# Automated segmentation of lacunes of presumed vascular origin in brain MRI scans

Eline Ooms

Delft University of Technology, The Netherlands

August 13, 2020

# Table of contents

# Table of Contents

# Introduction - cerebral small vessel disease

**cerebral small vessel disease**

:= changes in the brain due to damaged small vessels

# Introduction - cerebral small vessel disease - lacune as biomarker

**cerebral small vessel disease**

:= changes in the brain due to damaged small vessels

Resulting lesions (biomarkers)

- Lacunes
- Recent small subcortical infarcts
- White matter hyperintensities
- Perivascular spaces
- Cerebral microbleeds

# Introduction - cerebral small vessel disease - consequences

**cerebral small vessel disease**

:= changes in the brain due to damaged small vessels

Psychological and physical inabilities

- Cognitive decline
- Depressive symptoms
- Gait disturbances
- Urinary problems
- Dementia

# Introduction - relevance

**Relevance of an automated lacune segmentation method**

- Segmentation of lacunes
  - Detecting the disease
  - Giving a relevant treatment
  - Unraveling the cause of the disease
- Automating the segmentation procedure
  - Saving time of the radiologist

# Table of Contents

# Background on lacunes - lacunes - definition[1]

**Definition**

"a round or ovoid, subcortical, fluid-filled cavity of between 3 mm and about 15 mm in diameter, consistent with a previous acute small deep brain infarct or haemorrhage in the territory of one perforating arteriole"

Figure: Lacune on a FLAIR image

[1] J. Wardlaw et al. (2013). "Neuroimaging standards for research into small vessel disease and its contribution to ageing and neurodegeneration.". In: *The Lancet Neurology* 12.8, pp. 822–838. DOI: 10.1016/S1474-4422(13)70124-8.

# Background on lacunes - lacunes - appearance

**FLAIR image:**
hypointense with a hyperintense rim

**T1-weighted image:**
hypointense

**T2-weighted image:**
hyperintense

Figure: Lacune on a FLAIR image

# Background on lacunes - previous work

**Yokoyama et al.** & **Uchiyama et al.** & **Ghafoorian et al.**

- All two-staged methods
  - Candidate detection
  - False positive reduction
- Important findings[2][3]
  - Perivascular spaces are often among false positives
  - Location appears to be important in differentiaton with perivascular spaces

[2] Y. Uchiyama, R. Yokoyama, H. Ando, T. Asano, H. Kato, H. Yamakawa, H. Yamakawa, T. Hara, T. Iwama, H. Hoshi, and H. Fujita (2007). "Computer-Aided Diagnosis Scheme for Detection of Lacunar Infarcts on MR Images.". In: *Academic Radiology* 14.12, pp. 1554–1561. DOI: 10.1016/j.acra.2007.09.012.

[3] M. Ghafoorian, N. Karssemeijer, T. Heskes, M. Bergkamp, J. Wissink, J. Obels, K. Keizer, F. de Leeuw, B. van Ginneken, E. Marchiori, and B. Platel (2017). "Deep multi-scale location-aware 3D convolutional neural networks for automated detection of lacunes of presumed vascular origin.". In: *NeuroImage: Clinical* 14, pp. 391–399. DOI: 10.1016/j.nicl.2017.01.033.

# Background on lacunes - data: Rotterdam scan study

Scans

- 734 scans with lacunes
- ~ 4000 scans without lacunes
- Image size of 512x512x192

Available modalities

- FLAIR images
- T1-weighted images
- T2-weighted images

Figure: Lacune on a T1-weighted image from the Rotterdam scan study.

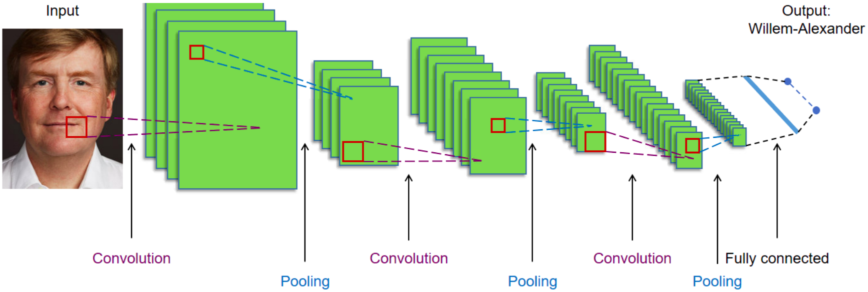
# Background on lacunes - data: Rotterdam scan study

Figure: Lacune on a T1-weighted image.

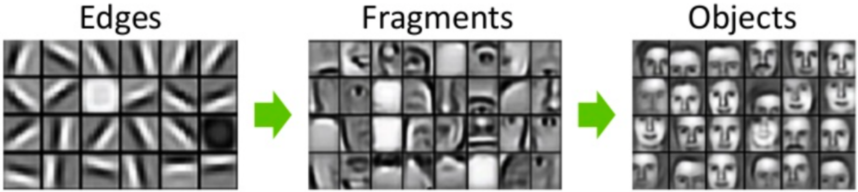
Figure: Annotated lacune on a T1-weighted image.

# Table of Contents

# Convolutional neural network



# Convolutional neural network

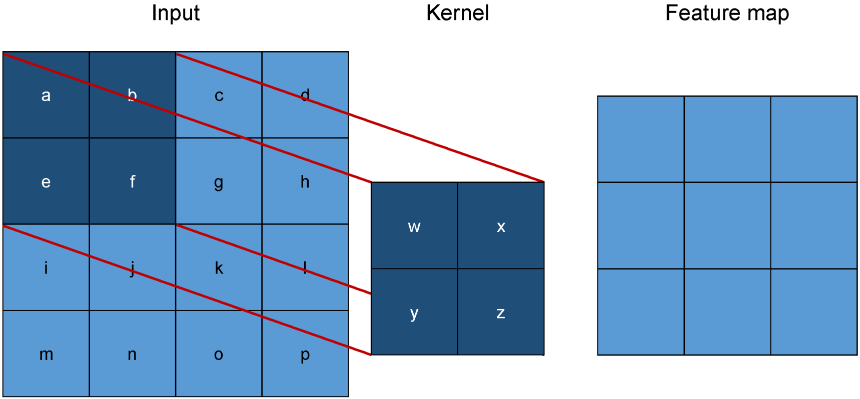


# Convolutional neural network - components

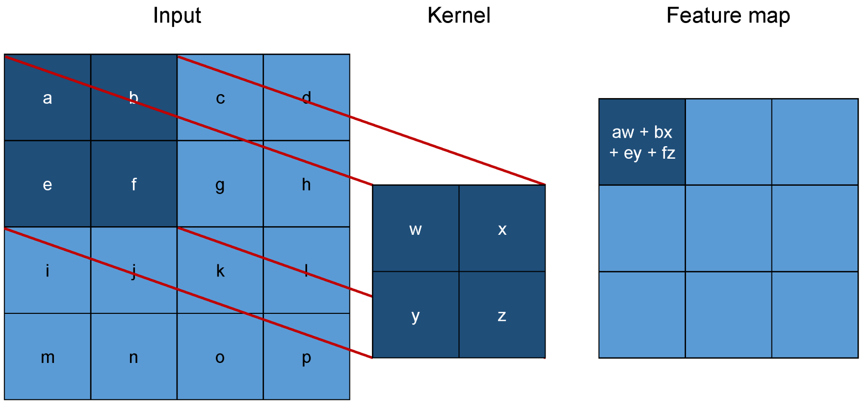
A convolutional neural network consists of

- A convolution layer
- An activation function
- A pooling layer
- A fully connected layer
- An end activation function

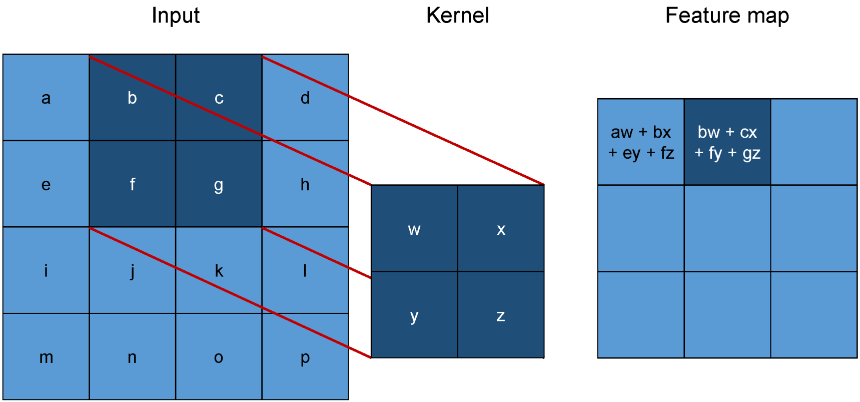
# Convolutional neural network - convolution layer



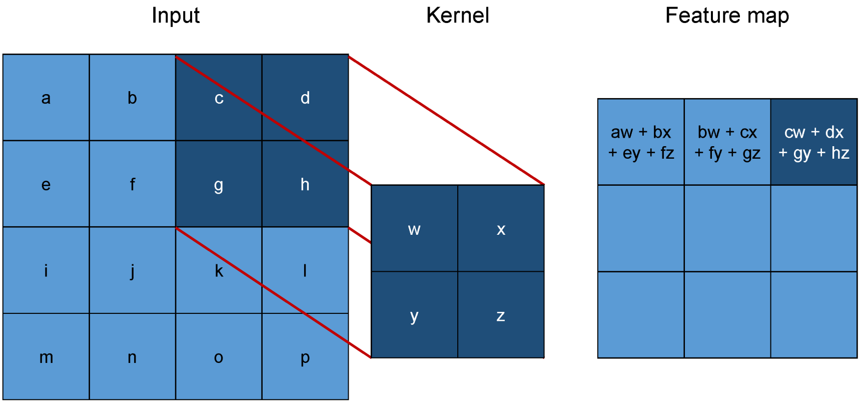
# Convolutional neural network - convolution layer



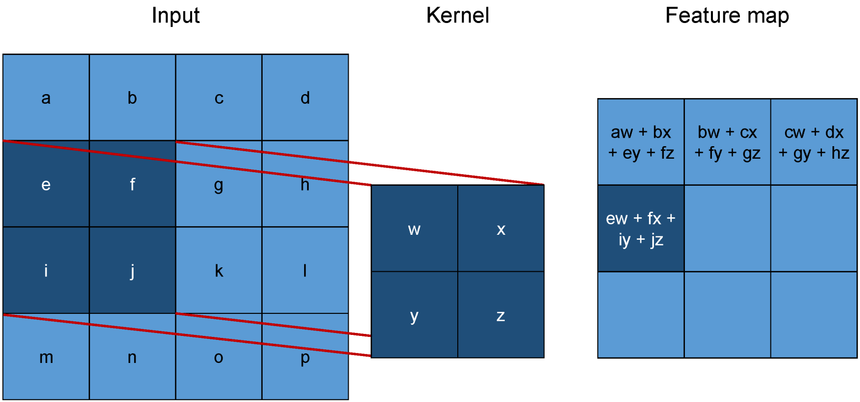
# Convolutional neural network - convolution layer



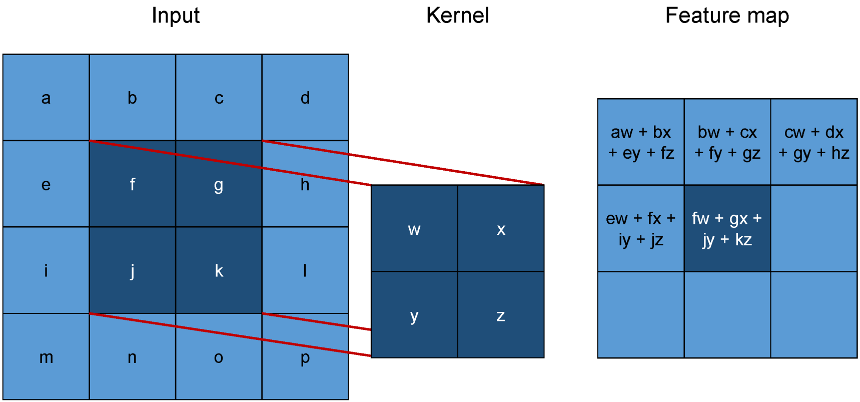
# Convolutional neural network - convolution layer



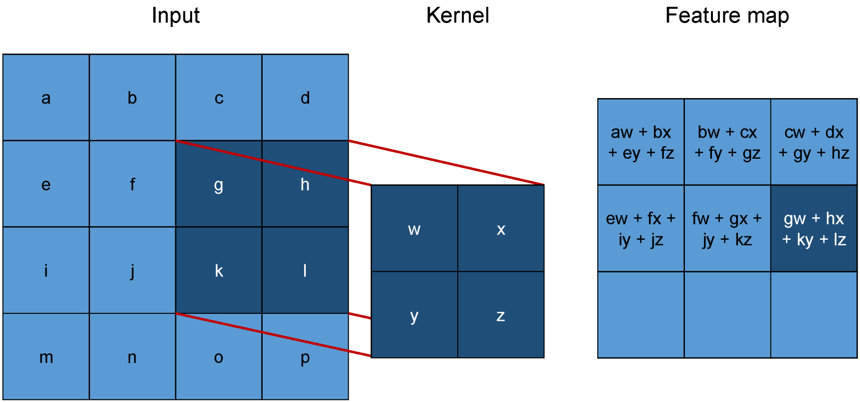
# Convolutional neural network - convolution layer



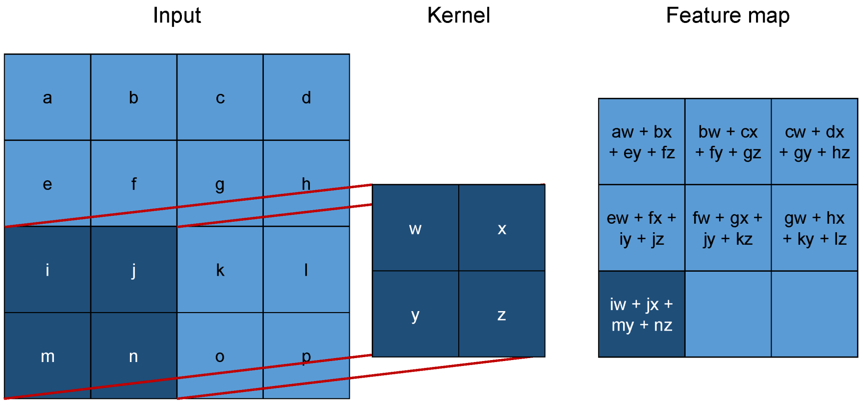
# Convolutional neural network - convolution layer



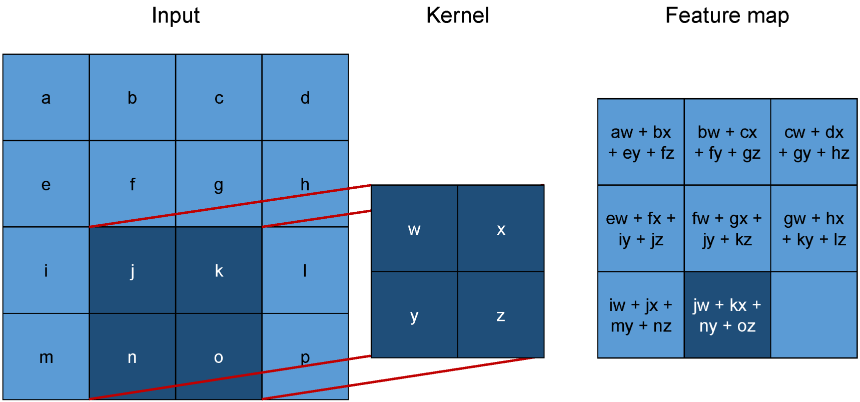
# Convolutional neural network - convolution layer



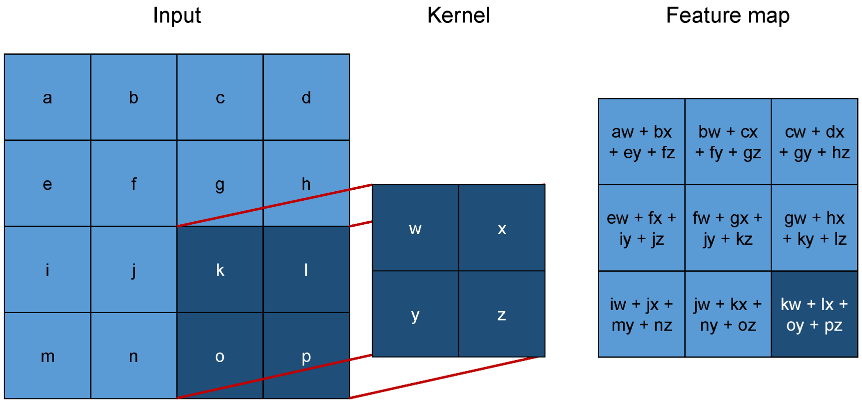
# Convolutional neural network - convolution layer



# Convolutional neural network - convolution layer



# Convolutional neural network - convolution layer



# Convolutional neural network - convolution layer

Let $I$ be the input feature map, $K$ be the kernel and $S$ be the output feature map.

$$S(i,j) = (K * I)(i,j) = \sum_{m}^{M} \sum_{n}^{N} I(i+m, j+n) K(m,n).$$

# Convolutional neural network - activation function

A non-linearity, which is applied elementwise such that for $X \in \mathbb{R}^{m \times n}, f : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$

$$(f(x))_{ij} = f(x_{ij})$$

# Convolutional neural network - activation function

**Rectified linear unit (ReLU)**[4]

$$f(x_{ij}) = \begin{cases} 0 & \text{for } x_{ij} \leq 0, \\ x_{ij} & \text{for } x_{ij} > 0, \end{cases}$$

**Leaky ReLU**[5]

$$f(x_{ij}) = \begin{cases} 0.01x_{ij} & \text{for } x_{ij} \leq 0, \\ x_{ij} & \text{for } x_{ij} > 0, \end{cases}$$

[4] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun (2009). "What is the best multi-stage architecture for object recognition?" In: *IEEE 12th International Conference on Computer Vision*, pp. 2146–2153. DOI: 10.1109/ICCV.2009.5459469.

[5] A. Maas, A. Hannun, and A. Ng (2013). "Rectifier nonlinearities improve neural network acoustic models". In: *Proceedings of the 30th International Conference on Machine Learning*. Atlanta, United States of America.

# Convolutional neural network - activation function

**Exponential linear unit (ELU)**[6]

$$f(x_{ij}) = \begin{cases} \alpha(e^{x_{ij}} - 1) & \text{for } x_{ij} \leq 0, \\ x_{ij} & \text{for } x_{ij} > 0, \end{cases}$$

where $\alpha > 0$.

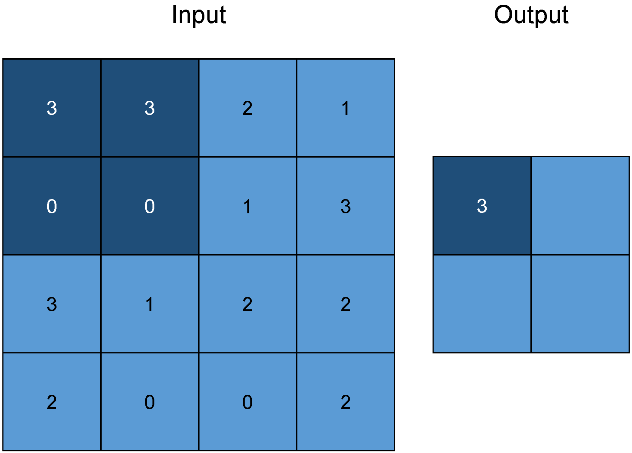
[6] D. Clevert, T. Unterthiner, and S. Hochreiter (2016). *Fast and accurate deep network learning by exponential linear units (ELUs)*.

# Convolutional neural network - pooling layer

**Pooling function**

- Downsizing the image for
  - Efficiency
  - Translational invariance
- Types of pooling functions
  - Max pooling
  - Average pooling

# Convolutional neural network - pooling layer - max pooling



# Convolutional neural network - pooling layer - max pool

Input

| | | | |
|---|---|---|---|
| 3 | 3 | 2 | 1 |
| 0 | 0 | 1 | 3 |
| 3 | 1 | 2 | 2 |
| 2 | 0 | 0 | 2 |

Output

| | |
|---|---|
| 3 | 3 |
| | |

# Convolutional neural network - pooling layer - max pool

Input

| | | | |
|---|---|---|---|
| 3 | 3 | 2 | 1 |
| 0 | 0 | 1 | 3 |
| 3 | 1 | 2 | 2 |
| 2 | 0 | 0 | 2 |

Output

| | |
|---|---|
| 3 | 3 |
| 3 | |

# Convolutional neural network - pooling layer - max pool

Input

| | | | |
|---|---|---|---|
| 3 | 3 | 2 | 1 |
| 0 | 0 | 1 | 3 |
| 3 | 1 | 2 | 2 |
| 2 | 0 | 0 | 2 |

Output

| | |
|---|---|
| 3 | 3 |
| 3 | 2 |

# Convolutional neural network - pooling layer - average pool

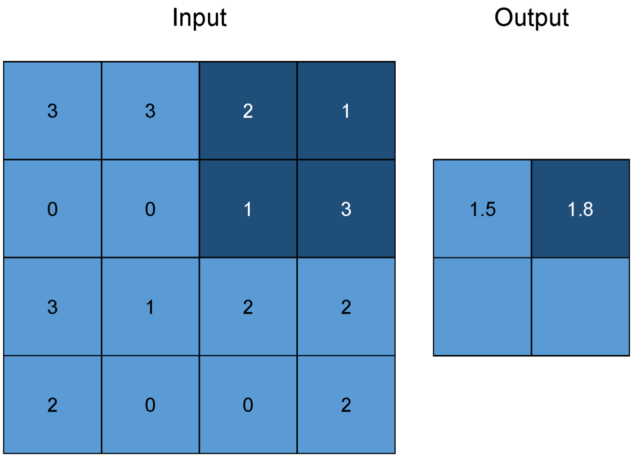
**Input**

| | | | |
|---|---|---|---|
| 3 | 3 | 2 | 1 |
| 0 | 0 | 1 | 3 |
| 3 | 1 | 2 | 2 |
| 2 | 0 | 0 | 2 |

**Output**

| | |
|---|---|
| 1.5 | |
| | |

# Convolutional neural network - pooling layer - average pool



# Convolutional neural network - pooling layer - average pool

Input

| | | | |
|---|---|---|---|
| 3 | 3 | 2 | 1 |
| 0 | 0 | 1 | 3 |
| 3 | 1 | 2 | 2 |
| 2 | 0 | 0 | 2 |

Output

| | |
|---|---|
| 1.5 | 1.8 |
| 1.5 | |

# Convolutional neural network - pooling layer - average pool

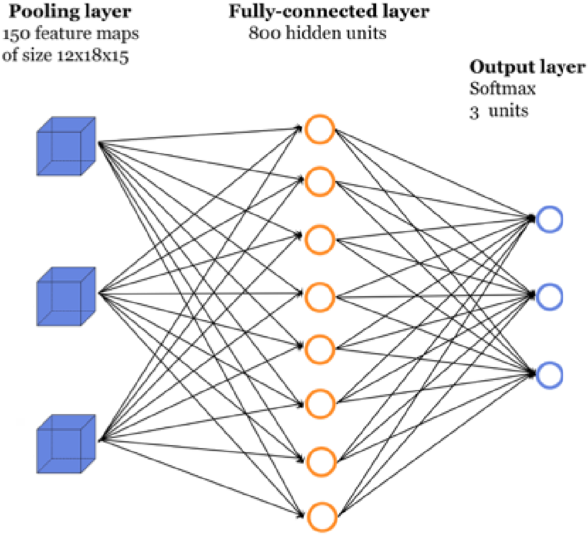
Input

| | | | |
|---|---|---|---|
| 3 | 3 | 2 | 1 |
| 0 | 0 | 1 | 3 |
| 3 | 1 | 2 | 2 |
| 2 | 0 | 0 | 2 |

Output

| | |
|---|---|
| 1.5 | 1.8 |
| 1.5 | 1.5 |

# Convolutional neural network - fully connected layer



# Convolutional neural network - end activation function

Let $\mathbf{x} \in \mathbb{R}^c$, where $c$ represents the number of classes

**Sigmoid function:** for a binary classification task

$$f(x_i) = \frac{1}{1 + e^{-x_i}}$$

**Softmax function:** for a multiclass classification task

$$f(x_i) = \frac{e^{x_i}}{\sum_j^c e^{x_j}}$$

# Table of Contents

# Training a convolutional neural network

Loss function

Optimizer

# Training a convolutional neural network - loss function

Suppose we want to train a dataset of $m$ examples $\{X^{(i)}\}_{i=1}^{m} \in \mathbb{R}^{V \times W}$ with labeled outputs $\{Y^{(i)}\}_{i=1}^{m} \in \mathbb{R}^{V \times W}$ and the predictions computed by a network $\{\hat{P}(X^{(i)};\Theta)\}_{i=1}^{m} \in \mathbb{R}^{V \times W}$ where $\Theta$ are the weights in the network

**Categorical cross-entropy loss**

$$L(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} Y_k^{(i)} \log\left(\hat{P}(X^{(i)};\Theta)\right), \tag{1}$$

where $K$ is the total number of classes

# Training a convolutional neural network - loss function

Suppose we want to train a dataset of $m$ examples $\{X^{(i)}\}_{i=1}^{m} \in \mathbb{R}^{V \times W}$ with labeled outputs $\{Y^{(i)}\}_{i=1}^{m} \in \mathbb{R}^{V \times W}$ and the predictions computed by a network $\{\hat{P}(X^{(i)}; \Theta)\}_{i=1}^{m} \in \mathbb{R}^{V \times W}$ where $\Theta$ are the weights in the network

**Binary cross-entropy loss**

$$L(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} Y^{(i)} \log\left(\hat{P}(X^{(i)}; \Theta)\right) + \left(1 - Y^{(i)}\right) \log\left(1 - \hat{P}(X^{(i)}; \Theta)\right) \tag{2}$$

# Training a convolutional neural network - optimizer

**Goal:** minimize loss function

$$\underset{\Theta}{\text{minimize}} \quad L\left(\hat{P}(X^{(i)};\Theta), Y^{(i)}\right)$$

# Training a convolutional neural network - optimizer

**Stochastic gradient descent (SGD) update**

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \nabla L(\boldsymbol{\theta})^T,$$

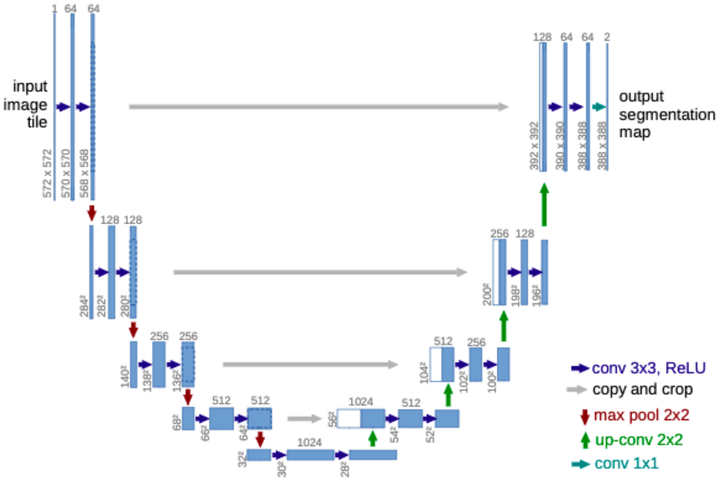
where $\epsilon$ is called the learning rate

**Other variants**

- SGD with momentum
- RMSProp
- RMSProp with momentum
- AdaDelta
- Adam

# Table of Contents

# U-Net architecture[7]




[7] O. Ronneberger, P. Fisher, and T. Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Proceedings of the 18th Medical Image Computing and Computer Assisted Intervention*. Munich, Germany, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

# U-Net architecture - upsampling

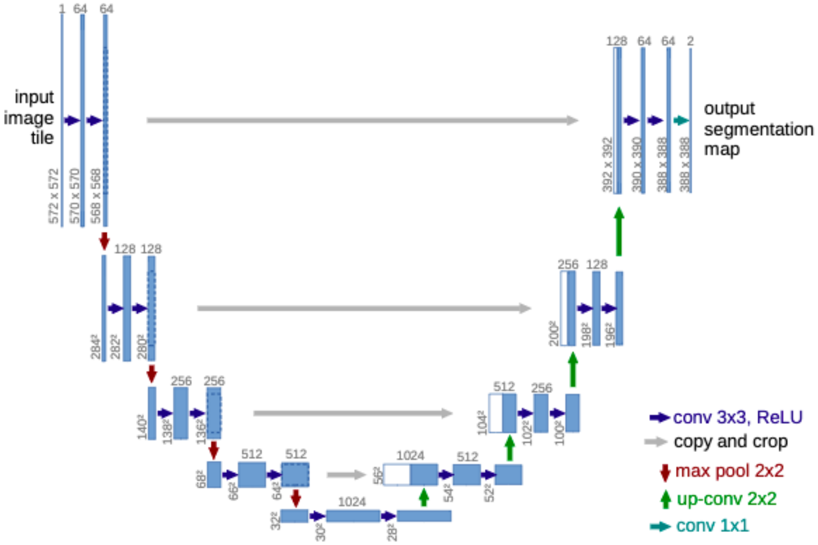
**Input**

| | |
|---|---|
| 3 | 2 |
| 2 | 1 |

**Output**

| | | | |
|---|---|---|---|
| 3 | 3 | 2 | 2 |
| 3 | 3 | 2 | 2 |
| 2 | 2 | 1 | 1 |
| 2 | 2 | 1 | 1 |

# U-Net architecture



# Table of Contents

# Segmentation challenges

The class imbalance problem

Differentiation with perivascular spaces

# Segmentation challenges - class imbalance

Difficult to optimize the network

- Slice of 512$x$512 $=$ 262,144 pixels
- 36 to 900 pixels with lacune
- Over-classification of non-lacune pixels

# Segmentation challenges - class imbalance - strategy

Let $y_l^{(i)}$ be the $l$th pixel value of the ground truth image $Y^{(i)}$ and let $\hat{p}_l^{(i)}$ be the $l$th pixel value of the predicted probabilistic map $\hat{P}^{(i)}$.

**Weighted binary cross-entropy loss (WBCE)**[8]

$$WBCE = -\frac{1}{L}\sum_{l=1}^{L} w y_l^{(i)} \log\left(\hat{p}_l^{(i)}\right) + \left(1 - y_l^{(i)}\right)\log\left(1 - \hat{p}_l^{(i)}\right),$$

where $w$ represents a weight

[8] O. Ronneberger, P. Fisher, and T. Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Proceedings of the 18th Medical Image Computing and Computer Assisted Intervention*. Munich, Germany, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

# Segmentation challenges - class imbalance - strategy

Let $y_l^{(i)}$ be the $l$th pixel value of the ground truth image $Y^{(i)}$ and let $\hat{p}_l^{(i)}$ be the $l$th pixel value of the predicted probabilistic map $\hat{P}^{(i)}$.

**Dice loss (DL)**[9]

$$DL = 1 - \frac{\sum_{l=1}^{L} \hat{p}_l^{(i)} y_l^{(i)} + \epsilon}{\sum_{l=1}^{L} \hat{p}_l^{(i)} + y_l^{(i)} + \epsilon} - \frac{\sum_{l=1}^{L} \left(1 - \hat{p}_l^{(i)}\right)\left(1 - y_l^{(i)}\right) + \epsilon}{\sum_{l=1}^{L} 2 - \hat{p}_l^{(i)} - y_l^{(i)} + \epsilon},$$

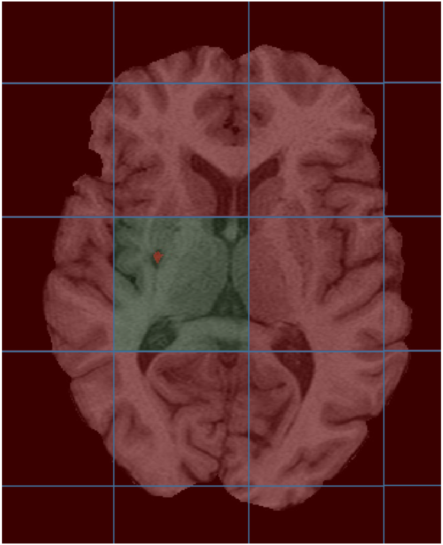
where $\epsilon$ is a small number

[9] F. Milletari, N. Navab, and S. Ahmadi (2016). "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *Proceedings of the 4th International Conference on 3D Vision*. Stanford, United States of America, pp. 565–571. DOI: 10.1109/3DV.2016.79.

# Segmentation challenges - class imbalance - strategy

Using patches instead of full image

- Divide image into patches
- Control ratio positive:negative samples



# Segmentation challenges - differentiation

**Similarities**

Lacunes and perivascular spaces

- Round or ovoid in shape
- Have the same intensity on
  - T1-weighted images
  - T2-weighted images

# Segmentation challenges - differentiation

## Differences

**Lacunes**

- $3 - 15$ mm in diameter
- Hyperintense rim on FLAIR image
- Spherical shape

**Perivascular spaces**

- $< 3$ mm in diameter
- No hyperintense rim on FLAIR image
- Elongated shape parallel to the course of the vessel

# Segmentation challenges - differentiation - strategy

Findings from previous papers on lacune detection/segmentation

- Perivascular spaces are often among false positives
- Location appears to be important in differentiation with perivascular spaces

Options to tackle the differentiation problem

- Put a constraint on the size of a candidate via the loss function
- Punish locations where lacunes cannot occur via the loss function
- Use different sized patches

# Table of Contents

# Conclusion

For the development of an automated lacune segmentation method

- U-Net might be a promising architecture
- Challenges we need to tackle
    - The class imbalance problem
        - Weighted cross-entropy loss
        - Dice loss
        - Using patches
    - Differentiation with perivascular spaces
        - Constraining the size
        - Punish locations
        - Using different sized patches

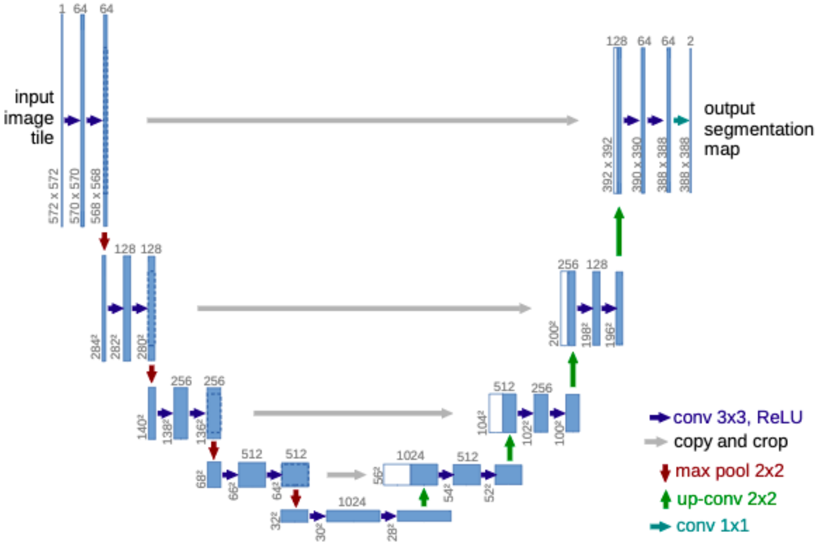
# Table of Contents

# Research proposal

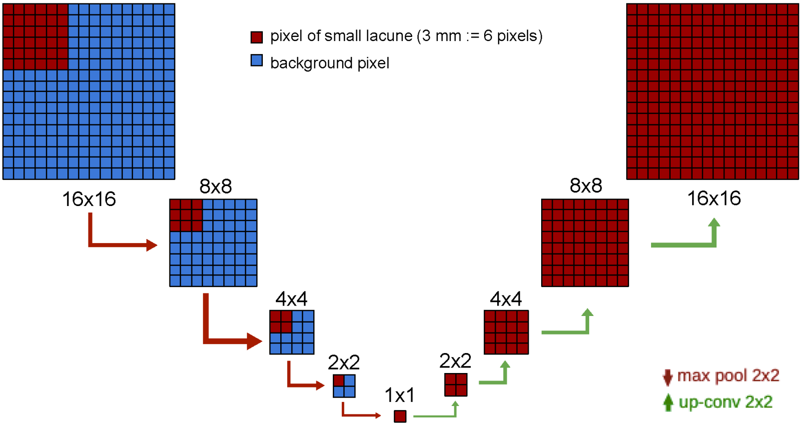
**Main research question:**

How can we develop an automated method that is able to segment lacunes of presumed vascular origin in brain MRI scans?

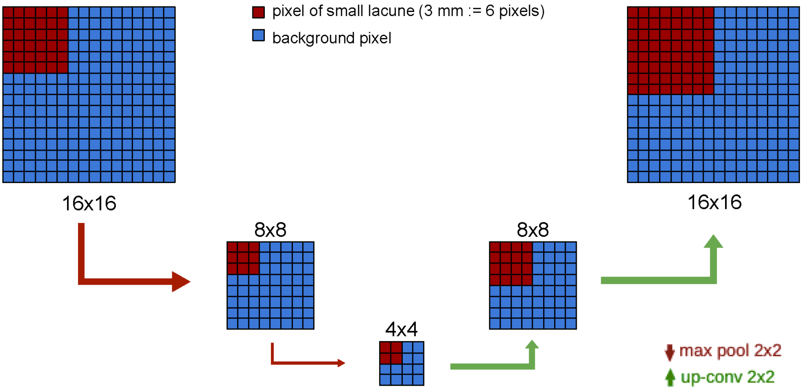
# Research proposal



# Research proposal



# Research proposal



# Research proposal

**Main research question:**

How can we develop an automated method that is able to segment lacunes of presumed vascular origin in brain MRI scans?

# Research proposal

**Supporting subquestions:**

Which options of general hyperparameters (e.g. activation function, optimizer) should be chosen to obtain the most accurate results?

Which approach should be used to tackle the data imbalance problem?

How can we make sure that the model is able to differentiate between lacunes and perivascular spaces?

Can we make the model applicable to another dataset as well?

# Research proposal

**Which options of general hyperparameters (e.g. activation function, optimizer) should be chosen to obtain the most accurate results?**

- Activation function: leaky ReLU, ELU
- Optimizer: Adam, AdamDelta

**Which approach should be used to tackle the data imbalance problem?**

- Weighted cross-entropy loss
- Dice loss
- Using patches

# Research proposal

**How can we make sure that the model is able to differentiate between lacunes and perivascular spaces?**

- Constraining the size via the loss function
- Punishing locations were lacunes cannot occur via the loss function
- Using different sized patches

**Can we make the model applicable to another dataset as well?**

- Haert-Brain Connection dataset

# Research proposal

**Timeline**

- Applying the U-Net architecture to the Rotterdam scan study data
- Fine-tune the network w.r.t. hyperparameters (e.g. activation function, optimizers)
- Tackle the data imbalance and differentiation problems
- Apply the method to another dataset