Benchmark results:

Figure 1 I think this diverging error is due to over fitting to our training dataset.

Idea 1: Reduce training data to only right side of the field because we only test the right side of the field

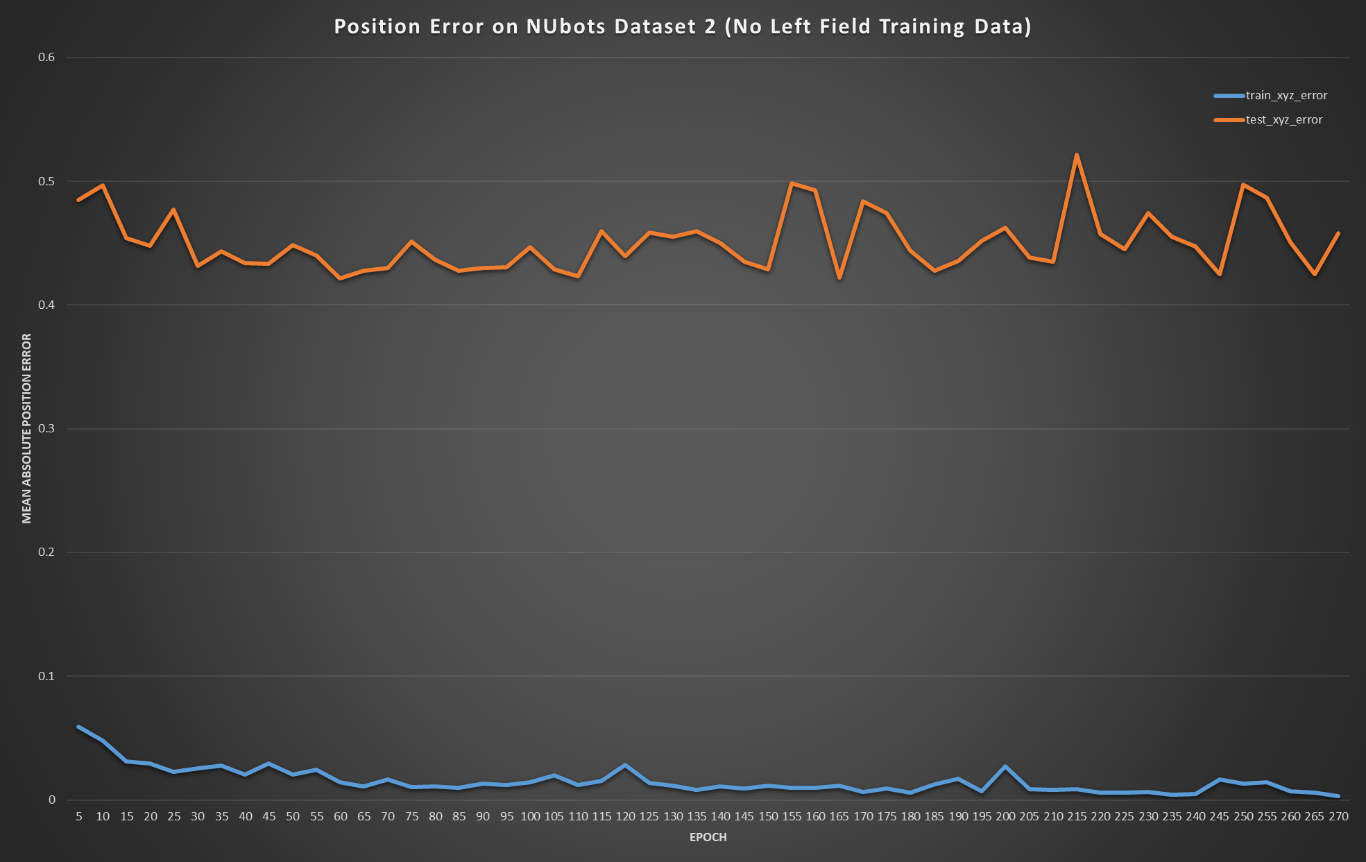


Figure 2 Removed the training data from the left hand side of the field that we are not testing in. This did not help.

Idea 2a: One way to address overfitting due to a small dataset is to reduce the amount of the pre-trained network you apply transfer learning to. For the first attempt, I made the whole ResNet network untrainable, leaving only the layers we added on the end: a fully connected layer and our 2 output layers.

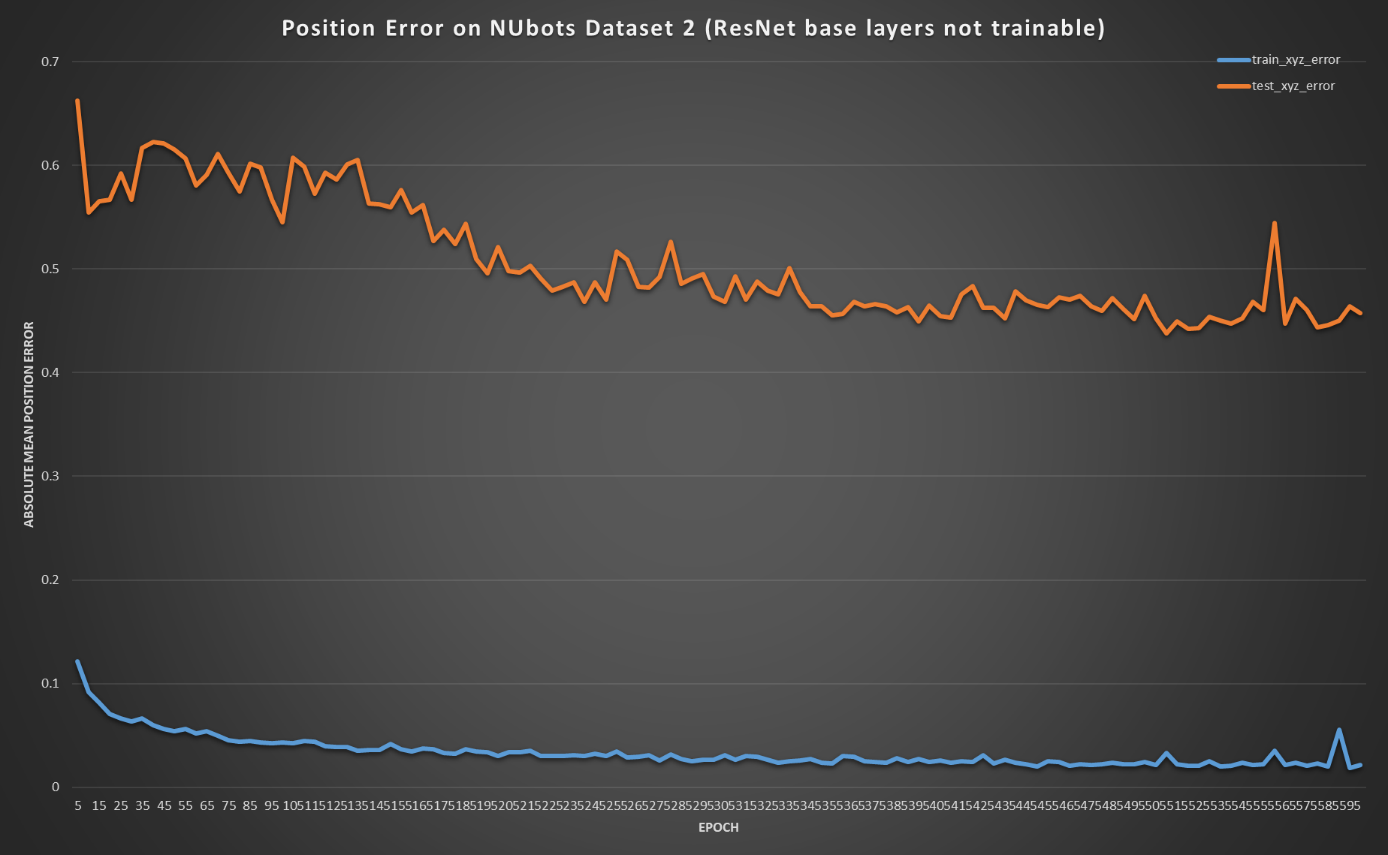


Figure 3 Keeping the reduced training data, I made the entire ResNet untrainable, leaving only the final 2 layers I added for pose regression. Had to train for twice as long as normal to stabilise. Slightly better results.

Idea 2b: Increase the amount of the ResNet network that is trainable to the last 20% of the network.

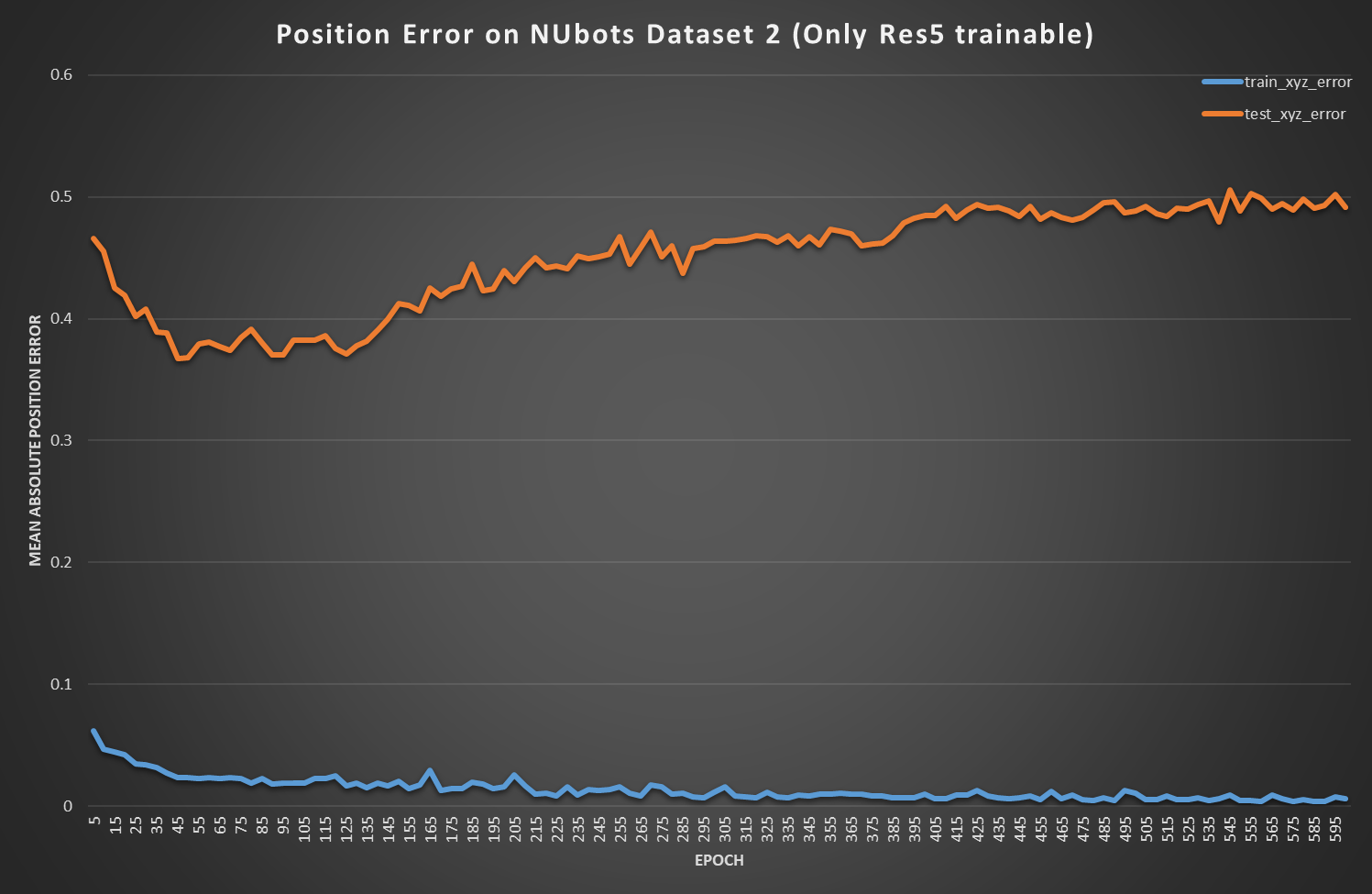


Figure 4 Allow training of the Res5 layers. Results are worse than not allowing any of the ResNet to be trainable, which is surprising.

Other Ideas:

* Use a better loss function as described in the PoseNet paper, instead of the naïve Root Mean Square loss currently used.
* Introduce dropouts in the fully connected layer we added to reduce overfitting.
* Report rotation error in degrees so it can be included in the graphs from now on.
* Augment the current dataset
  + Undistort the fish eye lens, and then reproject at a different angle?
  + Need some way to create artificial views that are a blend of two different training points.