Digital ICU : Robust Realtime Object Detection for Scene Understanding in Intensive Care Unit

General Info

Contact Person: Ee Heng Chen
Contact Email: eeheng.chen@tum.de

Project Abstract

Scene graph is a graph that models the relationship (edges) between objects (nodes) in a scene [1,2,3,4]. When incorporated into a system, the scene graph enables the system to understand what is happening in a scene. Such a situation-aware system is beneficial for settings where context is important, such as monitoring in the Intensive Care Unit (ICU). With the help of the scene graph, the workflow of clinicians in the ICU can be analyzed to understand what patient data is being examined by clinicians when monitoring patients. This workflow analysis can help to identify if clinicians missed any crucial patient data. In addition, a scene graph can also be used as a pseudo care planner that notifies clinicians what care has been provided and what care should be provided next. Furthermore, a scene graph could be used to model the dynamic context in the ICU and provide high-level information to reduce false-positive alerts in patient monitoring systems.

In order to build scene graphs, the objects in the scene need to be detected and identified first. This project aims to develop a robust object detector to achieve this. The detector should be able to (i.) handle the high degree of occlusion in the ICU, (ii.) achieve a high degree of detection accuracy, (iii.) run in real-time for real-world deployment, and (iv.) easily expandable to include new unseen objects.

Task Description

- Literature review on SOTA works on object detection [1] and real-time capable detectors [2].
- Develop/evaluate an object detection model/-s that gives the best trade-off between robustness (in terms of occlusion), accuracy, speed, and expandability (i.e. ease of adding new objects).
- Develop an evaluation metric for the model evaluation.
- Write and present about the work that is done.

Technical Prerequisites

- Intermediate or advanced programming experience with Python3. (C++ is a plus)
- Optional: Experience using the following libraries: OpenCV, Scikit-learn, Pytorch, Tensorflow.

References

[1] HE, Kaiming, et al. Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. 2017. S. 2961-2969.
[2] LIU, Wei, et al. Ssd: Single shot multibox detector. In: *European conference on computer vision*. Springer, Cham, 2016. S. 21-37.