ISIC 2018 Challenge: Skin Lesion Analysis Towards Melanoma Detection

Presentation by: Maria Camila Paris Diaz (dgk582)
Benjamin Heuser (krn362)
Alessandra Lucchetti (hft824)

(All group members have contributed evenly to the project)

Index

Background

- International Skin Imaging Collaboration (ISIC): an international effort to improve melanoma diagnosis
- The ISIC Archive contains the largest publicly available collection of quality controlled dermoscopic images of skin lesions [1]

Goal: develop image analysis tools to enable the automated diagnosis of melanoma from dermoscopic images.

U-Net Architecture

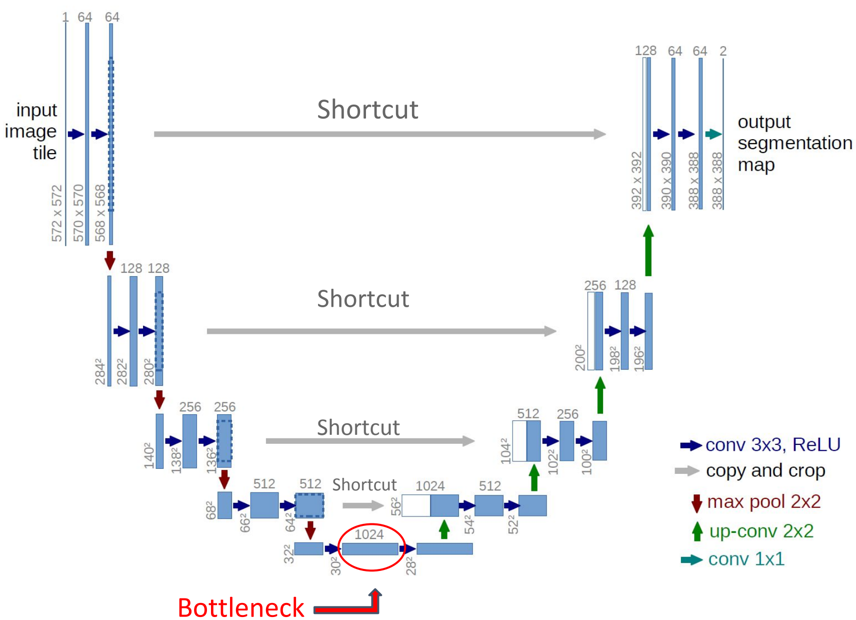


Figure 1: Adapted from [2]

<u>FCN (Fully convolutional network)</u>

Encoder: maps raw image pixels to a rich representation of a collection of feature vectors.

Decoder: produces an output and maps the output back into the "raw" format

+ shortcut connections from encoder to decoder to help decoder recover the object details better

Pros: good for biomedical imaging (small datasets)

Can be used with data of different sizes (no fully connected layers)

Cons: low resolution output

Task 1: Lesion Boundary Segmentation

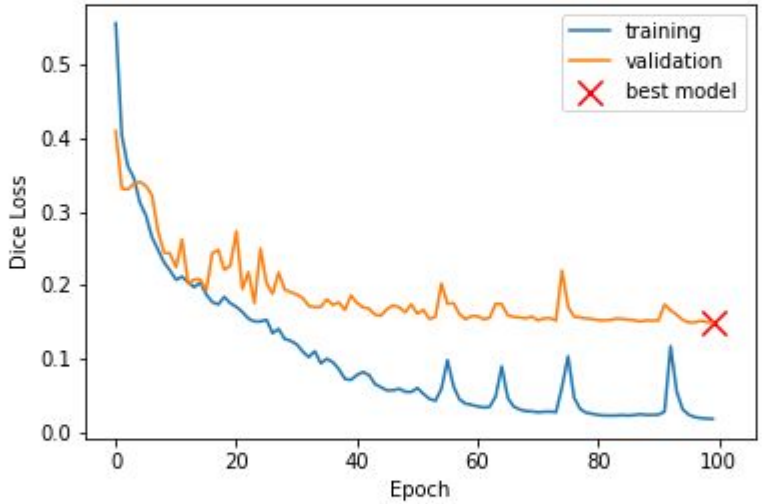
- Goal: Predict a segmentation mask covering the mole
- Methodology:
  - Architecture: two Unet implementations with different filter numbers
  - Preprocessing: Images are rescaled and normalized
  - Loss function: Mixture of binary cross entropy and dice loss

$$\mathcal{L} = \mathrm{BCE} + \left[1 - \frac{2 \cdot \sum y\hat{y} + 1}{\sum y + \sum \hat{y} + 1}\right]$$

- Experimental Setup:
  - Training on 2205 images and evaluating on 389 test images

Small U-Net implementation

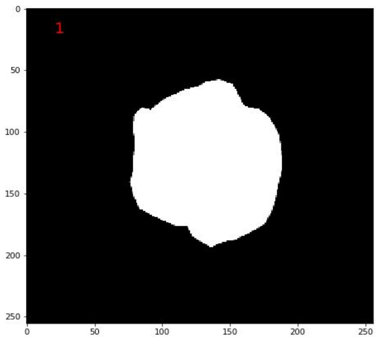
- Filter numbers: **16**, **32**, **64**, **128** and a **bottleneck** of **256**
- Convolutional layers are **3x3** filters with stride **(1,1)** and padding mode "**same**" and "**relu**" activation
- Followed by Max pooling layer with stride **(2,2)**
- Total parameter number: 1.9×10^6
- Training time per epoch: 75s

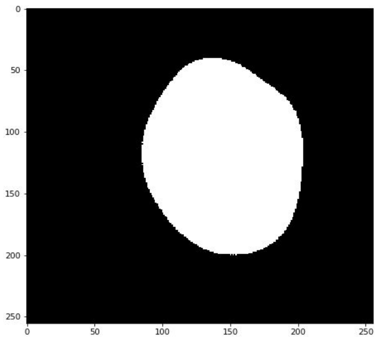


Validation values	Dice Coefficient	Jaccard index	Sensitivity	Specificity	Accuracy
ISIC 2018 winner	**0.898**	**0.838**	0.906	**0.963**	**0.942**
Our implementation	0.857	0.753	**0.970**	0.955	0.941

Large U-Net implementation

- Filter numbers: **32**, **64**, **128**, **256**, **512** and a **bottleneck** of **1024**
- Convolutional layers are **3x3** filters with stride **(1,1)** and padding mode "**same**" and "**relu**" activation
- Followed by Max pooling layer with stride **(2,2)**
- Total parameter number: 31.1×10^6
- Training time per epoch: 630s

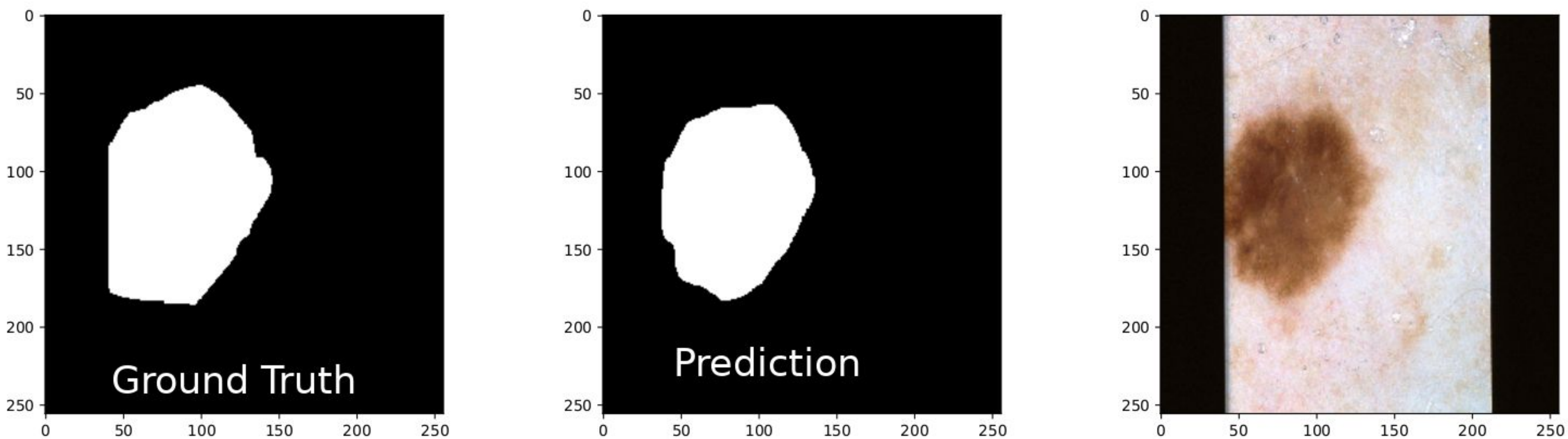




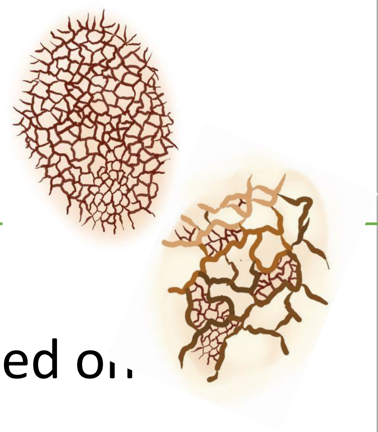
Validation values	Dice Coefficient	Jaccard index	Sensitivity	Specificity	Accuracy
ISIC 2018 winner	**0.898**	**0.838**	0.906	**0.963**	**0.942**
Our implementation	0.872	0.793	**0.978**	0.941	**0.948**

Problems and difficulties

- Small dataset
- Extremely complex models required
- Difficult evaluation metrics
- Bad training data



Task 2: Lesion Attribute Detection



- Goal: segmentation of several features (we concentrated on "pigment network - but the method is extendible)
- Challenges:
  - Limited dataset and very imbalanced
  - Sometimes tiny features not clearly distinguishable from rest of the mole
  - Noisy pictures (hair, plasters...)

- Experimental setup: 2594 pictures in jpg format + 2594 binary masks in png format
  - Shuffled and split in 0.9:0.1 train/test
  - All pictures and masks normalized and downsized (256x256 pixels)
- Architecture: U-Net

History:

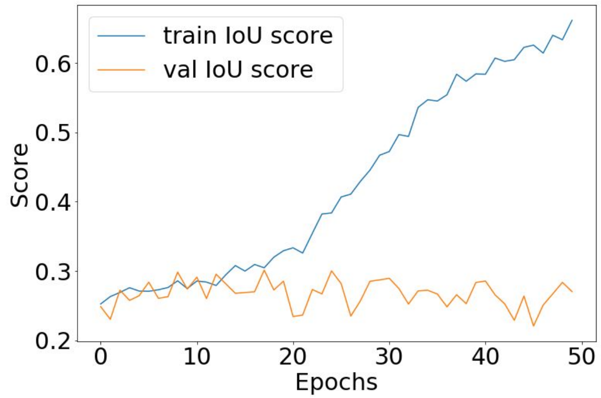
- Simple implementation [4] → It seemed like it had a hard time to even predict something inside the mole
- Substitute the encoder with InceptionResnetV2 trained on ImageNet and use augmentation of the pictures (random flip, rotations, zooming…) → Almost 60 millions parameters and more than 400 layers (10 times higher computation time and max batch size 2 pictures) [5]
- How to give the network the information of the mole boundaries?
  - Additional input channel
  - Valve filter approach [6] (GPU memory limitation)
  - <u>Go back to the simple implementation with pre-training on segmentation masks from Task 1.</u>
- Trained the network on those masks for 20 epochs.
- Loss function: Mixture of binary cross-entropy and Jaccard Loss

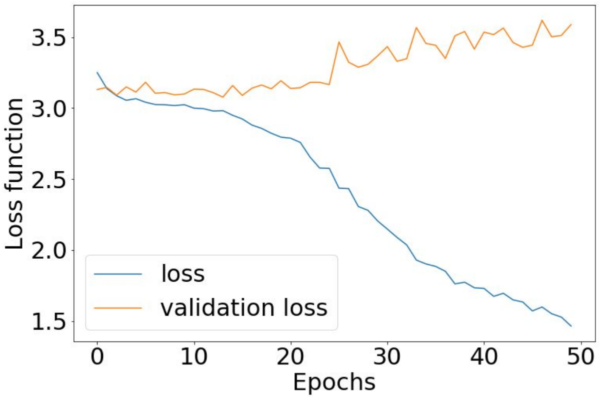
$$L = 0.5\,BCE + \left[1 - \frac{\sum y\hat{y}}{\sum y^2 + \sum \hat{y}^2 - \sum y\hat{y} + 1}\right]$$

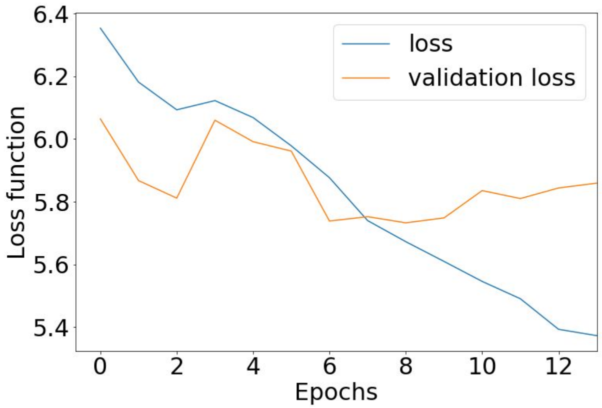
- Accuracy: Intersection over Union (IoU) with 0.5 threshold

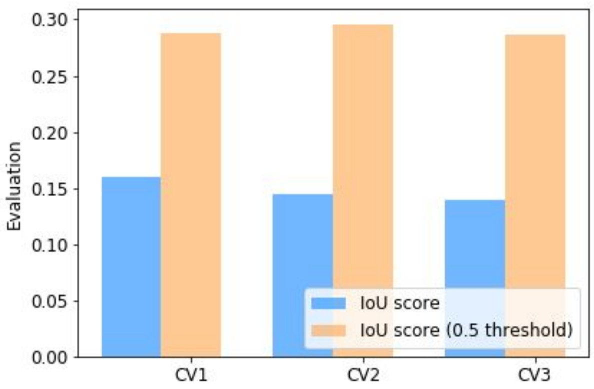
Training and validation

- Optimizer: Adam (default parameters) with reduced learning rate on plateaux of loss function and early stopping
- 3-fold cross-validation on the training dataset

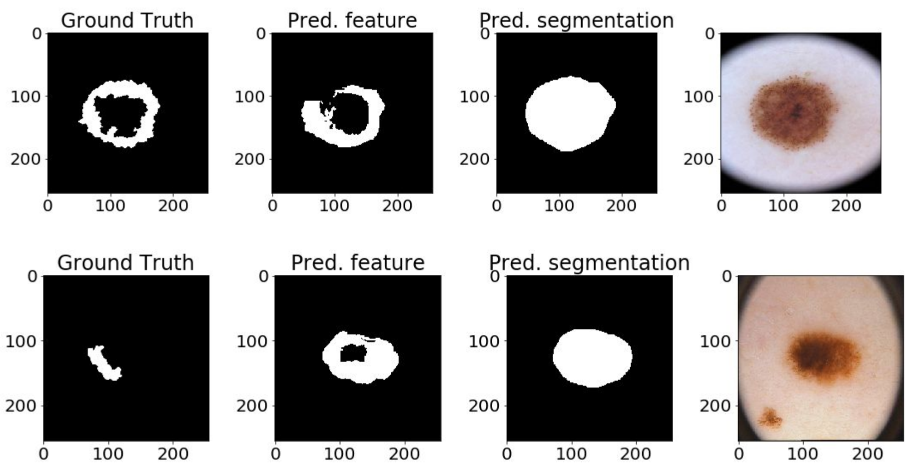




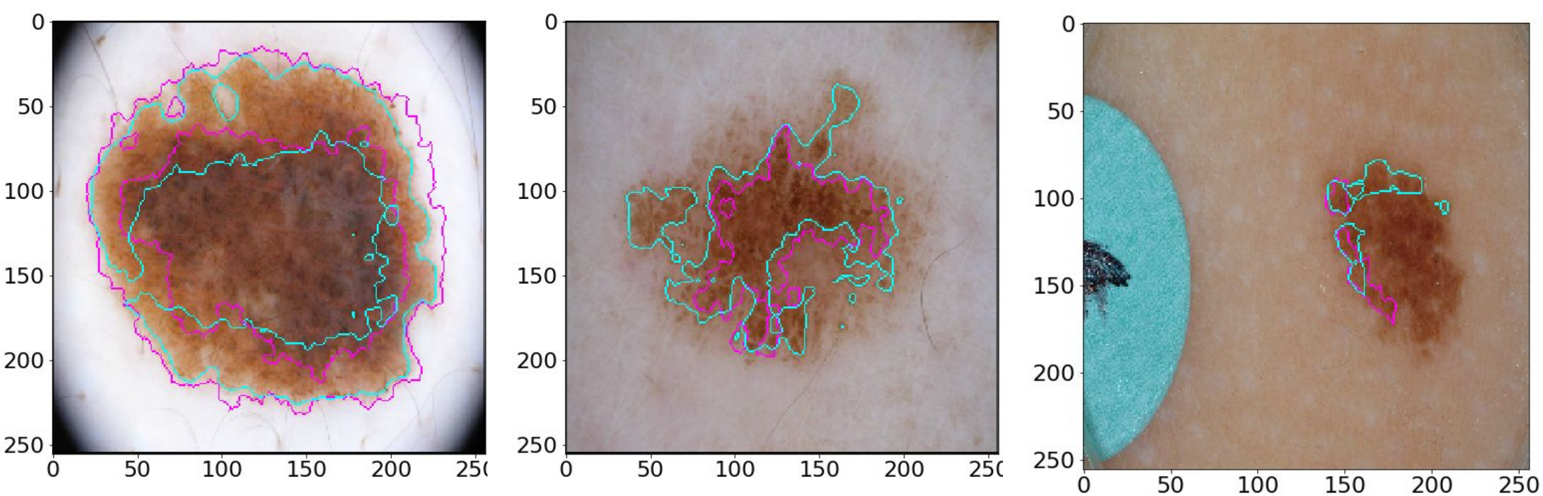




Results in the test set



Test values	Dice Coefficient	Jaccard index
ISIC 2018 winner	**0.690**	**0.527**
Our implementation	0.47	0.31



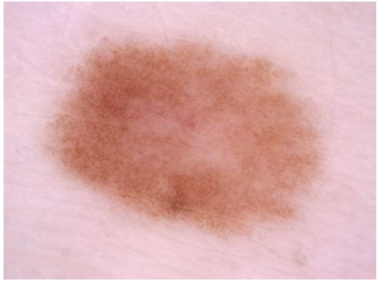
Pink: ground truth
Cyan: predicted feature mask

Task 3: Lesion Diagnosis

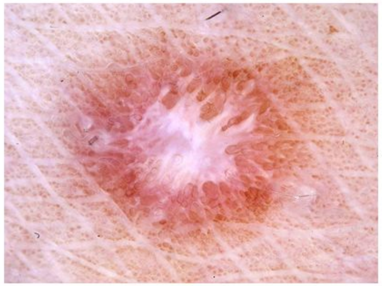
- Goal: automated predictions of disease classification
- Challenges:
  - Imbalanced dataset
  - Similarities between disease categories
- Methodology:
  - Image preprocessing with a target size of (350,350)
  - Separate loss function for each output (7)
  - Softmax: last activation function, increasing score for one label and lower the others
  - Loss function: binary cross entropy

Disease Categories

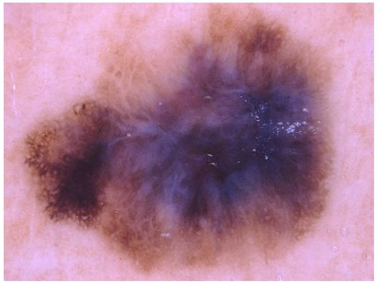
Nevus



Dermatofibroma

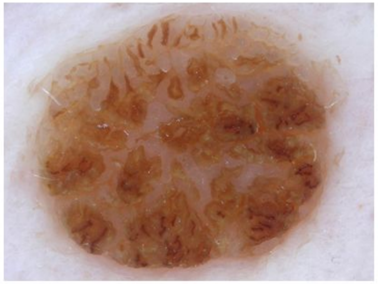


Melanoma

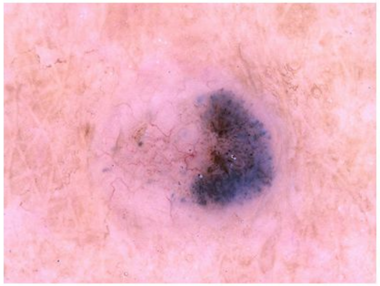


Pigmented Bowen's

Pigmented Benign Keratoses



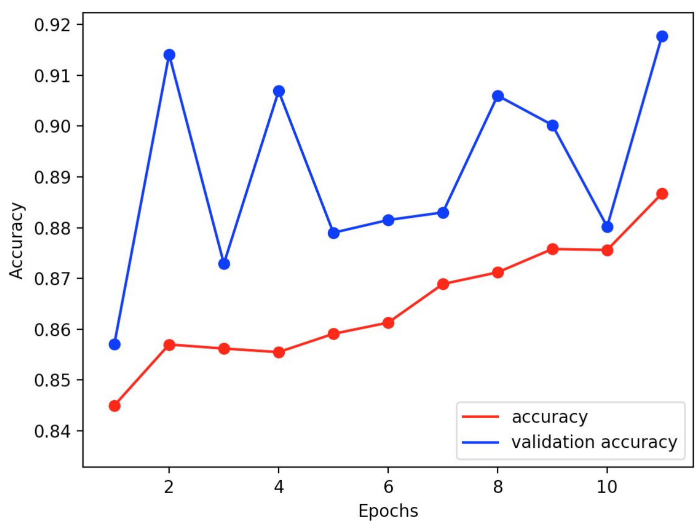
Basal Cell Carcinoma

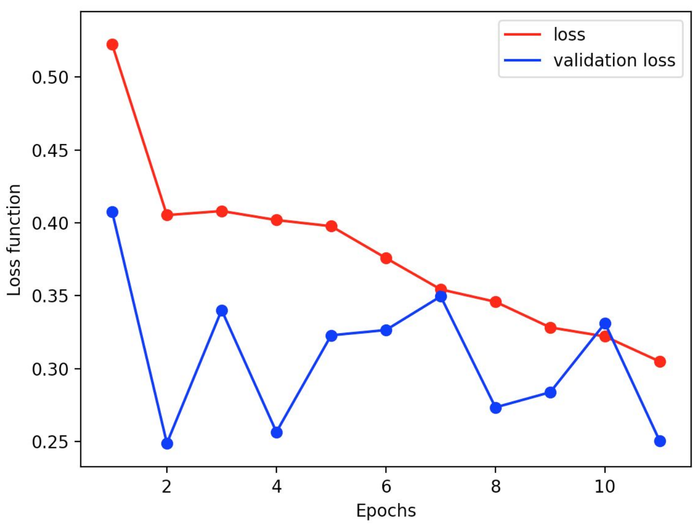


Vascular

Results

- Undersampling due to imbalanced dataset
- Cannot compare with test results
- Best: 92% accuracy with 1000 images





Results

- Used 3 convolutional layers
- Alternated with:
  - Nonlinear layers - ReLU
  - Pooling layers
  - Bound layers
- Better results- undersampling, imbalanced dataset

Category Metrics	Accuracy Value
ISIC Balanced multiclass Accuracy	0.885
Threshold Metrics (0.5)	0.958
Our implementation	0.918

Conclusions

- To sum up:

Task 1: Lesion Boundary Segmentation	Task 2: Lesion Attribute Detection	Task 3: Lesion Diagnosis
IoU = 0.793	IoU = 0.31	Accuracy = 92%

- Further improvements:
  - Using full size resolution pictures and more data
  - Using pictures augmentation
  - More complex networks with more powerful GPUs

References

[1] ISIC Challenge (dataset): https://challenge2018.isic-archive.com/

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI*. URL: https://arxiv.org/pdf/1505.04597.pdf

[3] Explanatory video of U-Net: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/u-net-teaser.mp4

[4] Python implementation (1): https://github.com/nikhilroxtomar/UNet-Segmentation-in-Keras-TensorFlow/blob/master/unet-segmentation.ipynb

[5] Python implementation (2):

https://segmentation-models.readthedocs.io/en/latest/index.html#

[6] Eppel, S. (2017). Setting an attention region for convolutional neural networks using region selective features, for recognition of materials within glass vessels. *CoRR, abs/1708.08711*. URL: https://arxiv.org/ftp/arxiv/papers/1708/1708.08711.pdf

Appendix

Formal definition of the dice coefficient and the IoU (or Jaccard index) reported here for clarity.

$$\mathrm{DC} = \frac{2TP}{2TP + FP + FN} = \frac{2|X \cap Y|}{|X| + |Y|}$$

$$\mathrm{IoU} = \frac{TP}{TP + FP + FN} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$

Appendix

We report for comparison the loss function and the score registered for Task 2 without the pre-trained weights. Doing the pre-training gave us an advantage since the starting point of the loss function was lower and the score higher even if then the over-fitting came very soon.