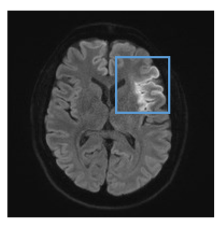
STROKE CLASSIFICATION USING DEEP LEARNING APPROACHES

Lisa Herzog[1,2], Elvis Murina[2], Oliver Dürr[2], Susanne Wegener*[4], Beate Sick*[1,2]
[1]UZH, [2]ZHAW, [3]HTWG , [4]University Hospital Zurich

UniversitätsSpital Zürich

Motivation



Extract predefined features from ROI → **Traditional Approach**: Prediction model based on extracted features

Deep Learning Approach: Prediction using a CNN considering the whole image

Dropout at test time → **Distribution of predictions per image (Uncertainty)** → How to best classify the patients?

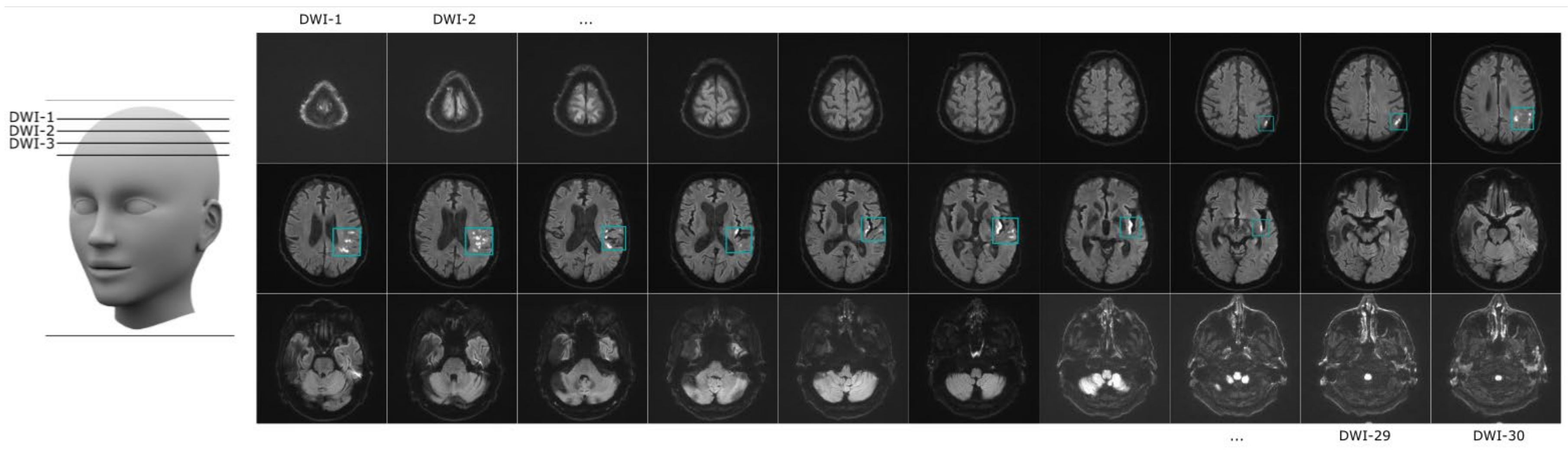
No dropout at test time → One prediction per image

- Using 2D images for 3D data results in one prediction per image. Since there are multiple images per patient, the image-based predictions have to be combined to one patient-level prediction. To know how certain the model is about each prediction might help to improve the aggregation.

1. We propose an adapted CNN architecture that considers the information from neighboring images.
2. We apply MC-dropout during training and test time to obtain uncertainties.
3. We determine patient-level predictions by aggregating image-based predictions and the corresponding uncertainties

Data

- We have n=397 patients: n=259 stroke and n=138 transient ischemic attack (TIA) patients
- We have ~30 DWI slices per patient and each image is expert-labeled as "stroke" or "non-stroke"



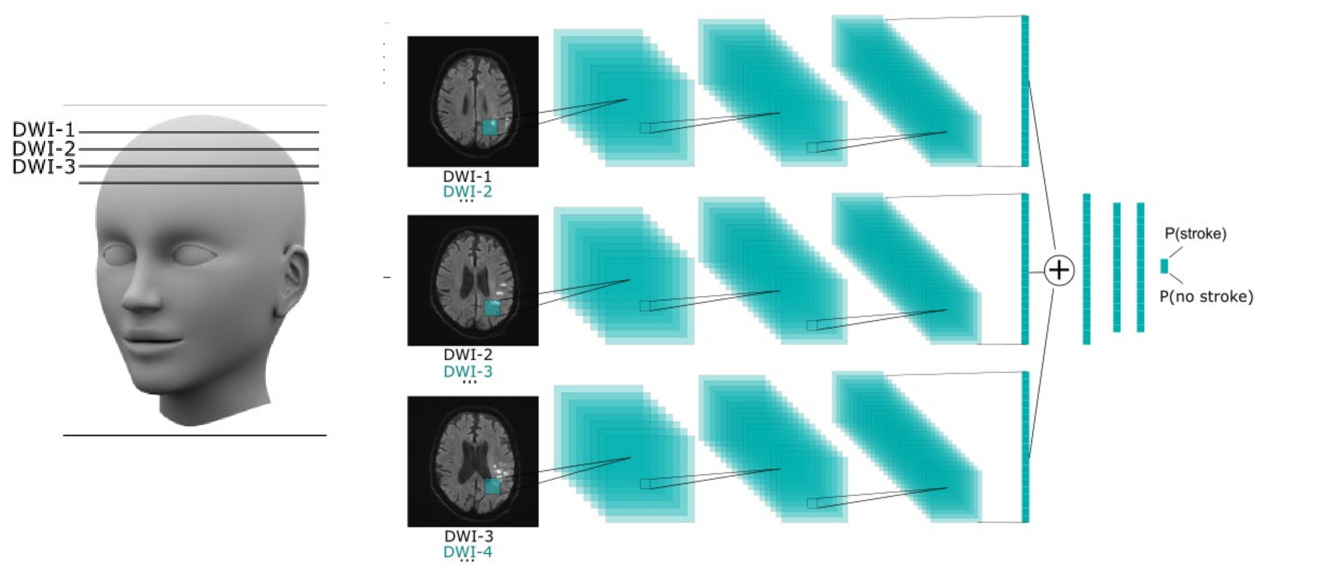
- For image and patient level predictions we used:

	For image level prediction		For patient level predictions		
	train1	valid1	train2	valid2	test
Patients	257	40	40+40	20	40
Images	7598	1209	1209+1213	594	1173

Methods I

Image level predictions

- We developed a VGG-like architecture which is repeated three times in three parallel pathways. We then feed in three subsequent images while we predict the image in the center.



- We train from scratch using the Adam SGD algorithm and apply data augmentation (zooming, rotating, shearing, shifting)
- We use MC dropout[2] to obtain uncertainty measures for the predicted class probabilities (VR*, PE*, MI*).

Architecture
2 Conv-Stacks*-32
Max Pooling
Dropout
2 Conv-Stacks-64
Max Pooling
Dropout
3 Conv-Stacks-128
Max Pooling
Dropout
3 Conv-Stacks-256
Max Pooling
Dropout
3 Conv-Stacks-512
Max Pooling
Dropout
3 Conv-Stacks-512
Max Pooling
Dropout
FC-400
Dropout
FC-100
Dropout
FC-2
Softmax

Aggregation to patient level predictions

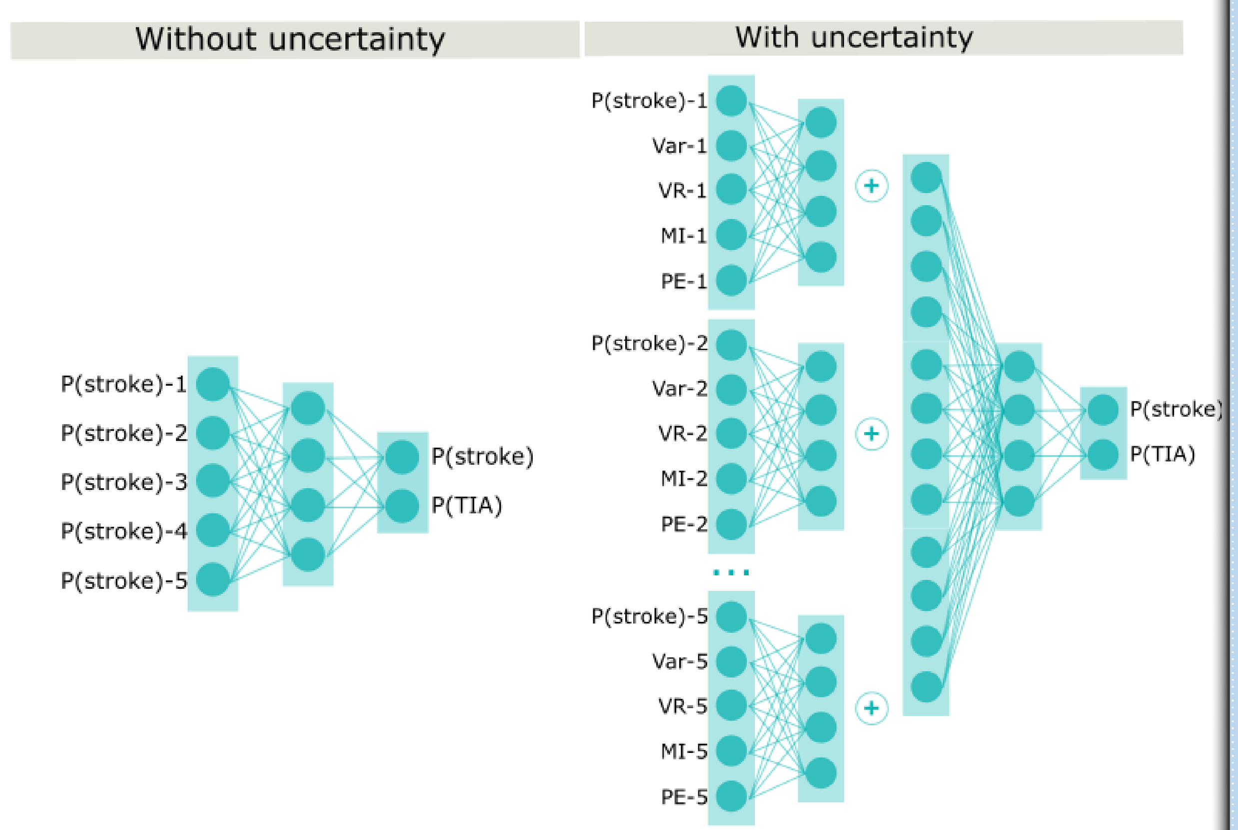
In order to aggregate the predicted class probabilities of the images to one patient level prediction we used:

- **Maximum**: As baseline aggregation method we use the maximum. If any stroke is observed on at least one image, we classify the patient as stroke patient.

Methods II

Aggregation to patient level predictions

- **Neural Network I:** We feed in the 5 predictions (p1,…,p5) that are most likely to show a stroke
- **Neural Network II:** We feed in the same 5 predictons as in the Neural Network I and add the corresponding uncertainties



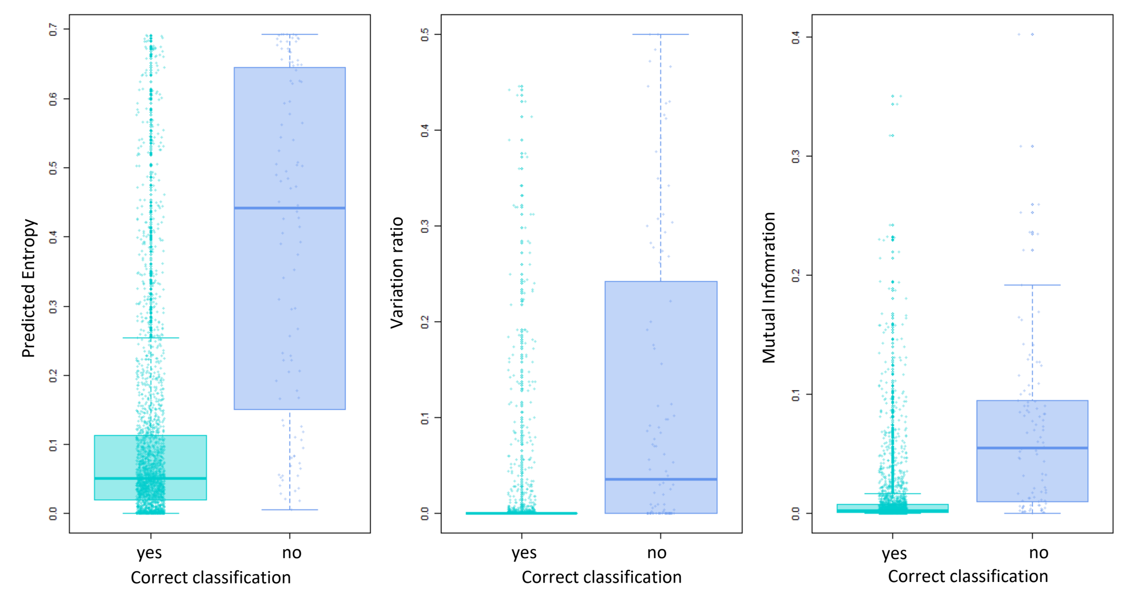
Results

Image level predictions (valid1)

%	Accuracy	Specificity	Sensitivity
Parallel CNN	96.01 [94.71,97.01]	99.00 [98.11,99.47]	84.28 [79.00,88.42]

Table1: The table shows the results (with wilson CIs) based on the customized parallel architecture. The model shows good classification performances on the image level.

Figure1: Comparing the uncertainty measures of the images, we see, that wrongly classified images are predicted with higher uncertainties. Therefore, considering the uncertainties toa ggregate the image level predictions should allow us to get better patient level predictions.



Aggregation to patient level predictions (valid2)

	Maximum and Neural Net II			Neural Net I		
	Acc	Spec	Sens	Acc	Spec	Sens
Parallel CNN (#patient)	95.00 [76.39,99.74]	100 [56.55,100.0]	93.33 (1) [70.18,99.66]	90.00 [69.90,97.21]	100 [56.55,100.0]	86.6 (2) [62.12,96.26]

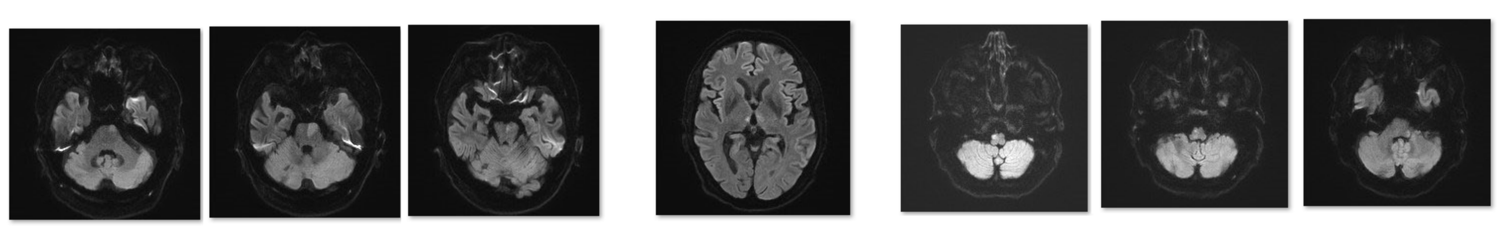
Table2: We compare the models on valid2. We find no differences between the bl aggregation method (maximum) and NN2 that considers the uncertainties. However, the dataset is so small, that we should not conclude that taking uncertainties into account doesn't help.

Final outcome predictions (test)

%	Accuracy	Specificity	Sensitivity
Image	95.33 [93.92,96.43]	98.84 [97.88,99.37]	82.10 [76.61,86.52]
Patient (NN II)	92.50 [80.14,97.42]	100.00 [79.61,100.00]	88.00 [70.04,95.83]

Table3: The model seems to generalize well. On the patient level, we don't recognize three stroke patients.

The following stroke patients were wrongly classified because the model was not able to identify the stroke on the images:



Patient 91 Patient 369 Patient 386

Conclusion

- We proposed an adapted CNN architecture for 2D images that takes into account the 3D data structure.
- Taking the maximum as aggregation method seems to yield promising results. However, at this stage we are not able to decide which method is superior due to the limited number of patients.
- Since we observe a shift in the uncertainty distribution between wrongly and correctly classified images, considering the uncertainty measures should improve the results. This should be investigated further using more data.

[1] J. Bernal, Deep Convolutional Neural Networks for brain image analysis on magnetic resonance imaging: a review, 2017
[2] Y.Gal, Uncertainty in Deep Learning, 2016

* Conv-Stack-f = convolutional, batch normalization layer and a ReLU nonlinearity. -f represents the number of filters, MP = Max Pooling, BN = Batch Normalization, FC = Fully connected layer followed by a batch normalization layer, SD = Standard deviation, VR = Variation Ratio, PE = Predicted Entropy, MI = Mutual Information