

CAI Classification Problem

Report 1: Planning Phase

Nil Crespo-Peiró, Paco Rahn, Amin Ranem

12. January 2021

1 Introduction

This report covers the planning phase of the CAI Classification Problem. In the following section, an outline regarding the implementation of this project will be presented shortly. This planning includes the basic idea of how this project will be structured by the developers, given a Gantt Chart showing the defined deadlines for Prototypes and submissions. Lastly, the defined Network architecture provided at our initialised GitHub repository will be briefly presented.

2 Planning Phase

In this section, the planning of the project will be presented, whereas the basic idea will be sketched in a first step, underpinned by a Gantt Chart representing the predefined deadlines for prototypes and further reports. Additionally, the implemented Network Architecture from our GitHub repository so far will be described, followed by a representation of the repository directory structure in a final step.

2.1 Sketch of the basic idea

The main goal of our project is to implement a Deep Learning Method for tool detection in surgical operations. As in [1] we will use a CNN and the **Cholec80** Dataset. Due to the lack of more data we want to use transfer learning and if the time allows it try out different network architectures.

Furthermore, we will try to tackle the problem of small dataset by experimenting with data augmentation and dataset balancing. As shown in [1] the results from the tool detection task can be used as tool usage signals to perform phase recognition. With just two additional layers, a support vector machine and a hierarchical hidden markov network, Twinanda et al. were able to do tool detection and phase classification with the same network simultaneously. Depending on the results for the tool detection task we consider to implement a phase classification as well.

2.1.1 Gantt Chart

Figure 1 shows a Gantt chart with our plan for the next weeks. Beside the weekly tasks from the lecture we added some subtasks that we wish to accomplish. The list is not complete and some subtasks could be removed again.



Figure 1: Schedule for the Project

2.2 Network Architecture

In this section we will be creating a tree graph in order to represent the network architecture of our GitHub Repository.

In the first layer there are some documents of interest like the `README.md`, the `requirements.txt`, the `setup.py`, or the `code_style_conventions.py`, which ensure that everyone follows the same style while coding, states the requirements we will need to download in order to work in this project on how to set it all up before starting.

Then there are 3 major folders:

- **docs:** A folder, where one can find the presentation, the video tutorial and the reports that have been created during the project.
- **test:** The folder in which the python tests will be saved in order to exclude as many bugs in our code as possible.
- **cai:** The main folder in which the structure of the data and the model itself will be defined, necessary functions for visualizations of the results will be saved and other important aspects like the GUI and the agent that trains the model.

2.2.1 Directory tree structure of GitHub repository

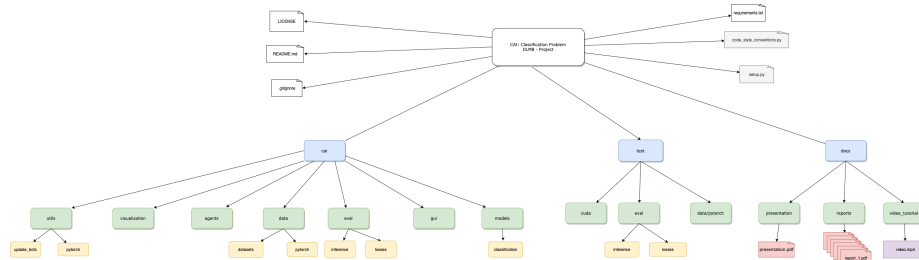


Figure 2: Diagram of our GitHub Repository

References

- [1] Andru Putra Twinanda et al. “EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos”. In: *CoRR* abs/1602.03012 (2016). arXiv: 1602.03012. URL: <http://arxiv.org/abs/1602.03012>.