1. Nice to meet you, everyone! My name is Robin Pierschke. And I will tell you today something about the task of image matching and my corresponding project “SuperBrain”.
2. So let's first get a proper understanding of what we want to do in the first place. After cutting a brain into multiple slices we reconstruct a digital 3 dimensional model of these 2 dimensional brain-slice scans. This is a two-step process consisting of image matching and image registration: Image matching gives us related points in 2 images, and image registration then aligns them. The needed mapping for the brain scans, the vertical lines in this graphic, is highly non-linear.
3. This work thematizes just the step of image matching.
4. The definition of image matching is the following:
5. The notion of finding key points is ill-defined because there is nothing like a correct or wrong key point. This makes the formulation of this problem in a mathematical way harder than for example for image classification.
6. Unfortunately, the classical understanding of image matching differs from the Brain matching task we consider in this present problem. While classical image matching matches natural images of the same object from different perspectives, brain matching matches medical images of different brain slices from the same perspective.
7. It is important to be aware of that because some of the following image-matching approaches make use of image augmentation which may be only suitable for natural images
8. The first method from 2015, called MatchNet, simply just classifies patches of images binary way. There are positive and negative patches.
9. This yields in patch-correspondences, not pixelwise. At test time you also have to compare a patch from the first image with each patch from the second one.
10. One year later, another approach was published, called QuadNetworks. This method works unsupervised by using augmentations. We map patches of the 2 images to single real-valued response. The goal is that corresponding patches from images should have the same position in their sorted intra-image ranking. Sounds complicated first but this just means that if a response of one patch is higher in the ranking than another, it should be so after a transformation
11. At training-time this looks like this: We compare 2 patches from the same image and make sure that the first and third patches both have a larger or smaller response than both, the second and fourth.
12. 2017 the model named Superpoint was published. This one is pretty relevant for todays research and i will come back to it again. Also, this paper is the reason why i named this project for my master-thesis “SuperBrain”. SuperPoint works in 3 steps: They first synthesized a data-set with simple geometrical forms as you can see here. They defined that all the corners are keypoints and train the base detector to find these keypoints of the synthesized dataset. The second step is to create pseudo-ground truth interest points for the unlabeled images we want to work with. This blue box just means that create pseudo-ground truth interest points not just for the original image, but also for transformed versions of it and combines these interest-points at the end. The last step is to predict and match the pseudo-interest points under different transformations.
13. D2-Net from 2019. Approaches like Sift detect key points to describe them afterward. D2-Net instead is a detect-and-describe approach. We have a local descriptor for

each pixel as you can see in the left part of the image. In order to detect a keypoint it must be the argmax in this red vector on the right side of the image and it must be the local max in this blue-marked neighborhood. It uses a version of the triplet loss, which is similar to contrastive learning. But following my first impression, this approach is not really suitable for Brain matching, because it works better for less strict thresholds. And by threshold i mean how far away a predicted match is allowed to be from the ground truth (in pixels).

14. One drawback of the approaches i showed you so far is the following: They all used (more or less) a pretty naive way of assigning matches by computing the mutual nearest neighbors in description-space. But they ignored the overall assignment-structure of keypoints.
15. The first paper known to me which addressed this problem is SuperGlue. The name is an allusion to SuperPoint, the approach i showed you before. SuperGlue does nothing else than learning the assignment structure, given local descriptors and their position from another approach, e.g. SuperPoint. They used self-attention and cross-attention to pay attention to intra-image and inter-image structures. The sinkhorn algorithm finds the optimal partial assignment at the end. I am not aware of how this algorithm works in detail but you could also use a dual-softmax function here to normalize rows and columns.
16. A really nice plot from the SuperGlue paper: The visualization of self- and cross-attention.
17. All of the approaches so far used CNNs for feature description/detection. These approaches are limited in their receptive fields. As you can see in the given image: there are a lot of texture-less points, especially in the second image on the right side. We are still able to connect these points by taking their relative position with respect to other points into account.
18. This is done by using a transformer. The first row shows the results using a transformer-based approach that i will show you in a second, the second row is done by SuperGlue.
19. LoFTR is one of the most promising approaches I read so far. You already saw the results in the slide before. The interesting part of this method is, that it doesn't use any kind of detection step. You remember: The methods so far, also methods like SIFT, detect key points and describe them. Instead of detecting key points, this approach maps the original images to a smaller feature space and matches these features. One could also interpret this as “a grid of key points”. The third part, the matching module, computes the confidence matrix. We choose a confidence threshold here and match features according to these confidence values and the mutual neighbor criteria. We call these matches coarse matches because we only know the rough neighborhood of their positions. This comes from the fact that they used the downscaled feature space instead of the original image space. For this reason, there is another step, called “coarse-to-fine module”. This last step refines the previously computed matches.
20. There are many more papers about image matching. For now, i think a first test using LoFTR would be nice. I also have a meeting with Pawel Swoboda next week, a guy from HHU who already researched this topic and is very interested in this project. Just to be sure, is this okay?