section, additional graphs, where the performance is analysed per quadrant were created in Figures 6.12-6.16.

In Figure 6.12 the overall evaluation dice is shown and in Figure 6.13 are results with the postprocessing applied. Looking at the dice results it is clear that the postprocessing neealy always helps increase the metric's value, but it is rather on the second decimal position. And the effect is largest on the third quadrant.

If we were to just focus on the nicely segmented tunnels i.e. those that were matched we can see that the overall dice score jumps to higher values, reaching even around 0.68 in the second quadrant. Hence, it looks like the overall Dice score is being lowered by false predictions, which even the postprocessing is not able to remove.

Lastly, we will talk about the Dice score on the level of tunnel detection as was described in Section 6.1. Compared to the overall Tunnel scores in Figure 6.16, where it seemed according to the average values, that the 2D version have slight advantage, here it is not so clear, there also seems not be any clear clearly better architecture than the rest.

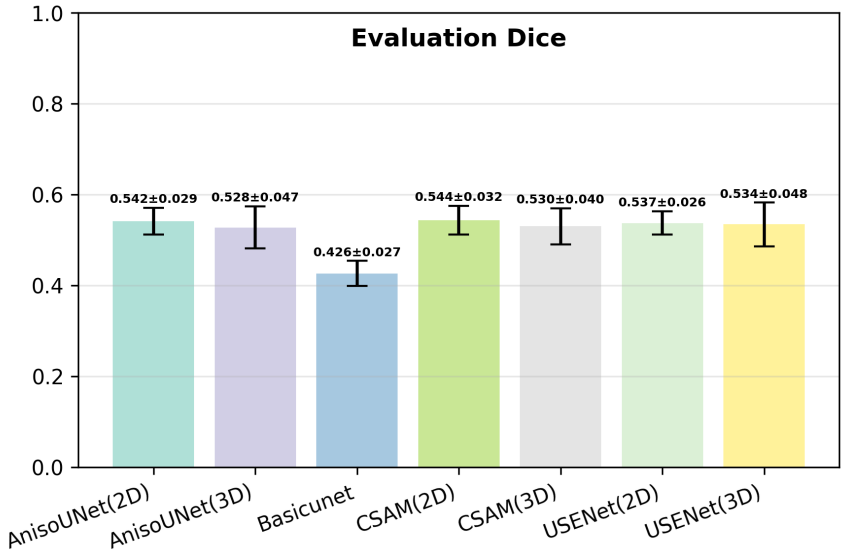

Figure 6.7: dice

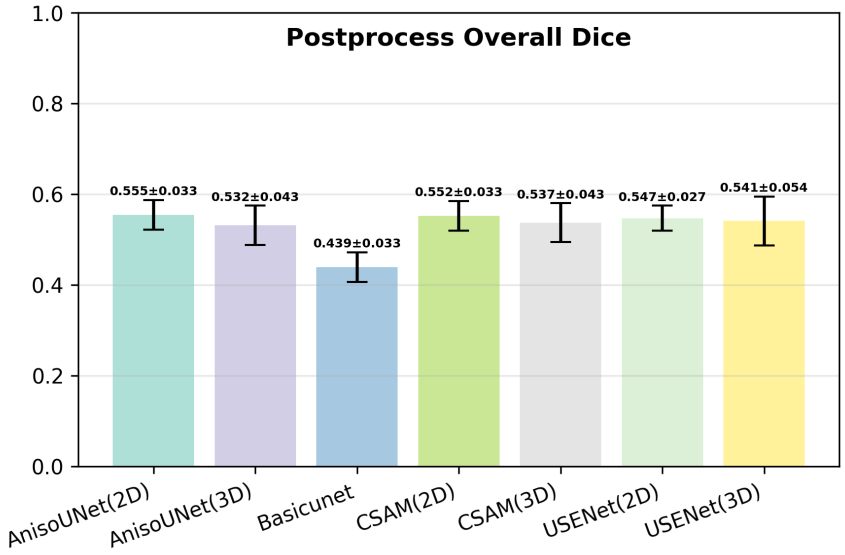

Figure 6.8: overall dice

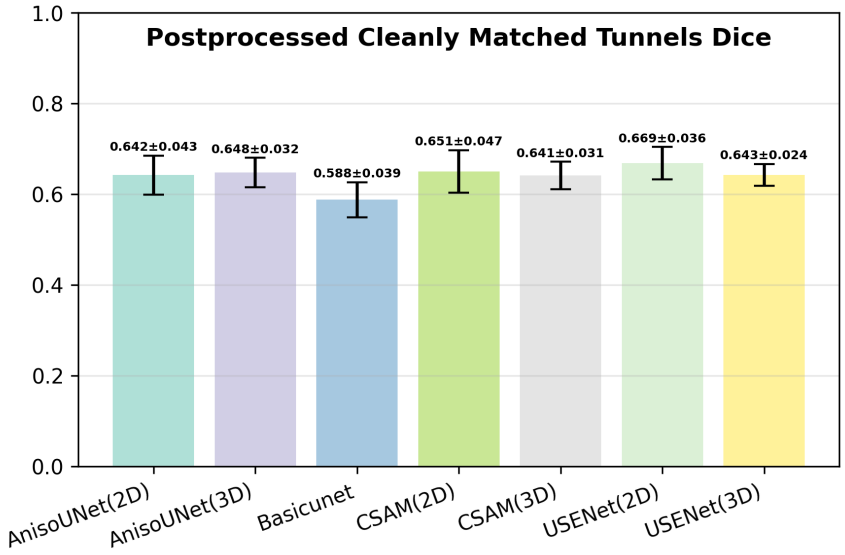

Figure 6.9: dice on cleaned matches

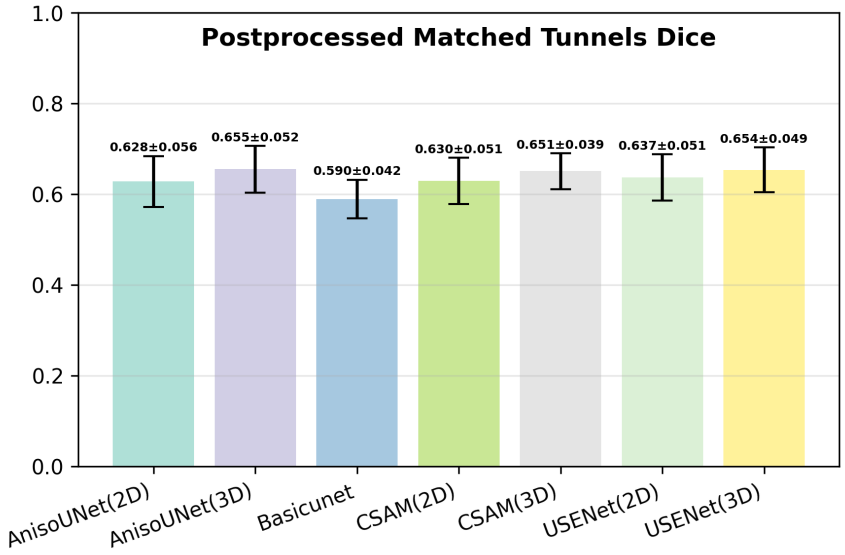

Figure 6.10: dice on all matches

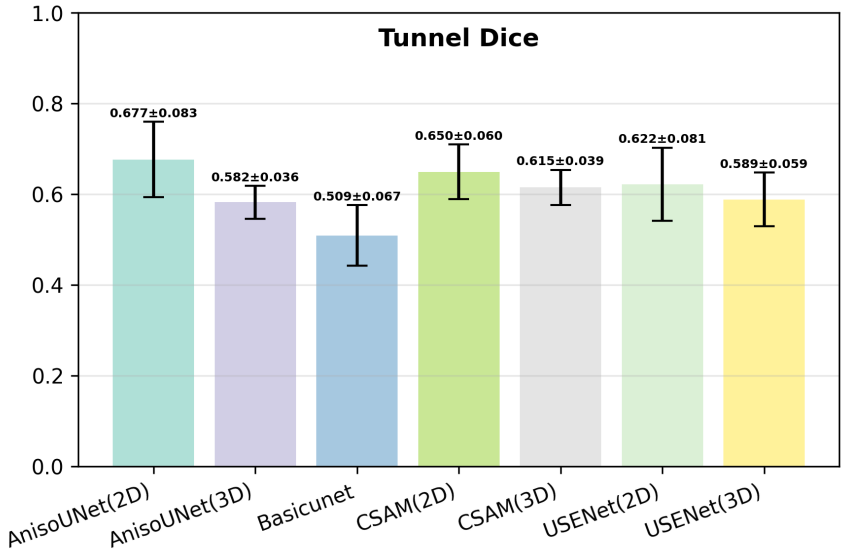

Figure 6.11: dice on tunnel level

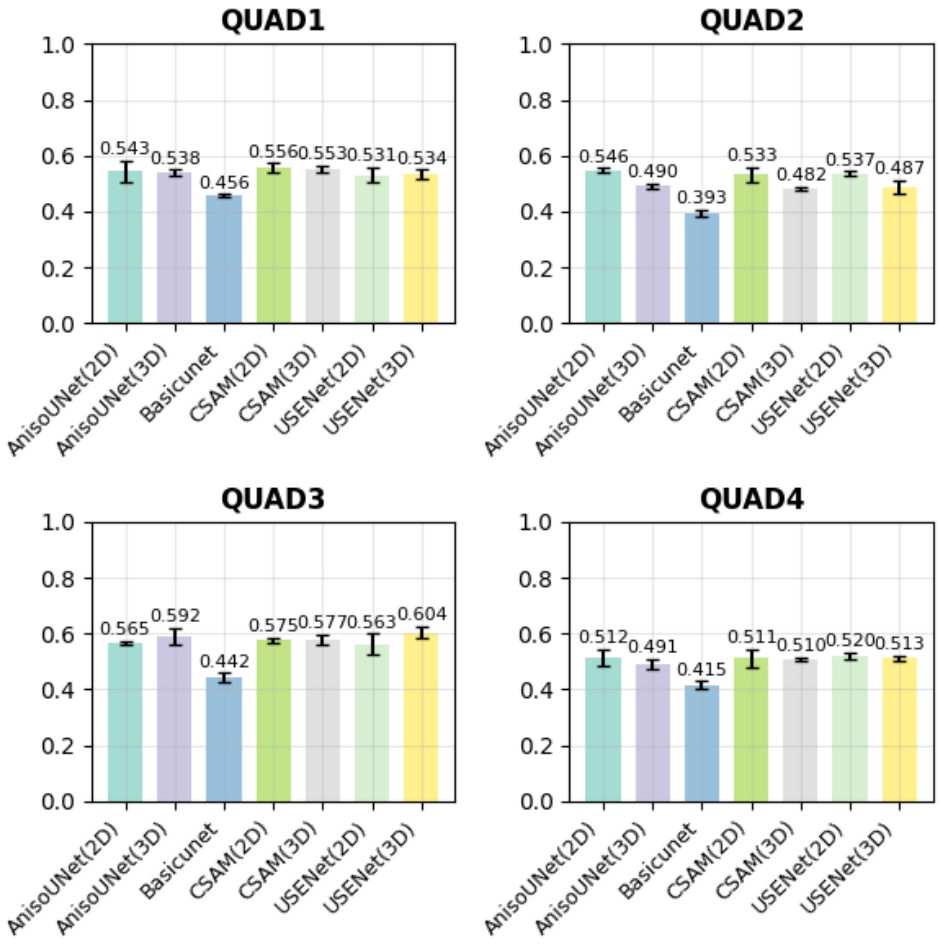

Figure 6.12: eval dice

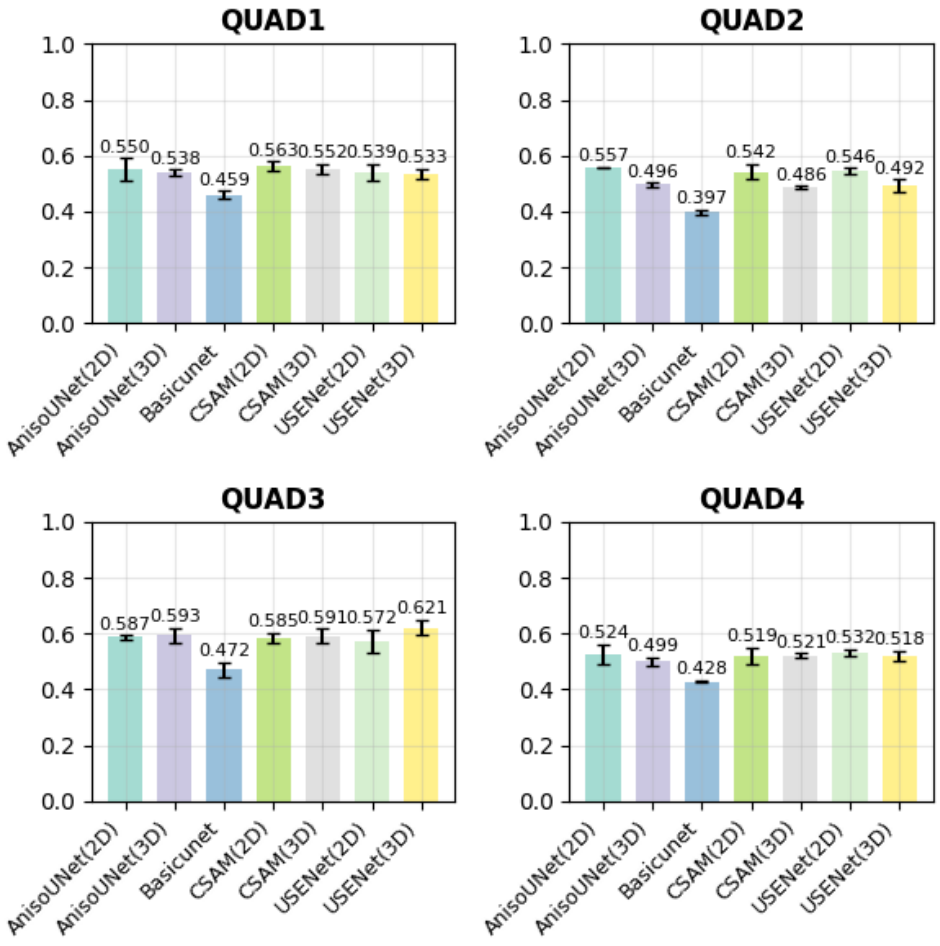

Figure 6.13: overall dice

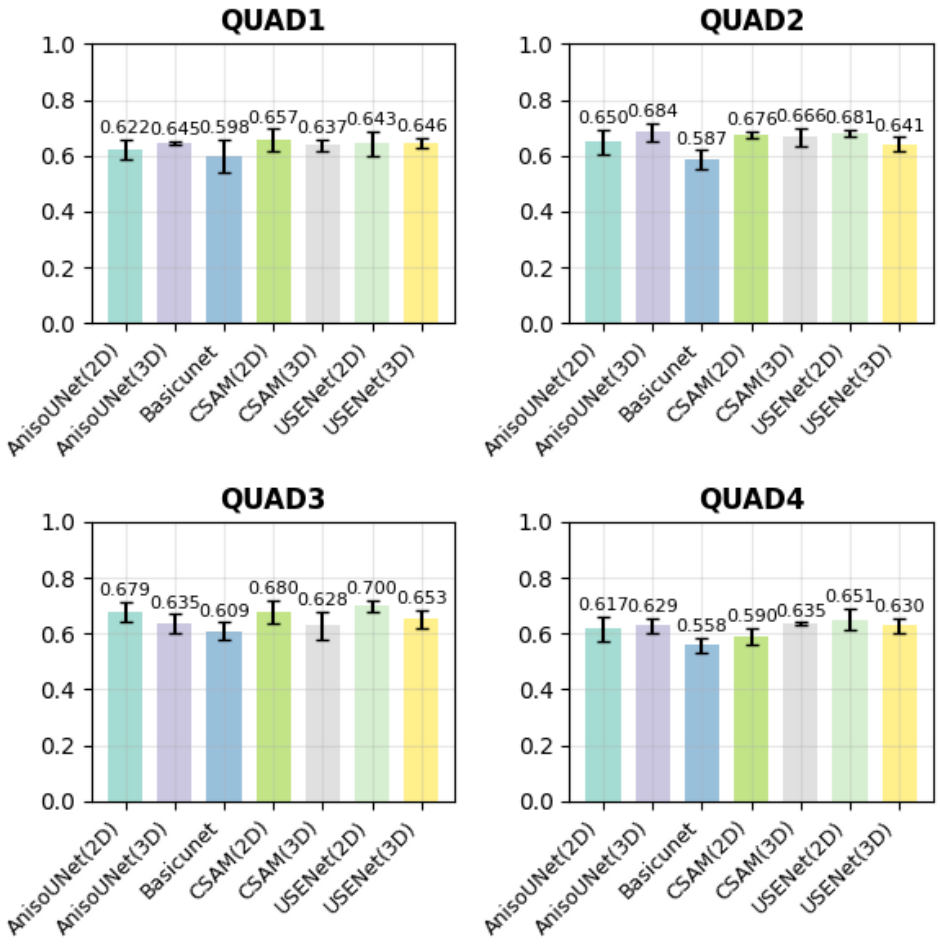

Figure 6.14: clean matched

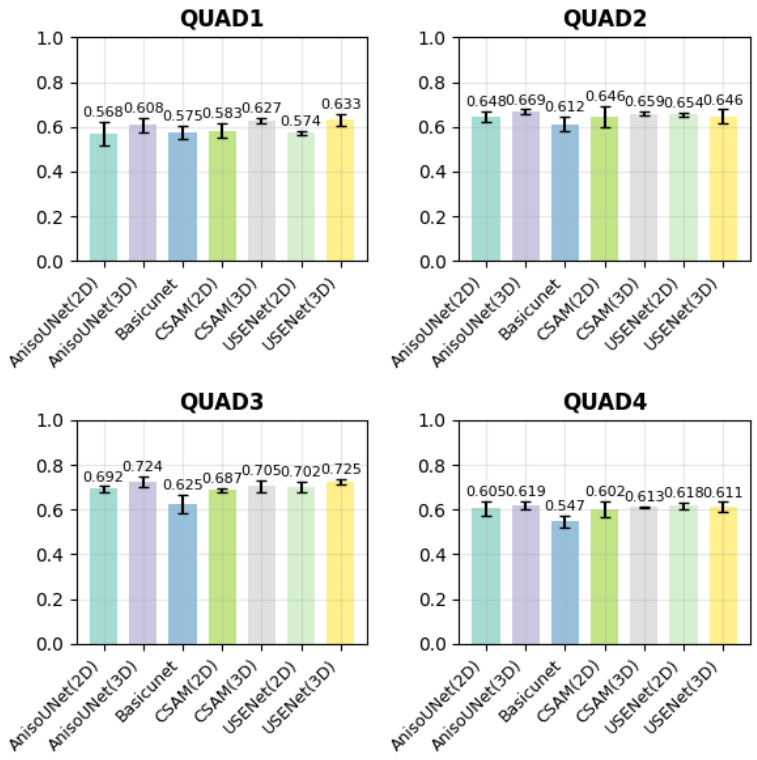

Figure 6.15: matched dice

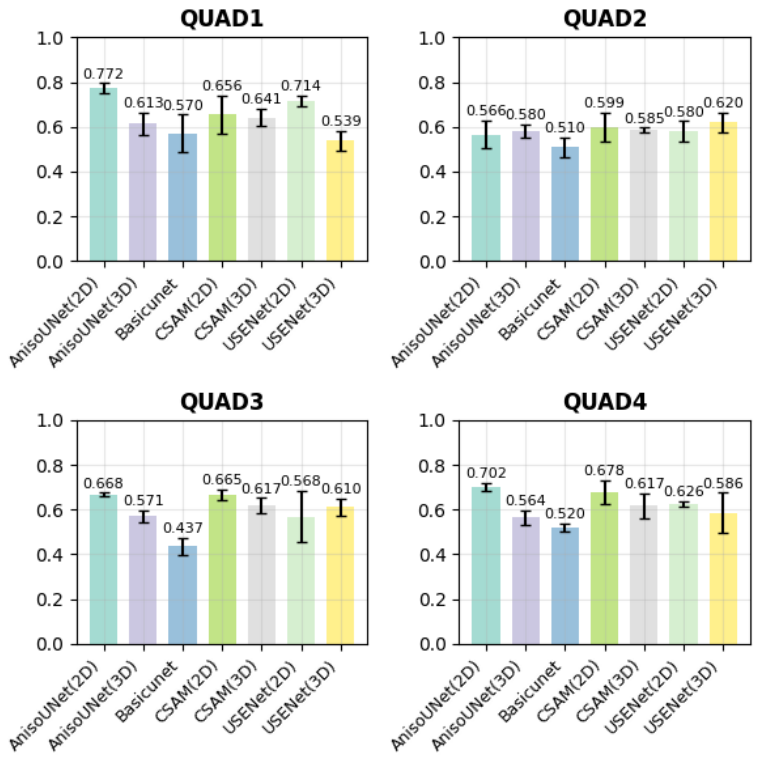

Figure 6.16: tunnels