Advanced Surgical Planning – Implant Recognition

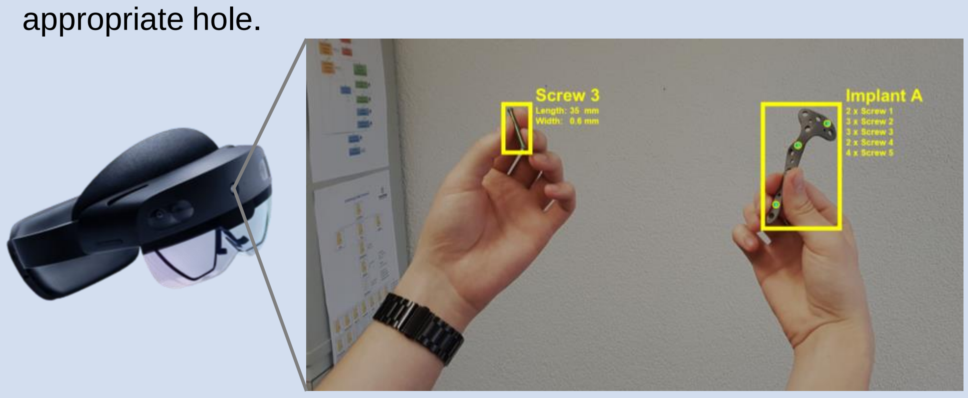
Thorbjörn Höllwarth, Lara Nonino, Dario Dittli

Supervised by: Thomas Zumbrunn, Martin Ralf Oswald

1 Introduction & Goal

Our project aims to achieve **implant recognition** and **length measurement** for medical purposes.

Specifically, during surgical procedures for bone repair, surgeons need to attach an implant to the bone using screws. Our goal is to assist the surgeon by **identifying** which screw corresponds to a specific implant hole. To accomplish this, we need to efficiently detect the screws, measure their length, and match them to the appropriate hole.



2 On / Off-device

	PROs	CONs
On device	• **Reliability** • **Simplicity for users** • Privacy • No latency due to data transmission	• Power consumption • Storage limitations • Complexity • Inference time (Yolov7-tiny ~ 5s)
Off device	• Power consumption • Storage expansion • Ease of development • **Inference time** (Yolov5 ~ 0.15s)	• Unreliability • Privacy gaps • Increased latency

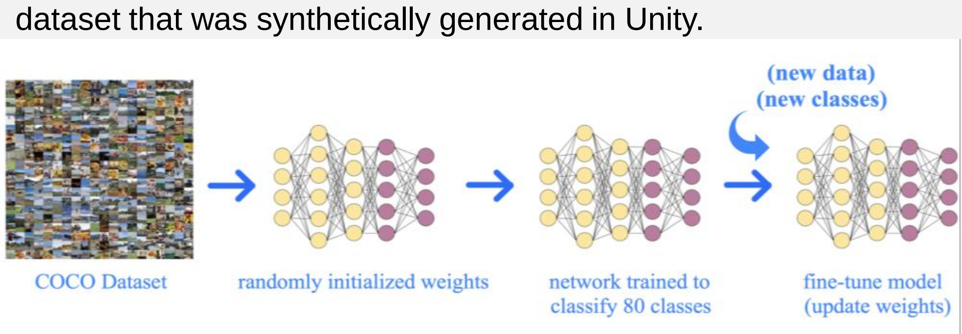
3 Yolo implementation & Fine-tuning

To identify the desired objects, the **YOLOv5** network was employed.

Training the network from scratch would have required a significant amount of time and computational resources. Hence, a **transfer learning** approach was adopted.

The network's backbone, along with its pretrained weights, were obtained from [1] in PyTorch format. The YOLOv5 model was initially pretrained on the COCO dataset, a vast collection of images encompassing objects from 80 distinct categories.

Subsequently, **fine-tuning** was conducted using a custom dataset that was synthetically generated in Unity.



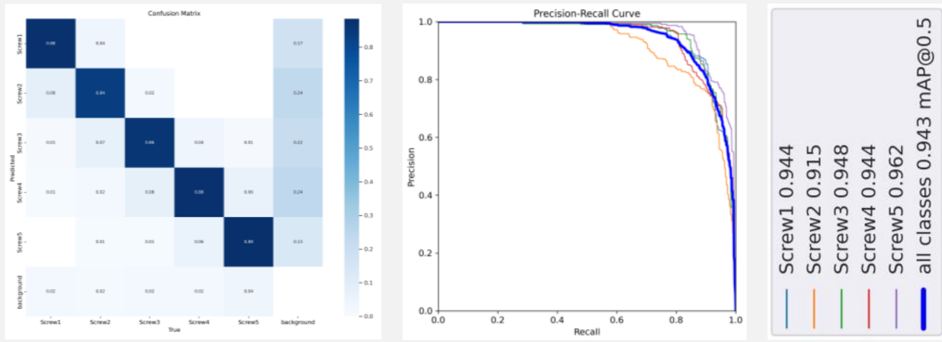
4 Dataset generation

The process of generating the dataset was accomplished using the **Unity** platform. The **background** of the dataset comprises multiple images manually collected from the internet, while the **foreground** consists of one or more screw objects positioned at various orientations and scales. To enhance the detection's resilience, the **light** source was randomized across the dataset samples.

Information pertaining to the **detected classes** and their corresponding **2D bounding boxes** is recorded in a text file. However, as YOLO necessitates a distinct data format for this information, a Python script was developed to parse the file and generate a compatible format that maintains data coherence.


```
Label, x, y, width, height
2 0.29 0.69 0.08 0.15
3 0.84 0.28 0.27 0.48
4 0.50 0.19 0.12 0.22
```

5 Results



- High true positive rates → Low screw misclassification
- High Precision-Recall curve → Reliable and robust model

6 Conclusion & Future work

After implementing both versions of the application, it was found that the off-device solution aligns better with our requirements.

By training the pretrained yolov5 network with our specific objects, we successfully developed a reliable and robust model with minimal misclassification of screws.

Future work will revolve around augmenting the training process by incorporating real images in addition to the synthetic data. Furthermore, leveraging the depth data from the Hololens holds significant importance as it will facilitate the alignment of 2D bounding boxes with the depth map, enable the filtration of irrelevant 3D data points, and ultimately enable accurate measurement of screw lengths.

References

1. https://github.com/ultralytics/yolov5
2. https://www.synergiz.com/en/produit/hololens-2-industrial-edition-2/
3. https://learn.microsoft.com/en-us/uwp/api/windows.ai.machinelearning?view=winrt-22621
4. https://pjreddie.com/darknet/yolo/
5. S. Borkman, A. Crespi, and N. Yadav, “Unity perception: Generate synthetic data for Computer Vision,” arXiv.org, https://arxiv.org/abs/2107.04259 (accessed May 16, 2023).