Preliminary
I hope that I will actually manage to have some runable version of this project and not do the adhd thing and abandon it halfway through. So keep in mind that this will be an evolving document and more tailored to keep me on track

Goal
For some reason i have the obsession on having an automatically controlled model railway with accurate dispatching. For this you need an intensive sensor arangement, which can be quite combersome to install many block detectors. Additionally in a station application it is infeasable/expensive to have that many blocks. Point detection is another problem in that it is difficult to track direction and actual train identification via NFC is unreliable[verify]. This doesnt even mention the need to hide these detectors.

My intent is to have A model which can be deployed to a small device and will be capable of the following things:
- detection of wagons and theire position along a track
- detection of consits of wagons(train) as an entity
- identification of individual wagons for shunting operations

This is a rough roadmap of feature completeness. To achieve this substations changes, such as multiple cameras, may be neded.

Pros and Cons of existing solutions
Here there wil be a short summary of all positioning systems for model rail

Design considerations
In the current state of the project in will be happy on reliably positioning a single wagon. Much more important are the design contrains. I want to limit the computation power to a primitive SBC(PI Zero) or embedded controler(ESP CAM). I currently only have a ESP CAM so this will be the intented limit. The processing by the PC is not yet finalized. Therefore a simple text stream of `ID:[ID], pos:float`. This processor limit will contrain the size of the model and require special attention to dowsizing it.

The choice is also to limit the application to a single static know camera. A single camera simplyfies the data processing as no observations need to be aligned. While multiple cameras allow for 3D vision, trigonometry should be able to substitute it. Being a static know camera whose position and focal length is known positions in 2D space should be easily translated to a position on the base plate. To make this easier this project will limit itselt to train track of the same height. An alternative is also the existince of homing points of know distance and size. This is to be investigated.

Existing tech
This section will be the literature review

Computer vision tasks can be subdivided into object detection, what object is in an image, or object segmentation, color in the surface of that object. Object detection in its most primitive form is unable to locate the subregion where the object is supposed to be. More modern variants such as yolo are able to provide bounding boxes. Yet these bounding boxes suffer as they do not conform to the actual object. As getting the trains orientation a simple bounding box is not sufficient. Image segmentation instead tries to either seperate all instances of a class from other classes, or in the panoptic variant to also seperate individual class nstances from another. But here again the problem of getting the orientation from the 2D image is difficult. In an ideal case with a perfect mask one could try to fit a diagonal into the resulting form. This is to be investigated on if OpenCV offers such out of the box tools. current assumption is no.

Keypoint detection offers the benefit of an end2end detection of the relevant components of the train. The front and back in 2D space.

Training: Training is a special consideration due to two specific aspects. The camera is from a ca 30° elevation. This is a diversion from most training datasets such as COCO who have front focused images. This

may cause inaccuracies lateron. The second is that there exists no data which i could find of (model) trains with their keypoints labeled. Special attention will therefore be placed on models which can learn these features in a self-supervised maner.

Models

I plan on scouring the web in this sequence to find a collection of suitable base architectures and backbones. A sufficnetly trained backbone should boost detection accuracy by having learned from a highly varied dataset and will keep my training efforts low.

1. Shitty AI blogs
   - the offer a quick list of popular open source projects and the general concensus
2. Papers with Code
   - Benchmarks of the most popular models
3. actual research in the literature
   - oh god, help me :(