Image of Brain Stroke Lesion segmented with X-Net and Quaternion Neural Network

Master's degree in Artificial Intelligence and Robotics

SAPIENZA
UNIVERSITÀ DI ROMA

Authors:
Denise Landini - 1938388
Alessandro Lambertini - 1938390

Course: Neural Network
Professor: Aurelio Uncini

Academic Year: 2020-2021

Pipeline followed

Pre-processing

Split

Models

Evaluation

Model 1:
X-Net

Model 2:
X-Net with quaternions

Dataset: Brain CT images with Interacranial Haemorrhage Masks

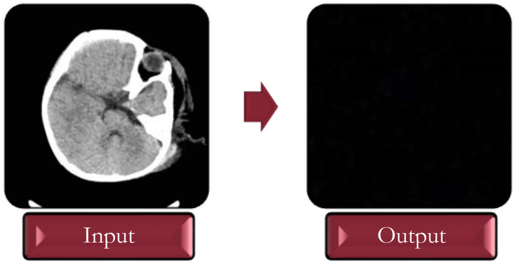
Pre-processing and split steps

The **dataset** is organized as follow:

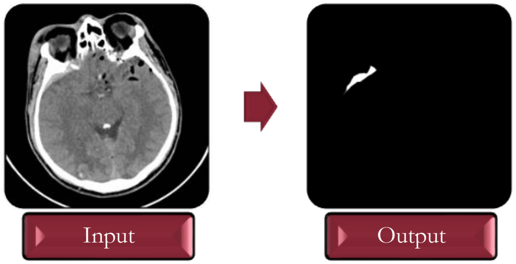
- Number of patients: **82**
- For each patient there are a few images that are related to different parts of the brain.
- Total images: **2501**

In **pre-processing step**, we obtain:

If the input image is a part of the brain where a stroke lesion wasn't found we obtain a black image.



If the input image is a part of the brain where a stroke lesion was found we obtain an image in which will be represented the stroke lesion.



In **split step** we consider that is not correct to separate completely randomly the images that we have. So, we have splitted the 82 patients. randomly. The data are splitted in 80% train set and 20% test set.

Pre-processing → Split → Models → Evaluation

Model 1: X-Net

Complete architecture implemented

The architecture is composed by:

- **Encoder**:
    - X-blocks have number of filters that vary at each step: 64, 128, 256, 512.
    - Maxpooling between X-blocks is always 2×2.
    - X-blocks 1024

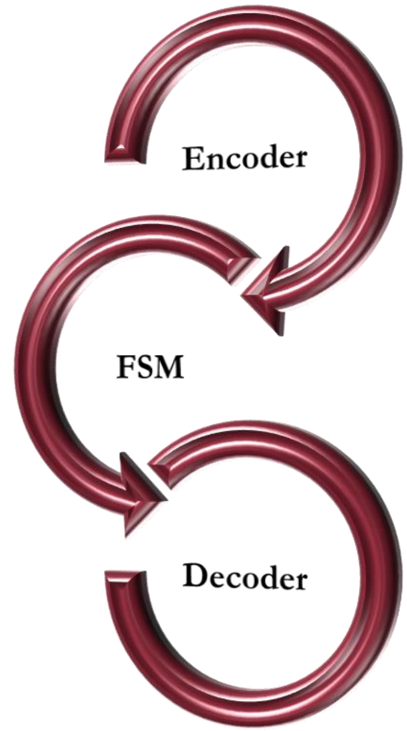
    Aim: produce high-dimensions feature maps.

- **FSM** (Feature Similarity Module):
    - Non-local operation
    - Connects the Encoder to the Decoder

    Aim: used to capture long-range spatial information

- **Decoder**:
    - Upsampling 2×2 and Convolution 3×3
    - Concatenate the previous block in the Encoder
    - X-blocks have number of filters that vary at each step: 512, 256, 128, 64
    - Convolution 1×1 with "sigmoid" as activation function.

    Aim: recover the spatial resolution.



Model 1: X-Net

Motivation of methods used

- We use **Maxpooling** because is a method of reducing the size of an image by splitting it into blocks and keeps only the one with the highest value. Doing this we reduce the problem of overfitting and only the area with grater activation are maintained.
- We use **UpSampling** to double the dimensions of input.
- The C**oncatenation** is used to improve segmenting performance.

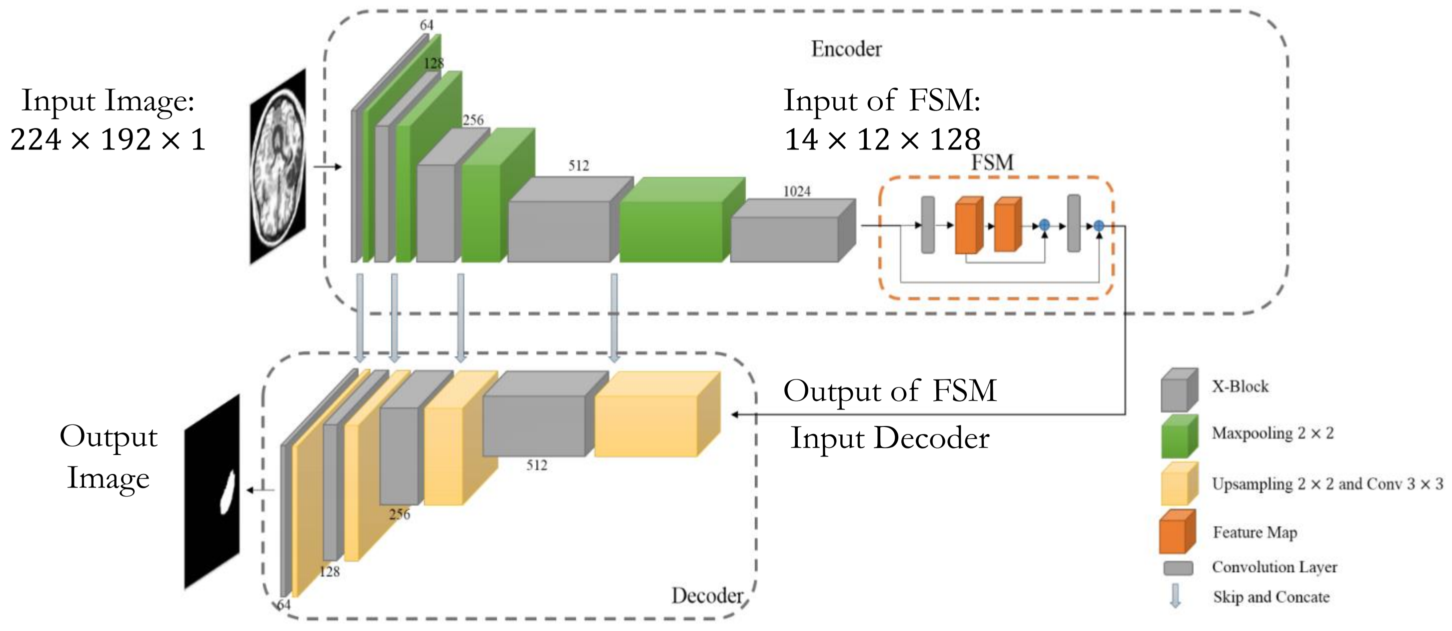


Figure: X-Net architecture

Model 1: X-Net

X-block structure

For each convolutional layer, both in the encoder and decoder, we use:

- **Batch Normalization** that is a technique that allow to improve the velocity, performance and stability of a Neural Network. We use this in order to normalize the input.
- **Activation function: ReLU**. We use this to introduce non linearity in the network and also because the computation of the derivative is fast and its value is zero only if the input is less than zero.

In **X-block** we have two paths:

1. **First path**: the residual connection consist in 1×1 convolution layer to guarantee that the number of channels in output is equal to C_O.
2. **Second path**:
   - Three depthwise separable convolution layers in cascade with kernel size 3×3
   - Convolution of size 1×1

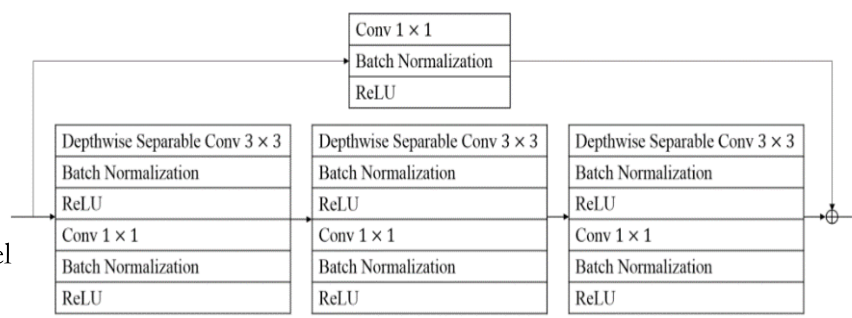
Pre-processing

Split

Model 1

Evaluation

Input feature map:
$I \in R^{H \times W \times C_i}$
C_i number of input channel



Output feature map:
$O \in R^{H \times W \times C_O}$
C_O number of output channel

Figure: X-block structure

Depthwise Separable Convolutions (DSC)

Convolution with DSC

The **process** of convolution is broken in two different operations:

1. Depthwise convolutions: Filtering Stage
In this operation, instead of done the convolution for all the M channel, the convolution is applied to a single channel at a time.

2. Pointwise convolutions: Combination Stage
It involves performing the linear combination of each of these layers.
In this operation, we apply on the M channels, an operation of 1×1 convolution.

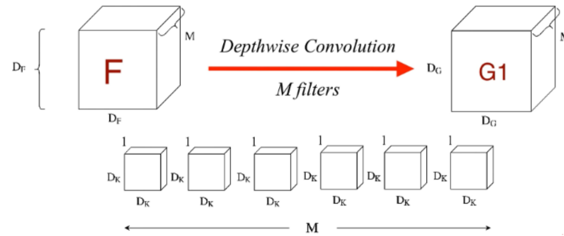


Figure 1: Depthwise convolutions

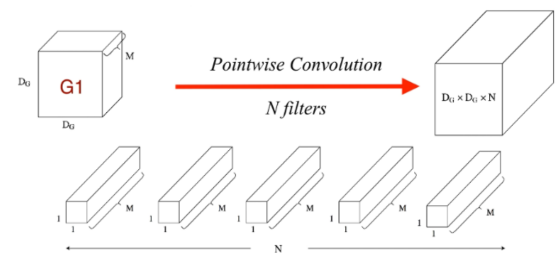


Figure 2: Pointwise convolutions

Advantages:
- Reduce the number of trainable parameters
- Reduce the computation time
- Manage to improve efficiency without reducing effectiveness

Disadvantage:
- Not used in small network due to the small number of trainable parameters

Pre-processing

Split

Model 1

Evaluation

Model 1: X-Net

FSM: Details

FSM is designed to explore dense context information for effective brain lesion segmentation through extracting long-range dependencies.

- Input feature map: $X_0 \in R^{H \times W \times C_0}$.
- Convolution 3×3 over I → B1
  Aim: used to filter out the irrelevant feature from the input I, because of this we obtain a feature map with depth $C < C_0$
- Three convolutions:
  - 1×1 → B2
  - 1×1 → B3
  - 1×1 → B4 indicates the representation of the input signal.
- B2 ⊗ B3 with softmax function → B5.
  Aim: relation map $f(x_i, x_j)$ which is the combination of dot product and softmax.
- B4 ⊗ B5 → B6
  B6 has a wide range of contextual view and aggregates the long-range context.
- Convolution 1×1 over B6, B1 ⊕ B6
- Convolution 3×3 over B6 → B7.
  Aim: obtain the initial dimension of the feature map.
- I ⊕ B7 → output of FSM (RIT)
  Aim: avoid overfitting.

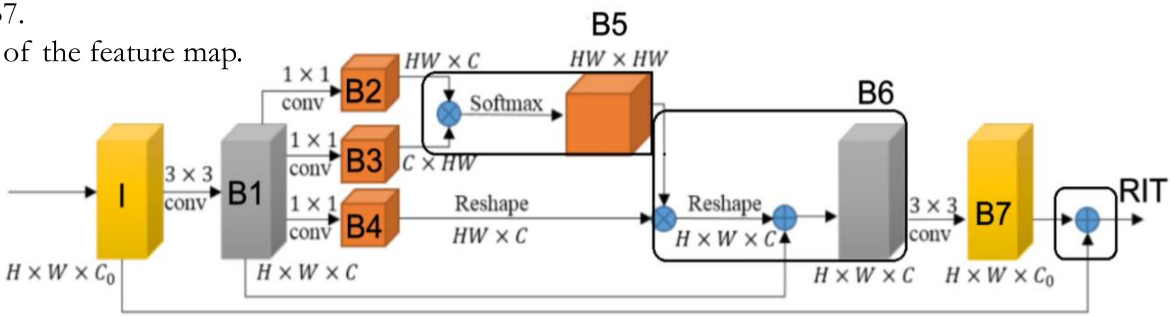


Figure: FSM architecture

Pre-processing
Split
Model 1
Evaluation

Model 2: X-Net with Quaternions

Introduction

Aim: process and build entities that are composed at maximum by four elements.

- Quaternions uses Hamilton product
- Can reconstruct the spatial relation within 3D coordinates and within colour pixels.

Advantages:

- QNN has a fourfold memory than NNs.
- QNN made efficient model in multidimensional space, because of the natural multidimensional representation of quaternions and their ability to reduce the number of parameters.
- QNN speed up computation

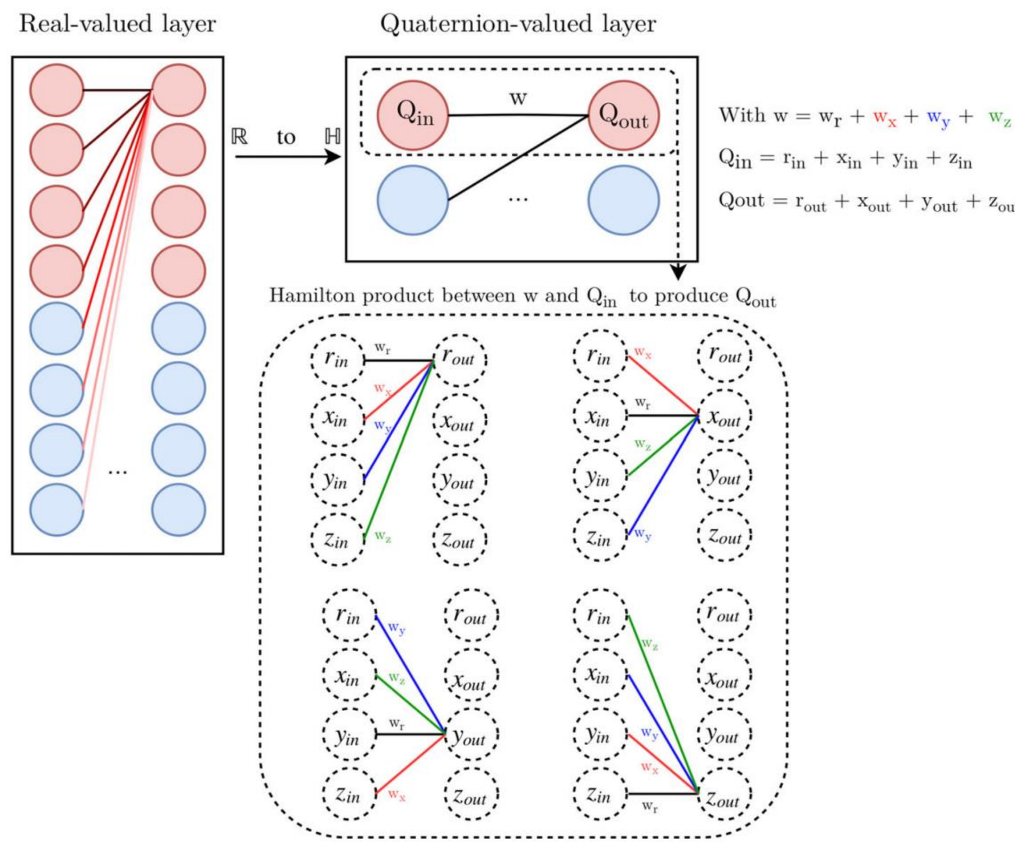


Figure: Quaternion architecture

Model 2: X-Net with Quaternions
Implementation

Pre-progessing step
The black and white image were represented in the quaternion domain in the three complex channels in which we have replicated the gray image, and the real channel was set to zero.
We have done this choice because we have images in black and white and not with colors.
If we have had the color images, we would set the primary color component for each channel.

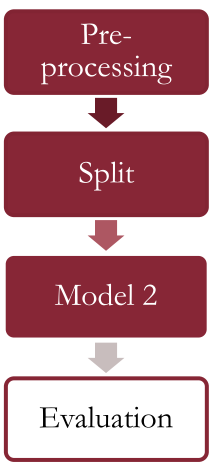
Split step
The data are splitted in 80% train set and 20% test set.

Architecture
The implementation of the X-Net is the same that we have done for the first model.
So, we have the **same structure** of:

- **X-block**
- **Encoder**
- **FSM**
- **Decoder**

The **crucial differences** are that:

- we have adapted the Depthwise Separable Convolution class in order to work with quaternions. In particular, we have substitute the Convolution with quaternions convolutions.
- in all this implementation we have added the forward function and we have modified the dimension of all the element in the network to fit the quaternion.



Evaluation with Model 1

X-Net: metrics and experiments

Metrics used:

- **Dice score:**

$$\frac{2 \cdot Area\ of\ Overlap}{total\ pixels\ combined}$$

- **Loss function:**

$$BinaryCrossEntropy + (1 - Dice)$$

Experiments done:

- Train of X-Net with and without Early Stopping
- Train of X-net with Early Stopping considering FSM and without considering FSM.

Pre-processing → Split → Train → Evaluation 1

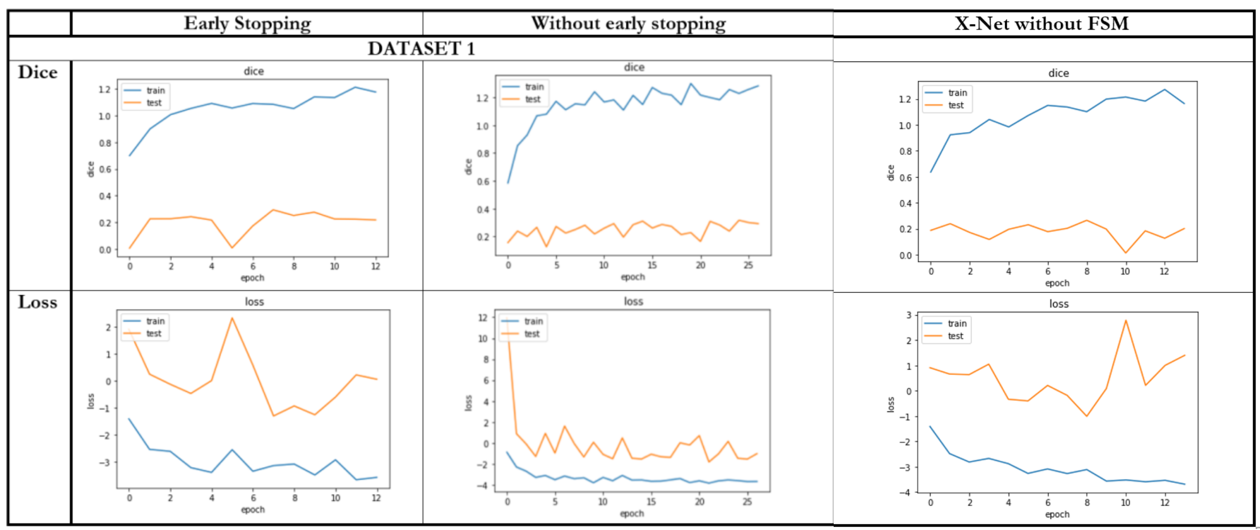


Figure: Comparisons

Evaluation with Model 1

X-Net: the importance of FSM for the output

If we cut the bone from the end image, we will have a perfectly segmentation of the lesion.
We can cut the bone because we have a folder in which are store the images.

Even if we cut the bone from the end image, we will not have a perfectly segmentation of the lesion because the problem is that we have a big area that are marked in our output and not only those of the bones

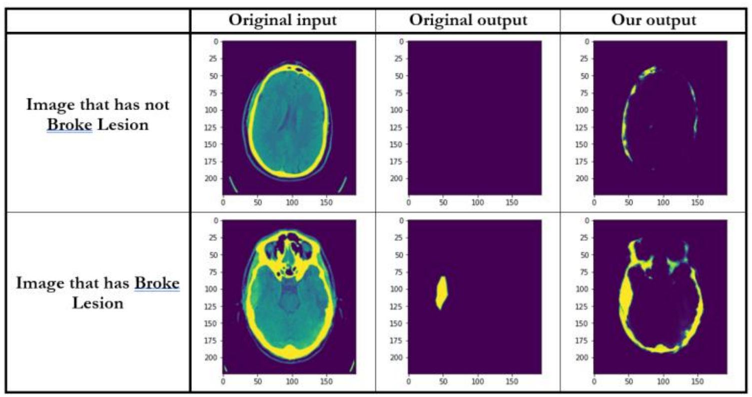


Figure 1: Random samples X-Net with FSM

BETTER

Problems

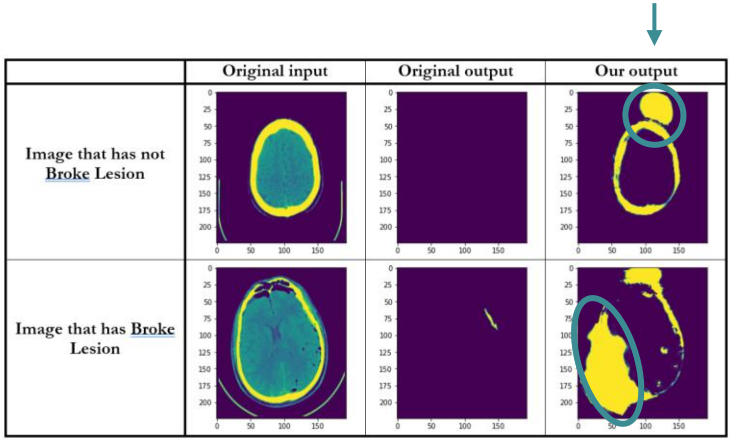


Figure 2: Random samples X-Net without FSM

The work of FSM is very important in order to avoid the problem in Figure 2.

Pre-processing
Split
Train
Evaluation 1

Evaluation with Model 2

X-Net with Quaternions: metrics and experiments

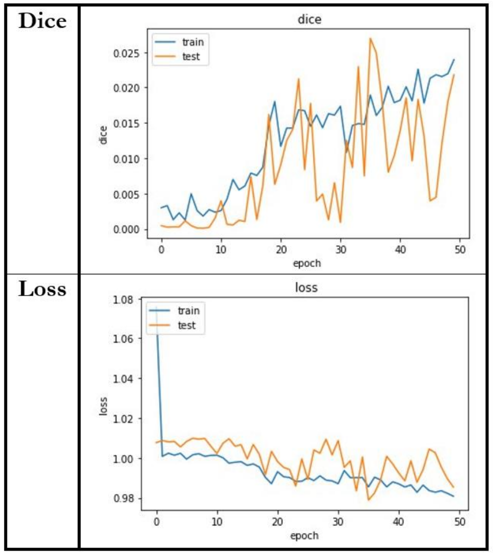


Figure 1: Dice and loss for X-Net using Loss function 1

$$Loss\ function\ 1\ = BinaryCrossEntropy + (1 - Dice)$$

- For dice we have the same behaviour that we have in Figure 2
- Better result for loss function

<u>BETTER</u>

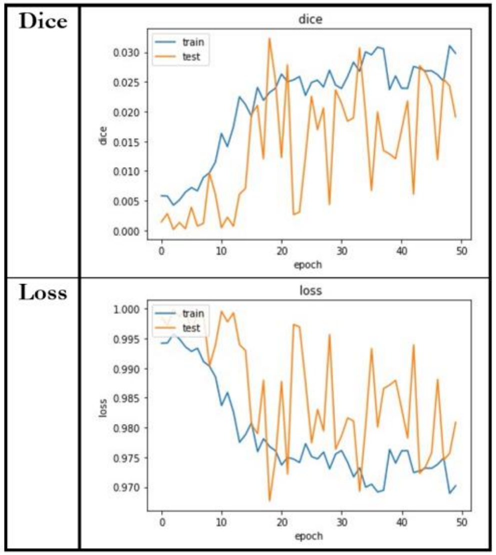


Figure 2: Dice and loss for X-Net using Loss function 2

$$Loss\ function\ 2 = 1 - Dice$$

- The results is not so good.
- Train set increasing the dice and decreasing the loss.
- Test set has the same behaviour of the train set, but it presents a lot of peaks.

<u>Conclusion</u>: if we train more the QNN, we can obtain better results.

Pre-processing

Split

Train

Evaluation 2

Evaluation with Model 2

X-Net with Quaternions: output obtained

Motivation of results obtianed:

- the number of epochs that we have chosen is not sufficient for a quaternion network
- it is probably that our dataset is not big enough
- this structure of the network might not adapt to quaternions
- the choices that we have done to represent the images in the quaternion domain is not adapt.

Comments:

- image that has not the broke lesion, the result is not good, because we have a white point so marked
- image that has broke lesion is good, because it highlights the problem that the patient has

Problem: we have a white point so marked

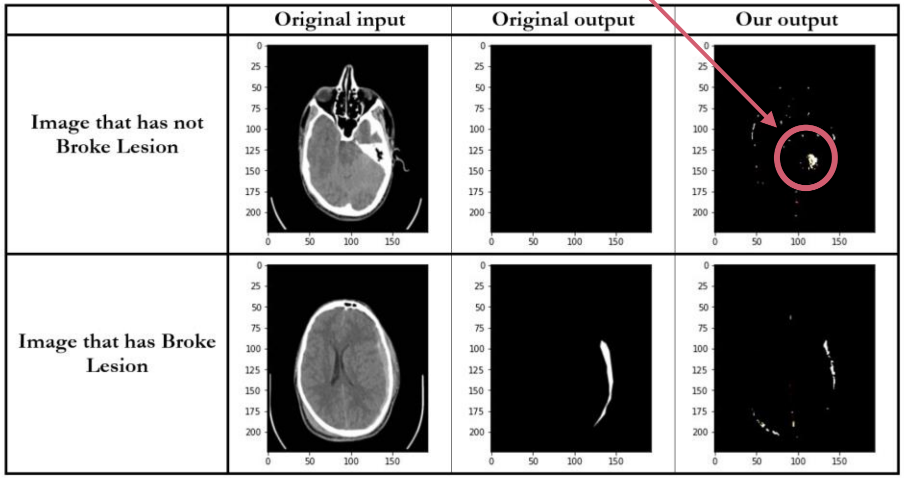


Figure: Random samples X-Net with quaternions

Pre-processing

Split

Train

Evaluation 2

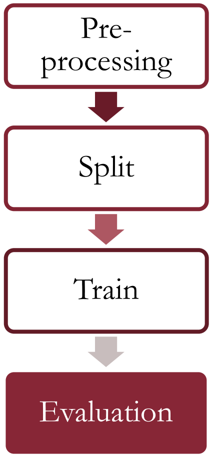
Comparisons of results obtained

Conclusion

Model	Dice	#Parameters
X-Net	0.2188	15.1M
X-Net (paper)	0.4867	15.1M
X-Net quaternion	0.0230	26.5M

Table: Comparisons between models

- We have only the 6% of data with respect to ATLAS (dataset used in paper). So, we have obtained the final dice = 0.2188 that is good compared to 0.4867 (paper)
- X-Net has a better Dice value with respect to X-Net with quaternions, but maybe is not true if we will train again X-Net with quaternions.
- X-Net with quaternions has the number of parameters that is almost double respect to which of the X-Net.



Thanks for the attention!

Sapienza
Università di Roma

Denise Landini
Alessandro Lambertini