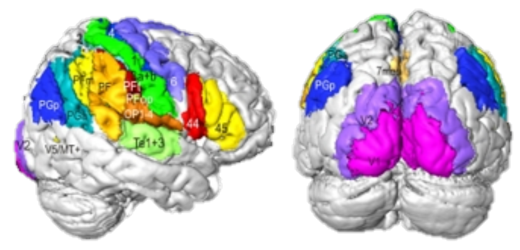
Supporting Cytoarchitectonic Mapping on Histological Brain Sections using Transfer-Learning with Convolutional Neural Networks

Christian Schiffer[1], Hannah Spitzer[1], Kai Kiwitz[2], Katrin Amunts[1,2], Timo Dickscheid[1]

[1]Institute of Neuroscience and Medicine (INM-1)
[2]Cécile and Oskar Vogt Institute of Brain Research, Medical Faculty, Heinrich Heine University, Düsseldorf, Germany
Contact: c.schiffer@fz-juelich.de

JÜLICH Forschungszentrum



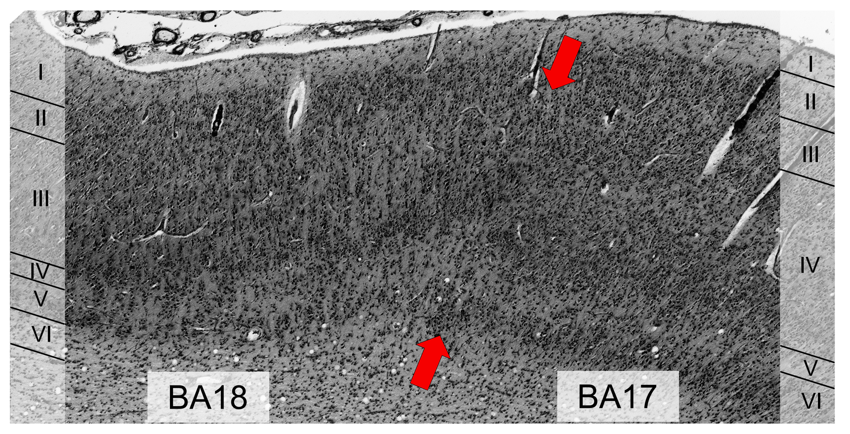
Introduction

Cytoarchitectonic mapping of cortical areas is a key aspect in creating a multimodal brain atlas [1]. Currently used semi-automatic methods [2] to detect cortical boundaries are precise, but insufficient to handle the steadily increasing quantity of histological brain sections. This motivates the development of an automated approach.To this end, a **Convolutional Neural Network** (CNN) was trained [3], which can automatically segment 13 cortical areas across different brains. We try to improve the accuracy of the above CNN by focusing on just **one specific area** in a few, **spatially close sections**. Knowing that spatially close sections share a similar texture and geometric structure, we simplify the objective in this way and expect to achieve better results.

Cytoarchitectonic mapping

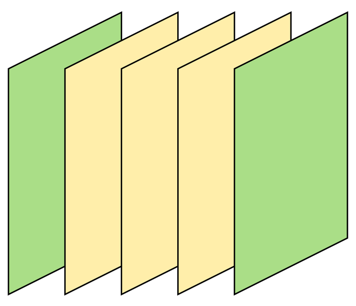
Cytoarchitectonic areas are distinguished by variations of cell distributions in cortical laminae and with respect to columnar organization

Semi-automatic method based on [2]
- Based on statistical criteria, results in reliable border definition
- **Time-consuming** delineation of inner and outer contour of the cortical ribbons



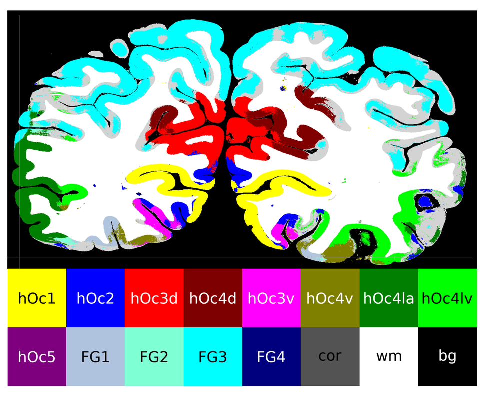
Supporting workflow using CNN model

Use spatially close sections for training
- Every 60th section manually annotated
- Spatially close sections share similar **texture and geometry**
- Focus on **one specific area** (e.g. hOc1)
- **Train** on **outer sections**
- **Predict** on **inner sections**

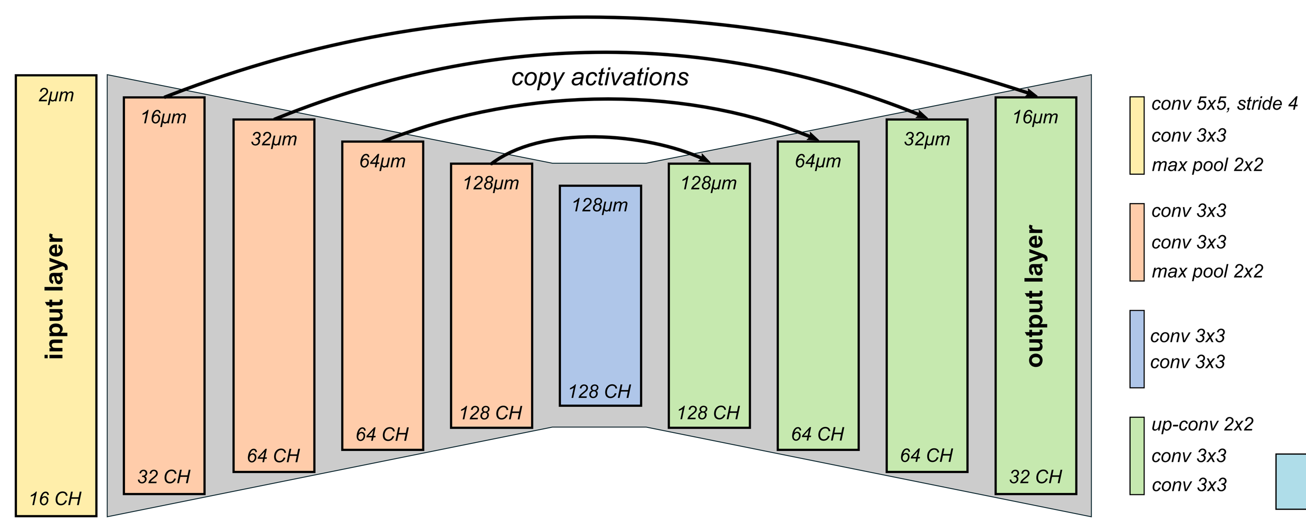


train

Transfer-learning using pre-trained network [3]
- **U-Net [4]** like network, pre-trained on 4 brains, 13 cortical areas and 111 sections
- Compensate **low amount of training data** by fine-tuning existing model



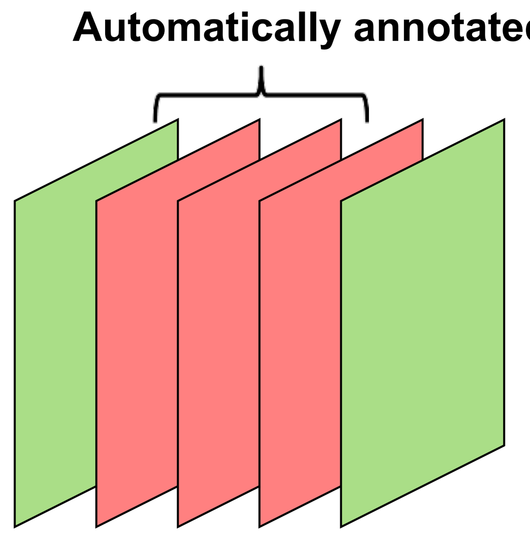
init



Convolutional Neural Network: U-Net [4]
- Learn to predict cortical areas based on **cell-body stained images**

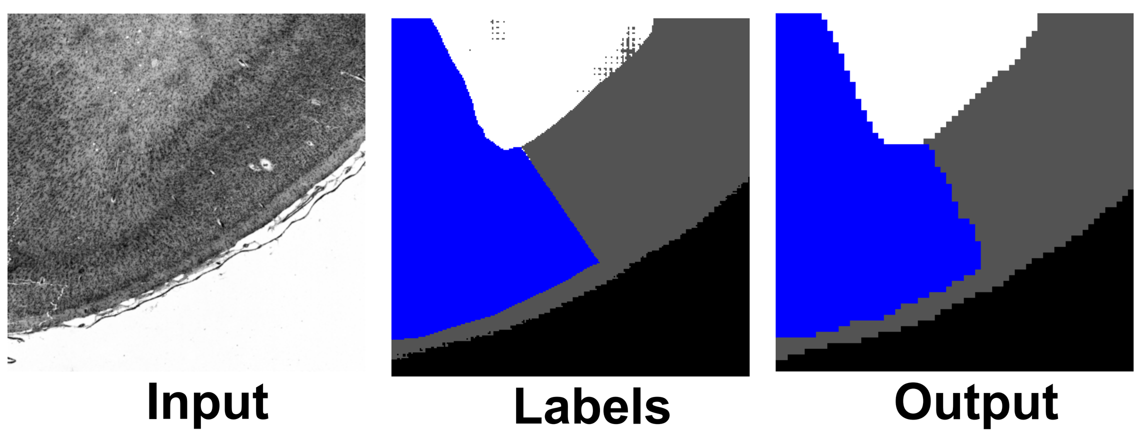
Carefully preprocess training data to simplify objective
- Train on **immediate surrounding** of annotations
- Handle **class imbalance** by adjusting sampling probability based on class frequency

predict



Automatically fill gaps
- Automatically create annotations for sections **between training sections**
- Reduce manual labor to annotating **two instead of 120** sections

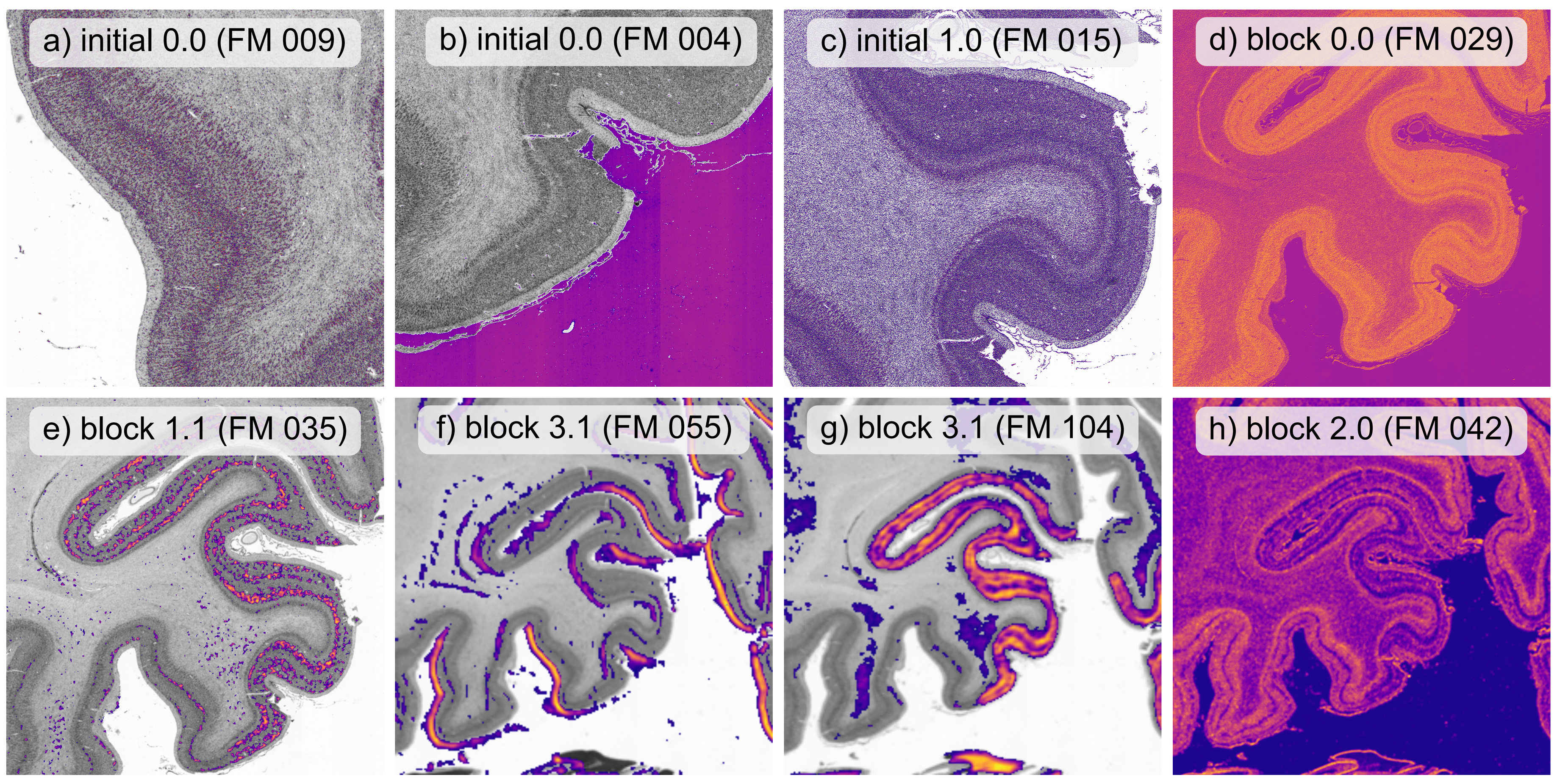
Training procedure
- Network input: grayscale images of **cell-body stained histological brain sections**
- Sample **patches** of 2025x2025 pixels on 2µm spacing
- **Purposeful overfitting** by reducing variance of training examples



Feature map visualization

Feature Maps
- **Question:** How does the model make decisions?
- Feature maps are outputs of **internal layers**
- **Internal representations** show what was learned
- **Transform** internal represenations to input image
- Use **color coding** to visualize presence of features
- Compare learned features to features used by **human mappers**
- **Deeper layers** learn more complex features

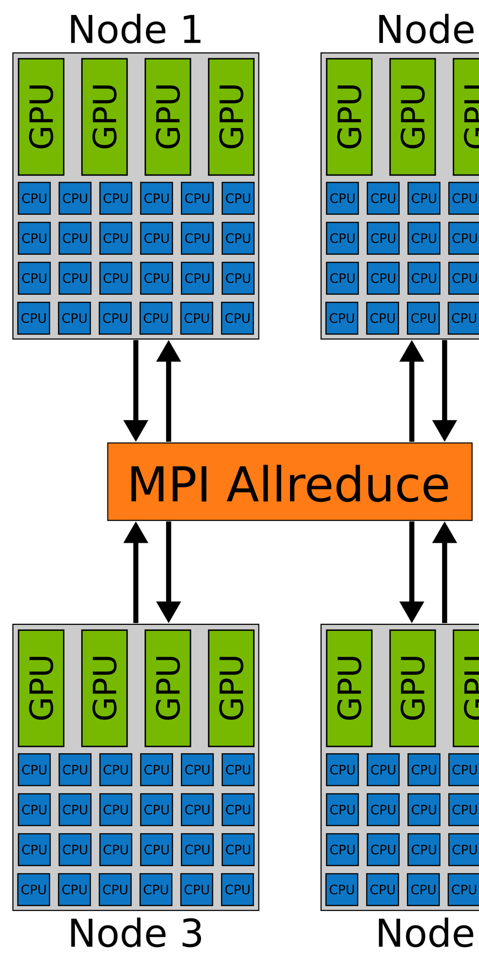
Interpretation
a) Detects **large cells** inside cortex
b) Detects **image background**
c) Visible **cell density** variations along cortex
d) High activation **inside cortex**
e) Stripes in **primary visual cortex** (*hOc1*)
f) South-West edges **tissue to background**
g) High activation for **primary visual cortex**
h) Activation for **higher visual areas**

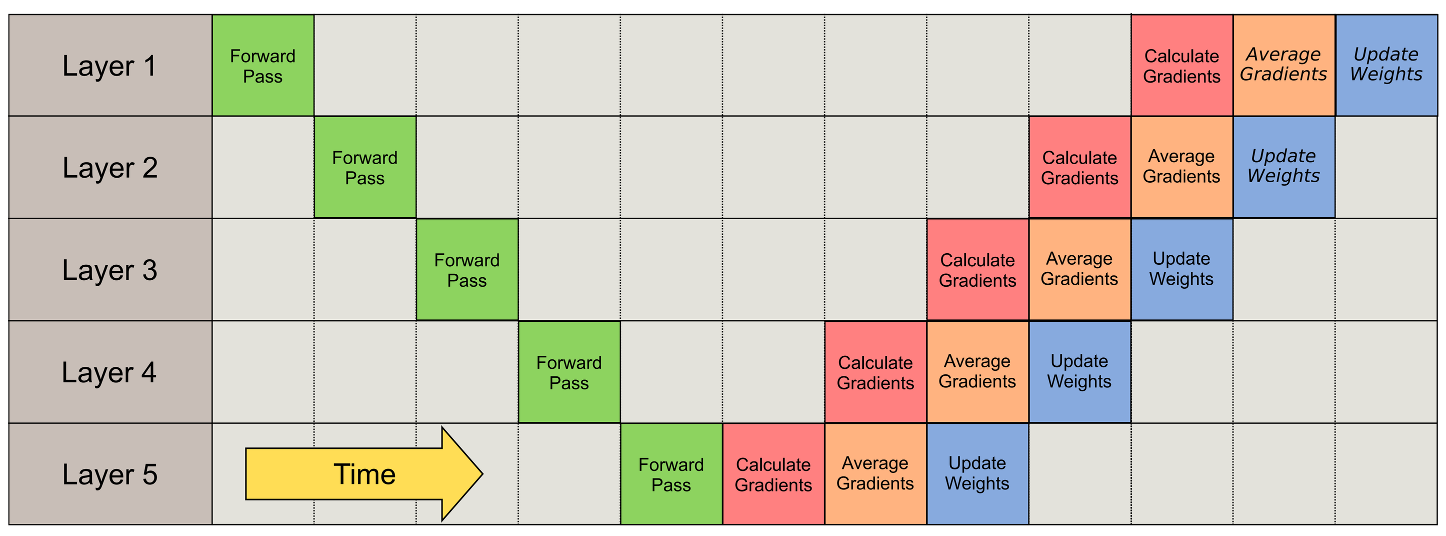


Deep Learning on HPC systems

Requirements for Deep Learning
- **GPUs** for massively parallel tensor operations
- **CPUs** for data pre-processing and data augmentation
- **I/O** and **memory** for reading training data on demand
- **Network** for inter-node communication during training

Data parallelism and synchronous training
- **Replicate model** on all compute nodes
- Process **different training samples** on each node
- Calculate forward pass on each node
- **Calculate gradients** for backpropagation on each node
- **Average gradients** across nodes before weight update
- Use **Pipelining** for efficient communication
- Computing time granted on **JURECA** [5]
- **Resources:** 250.000 CPU + 50.000 GPU core-h
- GPUs are entirely dedicated to Deep Learning
- 4 **NVidia Tesla K80** GPUs & 24 **Intel Xeon** cores per node





Conclusion

Feature map visualization
- **Feature map visualization** gives insight into the inner machinery of a trained model
- Model learns **intuitive baseline** features and combines them to useful **higher level abstractions** which are similar to features used by **human mappers**
- Future investigations to analyse the **dependencies** between features in different layers

Deep Learning on HPC systems
- Training across multiple nodes **reduces training time** and improves iterative workflow
- **Distributed training** across nodes allows more efficient usage of available HPC systems
- **Future work:** Investigate possible use cases of new prototype systems like **JURON**

References
[1] K. Amunts, K. Zilles (2015) Architectonic mapping of the human brain beyond brodmann, Neuron, vol. 88, no. 6, pp. 1086-1107.
[2] A. Schleicher et al. (1999) Observer-independent method for microstructural parcellation of cerebral cortex: a quantitative approach to cytoarchitectonics, Neuroimage, vol. 9, no. 1, pp. 165-177.
[3] H. Spitzer et al. (2017) Parcellation of visual cortex on high-resolution histological brain sections using Convolutional Neural Networks, IEEE International Symposium on Biomedical Imaging.
[4] O. Ronneberger et. al. (2015) U-Net: Convolutional neural networks for biomedical image segmentation, MICCAI, Springer, 2015, pp. 234-241.
[5] Jülich Supercomputing Centre. (2016). JURECA: General-purpose supercomputer at Jülich Supercomputing Centre, Journal of large-scale research facilities, 2, A62.

Acknowledgements
This project was supported by the Helmholtz Association through the Helmholtz Portfolio Theme “Super-computing and Modeling for the Human Brain”, and by the European Unions Horizon 2020 Framework Research and Innovation under Grant Agreement No. 7202070 (Human Brain Project SGA1). Computing time was granted by the John von Neumann Institute for Computing (NIC) and provided on the supercomputer JURECA at Jülich Supercomputing Centre (JSC).

SMHB Human Brain Project NIC