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

Eline Ooms

TU Delft & Erasmus MC, The Netherlands

May 14, 2021

Table of contents

Table of Contents

Introduction - cerebral small vessel disease

cerebral small vessel disease

:= changes in the brain due to damaged small vessels

Resulting lesions

- **Lacunes of presumed vascular origin**
- Recent small subcortical infarcts
- White matter hyperintensities
- Perivascular spaces
- Cerebral microbleeds

Introduction - lacunes of presumed vascular origin[1]

Definition

a round or ovoid, fluid-filled cavity of between 3 mm and about 15 mm in diameter

Figure: Example of a lacune.

[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.

Introduction - relevance

Relevance of finding lacunes

- Helps to detect the disease
- Can give more information about the disease

Relevance of an automated method

- Helps speeding up the process

Figure: Example of a lacune.

Introduction - segmentation

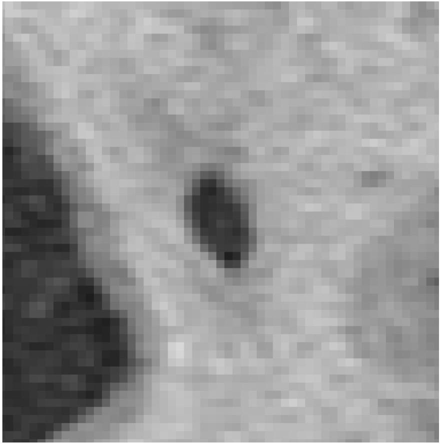


Figure: Example of a lacune

Introduction - segmentation

Segmentation methods

- Label every voxel

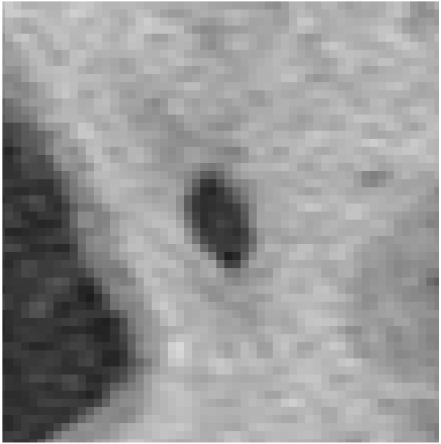


Figure: Example of a lacune

Introduction - segmentation

Segmentation methods

- Label every voxel

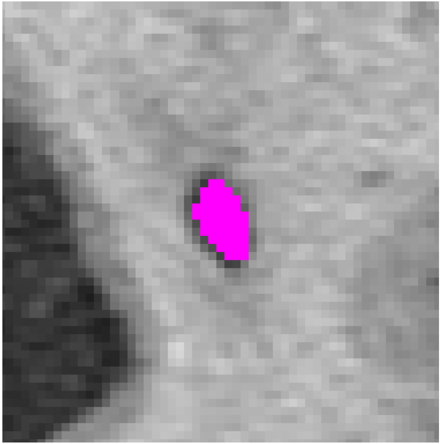


Figure: Example of a segmented lacune

Introduction - segmentation

Segmentation methods

- Label every voxel

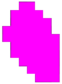


Figure: Lacune segment.

Introduction - segmentation

Segmentation methods

- Label every voxel

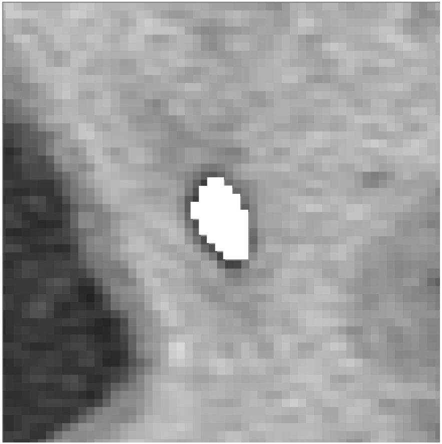


Figure: Background segment.

Introduction - segmentation

Gives information about

- Location
- Shape
- Size

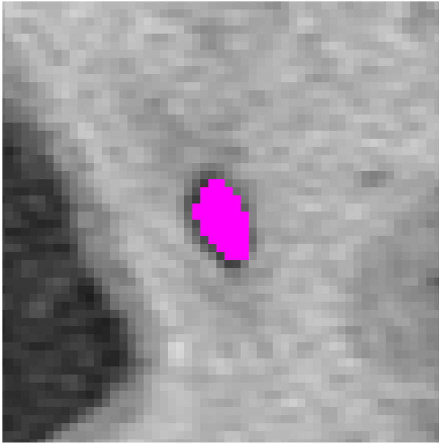


Figure: Example of a segmented lacune

Introduction - contribution

Previous lacune methods

- Only on 2D images or 3D sub-images
- Analyzing entire 3D images with these previous methods requires
  - more time
  - more computational cost
  - additional manual labour

Our methods

- detect and segment lacunes in 3D MRI images at once

Introduction - challenges

Class imbalance

Differentiation with similarly looking structures

Introduction - challenges - class imbalance

Imbalance between lacune and background voxels

- Scan of $512x512x192 =$ 50,331,648 voxels
- 74 to 9200 voxels with lacune
- Over-classifying the background

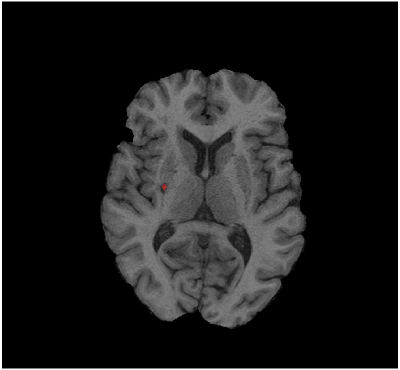


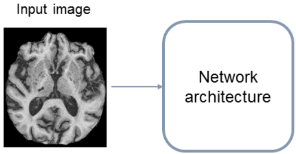
Figure: Example of a lacune.

Introduction - challenges - differentiation

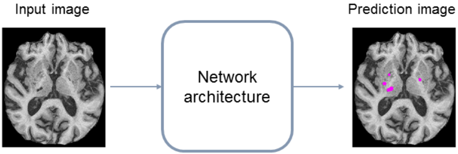
Differentiate lacunes from (parts of) brain structures with similar

- shape
- size
- intensity

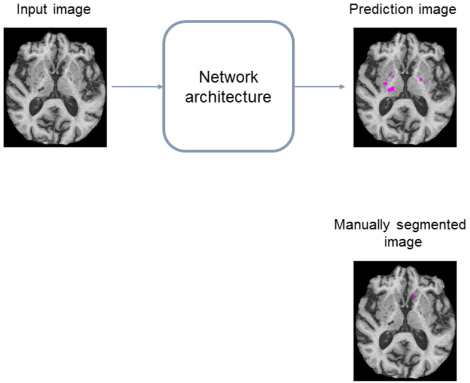
Introduction - components of a method



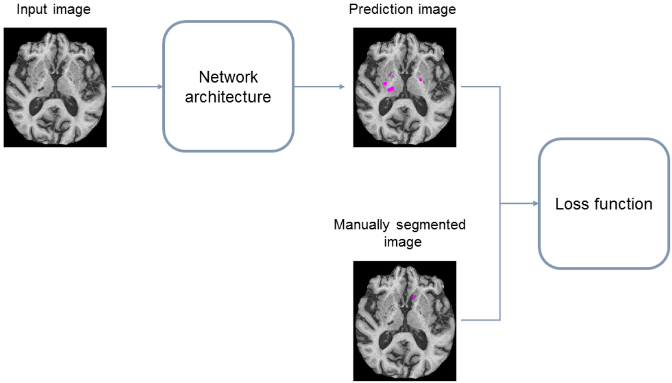
Introduction - components of a method



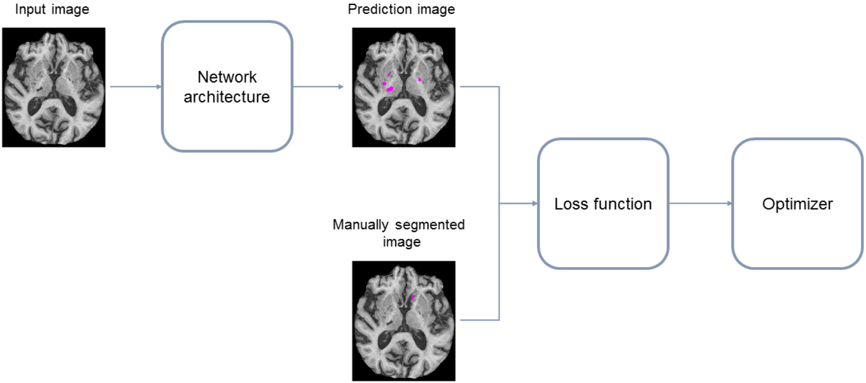
Introduction - components of a method



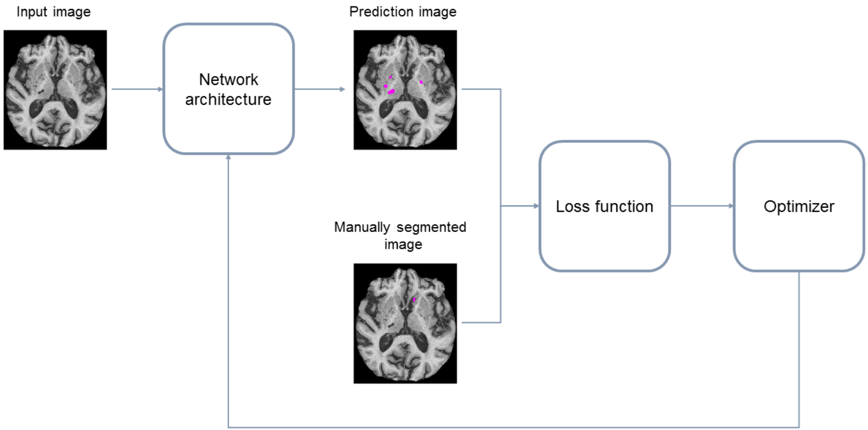
Introduction - components of a method



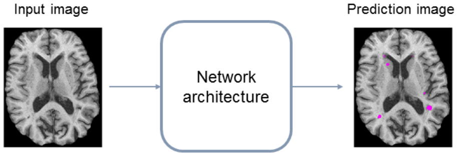
Introduction - components of a method



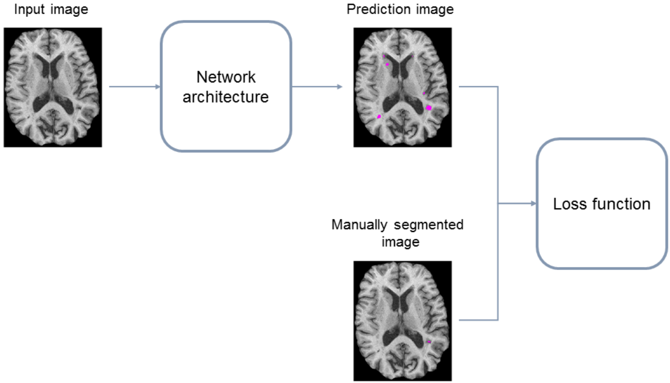
Introduction - components of a method



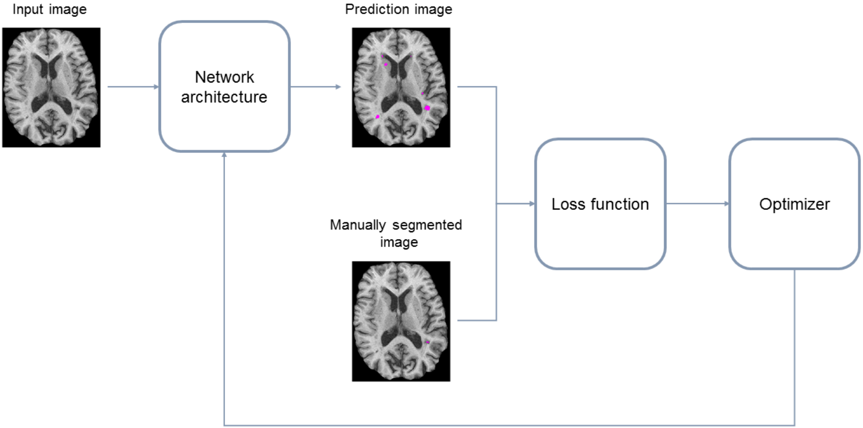
Introduction - components of a method



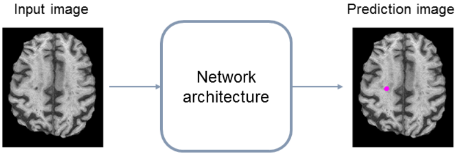
Introduction - components of a method



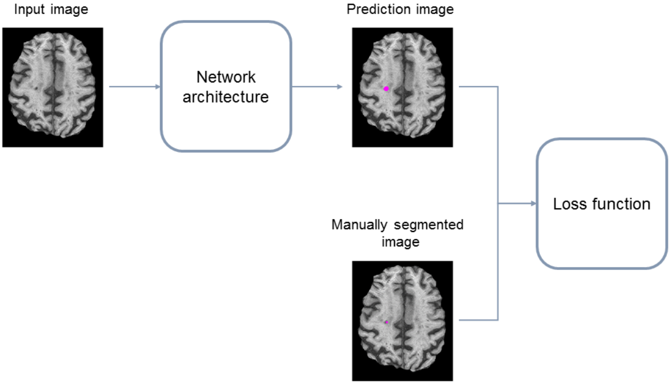
Introduction - components of a method



Introduction - components of a method



Introduction - components of a method



Introduction - components of a method

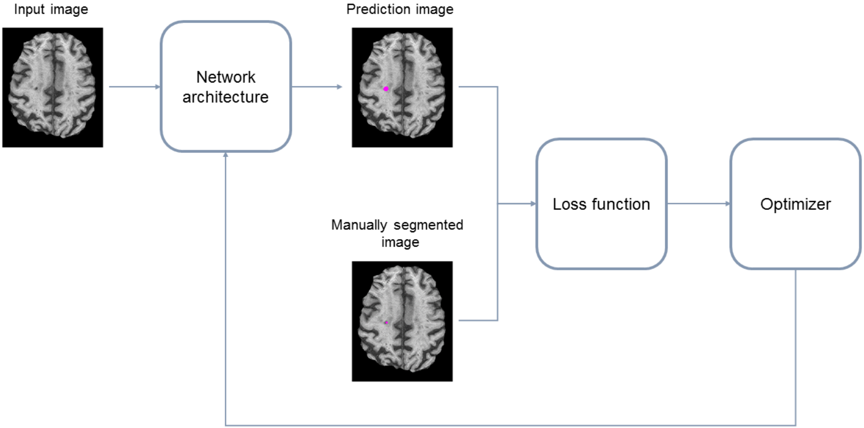


Table of Contents

Data - Rotterdam scan study

Scans

- 222 manually segmented lacune scans
- Image size of 512x512x192 voxels
- T1-weighted MRI images

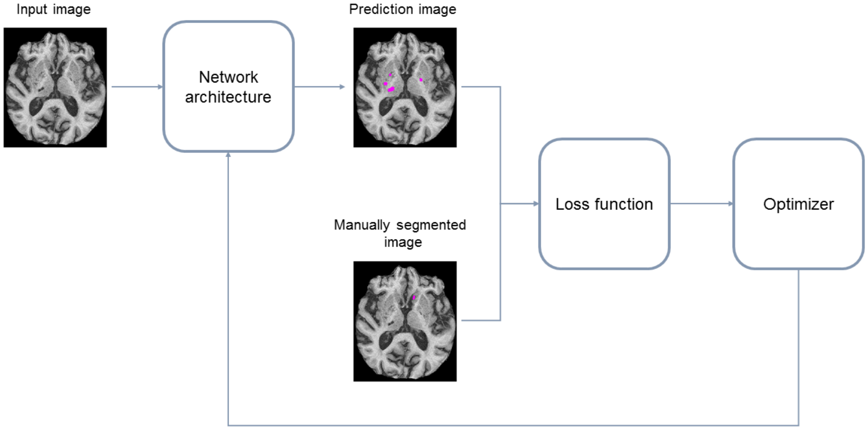
Data split

- 89 images for training
- 22 images for validation
- 111 images for testing

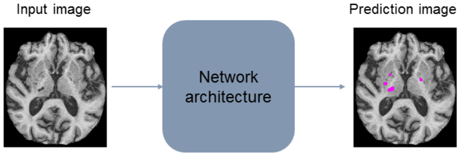
Figure: Lacune on a Rotterdam scan study scan.

Table of Contents

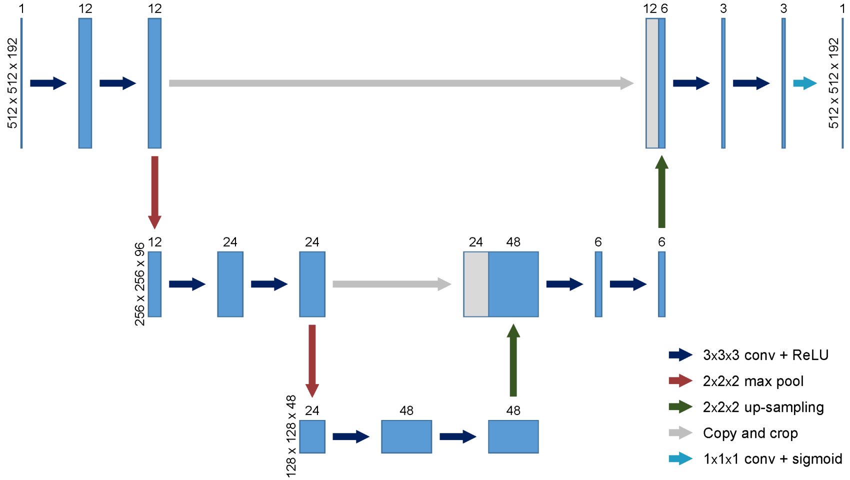
Method - network architecture



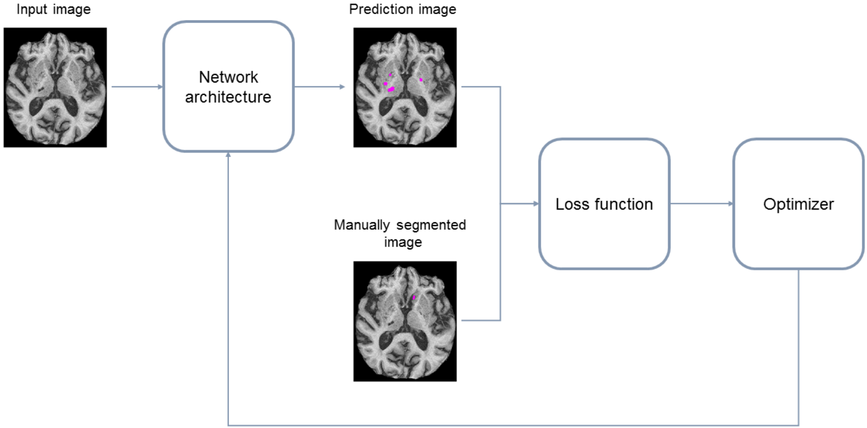
Method - network architecture



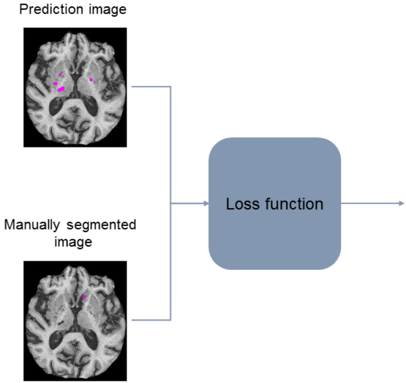
Method - network architecture



Method - loss function



Method - loss function



Method - loss function

To cope with the challenges, 5 loss functions are compared

- Binary cross-entropy loss
- Weighted binary cross-entropy loss
- Dice loss
- Dice-ReLU loss
- Constrained Dice-ReLU loss

Method - loss function

To cope with the challenges, 5 loss functions are compared

- Binary cross-entropy loss
- **Weighted binary cross-entropy loss**
- **Dice loss**
- **Dice-ReLU loss**
- Constrained Dice-ReLU loss

Method - loss function

To cope with the challenges, 5 loss functions are compared

- Binary cross-entropy loss
- Weighted binary cross-entropy loss
- Dice loss
- Dice-ReLU loss
- **Constrained Dice-ReLU loss**

Method - loss function

Manually segmented image

$$Y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Prediction image

$$\hat{P} = \begin{bmatrix} 0 & 0.10 & 0.13 & 0.25 & 0.15 & 0.06 \\ 0.09 & 0.14 & 0.34 & \mathbf{0.89} & 0.55 & 0.28 \\ 0.21 & 0.44 & \mathbf{0.83} & \mathbf{1} & \mathbf{0.86} & 0.36 \\ 0.32 & \mathbf{0.76} & \mathbf{0.96} & \mathbf{1} & \mathbf{0.91} & 0.34 \\ 0.16 & 0.47 & \mathbf{0.78} & \mathbf{0.93} & 0.38 & 0.22 \\ 0.04 & 0.21 & 0.29 & 0.36 & 0.18 & 0 \end{bmatrix}$$

Method - loss function - binary cross-entropy loss

n = number of voxels in an image
$y_k = k^{th}$ voxel value of the manually segmented image
$\hat{p}_k = k^{th}$ voxel value of the prediction image

$$BCE = -\frac{1}{n}\sum_{k=1}^{n}\left(y_k \log\left(\hat{p}_k\right) + \left(1 - y_k\right)\log\left(1 - \hat{p}_k\right)\right)$$

Method - loss function - binary cross-entropy loss

n = number of voxels in an image
$y_k = k^{th}$ voxel value of the manually segmented image
$\hat{p}_k = k^{th}$ voxel value of the prediction image

$$BCE\ loss = -\frac{1}{n}\sum_{k=1}^{n}\left((y_k \boldsymbol{\log(\hat{p}_k)} + (1-y_k)\log(1-\hat{p}_k)\right)$$

Method - loss function - binary cross-entropy loss

n = number of voxels in an image
y_k = k^{th} voxel value of the manually segmented image
$\hat{p}_k$ = k^{th} voxel value of the prediction image

$$BCE\ loss = -\frac{1}{n}\sum_{k=1}^{n}\left((y_k \log(\hat{p}_k) + (1 - y_k)\,\mathbf{log}\,(\mathbf{1} - \boldsymbol{\hat{p}_k})\right)$$

Method - loss function - binary cross-entropy loss

n = total number of voxels in an image
y_k = k^{th} voxel value of the manually segmented image
$\hat{p}_k$ = k^{th} voxel value of the prediction image

$$BCE\ loss = -\frac{1}{n}\sum_{k=1}^{n}\left(y_k \log\left(\hat{p}_k\right) + \left(1 - y_k\right)\log\left(1 - \hat{p}_k\right)\right)$$

Method - loss function - weighted binary cross-entropy loss

q = number of lacune voxels in the manually segmented image
r = number of background voxels in the manually segmented image
$\hat{p}_s$ = s^{th} prediction voxel that should be a lacune
$\hat{p}_t$ = t^{th} prediction voxel that should be background

$$WBCE\ loss = -\frac{1}{2}\left(\frac{1}{q}\sum_{s=1}^{q}\log\left(\hat{p}_s\right) + \frac{1}{r}\sum_{t=1}^{r}\log\left(1-\hat{p}_t\right)\right)$$

Method - loss function - weighted binary cross-entropy loss

q = number of lacune voxels in the manually segmented image
r = number of background voxels in the manually segmented image
$\hat{p}_s = s^{th}$ prediction voxel that should be a lacune
$\hat{p}_t = t^{th}$ prediction voxel that should be background

$$\mathit{WBCE\ loss} = -\frac{1}{2}\left(\boldsymbol{\frac{1}{q}\sum_{s=1}^{q}\log\left(\hat{p}_s\right)} + \frac{1}{r}\sum_{t=1}^{r}\log\left(1-\hat{p}_t\right)\right)$$

Method - loss function - weighted binary cross-entropy loss

q = number of lacune voxels in the manually segmented image
r = number of background voxels in the manually segmented image
$\hat{p}_s = s^{th}$ prediction voxel that should be a lacune
$\hat{p}_t = t^{th}$ prediction voxel that should be background

$$WBCE\ loss = -\frac{1}{2}\left(\frac{1}{q}\sum_{s=1}^{q}\log\left(\hat{p}_s\right) + \boldsymbol{\frac{1}{r}\sum_{t=1}^{r}\log\left(1-\hat{p}_t\right)}\right)$$

Method - loss function - weighted binary cross-entropy loss

q = number of lacune voxels in the manually segmented image
r = number of background voxels in the manually segmented image
$\hat{p}_s = s^{th}$ prediction voxel that should be a lacune
$\hat{p}_t = t^{th}$ prediction voxel that should be background

$$WBCE\ loss = -\mathbf{\frac{1}{2}}\left(\frac{1}{q}\sum_{s=1}^{q}\log\left(\hat{p}_s\right) + \frac{1}{r}\sum_{t=1}^{r}\log\left(1-\hat{p}_t\right)\right)$$

Method - loss function - Dice loss

The Dice loss is derived from the Dice similarity coefficient(DSC)

$$DSC = \frac{2|Y \cap P|}{|Y| + |P|},$$

where $|Y|$ is the cardinality of set Y and $|P|$ is the cardinality of set P

Method - loss function - Dice loss

n = number of voxels in an image
$y_k = k^{th}$ voxel value of the manually segmented image
$\hat{p}_k = k^{th}$ voxel value of the prediction image

$$Dice\ loss = 1 - \frac{2\sum_{k=1}^{n} y_k \hat{p}_k}{\sum_{k=1}^{n} y_k + \sum_{k=1}^{n} \hat{p}_k + \epsilon}$$

Method - loss function - Dice loss

n = number of voxels in an image
y_k = k^{th} voxel value of the manually segmented image
$\hat{p}_k$ = k^{th} voxel value of the prediction image

$$Dice\ loss = 1 - \frac{2\sum_{k=1}^{n} y_k \hat{p}_k}{\sum_{k=1}^{n} y_k + \sum_{k=1}^{n} \hat{p}_k + \epsilon}$$

Method - loss function - Dice loss

n = number of voxels in an image
$y_k = k^{th}$ voxel value of the manually segmented image
$\hat{p}_k = k^{th}$ voxel value of the prediction image

$$Dice\ loss = 1 - \frac{2\sum_{k=1}^{n} y_k \hat{p}_k}{\boldsymbol{\sum_{k=1}^{n} y_k} + \sum_{k=1}^{n} \hat{p}_k + \epsilon}$$

Method - loss function - Dice loss

n = number of voxels in an image
$y_k = k^{th}$ voxel value of the manually segmented image
$\hat{p}_k = k^{th}$ voxel value of the prediction image

$$Dice\ loss = 1 - \frac{2\sum_{k=1}^{n} y_k \hat{p}_k}{\sum_{k=1}^{n} y_k + \boldsymbol{\sum_{k=1}^{n} \hat{p}_k} + \epsilon}$$

Method - loss function - Dice loss

n = number of voxels in an image
y_k = k^{th} voxel value of the manually segmented image
$\hat{p}_k$ = k^{th} voxel value of the prediction image

$$Dice\ loss = 1 - \frac{2\sum_{k=1}^{n} y_k \hat{p}_k}{\sum_{k=1}^{n} y_k + \sum_{k=1}^{n} \hat{p}_k + \epsilon}$$

Method - loss function - Dice-ReLU loss

Clip prediction values with a shifted ReLU function

$$f\left(\hat{p}_k\right) = \max\left(0.1, \hat{p}_k\right),$$

where $\hat{p}_k$ is the k^{th} voxel value of the prediction image

Method - loss function - Dice-ReLU loss

n = number of voxels in an image
$y_k = k^{th}$ voxel value of the manually segmented image
$\hat{p}_k = k^{th}$ voxel value of the prediction image

$$\textit{Dice-ReLU loss} = 1 - \frac{2\sum_{k=1}^{n} y_k \max(0.1, \hat{p}_k)}{\sum_{k=1}^{n} y_k + \sum_{k=1}^{n} \max(0.1, \hat{p}_k)}$$

Method - loss function - constrained Dice-ReLU loss

C_B = constraint on the volume of the prediction voxels that should be predicted as background

C_L = constraint on the volume of the prediction voxels that should be predicted as lacunes

μ = parameter defining the contribution of the constraint to the loss

$$\mathit{CDR\ loss} = \mathit{Dice\text{-}ReLU\ loss} + \mu\,(C_B + C_L)$$

Method - loss function - constrained Dice-ReLU loss

C_B = constraint on the volume of the prediction voxels that should be predicted as background

C_L = constraint on the volume of the prediction voxels that should be predicted as lacunes

μ = parameter defining the contribution of the constraint to the loss

$$\textit{CDR loss} = \textit{Dice-ReLU loss} + \mu\left(\boldsymbol{C_B} + C_L\right)$$

Method - loss function - constrained Dice-ReLU loss

$$V_B = \text{volume of the prediction voxels that should be predicted as background}$$

$$V_T = \text{volume of the manually segmented lacune voxels}$$

$$FP_{start} = \text{start volume of the prediction voxels that should be predicted as background}$$

Background constraint

$$C_B\left(V_B, V_T\right) = \begin{cases} \frac{\left(V_B - 0.25 V_T\right)^2}{\left(FP_{start} - 0.25 V_T\right)^2} & \text{if } V_B > 0.25 V_T, \\ 0 & \text{otherwise} \end{cases}$$

Method - loss function - constrained Dice-ReLU loss

C_B = constraint on the volume of the prediction voxels that should be predicted as background

C_L = constraint on the volume of the prediction voxels that should be predicted as lacunes

μ = parameter defining the contribution of the constraint to the loss

$$\textit{CDR loss} = \textit{Dice-ReLU loss} + \mu\,(C_B + \boldsymbol{C_L})$$

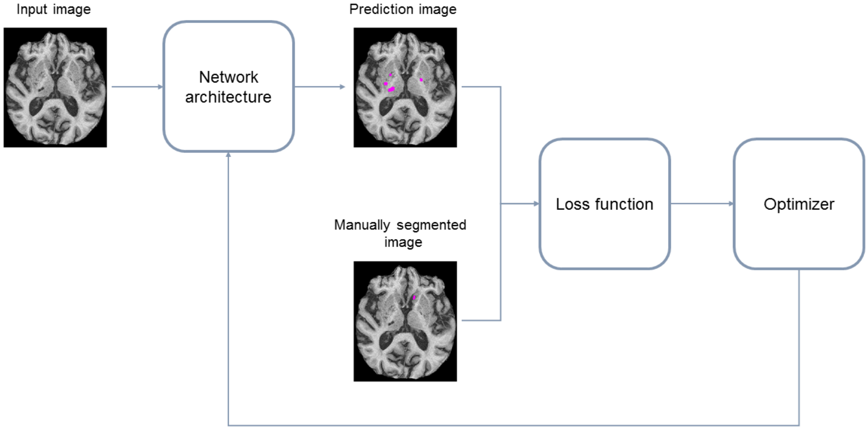
Method - loss function - constrained Dice-ReLU loss

V_L = volume of the prediction voxels that should be predicted as lacune
V_T = volume of the manually segmented lacune voxels

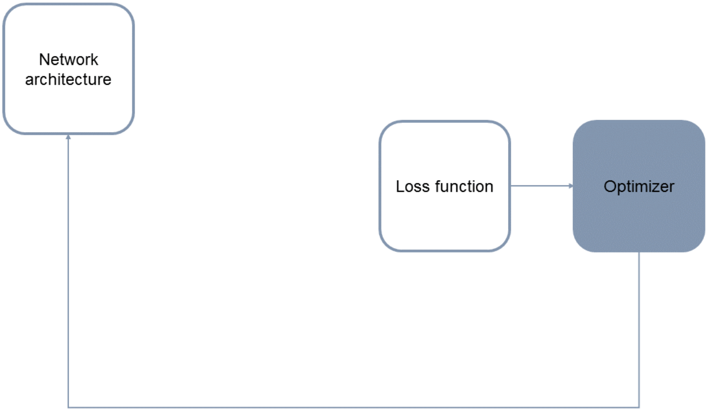
Lacune constraint

$$C_L(V_L, V_T) = \begin{cases} \frac{(V_L - 0.75 V_T)^2}{(0.75 V_T)^2} & \text{if } V_L < 0.75 V_T, \\ 0 & \text{otherwise} \end{cases}$$

Method - optimizer



Method - optimizer



Method - optimizer

AdaDelta

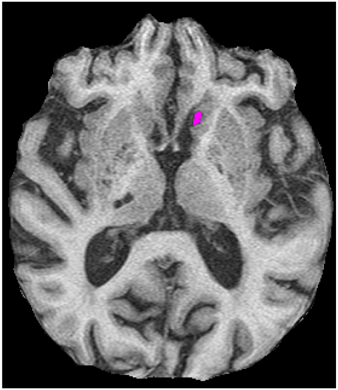
- Binary cross-entropy loss
- Weighted binary cross-entropy loss

Adam

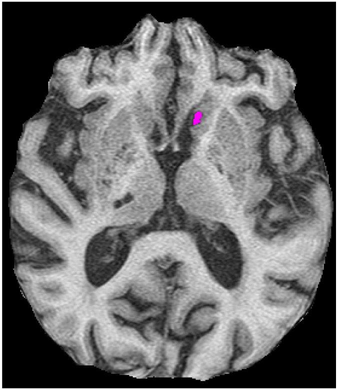
- Dice loss
- Dice-ReLU loss
- Constrained Dice-ReLU loss

Table of Contents

Results - terminology - true positive

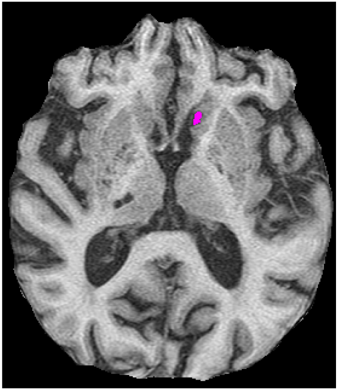


(a) Manually segmented image.

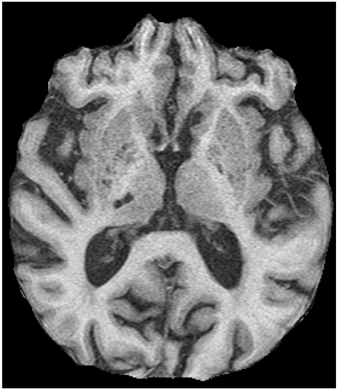


(b) Prediction image.

Results - terminology - false negative

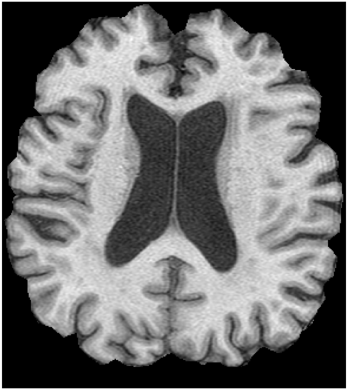


(a) Manually segmented image.

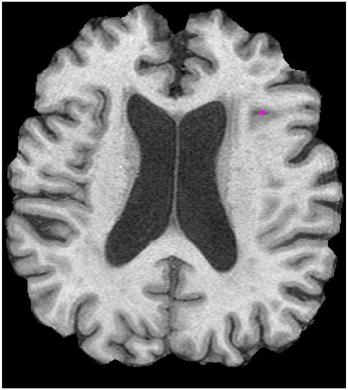


(b) Prediction image.

Results - terminology - false positive

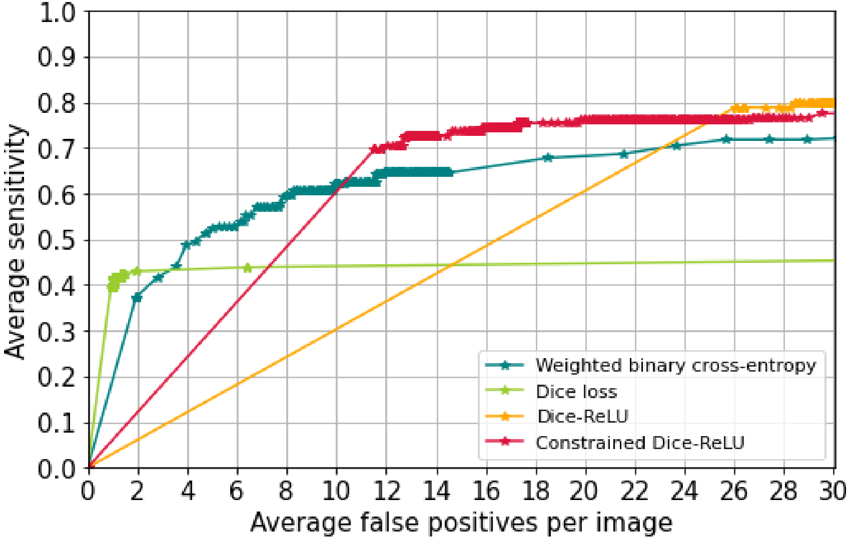


(a) Manually segmented image.



(b) Prediction image.

Results - detection performance



Results - overall segmentation performance

Loss function	DSC (mean ± STD)	Relative volume difference (mean ± STD)	Absolute volume difference (mean ± STD)
BCE	-	-	-
WBCE	0.14 ± 0.19	1.07 ± 1.37	243.10 ± 395.77
Dice	**0.19** ± **0.25**	**0.89** ± **1.69**	**204.01** ± **370.38**
Dice-ReLU	0.05 ± 0.05	42.28 ± 43.78	5409.05 ± 2445.62
CDR	0.08 ± 0.08	18.28 ± 20.59	2424.08 ± 1575.61

BCE = binary cross-entropy, WBCE = weighted binary cross-entropy, CDR = constrained Dice-ReLU

Results - segmentation performance of TP elements

Loss function	DSC (mean ± STD)	Relative volume difference (mean ± STD)	Absolute volume difference (mean ± STD)
BCE	-	-	-
WBCE	0.45 ± 0.21	0.75 ± 0.61	**106.74** ± **87.05**
Dice	**0.47** ± **0.23**	**0.49** ± **0.30**	132.88 ± 314.79
Dice-ReLU	0.28 ± 0.15	5.93 ± 5.65	738.66 ± 509.36
CDR	0.29 ± 0.14	4.77 ± 4.41	601.65 ± 428.37

BCE = binary cross-entropy, WBCE = weighted binary cross-entropy, CDR = constrained Dice-ReLU

Results - example of a true positive

(a) Unsegmented.

(b) Manually.

(c) WBCE loss.

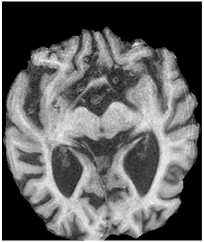
(d) Dice loss.

(e) Dice-ReLU loss.

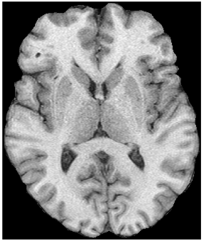
(f) CDR loss.

Results - examples of a false negative

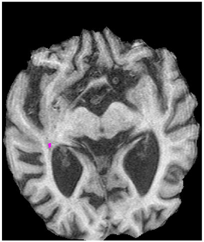

(a) Near ventricle.

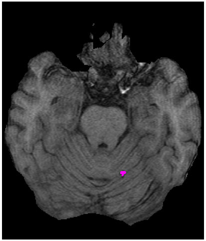

(b) Intensity.

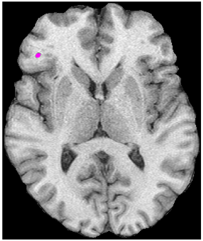

(c) In cerebellum.


(d) Outer part.

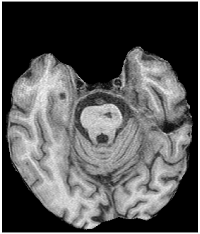

(e) Near ventricle.


(f) Intensity.


(g) In cerebellum.

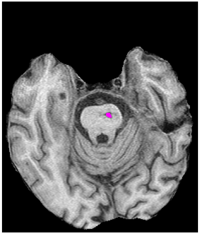

(h) Outer part.

Results - examples of a false negative

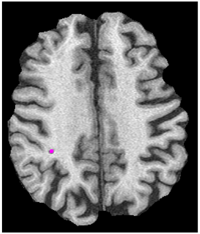


(a) In brainstem.

(b) Upper part.



(c) In brainstem.



(d) Upper part.

Results - examples of a false positive

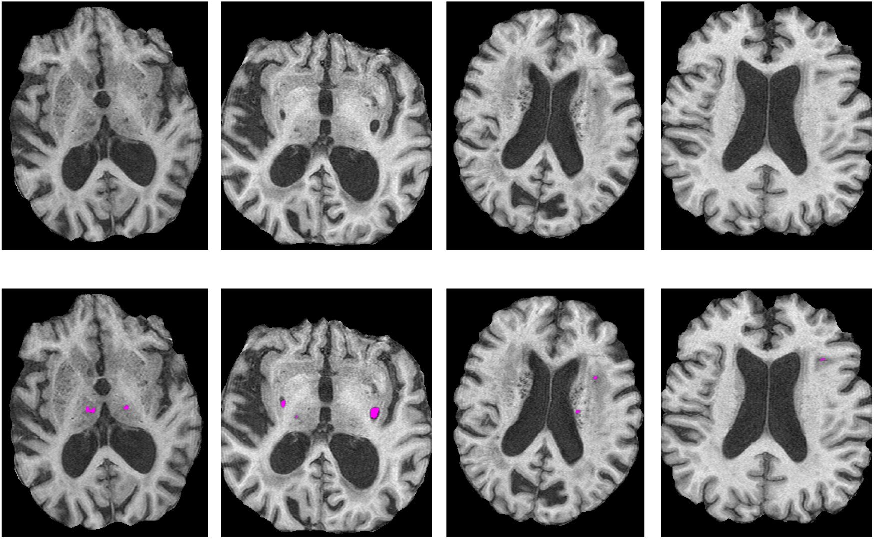


Table of Contents

Conclusion

Conclusions

- All loss functions, except the BCE loss were able to detect and segment lacunes
- Dice loss performed best on the number of FPs per image and on both segmentation performances, but worse on sensitivity performance.
- Clipping background values (Dice-ReLU loss) improved the sensitivity performance, but with many FPs.
- Adding a constraint (constrained Dice-ReLU loss) halved the number of FPs with only a limited decrease in sensitivity.

Conclusion

Aim

- Develop a deep learning method that is able to detect and segment lacunes in 3D brain MRI scans

Challenges

- Data imbalance
- Differentiation with similarly looking structures

Conclusion

Final conclusion

- All loss functions can cope with the data imbalance
- Clipping background values in the Dice loss (Dice-ReLU loss) helps in coping with the data imbalance
- Adding a constraint improves the differentiation with similarly looking structures
- The Dice-ReLU loss and the CDR loss are suitable for detecting cerebral small vessel disease
- The WBCE loss and the Dice loss are suitable for gaining more information into the cerebral small vessel disease

Table of Contents

Future work

- Add scans without lacunes
- Add FLAIR images
- Fine-tuning of constraint
- Constrain Dice loss to keep lacunes
- Constrain weighted binary cross-entropy to reduce false positives

Figure: Lacune on a FLAIR image