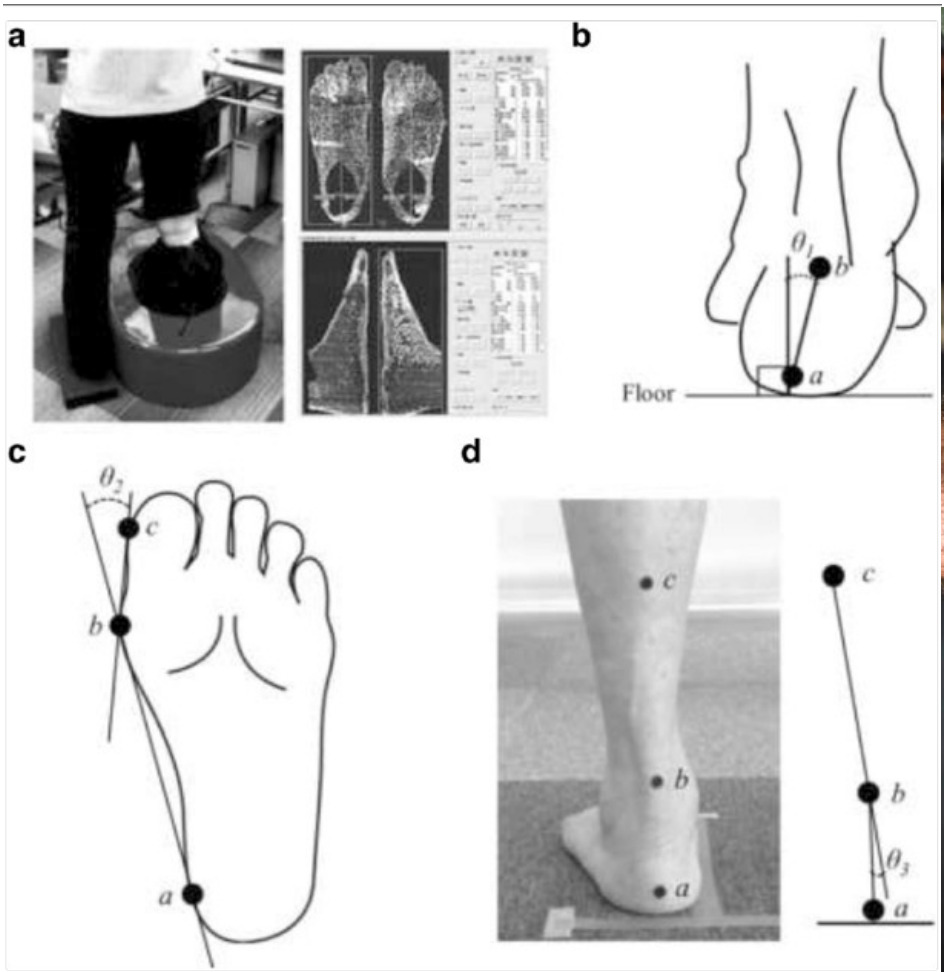
Climbing Shoe Fit

Data driven innovation challenge

Final report



Semester 7 - Advanced AI

Eniko Kakas

18-01-2024

Table of Contents

1. Introduction

Shoes for climbing have to fit their owners almost perfectly, they have to be tight enough to be able to feel the climbing holds/ rock structures well while they should not cause too much pain.

Since feet sizes and shapes are very diverse, climbing shoe manufacturers make many different shoes to fit all the distinctive needs. Navigating between all the choices can be challenging, often extensive research has to be done to find the best fit including trying them on.

It is not uncommon that even after a thorough investigation climbers end up with the wrong shoes which they only realize after climbing in them a couple of times. Therefore, having a tool that can give accurate suggestions for climbing shoes taking the foot size and shape into account could mitigate this problem.

2. Goal

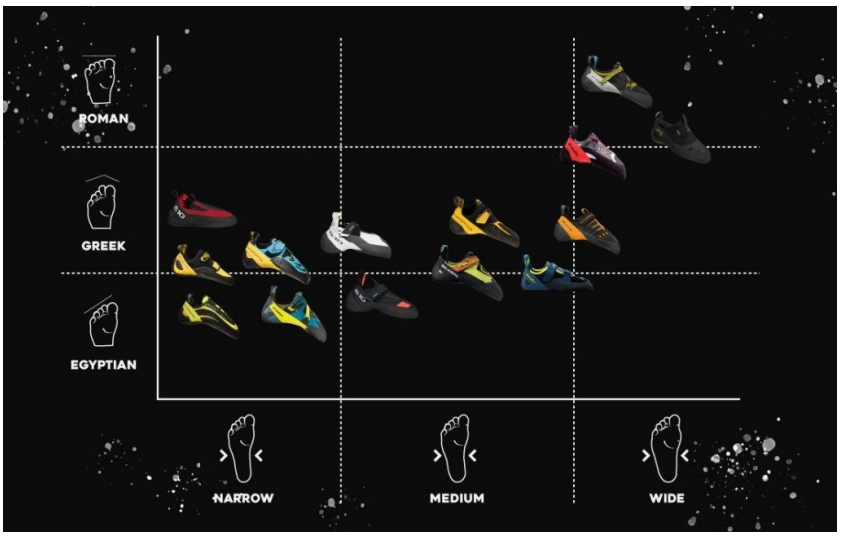
The goal of this project is to create a tool that can scan climbers feet and give accurate suggestions about which climbing shoes would fit them best.

3. Initial plan

Steps:

1. Collect data: pictures of feet
2. Try shoe measurement algorithm (https://github.com/wildoctopus/FeetAndShoeMeasurement) and fine tune it to be able to determine foot width
3. Label collected data according to 3 main foot shapes (roman, egyptian, greek)
4. Train model to be able to determine foot shape from images
5. Write algorithm that tells which shoe fits the given foot shape and width based on this chart:

   https://www.climbingshoereview.com/how-to-choose-climbing-shoes/



6. Make application that scans foot and makes suggestions for climbing shoes

3. Approach

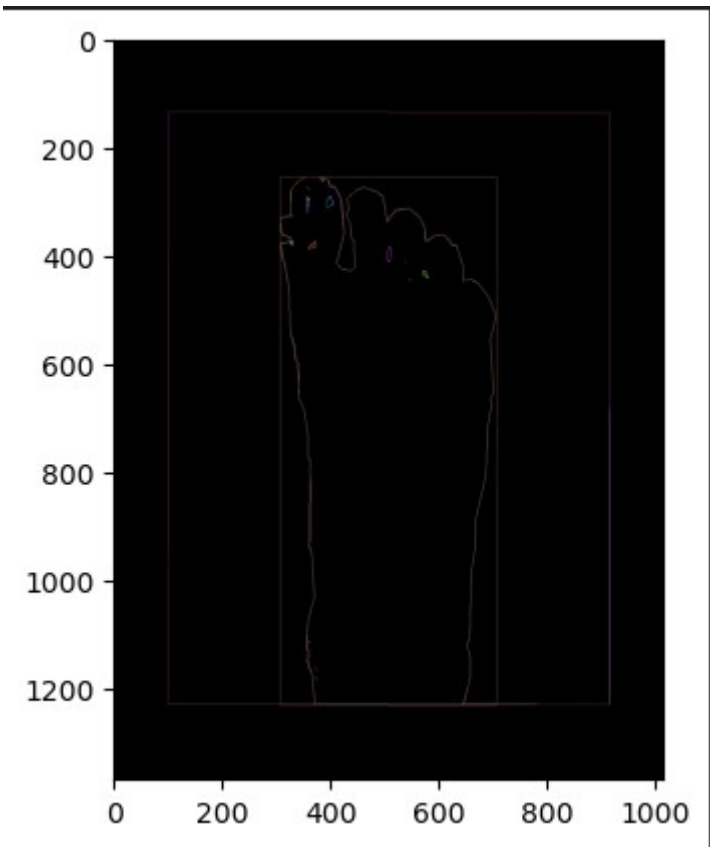
3.1. First status update

Up until now (30-11-2023), the first 4 steps from the plan were visited.

1. For step 1 pictures were collected that were taken on an A4 paper. These pictures are used to test and fine tune the shoe measurement algorithm.

   In addition, a dataset from Kaggle was found that contains ~6000 pictures of feet.

2. For step 2 the shoe measurement algorithm was tested. It seems like it works really well when it comes to determining the foot length but the calculation for foot width seems to be off. This step has to be revisited and the foot width calculation has to be fine tuned.




```
print("feet size (cm): ", calcFeet

Feet height:  1113
Feet Width:  500
Paper height:  1370
Paper Width:  1016
feet size (cm):  10.334645669291337

print("feet size (cm): ", calcFeet

Feet height:  1113
Feet Width:  500
Paper height:  1370
Paper Width:  1016
feet size (cm):  24.128540145985404
```

(For further information check FeetAndShoeMeasurement.html)

3. For step 3 the Kaggle dataset was labeled by hand (pictures were sorted into the 3 main foot shape categories). Since many pictures are a bit ambiguous, the labeling might have to be revisited in the future, it is possible that mistakes were made.

```
df
[4]  ✓ 0.0s
```

	Image Path	Label
158	./foot/Photos/OG/egyptian/84_1_41_1.JPG	egyptian
109	./foot/Photos/OG/egyptian/49_1_50_1.JPG	egyptian
131	./foot/Photos/OG/egyptian/15_1_37_48.JPG	egyptian
55	./foot/Photos/OG/roman/50_1_55_1.JPG	roman
94	./foot/Photos/OG/roman/63_1_28_33.JPG	roman
...	...	...
71	./foot/Photos/OG/roman/23_1_28_1.JPG	roman
106	./foot/Photos/OG/roman/45_2_28_1.JPG	roman
14	./foot/Photos/OG/greek/2_1_32_1.JPG	greek
92	./foot/Photos/OG/roman/10_1_38_42.JPG	roman
102	./foot/Photos/OG/roman/90_1_24_1.JPG	roman

162 rows × 2 columns

4. For step 4 the Kaggle dataset was used. First, the pictures were preprocessed with an edge detection algorithm and an ANN model was trained. Since this model did not perform well a CNN model was trained on the original images. Since the CNN model did not show better results the data labeling was revisited.

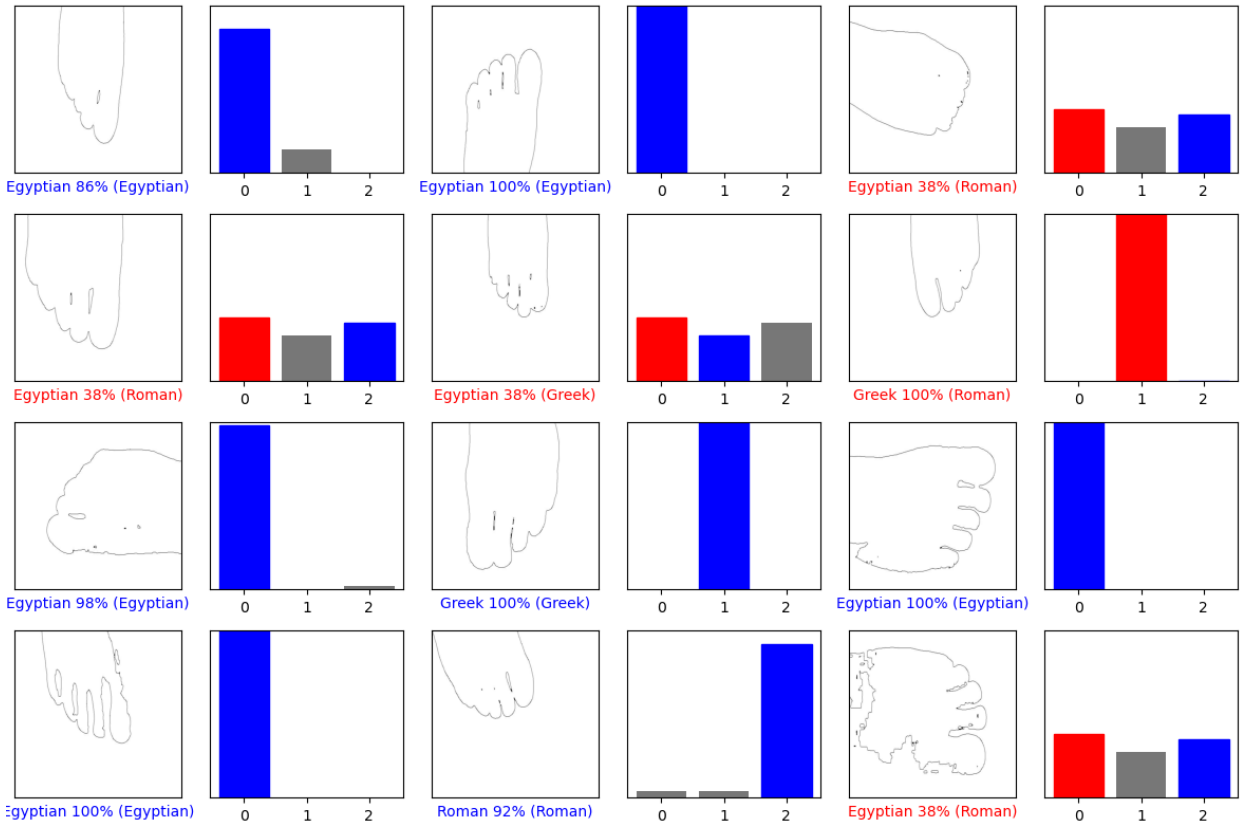
   After re-labeling the data, it was decided to enrich it. The original dataset contained ~6000 pictures but those were taken from 100 individuals so in the first model training only ~200 pictures were used (1 picture of left and and 1 picture of right foot from the 100 people), in the second round the dataset got enriched from the original dataset. This time ~2500 pictures were used (not more to save on processing power).

```
df
✓ 0.0s
```

	Image Path	Label
809	/home/kakas/Desktop/output/roman/22_1_26_42.JPG	roman
2356	/home/kakas/Desktop/output/egyptian/53_2_33_14...	egyptian
761	/home/kakas/Desktop/output/roman/13_1_29_38.JPG	roman
318	/home/kakas/Desktop/output/greek/46_1_39_5.JPG	greek
961	/home/kakas/Desktop/output/roman/12_2_27_26.JPG	roman
...	...	...
1638	/home/kakas/Desktop/output/egyptian/47_1_26_13...	egyptian
1095	/home/kakas/Desktop/output/roman/6_1_29_11.JPG	roman
1130	/home/kakas/Desktop/output/roman/90_1_24_84.JPG	roman
1294	/home/kakas/Desktop/output/egyptian/69_1_23_23...	egyptian
860	/home/kakas/Desktop/output/roman/90_1_24_55.JPG	roman

2580 rows × 2 columns

   Using the enriched dataset with ANN resulted in a model that is 70% accurate, which is a big improvement compared to previous attempts.



Since one of the categories (egyptian) is overrepresented in the dataset, in the next round attempt will be made to balance this out either with data augmentation or with removing images from that category.

(For further information check footshapeANN.html and footshapeCNN.html)

3.2. Second status update

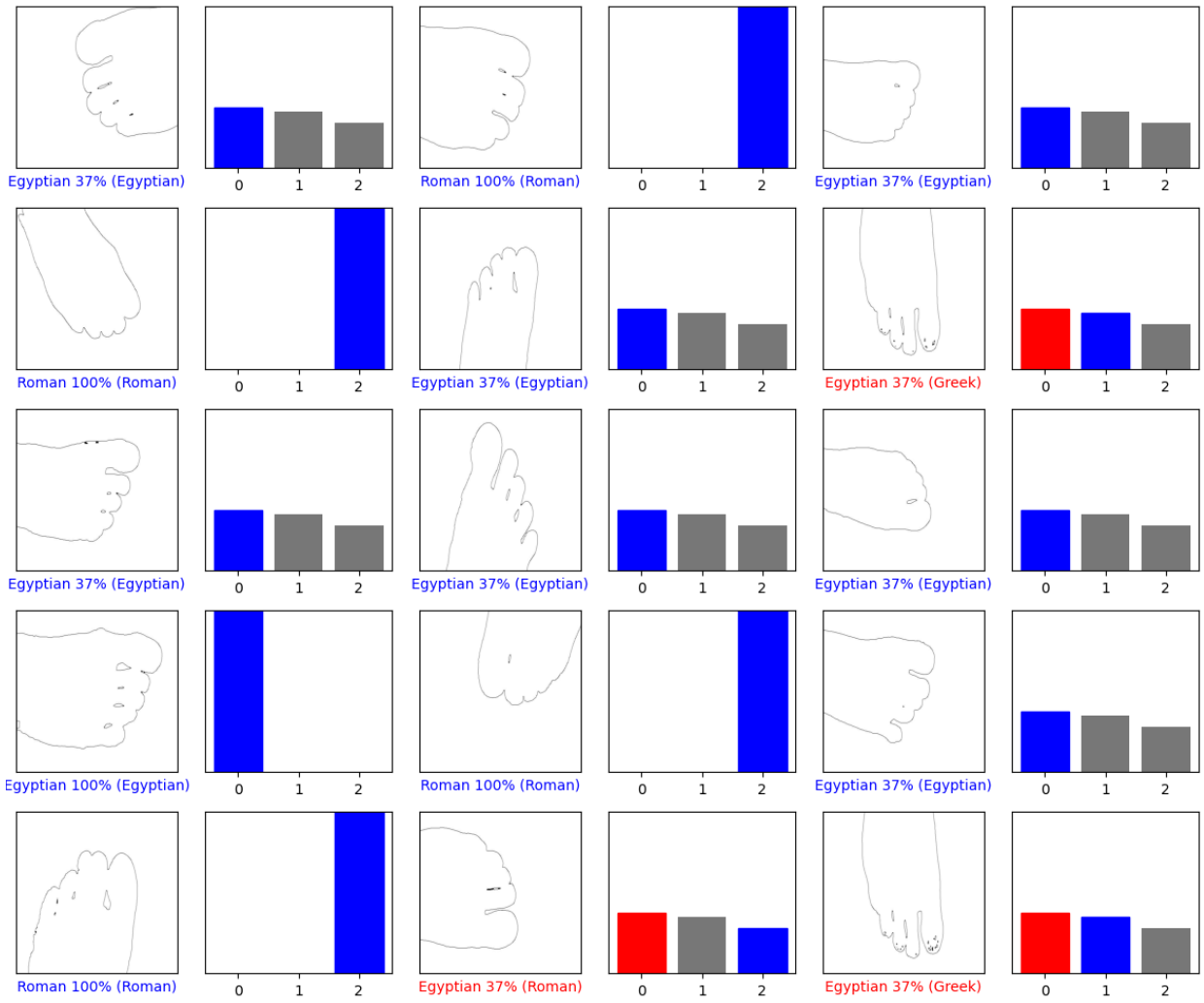
At this point of the project (22-12-2023) the following progression was made since the previous status update (30-11-2023):

1. For step one no additional action was taken.
2. For step two the shoe measurement algorithm was replaced first with the DETR model which was capable of finding the bounding box around a foot accurately, then with a yolo model (https://huggingface.co/hustvl/yolos-tiny) that could do the same but a bit faster.

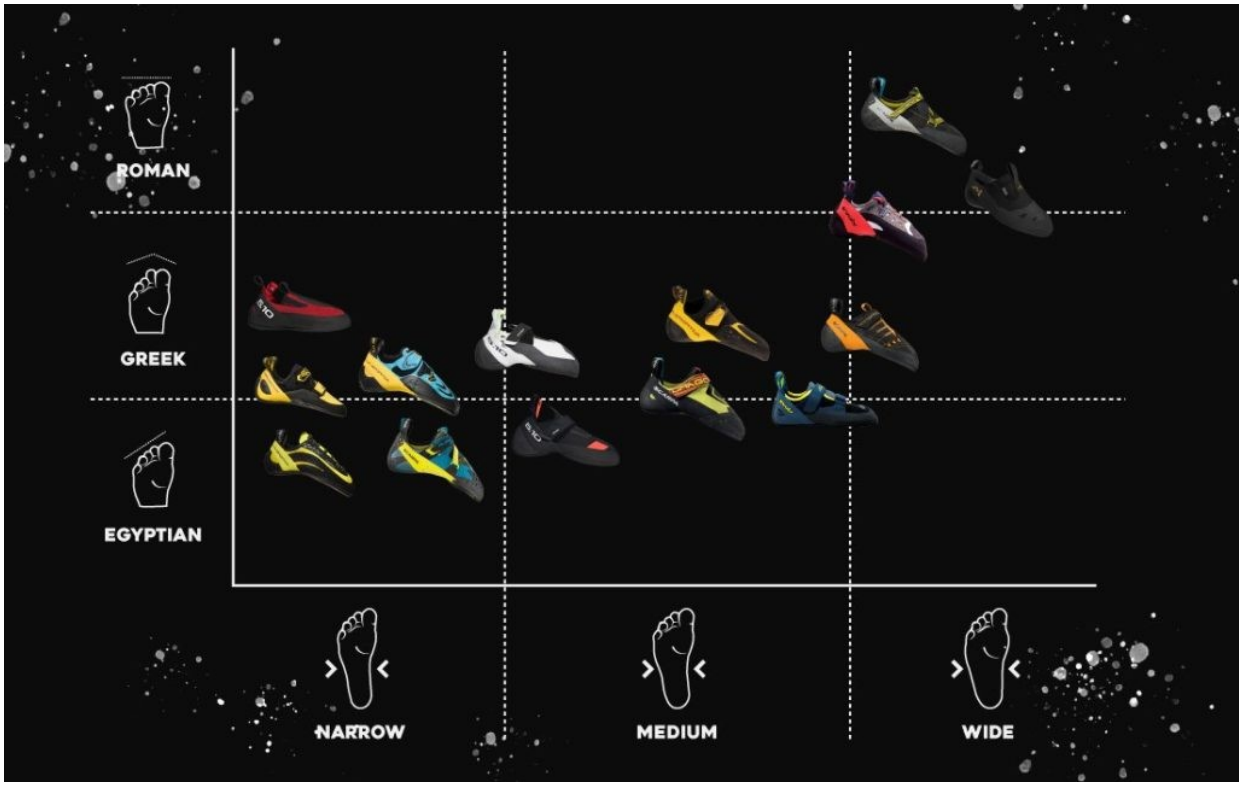
   This was done because it turned out that calculating the foot width category can be done by dividing the foot width with the foot length, so in this case dividing the bounding box width with the bounding box length. Therefore knowing the exact foot measurements in cm is not necessary, the previous approach could be replaced.
3. For step three, the Kaggle dataset for re-labeled once again, to remove previous mistakes. In addition, a step was added to the data pre-processing to balance the dataset along the 3 categories. Eventually 481 images were used per category.

4. For step four, the ANN model was re-trained and tuned a couple of times with the newly labeled and balanced dataset. The best performing model had a 75% accuracy which was decided to be used for the application.

   The diagram below shows how the model performs and how it predicts along the three categories:



5. For step five, the climbing shoes from these diagrams were identified and a dataset was created with their qualities:



https://www.climbingshoereview.com/how-to-choose-climbing-shoes/

The softness and shape values from the second diagram were used to determine whether a shoe is suitable for indoor (soft)/ outdoor (hard), for beginner (neutral)/ advanced (aggressive) and for slab climbing (neutral)/ overhang climbing (aggressive).

The created dataset looks like this:

Model	roman	greek	egyptian	narrow	medium	wide	beginner	intermediate	advanced	slabs	overhang	indoor	outdoor
Five Ten Moccasym	0	1	0	1	0	0	1	1	0	1	0	1	1
Five Ten Hiangle	0	1	0	1	1	0	0	1	1	1	1	0	1
Five Ten Crawe	0	0	1	0	1	0	0	1	1	1	1	0	1
Scarpa Drago	0	1	1	0	1	0	0	0	1	0	1	1	0
Scarpa Instinct VS Men's	0	1	0	0	1	1	0	1	1	1	1	0	1
Scarpa Furia Air	0	0	1	1	0	0	0	1	1	1	1	1	0
Scarpa Veloce Men's	1	0	0	0	0	1	0	1	1	1	1	1	0
La Sportiva Katana	0	1	1	1	0	0	1	1	0	1	0	0	1
La Sportiva Miura	0	0	1	1	0	0	0	0	1	1	1	0	1
La Sportiva Futura	0	1	1	1	0	0	0	1	1	0	1	1	0
La Sportiva Solution Comp	0	1	0	0	1	0	0	1	1	1	1	1	1
Evolv Oracle/ Shaman	1	1	0	0	1	1	0	0	1	0	1	0	1
Evolv Defy	0	1	1	0	1	1	1	1	0	1	0	1	1
Mad Rock Remora HV Tokyo edition	1	1	0	0	0	1	1	1	0	1	1	1	0

Based on this dataset a filtering algorithm was created that returns all suitable shoes for the detected foot shape and foot width along with the shoes additional qualities.

6. For step six, a prototype application was created with Streamlit that brings the whole process together.

First, the user can take an image of their foot. Then the yolo model draws a bounding box around the foot. After, the foot width category (narrow, medium or wide) is determined by dividing the box width with the box height. Then, the image is being pre-processed to get the foot edges. This is done by kMeans clustering and openCV. After, the edged image is given to the ANN model to determine the foot shape.Once both the foot width and the foot shape are known, the filtering algorithm returns the suitable climbing shoes from the shoes dataset.

The application has a version where the bounding box detection is done real time, but this process seem to be too computationally heavy, although it would be suggested to use real-time detection to overcome cases when the foot is not recognized from the taken image.

3.3. Third status update, final results

At the final stage of the project, only step two and six were revisited.

For step two, the tiny-yolo model was replaced with yolo-v8. It was also attempted to fine-tune this model to specialize in foot detection but eventually it was found that the fine-tuned version did not perform better than the original one. Nevertheless, using yolo-v8 seems to be the correct approach it works faster than any of the previous models and it can draw the bounding boxes around feet accurately and precisely.

For step six, the application got finalized, the model got changed in it according to step two and it got deployed on the Streamlit server.

4. Limitations and Recommendations

Even though the project reached its final goal and a tool got created that can scan climbers feet and give accurate suggestions about suitable climbing shoes, the application and the models have their limitations.

Therefore the followings are suggested to be taken into account when using the tool and/ or when further developing it:

- The model that draws the bounding box around the foot can detect other objects as well, therefore it would be a good idea to first check wether the detected object is a foot or not before moving on to further processes.
- Both the object detection and the edge detection algorithms are sensitive to shadows, therefore this limitation should be taken into account. To combat it, some ideas could be like using the camera flash automatically, or using real-time object detection in the camera.
- Calculating the foot width category was based on available charts, but no universal chart was found, therefore it is suggested to calibrate these numbers if sufficient amount of data becomes available.
- The dataset with the climbing shoes is limited to the ones that were mentioned on the chart (15 shoes in total). This data is sufficient for a prototype but for accurate suggestions it would be better to extend it with more shoes from more manufacturers.