Master project in Life Sciences Engineering

Label-Efficient Volumetric Deep Semantic Segmentation of Intracranial Hemorrhages

Carried out in the Laboratory of Biomedical Imaging at Kunglia Tekniska Högskolan (Stockholm, Sweden) Under the supervision of Chunliang Wang (Researcher)

Done by

ANTOINE SPAHR

Under the direction of
Prof. Jean-Philippe Thiran (Prof. EPFL)
In the Laboratory of Signal Processing 5 (LTS5)

EPFL
Lausanne, EPFL 2021
Contents

Summary 1

1 Introduction 2

2 Related Work 3
  2.1 ICH Detection & Segmentation 3
  2.2 Label-Efficient Approaches 4
  2.3 Anomaly Localization Algorithms 5

3 Datasets 6
  3.1 public ICH Segmentation Dataset 6
  3.2 RSNA 6
  3.3 CQ500 6

4 Methods 7
  4.1 Evaluation Metrics 7
  4.2 Pre-processing 8
    4.2.1 Contrast Window 8
    4.2.2 Data augmentation 8
    4.2.3 Brain Extraction 9
  4.3 Fully Supervised 2.5D U-Net 10
    4.3.1 ICH only 11
    4.3.2 Mixt 11
  4.4 Self-Supervised and Weakly-Supervised Pre-Training 11
    4.4.1 U-Net 2-5D pre-training on Context Restauration 12
    4.4.2 U-Net 2-5D pre-training on Global-Local Contrastive 12
    4.4.3 U-Net 2-5D pre-training on Binary Classification 15
    4.4.4 U-Net 2-5D pre-training on Multi-Label Classification 16
  4.5 Unsupervised Inpainting Based Segmentation 17
    4.5.1 General Idea 17
    4.5.2 SN-PatchGAN Inpainter 17
    4.5.3 Inpainting Anomaly Detection Module 19
    4.5.4 Auto-Encoder Baseline 23
    4.5.5 FCDD Baseline 24
  4.6 Semi-Supervised Inpainting-attention U-Net 2.5D 25

5 Results & Discussion 26
  5.1 Brain Extraction 26
  5.2 Inpainting Anomaly Detection Validation 27
  5.3 ICH Segmentation 31

6 Conclusion & Future Works 42

Acknowledgements 44

References 44

Annexes 48

Antoine Spahr 2021
Summary

Traumatic brain injuries is one of the major cause of death and disability worldwide especially among young people. The initial injury can later develop into a secondary injury that is particularly lethal. The patient’s survival relies on fast and proper anticipation of secondary injuries by medical worker. However, it is a demanding task for radiologist, that could be eased by automatizing it. In this context, Deep Learning algorithms pipeline would be well adapted to efficiently process the high-dimensional CT-scans. In this context, the very first step of such a pipeline is to locate intracranial hemorrhages (ICH) form CT-scans. Because ICH comes in various shapes, sizes and location, a large number of labelled data is required to train robust and trustworthy models. However it is prohibitively expensive and time consuming to acquire many volumetric labels from trained radiologist. As a results, there is a need for label-efficient algorithms that can exploit easily available unlabelled and weakly labelled data. In this work we explore a varieties of methods that harness unlabelled and weakly labelled data to improves the ICH segmentation. Unlike most studies on ICH, our work relies exclusively on publicly available datasets which allow to easily compare performances with future studies. We further explore unsupervised ICH localization through the anomaly detection principle and we propose a novel approach based on image inpainting. This study demonstrates that unlabelled and weakly labelled data can be used to greatly improve the ICH segmentation performances with a scarce amount of annotations. We further show that our anomaly localization methods enables to extract meaningfully hemorrhages in an unsupervised way, and demonstrate its general capabilities on an anomaly localization benchmark, the MV-Tec dataset.¹

¹The python code is available on Github at https://github.com/antoine-spahr/Label-Efficient-Volumetric-Deep-Semantic-Segmentation-of-ICH
1 Introduction

Traumatic brain injuries (TBI) is one of the major cause of death and of disability worldwide. For example, in the US, 1 to 6 million people suffer a TBI each year, among those more than 200’000 are hospitalized, and around 50’000 later die, while around 80’000 are disabled by it [1, 2]. TBI is also heavily affecting young person and is one of the leading cause of death for this population range [3]. For example, in New Zealand it has been estimated that by 25 years old, around 30% of the population has experienced a TBI requiring medical attention [2]. The mechanical force of impact in TBI often results in intracranial hemorrhages (ICH) and an augmentation of the intracranial volume known as mass effect. Those primary injuries can later evolve in secondary injuries that are the leading cause of death after hospital admission [4]. As a results, upon admission of a patient with TBI, radiologists try to predict how the injuries may evolve to administrate the best treatment possible. The diagnosis is usually performed through a non-contrast CT (computer tomography) scan [5]. This predictive task from the CT-scan is time-consuming but also error prone.

Automatizing this procedure would greatly help the medical workers in acting efficiently and rapidly with TBI patients. The very first step in automatizing the TBI diagnostic would be to localize ICH in the CT-scan as well as the mid-line shift (an indicator of the mass effect). Meaningful decision would then be made using those extracted features. The pixel-wise localization of structures is commonly known as a semantic segmentation task and can be a tough task. However, the rise of deep learning in the past years [6] has enabled the efficient processing of high-dimensional input such as images and has shown great performances in the medical domain in a diversity of tasks including segmentation tasks [7, 8, 9]. Nonetheless, one of the major pitfall of deep learning is the need for large labelled datasets to avoid overfitting of the models. The access to label data is even more problematic in the medical imaging field in which labels must be provided by trained radiologist which is often time consuming and expensive. On top of that, pixel-wise label for segmentation task are even more time-consuming and thus even harder to get in a large number. In the particular case of ICH segmentation, ICH comes in different shapes and sizes at different location in the brain, it is therefore quite an heterogeneous structure to segment. As a results, a segmentation algorithm would need a large amount of labeled scans to provide robust predictions. It is therefore desirable to develop and use algorithms using fewer labelled data.

In light of the aforementioned challenges, the present work focuses in the very first step of the TBI automatic diagnostic: the segmentation of ICH in CT-scans in a low label regime. Different label-efficient approaches are explored and compared on a dataset composed of 36 CT-scans with a variety of different ICH. Specifically, we explored supervised methods, semi-supervised methods based on weights pre-training with different tasks self-supervised and weakly-supervised tasks. We further proposed a novel general approach for anomaly segmentation, validate it on a benchmark dataset, and explore its ability to segment ICH in an unsupervised way.

This work is articulated as follows: we first provide a brief literature review in section 2 about automated ICH detection and segmentation, label-efficient segmentation, and anomaly localization algorithms. Then, in section 3 we introduce the datasets used throughout the study. Section 4 describes the different label-efficient algorithms explored in this work as well as the proposed anomaly localization method (section 4.5). Finally, the results are presented and discussed in section 5.
2 Related Work

2.1 ICH Detection & Segmentation

Several studies have already explored the use of deep learning method both for the detection and localization of ICH following a TBI, mostly using CT-scan volumes as input. Chilamkurthy et al. [10] conducted an extensive study on harnessing deep learning for automated detection of ICH (as well as the five sub-types), calvarial fractures, midline shift, and mass effect using CT scans. They trained one classification model for the hemorrhages detection: a ResNet18 with 5 MLP branches for each type assuming that the feature map for each hemorrhages would be quite similar. They used a similar strategy for the midline shift and the mass effect classification (ResNet18 with 2 MLP branches). The clavarial fractures were detected and localized using a DeepLab architecture. For the hemorrhages detection, they also explored hemorrhages localization for three type of hemorrhages using a 2D U-Net with manually labeled slices (around a 1000 slices with ICH and 750 without).

Patel et al. [11] tackled the segmentation problem in 3D using a 3D U-Net to segment spontaneous ICH (i.e. not associated with a TBI). They trained their model on 21 volumes and evaluated it on 30 volumes. With their settings they reached a median dice coefficient of 0.91. Sharrock et al. [12] explored the segmentation of ICH (intraventricular and subdural) with 2D and 3D V-Nets [13]. They trained their network on 100 scans and reported a best mean dice of 0.911 when training a 3D V-Net with 64x64x64 patches on the dice loss.

Dataset for ICH segmentation are often private ones, and until recently there was no benchmark dataset for ICH segmentation making the comparison of methods often impossible. Hssayeni et al. [14] collected and prepared a public dataset of 75 CT-scans with associated segmentation ground truth for ICH. 36 of the 75 scans present an ICH. Each scan has around 35 slices. They further provide segmentation results with a 2D U-Net as a proof of concept. They fed the network with 160 × 160 pixels portion of the CT-slice, each portion having an 80 pixels overlap with the preceding one. They achieved a mean slice dice score of 0.31 on their dataset.

Kuo et al. [15] explored the use of a single neural network to perform both detection and segmentation of intracranial hemorrhages in CT scans. They used a PatchFCN, a fully convolutional network applied to patches of the input scan. Their model achieved an impressive 99.1 ± 0.6% AUC for the classification task and an average dice score of 0.75 for the segmentation. They used CT volumes of 4396 patients among which 1131 have hemorrhage. Their model is designed for processing 2D input, but they included depth information by adding the superior and inferior slices (input with 3 channel) to the slice of interest.

In the same spirit, Guo et al. [16] showed that combining simultaneously the classification task (5 ICH sub-types) with the segmentation task improves the performances. Their model processes a 3D input in a 2D fashion (with 2D convolution), the inter-slice information is processed with the aid of a ConvLSTM module. With a private dataset of 1176 scans (training on 706 scans among which 370 have ICH), they obtained a segmentation dice of 0.951 for ICH compared to 0.919 with a 3D U-Net. They also showed that simultaneous training improves the classification task reaching an ICH-detection AUC of 0.995.

Although, the studies presented above report impressive segmentation performances, they usually get there with large private labelled datasets. Only Hssayeni et al. [14] works with a small dataset and consequently report lower segmentation performances. In addition, they essentially focus on fully supervised approaches. To our knowledge, only Wang et al. [17] explored a semi-supervised method for ICH segmentation. They improved the multi-tasked attention-
based semi-supervised method developed by Chen et al. by stabilizing the optimization process and they evaluate their method using the challenging public dataset of Hssayeni et al. [14] combined with the large weakly labelled RSNA dataset [18] (see section 3). The method consists of a combination of a segmentation network (U-Net) and a reconstruction decoder. Using the U-Net segmentation, the detected ICH and the background are reconstructed separately and those two reconstruction losses enable to regularize the U-Net encoder’s weight with an attention mechanism. In a 5-fold cross validation scheme, Wang et al. reached a segmentation dice of 0.67 for the segmentation of slice (and not volumes).

2.2 Label-Efficient Approaches

While label efficient approaches applied to ICH segmentation are limited, a large number of studies proposed label efficient methods for other applications. In their presentation of the U-Net architecture, Ronneberger et al. [7] showed that simply using elastic transformation as data augmentation enabled them to segment efficiently cells with a small training set. Nonetheless, for more complex segmentation tasks, more complex approach may be necessary. It is often difficult to get labelled data for segmentation, but it is way easier to get unlabelled or weakly labelled data in a large quantity. Even if not labelled, those images still carry meaningful information that can be extracted to improve the segmentation performances. There are different strategies to exploit those data: sequential tasks, parallel tasks or data generation. The sequential tasks makes use of transfer learning. The rational is to learn features through a pretext task that does not need specific labels. The features learned in an self-supervised fashion are then used to initialize the segmentation network before it is fine tuned using the few labels available. It can be understood as if the model receive an introductory lesson about the subject before receiving the explicit lesson. The quality of the transfer depends solely on the nature of the pretext task and its ability to drive the learning of good features. The nature of the pretext task also imposes what parts of the network can be pre-trained (encoder only or both encoder and decoder). As a results, several studies have explored various pretext tasks. The most common and straightforward method to learn feature in a self-supervised fashion is the auto-encoder [19] which is a convolutional neural network composed of a compression and decompression trained with the objective of reconstructing the input. However the auto-encoder often tends to learn low-level features that have poor value for a discriminative task such as in medical image analysis [20]. In a similar fashion Hervella et al. [21] used a reconstruction task between two image modalities to pre-train a network for segmentation of joint disk and cup on the retina. Taleb et al. [22] proposed a self-supervised task to learn efficiently an encoder’s weights using a puzzle-solving task. Their approach make use of images in several modality (e.g. CT and MRI). Those input images are cut into patches that are mixed to generate a puzzle composed of several patches from different modalities. A network is then trained to learn a permutation matrix to solve the puzzle and reconstruct the input image. This way the network is enforced to understand the context and the object under different modalities leading to learning of good features. However their method implies the access to several modalities of the same image. In the similar direction of puzzle solving, Chen et al. [23] proposed a self-supervised method based on image restoration. The rational is to swap small patches of the input images producing an altered image. A convolutional neural network is then trained to restore the image which enforces the network to understand the spatial context. They showed that features learned with their task can be transferred to downstream classification, localization and segmentation task leading to improved performances compared to other
self-supervised methods on three medical image datasets. Recently, self-supervised contrastive learning tasks have shown promising capabilities to learn meaningful and detailed features of a dataset to improve semi-supervised classification of natural images [24, 25, 26]. The rational of a contrastive task is to learn features by comparing images in a set, an encoder is optimized to bring similar images closer in an embedding space while pushing dissimilar ones away from one another [27]. Initially, contrastive pre-training has been mainly used for classification task and recently for anomaly detection [20]. Recent multiple studies have explored its used in the segmentation task (for encoder-decoder networks). For example, Chaitanya et al. [28] proposed an extension of the contrastive loss able to train a part of the decoder in a segmentation networks. Their rational is to enforce similar local representations on the feature map generated after few deconvolutional layer. The contrastive loss takes as positive pairs two spatially identical patches of the feature map from two image representations. Instead of taking negative contrast from other samples, their loss formulation uses all other feature map patches as negative to enforce local representation that are locally distinctive. They assessed the relevance of their pre-training task on volumetric organ segmentation. Alternatively, Feng et al. [29] have proposed a self-supervised contrastive task using reconstruction. An input volume is cropped and resized several times to get several parts. An encoder-decoder network is then optimized to reconstruct the whole volume from the parts. They showed that this reconstruction approach enforce the network to learn contrastive representations.

Contrastive learning has also been used in a parallel tasks scheme. Iwasawa et al. [30] exploit unlabelled data through a regularization branch append to the segmentation network. They showed that enforcing a contrastive loss on this regularization branch improves the segmentation results more effectively than other regularization loss such as variational auto-encoder losses or boundary losses.

In a completely different direction, several studies for lesion segmentation relies on the observation that a lesion can be viewed as an anomaly. This observation enables to segment lesion in an unsupervised way using only healthy images during training. For example, Bowles et al. [31] used a combination of image synthesis, Gaussian mixture models and one class support vector machines to segment brain lesion on MRI images. Subsequently, Baur et al. [32] explored the use of different variants of the auto-encoder for detection of brain lesion in MRI images. The idea is to train an auto-encoder network to reconstruct healthy images only. This network should then have trouble in reconstructing a lesion as it is not a feature that it has seen before, resulting in a large reconstruction error for anomalous pixels. In the same fashion, Pawlowski et al. [33] used a Baysian convolutional auto-encoder to extract hemorrhage lesion from CT-scans. Baur et al. [34] further proposed to use the regions where the reconstruction error is large (unsupervised lesion segmentation) obtained through an auto-encoder as ground truth for non-labelled data. Those can be used for the optimization of a segmentation network together with labelled samples and they present improved results using this approach.

2.3 Anomaly Localization Algorithms

The use of anomaly localization algorithm for lesion segmentation appears to be a promising direction of unsupervised lesion segmentation. However, more methods have been proposed than the deep convolutional auto-encoder [35]. Schlegl et al. exploit the GAN framework [36] to model the normal sample distribution. For a new image, the algorithm search for the generated image that is closer to the input image, the difference between the generated one and the original one should be close to zero for a normal pixels and higher for anomalous pixels as the
generator should fail in generating structure it has not seen before. Li et al. [37] also recycled an inpainting network to localize anomalies. The network is trained to fill in missing region of normal image only and it should fail in reconstructing properly anomalies. During evaluation, they cut the image into super-pixels (chunk of image) and for each of them, they mask it, inpaint it and measure the mean reconstruction error over this super-pixel. As a results, their method yield an anomaly map with a resolution of super-pixel only, hence, rather coarse. In a completely different fashion, Liznerski et al. [38] proposed a method specifically designed for anomaly detection and localization that provides a heat-map of the anomalous region. During training, a fully-convolutional encoder network is trained to map normal samples close to a point in the embedded space while letting abnormal one away from it. Because, they use fully convolutional network, the final feature map is a downscaled representation of the image where high values highlight anomalies. A heat-map with similar size as the input is obtained by upsampling with a Gaussian kernel. It therefore results in anomaly heat-maps that are more roundish and less sharp.

3 Datasets

3.1 public ICH Segmentation Dataset

Hssayeni et al. [14] collected and prepared a public dataset of 75 CT-scans of patient with TBI, with the associated segmentation ground truth for ICH. 36 out of the 75 scans present an ICH. The CT scans were collected between February and August 2018 from Al Hilla Teaching Hospital, Iraq. The scans are anonymized by blurring the face of patient on the scans resulting in non-natural structure around the face. The dataset is available in NIfTI format and each scans contains approximately 35 slices, hence with a small depth resolution of around 5 [mm]. In total, there are 2814 CT slices among which 318 contain an ICH and the five different labelled ICH types are not equally represented. An overview as well as few slices (with the chosen contrast window) are presented on Fig.1. Noteworthy that the dataset contains CT-scans acquired right after the injuries and thus contains recent ICH. There is, however, one CT-scan from a patient admitted two weeks after the injury and thus contains an 'old' hemorrhage with a quite different texture as the other recent ICH.

3.2 RSNA

The Radiological Society of North America (RSNA) has recently released a large dataset of CT-scans slices labeled for ICH detection (classification task) [18]. The dataset contains 752’803 slices in DICOM format from more than 25’000 CT-scans. Each slice is labeled whether it contain an ICH or not and what type of ICH it is. 107’933 slices contains an ICH. An overview is presented on Fig.2.

3.3 CQ500

Chilamkurthy et al. [10] have made available their validation dataset from their study on ICH detection. The dataset contains 491 CT-scans labeled at the scan level for ICH presence as well as for the five different types of ICH. 171 scans are positive for ICH.
4 Methods

4.1 Evaluation Metrics

Semantic segmentation is a pixel-wise classification, therefore each segmentation can be viewed as a sample prediction which enables to report classification metrics sample-wise [39]. To quantify the capabilities of the model we decided to report the mean volume sensitivity (also known as recall) which highlights the model’s capabilities to detect hemorrhage. We also report the mean volume precision which highlights the model’s capabilities to avoid false detection of hemorrhage. Furthermore we are reporting the mean volume dice coefficient which is commonly used in segmentation task. The dice score is an overlap based metric quantifying the amount of intersection between the segmentation $p$ and the ground truth $t$: 
\[
\text{Dice} = \frac{2|p \cap t| + \epsilon}{|p| + |t| + \epsilon} = \frac{2TP + \epsilon}{2TP + FP + FN + \epsilon}
\]

\[
\text{Precision} = \frac{TP + \epsilon}{TP + FP + \epsilon}
\]

\[
\text{Recall} = \frac{TP + \epsilon}{TP + FN + \epsilon}
\]

where \(TP\) is the number of true positive, \(FP\) the number for false positives, \(FN\) the number of false negatives, and \(\epsilon\) ensure numerical stability when there are no true positives. We set it to \(\epsilon = 1\). The dice score is therefore equivalent to the F1-score (harmonic mean of the recall and precision) and requires both the recall and precision to be good in order to yield a high value. As a result, even a small number of false positives or negatives will affect the dice score.

### 4.2 Pre-processing

Before jumping straight to the segmentation task, the images must be prepared to facilitate the extraction of the features of interest.

#### 4.2.1 Contrast Window

The output of a CT scanner is a volume in which each voxel is assigned a physical measure of the light absorption relatively to water, given in Hounsfield Units (HU) [40]. Therefore, a value of zero is the signature of water while \(-1000\) [HU] represents the air. As a result, the pixel value is already standardized between samples but also between scanners. Therefore, there is no need for standardization within and between datasets. Nevertheless, each voxel can be assign a large spectrum of values and most part of this spectrum may not be useful at all for a given task. It is therefore common to adjust the contrast and rescale the intensity of the voxels only on a specific range of the spectrum (window) while clipping the values outside it. Different windows will extract different features, for example, voxels values between 700 [HU] and 3000 [HU] contains the signature of bone structure while soft tissues are closer to zero due to the presence of water. Hematoma usually lies in the range of 40 [HU] and 90 [HU] [41, 42].

In all the following ICH segmentation experiments we chose a window of \([-50, 150]\) [HU] to adjust the contrast of the scans in view of focusing on the ICH signature while keeping variety in pixel values to let the networks understand and learn good features. As a result, each scan is rescaled so that values below \(-50\) [HU] are clipped to 0, values above 150 [HU] are clipped to 1, and values inside the window are linearly scaled between 0 and 1.

#### 4.2.2 Data augmentation

Secondly, to reduce the overfitting on the rather limited amount of labeled data for the segmentation task, and in view of learning more robust features, we use data augmentation upon data loading. Because every CT-scans is acquired in the same orientation (head pointing upward and roughly in the center), we decided to use spatial transformations that would yield meaningful and realistic scans. The dataset is thus virtually enriched by translating (vertically and horizontally) the image by a factor uniformly sampled in \([-10\%, 10\%]\), followed by randomly rotating the image by an angle sampled uniformly in \([-15^\circ, 15^\circ]\), followed by a random scaling of the image with a factor sampled uniformly in \([0.9, 1.1]\), and finally by randomly flipping the
image horizontally 50% of the time. Finally, the image is resized to 256 × 256 pixels. Note that different augmentation are used for the contrastive experiment in section 4.4.2.

4.2.3 Brain Extraction

Naturally, ICH can only be present in the brain of the patient, and any prediction outside the brain would be meaningless. Therefore, we generate a mask of the brain for each CT-scan that allows us to ignore hemorrhage detection outside the brain if needed. Motivated by the work of Akkus et al. [43] we chose to generate the brain 3D-mask using a 2D U-Net applied on each slice of the CT-scan (we used the same U-Net architecture as in section 4.3). Due to the lack of public data for brain segmentation on CT-scan, we decided to manually labeled 10 scans from the CQ500 dataset using the ITK-SNAP software [44]. In order to obtain a more robust segmentation model for our use-case (with potential presence of ICH) We segmented 5 scans defined as healthy and 5 scans on which an hemorrhage is present. One of the manual segmentation ground truth is presented on Fig.3. Because those 10 scans have a good resolution on the z-axis, it allows us to obtain 2572 labeled image among which 2106 contains a part of the brain. However, due to the large variety in shape and size of ICH, using only 5 scans is unlikely to be enough to build a robust model. To cope with it without labeling more scans, we decided to train two models on different CT window : 1) One U-Net is trained on slices rescaled with a tissue window $HU \in [0, 600]$ 2) One U-Net trained on slices rescaled with a bone window $HU \in [150, 650]$. The model trained on the tissue window should be able to extract well the brain where the boundary is less clear (at the top and bottom of the skull) while the bone window should perform well when there are abnormal elements on the brain (e.g. ICH) since it is based principally on the skull structure. The final brain mask is then obtained as the union of the predictions of those two models.

The tissue and bone U-Nets are trained on the dice loss [13] (see sec. 4.3) for 50 epochs with a batch size of respectively 20 and 16. The weights are optimized using the Adam methods with default parameters and with an initial learning rate of 0.001 exponentially decayed with a base of 0.96 every epoch. The weight of $L_2$ regularization on the model’s parameters is set to 1e-6. In order to measure the segmentation performances, the two U-Nets are trained similarly through a 5-fold cross-validation with partition made at the scan level to avoid test leakage and ensure that the validation scans are never seen by the models upon training. The final models are then trained on all the data available.
4.3 Fully Supervised 2.5D U-Net

The U-Net architecture proposed by Ronnenberger et al. in 2015 [7] is well suited for semantic segmentation and has shown its potential on various tasks across various domains. Even though the U-Net has been successfully used with 3D volumes as input, such an approach requires the volumes to have relatively similar resolution along the three dimensions in order to learn meaningful convolutional filters. This not our case as the available CT-scans have a resolution of around 0.5mm in the $x$ and $y$ direction but only around 5mm between slices. In addition, the use of 3D convolutions is computationally intensive and hinder the possibility to use deeper and more complex architecture. That is why, we decided to use a U-Net 2.5D as main architecture. A U-Net 2.5D processes a volumes slice by slice (therefore in 2D) but the performances are computed over the whole volumes instead of separately for each slice. The U-Net 2.5D’s principle is summarized on Fig. 4. For all experiment, the same U-Net architecture is used for meaningful comparison of the different approaches. The U-Net is composed of 5 convolutional blocks for encoding and 4 deconvolutional blocks for decoding that integrate features of the corresponding block in the encoding path through skip connections. A convolutional block is a series of two convolutions, batch-normalization and ReLU layers over which the number of channels is increased by a factor two. Between two convolutional blocks a Max-pooling layer reduce the feature map size by a factor two. A deconvolutional block is composed of a transposed convolution that increases the feature map size followed by a convolutional block similar to the encoding path except that the number of channel is reduced by a factor two over the block. A detailed overview of the architecture is available in Fig. 21 in the annexes.

Because the number of labeled images in the public dataset is rather low ($n = 318$) and since the hemorrhages are quite heterogenous, the performances of the U-Net 2.5D are estimated...
through a 10-fold cross-validation scheme. Noteworthy that the dataset is split in a stratified way at the level of the volume to ensure that all slices of a scan end up in the same fold and that each fold contains volumes with hemorrhages.

4.3.1 **ICH only**

First of all, as a crude baseline, a U-Net 2.5D is trained only on slices containing hemorrhages in the training set. For each of the 10 folds, the models is trained for 100 epochs with a batch size of 16 and an initial learning rate of 0.001 that is then exponentially decayed at each epoch with a base of 0.96. The model’s parameters are optimized using the Adam optimizer with the default parameters and a weight of $L_2$ regularization set to 1e-6. The model is trained on the dice Loss [13] which directly enforces the model to maximize the dice score:

$$L_{\text{Dice}}(p, t) = 1 - \frac{2|p \cdot t|}{|p|^2 + |t|^2}$$

where $p$ is the output of the network passed through a sigmoid activation, and $t$ is the binary ground truth. The binary segmentation is obtained by thresholding the sigmoid output with $t = 0.5$.

4.3.2 **Mixt**

In evaluation mode, the model will have to process slices where no hemorrhages will be present since the CT-scan does not contain hemorrhages everywhere in the scan and the whole volume has to be processed. With the view to train a more robust model able to ignore hemorrhage-free slices we decided to include some of the normal slices in the training set in addition to the ones with hemorrhage. Moreover we suppose that those additional data would enable the model to learn better features as it will be optimized on a larger variety of data structure that should help in discriminating ICH from other tissues. However, to keep the model’s optimization focused on the detection of hemorrhage we decided to assign a smaller weight $\alpha$ to normal slices in the loss. The loss function thus becomes:

$$L_{\text{Dice}}(p, t) = 1_{t=0} \alpha L_{\text{Dice}}(p, t) + 1_{t \neq 0} L_{\text{Dice}}(p, t)$$

where $1_{\text{condition}}$ is a boolean gate giving 1 is condition is true, else 0. We trained the U-Nets on all the ICH train slices together with twice as much normal slices randomly selected in the training volumes. We set $\alpha$ to 0.2 and use the same training parameters as above.

4.4 **Self-Supervised and Weakly-Supervised Pre-Training**

In a settings where the amount of available annotation is scarce like ours, semi-supervised approaches are effective frameworks enabling to use unlabeled data to extract meaningful features that should in turn assist the model in performing better on the few annotations available. The simplest form of semi-supervised learning is transfer learning [45, 46] where the model’s weights are initialized with those learned using the unlabeled/weakly-labeled data on a pretext task. We chose this kind of semi-supervised method because of its simplicity and absence of cumbersome scaling when dealing with joint training of several tasks. In the experiments described below we uses the large RSNA dataset for the pretext task learning and we use the few available annotated data for supervised fine-tuning on the segmentation task.
Figure 5: Context Restoration. Stage 1 represents the unsupervised feature learning using the context restoration task with a U-Net as network $\phi(\cdot)$. The orange bracket represents the knowledge transfer to the segmentation network of Stage 2. The segmentation is a U-Net 2.5D trained on CT slices.

4.4.1 U-Net 2-5D pre-training on Context Restoration

Principle Chen et al. [23] proposed a self-supervised method based on image restoration. For an image $x$ from a set of unlabeled data $X$, the rational of the method is to generate $\hat{x}$, a corrupted version of $x$, by sequentially swapping $N$ patches of dimension $h_{\text{swap}} \times w_{\text{swap}}$ on the image $x$. The model $\phi(\cdot)$ is then trained to reconstruct $x$ from the corrupted version $\hat{x}$ by using the MSE loss (with $x$ as ground truth). This pretext task is supposed to enforce the network to understand the spatial context of the image and thus learn meaningful features. The learned features can thus be used to initialize the whole model for the segmentation task. An overview of the method is presented on Fig.5.

Settings We defined $\phi(\cdot)$ as a U-Net identical to the one used on the supervised settings (see section 4.3). We uses 107'933 slices of the RSNA dataset labeled with ICH and as much slices without ICH. We keep 5000 images for validation and use the rest (around 211'000 slices) to train the context restoration task. Each image is corrupted with $N = 20$ swaps of dimension $h_{\text{swap}} \times w_{\text{swap}} = 20 \times 20$ pixels. The network is trained for 50 epochs with a batch-size of 32 using Adam as optimizer with the default parameters. The learning rate is set to 0.001 and is exponentially decayed with a base of 0.96 every epoch. The network’s weights are regularized with a $L_2$-penalty weight of 1e-6. Then, the learned weights are transferred to a supervised U-Net 2.5D trained similarly as in sec. 4.3.2.

4.4.2 U-Net 2-5D pre-training on Global-Local Contrastive

Principle Global Contrastive The contrastive self-supervised task enables to learn features by comparing images in a set. An encoder network $\psi$ is optimized to bring similar images closer in the embedding space while pushing dissimilar ones away from one another [27]. To identify similar and dissimilar sample without prior knowledge, an images $x$ form a comparison
Figure 6: Global-Local Contrastive. The overview of the global-local contrastive approach is presented as three stages. **Stage 1** Images augmented by $T_G$ are used to optimize the encoder part of the U-net using the global contrastive task. **Stage 2** The partial U-Net encoder is initialized with the features from Stage 1 highlighted by the orange bracket. Those weights are then frozen. Images augmented with $T_L$ are used to optimize the partial decoder on the local loss. In this representation, $N_r = 6$. **Stage 3** The features of the partial U-Net are transferred in a complete U-Net and the whole network is trained on the supervised segmentation task using the labeled data. Note that in the case of the global contrastive approach only, the learned features of Stage 1 are directly transferred to Stage 3.

set of images $\mathcal{X}$ is heavily augmented twice by the transformation $T_G$, resulting in two versions of the same images. The network encoder $\psi(\cdot)$ yields a representation $z$ of each images. For each transformed image $x_i$ in the comparison set, the network $\psi$ is trained to identify the corresponding image’s ($x_j$) representation form the set of $2N - 1$ other transformed images $\{x_k\}_{k \neq i}$. It can be transcribed into the InfoNCE loss that we will call the global contrastive loss $\mathcal{L}_{Global}$ for it processes global representation:

$$
\mathcal{L}_{Global} = \sum_{i=0}^{N} -\log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1[k \neq i] \exp(sim(z_i, z_k)/\tau)}
$$

(6)

where $z_i = \psi(x_i)$ and $sim(u, v) = \frac{u^T v}{\|u\|\|v\|}$

$\tau$ is a hyper-parameter called temperature. It has been demonstrated by Chen et al. [24, 26] and He et al. [25] that large comparison set as well as strong augmentation yield better learned
features for downstream tasks. There are different approaches to define the comparison set: Chen et al. [24] use the mini-batch as comparison set while He et al. [25] use a data bank updated over the training allowing larger comparison set without the computational cost of large mini-batch. Chen et al. [24] further showed that defining $\psi(\cdot) = MLP(\psi_e(\cdot))$ as the encoder to trained $\psi_e(\cdot)$ yields better performances. The contrastive optimization is thus performed with an additional MLP projection head, but only the features of $\psi_e$ are transferred downstream. The features extractor $\psi_e(\cdot)$ can then be used to initialize the encoder part of the segmentation network. Such an initialization should improve the performances by including quality prior knowledge about the input images to segment. The segmentation model is then trained similarly as in a supervised settings with labelled data. An overview of the procedure is presented on Fig. 6 in which Stage 2 is ignored.

**Settings Global Contrastive** We defined $\psi(\cdot)$ as the encoder part of the U-Net (see section 4.3) followed by a 2-layer MLP ($512 \rightarrow 512 \rightarrow 128$). Similarly as for the context restoration task, we uses 107’933 slices of the RSNA dataset labeled with ICH and as much slices without ICH. We keep 5000 images for validation and use the rest (around 211’000 slices) to train the global contrastive task. We defined the transformation $T_G$ as a sequential combination of random translation in range $[-15\%, 15\%]$, random rotation in the range $[-90°, 90°]$, random scaling in the range $[0.8, 1.2]$, random horizontal flipping 50% of the time, randomly adjusting the contrast 50% of the time, randomly adjusting the brightness 50% of the time, randomly blurring (gaussian) 50% of the time, and randomly cropping and resizing the image. The temperature hyper-parameter is set to $\tau = 0.1$. The network is trained for 50 epochs with a batch-size of 60 using Adam as optimizer with the default parameters. The learning rate is set to 0.001 and is exponentially decayed with a base of 0.96 every epoch. The network’s weights are regularized with a $L_2$-penalty weight of 1e-6. After transferring the contrastive weight to a complete U-Net, the U-Net 2.5D is trained similarly as in section 4.3.2.

**Principle Local Contrastive** Global contrastive pretraining is designed for learning features of an encoder only making it well suited for tasks involving only an encoder structure such as for classification. However in a segmentation procedure, the network is expected to expand the representation into a segmentation mask. As a results, in the global contrastive approach, around half of the network remains randomly initialized when fine tuning for the segmentation. Chaitanya et al. [28] recently proposed an adaptation of the contrastive pretraining for the segmentation of organs in CT-scans. In their approach, a contrastive loss is applied locally in the feature map of the decoding path. The core of the idea is that the decoding feature maps of two versions of an image should present local similarities: a region of the feature map should be similar between two versions, but dissimilar to other region of the feature map. As a result, no comparison is made with other images, only local region are compared. Formally, for an input image $x$, two augmented versions are obtained by applying the transformation $T_L$ twice to obtain $x_1$ and $x_2$. Note that we choose $T_L$ to not contains transformation that would strongly impair the localization of feature (such as horizontal flipping) since the feature localization is the core of the comparison. In order to compare features that are still of rather low level of abstraction, the comparison is performed on the feature map of an intermediate stage of the decoding. As a result, the network $\psi(\cdot)$ is an encoder followed by a partial decoder. The feature map of interest is then passed through a convolutional projection head (series of $1 \times 1$ convolutions) that have the same purpose as the MLP head in the global contrastive.
Therefore the network output $F_1 = \psi(x_1)$ has dimension $[H \times W \times C]$ in the 2D case. To compare local features, $N_r$ non-overlapping regions of size $[K \times K \times C]$ are randomly extracted from the feature map obtained for both $x_1$ and $x_2$ ($F_1$ and $F_2$). It yields $2N_r$ comparison units $f_{ij}$ where $i$ is the index of image version and $j$ is the index of the region. The local contrastive loss can thus be written as:

$$L_{\text{Local}} = \sum_{i=0}^{N_r} -\log \frac{\exp(\text{sim}(f_{1i}, f_{2i})/\tau)}{\sum_{k=1}^{2N_r} 1_{[k\neq i]} \exp(\text{sim}(f_{1k}, f_{2k})/\tau)}$$  \hspace{1cm} (7)

In their study, Chaitanya et al. [28] first optimized the encoder part of $\psi$ using the global contrastive task, then froze the encoder weight to train only the partial decoding on the local contrastive task. The learned encoding and decoding features can then be used to initialize a segmentation network such as a U-Net and fine-tune it with the labeled data. Note that in the case of a U-Net, skip connection are conserved in the partial network.

**Settings Global-Local Contrastive** We defined $\psi(\cdot)$ as the encoder part of the U-Net (see section 4.3) followed by three decoding blocks and two $1 \times 1$ convolution with $(64 \rightarrow 128 \rightarrow 32)$. Again, we uses 107’933 slices of the RSNA dataset labeled with ICH and as much slices without ICH. We keep 5000 images for validation and use the rest (around 211’000 slices) to train the local contrastive task. We defined the transformation $T_L$ as a sequential combination of random translation in range $[-15\%, 0.15\%]$, random rotation in the range $[-45^\circ, 45^\circ]$, random scaling in the range $[0.8, 1.2]$, randomly adjusting the contrast 50% of the time, randomly adjusting the brightness 50% of the time, randomly blurring (gaussian) 50% of the time, and randomly cropping and resizing the image. The temperature hyper-parameter is set to $\tau = 0.1$. We extract $N_r = 20$ regions of size $K \times K = 3 \times 3$ from the feature map to compute the local loss. The network is initialized with the weights learned with the global contrastive task (see above) and is then trained for 50 epochs with a batch-size of 24 using Adam as optimizer with the default parameters. The learning rate is set to 0.001 and is exponentially decayed with a base of 0.96 every epoch. The network’s weights are regularized with a $L_2$-penalty weight of $1e-6$. After transferring the contrastive weight to a complete U-Net, the U-Net 2.5D is trained similarly as in section 4.3.2.

### 4.4.3 U-Net 2-5D pre-training on Binary Classification

**Principle** Self-supervised pre-training approaches are designed to learn relevant features in an unsupervised way. Those feature should help the segmentation network to get an initial idea of what an ICH is compared to healthy scans. In our case, we have access to the large RSNA dataset that is labeled for classification. We can therefore pre-train the encoder part directly on a binary classification task which provide an explicit learning signal to discriminate the hemorrhages from the rest. We suppose this weakly-supervised pre-training would lead to better initial features for the segmentation that would transpose into better segmentation performances. This approach can be categorized as a weakly-supervised model since we make use of the classification labels of the RSNA dataset. Practically, the classifier $\psi(\cdot)$ is trained on the binary cross entropy. The learned convolution weights are then transferred to the segmentation network that is then optimized on the segmentation task. An overview of the method is presented on Fig.7.
Stage 1 Classification

Figure 7: Classification Pre-Training. Stage 1 represents the weakly-supervised feature learning using a classification task with the encoder part of a U-Net and a MLP as network $\psi(\cdot)$. The orange bracket represents the knowledge transfer to the segmentation network of Stage 2. The segmentation is a U-Net 2.5D trained on slices.

**Settings** We defined the classifier $\psi(\cdot)$ as the encoder part of the U-Net (see section 4.3) followed by an average pooling layer, a 3-layer MLP ($512 \to 1024 \to 256 \to 2$) and a softmax layer as final activation. All the ICH positive images of the RSNA dataset are used together with as much of the ICH negative images (i.e. 1:1 ratio). We used 5000 images for validation and use the rest for training. The model is thus trained with a similar setting as for the self-supervised pre-training tasks which enables a fair comparison of the tasks. We weighted the two classes accordingly in the cross-entropy loss. The model is then trained for 50 epochs with a batch-size of 64 using Adam as optimizer with the default parameters. The learning rate is set to 0.001 and is exponentially decayed with a base of 0.96 every epoch. The network’s weights are regularized with a $L_2$-penalty weight of 1e-6. After transferring the classification weight to a complete U-Net, the U-Net 2.5D is trained similarly as in section 4.3.2.

4.4.4 U-Net 2-5D pre-training on Multi-Label Classification

**Principle** The RSNA dataset is not only labeled for the binary classification task but also for the classification of ICH types allowing more complex weakly-supervised pre-training scheme. There are five types of ICH (Intraparenchymal, Intraventricular, Epidural, Subdural and Subarachnoid) and each CT slices may contain more than one type. Such a classification task is known as multi-label classification in which the classifier $\psi(\cdot)$ is trained to maximize its sigmoid outputs response for all non-exclusive positives classes through class-wise binary cross-entropy loss or through a dice loss. In our case, we chose to use 7 different classes: no-ICH, ICH and the five ICH types. We decided to include the ICH class to train the model explicitly in differentiating non-ICH and ICH slices as in the binary classification task. On top of that, the classification of the five types should push the models in differentiating the nature of the hemorrhage and therefore in learning better features. Similarly as in the binary case, the learned weights are then transferred to the segmentation network that is then optimized on the segmentation task.

**Settings** We defined the classifier $\psi(\cdot)$ as the encoder part of the U-Net (see section 4.3)
followed by an average pooling layer, a 3-layer MLP ($512 \rightarrow 1024 \rightarrow 256 \rightarrow 7$) and a sigmoid layer as final activation. We used the same RSNA subset as in the binary classification pre-training (see section 4.4.3). We trained the model either on a weighted binary cross-entropy (CE) loss where each possible class is seen as a binary classification task, or on the dice loss (DL) where the model is trained to generate an output that matches the 7-neurons ground truth. The model is then trained with the same settings as in the binary classification task (see section 4.4.3). After transferring the classification weight to a complete U-Net, the U-Net 2.5D is trained similarly as in section 4.3.2.

4.5 Unsupervised Inpainting Based Segmentation

4.5.1 General Idea

The methods described above are relying on a low number of labelled samples and exploit the availability of many unlabeled or weakly labeled ones. In this section we propose an approach for a more drastic set up in which there is no access to labeled data for segmentation. The ICH is therefore detected in an unsupervised way. The core of the approach boils down to the observation that an ICH (i.e. a lesion) can be viewed as an anomaly in a brain CT-scan [31, 32, 33, 34]. Therefore, being able to detect and localize anomalies should allow to detect brain hemorrhages. Many algorithms for anomaly localization already exists, such as the autoencoder [35], the AnoGAN [47], the SMAI [37], or the recent FCDD [38]. However, we propose here a different approach based on image inpainting, a task in which the computer has to fill in masked regions of an image to restore the original image [48]. We suppose that a neural network trained to inpaint only images considered as normal would not be able to restore properly an anomaly, which can then be detected through the larger reconstruction error. Our approach relies on a similar idea as in SMAI [37] but we tackle the problem at the pixel resolution instead of the super-pixel.

First the inpainter network and its training are described in section 4.5.2, followed by the description of our anomaly detection module in section 4.5.3. Then we present two other approaches for anomaly localization: the Auto-Encoder (AE) and Fully-Convolutional-Data-Descriptor (FCDD) that we use as baseline.

4.5.2 SN-PatchGAN Inpainter

**Architecture** To inpaint the images, we adopt the SN-PatchGAN architecture presented by Yu et al. [49] with the further improvements presented in [50]. The inpainter is the generator of the GAN framework and is composed by a sequence of two encoder-decoder architectures. The first encoder-decoder takes as input the masked image as well as the mask itself and aims at producing a coarse inpainting of the image, that is then improved by the second encoder-decoder. Noteworthy that the final corrected image is kept only where the image is masked, while the rest of the image is kept as the original. Instead of using standard convolutions the encoder-decoders use gated-convolution [50] to better process and integrate the mask channel in the inpainting. In a gated convolution, the input is processed by two parallel paths (1. a sequence of a convolution, batch normalization and leaky-ReLU 2. a sequence of a convolution and sigmoid) that are then merged together. In the second encoder-decoder, an additional attention layer is used to promote the use of distant features (and not just proximal ones) with the goal of letting the inpainter borrow textures from further away in the image and yield a
Figure 8: SN-PatchGAN. The overview of the used SN-PatchGAN framework is presented here with the generator on top and the discriminator below. The discriminator is fed either with the original image or with the inpainted one. The number in gray below the convolution block represents the number of channels. The relative heights of the convolution blocks are proportional to the feature map size.

more realistic inpainting. Yu et al. used a complex contextual attention layer in their implementation [49] that we decided to replace with a simpler self-attention layer proposed by Zhang et al. [51] detailed on Fig. 8.

The discriminator used to train the generator under the GAN framework [36] is the same as the one used by [50], it takes as input an image (real or fake) as well as the mask and is composed of standard convolution (kernel size of 5 with stride of 2) followed by leaky-ReLU activations. To stabilize the training of the discriminator, the convolutions’ weights are spectrally normalized [52] similarly as in [50]. In addition, unlike in [50], before the last convolution, a self-attention layer is added to give a broader receptive-field to each output neuron. An overview of the SN-PatchGAN is presented on Fig. 8.

Training Similarly as in [49, 50], the generator’s weights are optimized by a combination of two loss functions. The first one enforces the coarse and final inpainted images to be close to the original one through a $L_1$ loss. However, as we want the inpainter to inpaint accurately the border of the mask (junction with the real image) while letting him hallucinate what’s in the middle of the mask, we use a discounted $L_1$ loss [49] that gives more weight to errors on the border: $L_{id}(\cdot) = L_1(\cdot) \odot \gamma I(mask)$ where $I(x)$ is the euclidean distance of the pixel $x$ to the nearest known pixel, and $\gamma$ is a hyper-parameter. The second loss functions is the Hinge loss to motivate the generator in fooling the discriminator: $L_G = -\mathbb{E}_{z \sim P_{masked}(z)}[D(G(z))]$ where $P_{masked}(\cdot)$ is the distribution of masked images, $G(\cdot)$ is the generator and $D(\cdot)$ is the discriminator. The overall loss function on the generator is a weighted sum of those two loss functions: $L_{gen}(I_{coarse}, I_{fine}) = \lambda_1 L_{id}(I_{coarse}) + \lambda_1 L_{id}(I_{fine}) + \lambda_2 L_G(I_{fine})$ where $I_{coarse}$ and $I_{fine}$ stands respectively for the intermediate and final inpainted image where the outputs is only kept on the masked part of the input. The discriminator’s weights are optimized to distinguish inpainted images from real ones through a Hinge loss: $L_D = \mathbb{E}_{x \sim P_{real}(x)}[ReLU(1 - D(x))] + \mathbb{E}_{z \sim P_{masked}(z)}[ReLU(1 + D(G(z)))]$
where \( P_{\text{masked}}(\cdot) \) is the distribution of masked images and \( P_{\text{real}}(\cdot) \) is the distribution of real images.

**Settings** We trained the SN-PatchGAN to inpaint normal CT-scans (i.e. without hemorrhage) with a sample of 20,000 normal images from the RSNA dataset. The free-form inpainting masks are generated on the fly to mimic free-form hand drawing. A free form mask is a set of \( N \) elements, each composed of \( N_v \) vertexes (segments) of width \( w \) with various length \( l_i \) (for each segment) and separated by an angle \( \alpha_i \). In addition, \( N_s \) spheres of radius \( r_i \) are added to the mask. In our set-up, \( N \) is sampled uniformly from \([1,4]\), \( N_v \) is sampled uniformly from \([15,30]\), \( w \) is sampled uniformly from \([15,25]\), \( l_i \) is sampled uniformly from \([15,40]\), \( \alpha_i \) is sampled uniformly from \([0,2\pi]\) (in radian), \( N_s \) is sampled uniformly from \([0,15]\) and \( r_i \) is sampled uniformly from \([1,6]\). Each drawing starting position is drawn from a 2D normal distribution to favor inpainting of the center part of the image. The GAN is trained for 50 epochs with a batch-size of 7 (\( \approx 140\,000 \) gradient steps). The generator and discriminator are optimized with an initial learning rate of respectively \( 1e^{-4} \) and \( 4e^{-4} \), both exponentially decayed with a base of 0.97 at every epoch. Both are optimized using the Adam algorithm with \( \beta_1 = 0.5 \) and \( \beta_2 = 0.999 \) to give more freedom in the gradient steps (less momentum). \( \gamma \) is set to 0.995 for the \( L_1 \) weights computation. In the generator loss \( L_G \), \( \lambda_1 \) and \( \lambda_2 \) were empirically chosen to be respectively 1.0 and 0.05. The generator and discriminator’s weights are regularized with a \( L_2 \)-penalty weight of \( 1e^{-6} \).

4.5.3 Inpainting Anomaly Detection Module

**Method Principle** Let \( x \) be an input 2D image with an arbitrary number of channels. Let \( \Psi(x_m, m) \) be an inpainting neural network trained on normal images and taking a image \( x_m \) masked and a binary mask \( m \) as input to output an inpainted version \( \hat{x} \) on the mask \( m \). The core idea of the anomaly detection module is that anomalous pixels should display a large reconstruction error. In order to compute the pixel-wise inpainting error, a sliding occlusion window can be applied to the input image and for each occlusion, the image is inpainted and the error is computed as the difference with the original image. Assuming that the inpainter relies mostly on local structure to inpaint the image and to speed up the process, we propose to use a grid of occlusion instead of a single window. As a results the image \( x \) is inpainted with a set of grids of dimension \( (g_h, g_w) \) shifted by \( g_s \) pixel between each grid. The consecutive inpainting with shifted grids ensure that each pixel is inpainted several times. We therefore obtain multiple error measures for each pixels: \( \epsilon \). Note that with multiple color channels, the inpainting errors \( \epsilon \) is averaged over the channel to end up with a 3-dimensional array. With a sample of error measures per pixel, we can estimate the underlying distribution of the pixel-error and compare it to what is assumed to be the distribution of normal pixel. Large differences should highlight anomalous regions while normal region should present small differences. Practically, we assume that the observed error distribution per pixel is normally distributed with the observed mean and variance \( (\mu_{obs}, \sigma_{obs}) \). We also assume that a non-anomalous pixel error should be normally distributed with a mean of 0 and a given standard deviation \( \sigma_0 \). We estimate \( \sigma_0 \) as the 25% quantile of \( \sigma_{obs} \) (\( \sigma_0 = q_{25}(\sigma_{obs}) \)) with the underlying assumption that there are at least 25% of the image that is not anomalous. The distance between the observed and expected error is then measure through the Kullback-Leibler divergence \( D_{KL}(\cdot, \cdot) \) for each pixels. With the above assumptions of normally distributed errors we can compute efficiently the pixel-wise \( D_{KL}(\cdot, \cdot) \) with the approximation for normally distributed samples:
Figure 9: Inpainting AD Module. The Inpainting Anomaly Detection Module (IAD) processing pipeline is summarized as a flowchart. A Detailed flow chart of $I_{\text{Inpainting AD}}(\cdot)$ which process one image to output one segmentation of the anomalies. It is composed of the generation of the initial anomaly mask (top row) followed by its iterative adjustments (bottom row). Note that the bottom row is repeated multiple times to get the final anomaly mask. B Overview of the robust procedure that compute an anomaly mask for a set of transformed image $T_i(x)$. The mean of all the anomaly masks generate an anomaly heatmap. $T(t_l, t_h, \cdot)$ represent the hysteresis threshold. $\oplus$ is the morphological dilation. $\bullet$ is the morphological closing. $\odot$ is the morphological opening. $\oslash$ is the Hadamard product. $D_{KL,N}(\cdot)$ is the Kullback-Leibler divergence approximation for normal samples.

$$D_{KL,N}(\mu_{obs}, \sigma_{obs}, \mu_0, \sigma_0) = \log \left( \frac{\sigma_0}{\sigma_{obs}} \right) + \frac{\sigma_{obs}^2 + (\mu_{obs} - \mu_0)^2}{2\sigma_0^2} - \frac{1}{2}$$

The obtained distance map, $D_0$, should present large values for abnormal pixels and lower value for normal ones. We can therefore threshold it to obtain an anomaly segmentation. Instead of applying a simple threshold, we make use of a hysteresis threshold $T(t_l, t_h, \cdot)$ which uses a lower ($t_l$) and an upper ($t_h$) threshold. The value of the pixel is set to 1 if the pixel value is above the upper thresholds. If the pixel value is between the lower and upper thresholds and the pixel is adjacent to pixels above the upper threshold, the pixel is set to 1, else it is set to 0. If the pixel value is below the lower threshold, it is set to 0. We decide to use fractions of the interquartile range of $D_0$ to define the lower and upper thresholds. Indeed, we want to extract upper outliers in $D_0$, if we observe the distribution of value in $D_0$ through a boxplot,
the upper outlier are usually defined as point lying further than \( q_{75}(D_0) + 1.5 \cdot IQR \) where \( IQR = q_{75}(D_0) - q_{25}(D_0) \) and \( q_i(D_0) \) is the \( i \)-th quantile of \( D_0 \). We thus define the lower threshold to be \( t_l = q_{75}(D_0) + \alpha_{01} \cdot IQR \) and the upper threshold to be \( t_u = q_{75}(D_0) + \alpha_{02} \cdot IQR \). \( \alpha_{02} \) is a hyper-parameter that allow to define what is considered an outlier, and \( \alpha_{01} \) allows to define what is accepted to be potentially anomalous depending on the relative proximity to other anomalies. For example, if \( \alpha_{02} \) is set to 1.5 and \( \alpha_{01} \) to 0.0, it means that pixels further away from the whisker of the boxplot will be selected and pixels between the box and the whisker will be selected only if they are close to pixel further away form the whisker.

The thresholding yields a first segmentation of the anomaly: \( M_0 \). However, since the inpainter network \( \Psi \) is trained to generate structures, it has some freedom in the structures it draws and those structures will most likely not perfectly match the original normal area of the image. There will therefore be potential large inpainting errors of normal structures and therefore many false positives in \( M_0 \). If we suppose that the anomalies have been detected in \( M_0 \), we can correct the original image by using \( M_0 \) as inpainting mask to produce an anomaly-free image \( \hat{x} = \Psi(x \odot (1 - M_0), M_0) \). To ensure that the anomalies are well covered by the mask, \( M_0 \) is morphologically dilated by a disk or radius \( r_{d0} \) before correcting the image \( x \). We can then repeat the same anomaly detection process on this normal image \( \hat{x} \) to obtain a mask \( M_{0 \text{normal}} \) that highlight regions that are naturally producing high inpainting errors. \( M_0 \) can then be updated to exclude those regions and reduce the number of false positives: \( M_1 = M_0 \odot (1 - M_{0 \text{normal}}) \).

In addition, the resulting segmentation mask, \( M_1 \), is cleaned to remove noisy structures by morphologically closing and opening it respectively with a disk of radius \( r_c \) and \( r_o \). The new anomaly segmentation \( M_1 \) is therefore more accurate but likely not perfect since we defined the normal image as the one corrected on \( M_0 \). However we can get a more accurate normal image \( \hat{x} \) and a better measure of what regions are naturally producing high inpainting errors by inpainting the original image with the new mask: \( M_1 \), and repeating the process and so on. We call this phase the iterative adjustment of the mask in which the estimation of region with naturally high inpainting error is repeated \( n_{\text{iter}} \) times to improve \( M_0 \).

The pipeline described above generate a single anomaly segmentation mask and its quality is likely to not be perfect since the inpainter \( \Psi \) is likely not perfect too. Therefore the inpainter may have trouble on some parts of an image because the arrangement of the image together with what is masked by the grid may make the task quite hard. For example, in a case where the network is trained to inpainted faces, if an eye is masked, the network will probably use the other eye to inpaint the masked one. But if the two eyes are masked, it is hard even for a human to come up with the right eye shape and color as the one that is hidden. To cope with this limitation, we decided to simply repeat the anomaly segmentation described above \( (F_{\text{InpaintingAD}}(\cdot)) \) on various spatially transformed versions of the image \( x \). This way the inpainter is presented with different context and should be able to successfully inpaint normal regions of the image in most cases while systematically having troubles in inpainting abnormal regions. This approach is similar to a majority voting scheme for robust classification. Practically, the input image \( x \) is transformed by \( N \) spatial transformation \( \mathcal{T}_j \) with \( j \in [1 : N] \). The resulting anomaly segmentation \( M_{aj} \) is transformed back into the original format by the inverse transformations \( \mathcal{T}_j^{-1} \). An anomaly heat-map \( x_{AD} \) is obtained by averaging all the \( N \) anomaly segmentations: \( x_{AD} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{T}_j^{-1}(\mathcal{T}_j(F_{\text{InpaintingAD}}(x))) \). A final anomaly mask can then be computed by applying a hysteresis threshold to \( x_{AD} \). The value chosen for \( t_l \) and \( t_u \) are intuitive as they represent the percentage of detection over the different transformations:
A pixel will be set as anomalous if it appears at least \( t_h \% \) of the time or if it appears at least \( t_l \% \) of the time and is close to region appearing at least \( t_h \% \) of the time. An overview of the whole processing is presented on Fig. 9.

**Method Validation** The proposed method is an unsupervised anomaly segmentation method. Prior of applying it to the special case of ICH segmentation, we decided to validate the workingness of our approach on a benchmark dataset. We chose to use the MV-Tec dataset [54] to validate our method. This dataset contains photographs of 15 different categories of industrial elements (10 objects and 5 textures) with a large variety of type of anomalies labeled at the pixel level. For each categories, the train set is composed of around 220 normal images while the test set contains normal and different type of abnormal images. This dataset has the advantage of offering a variety of anomalies which enables us to assess where our method works best, but it also enable to compare easily our approach with other anomaly segmentation methods that use this benchmark dataset.

We chose to apply our method separately on each of the 15 categories. We used a SN-PatchGAN as inpainter (see section 4.5.2) trained to inpaint the normal images from the train set. During training, all images are augmented with random translation and rotation. For texture, the output of the spatial transformations are filled with a reflection of the image, while for objects we just fill with zeros or the nearest value. We further transform the images with horizontal and vertical flip when they do not conflict with the possible anomalies (for example, in the metal nut category, a flipped image is considered as anomalous). Details of the SN-PatchGAN training parameters can be found in table 5 in the annexes. We then apply our anomaly detection methods separately for each categories. Parameters details can be found on table 6 in the annexes. As proposed in [54] we report the anomaly segmentation performances both by the mean intersection over union (IoU) of the generated masks \( M_a \) with the ground truth, and by the mean area under the ROC-curve (AUC) computed on the anomaly heat-map \( x_{AD} \) (i.e. the IoU and the AUC are computed for each image at pixel level, then the mean is taken over all the images). In order to select the method’s hyper-parameters without bias toward anomalies, we propose to empirically test the method on one or few normal images and tune the parameters to not detect anomalies. This way, we get parameters values that consider the inner variation of normal images structures. Nonetheless, it can still be hard to find the perfect set of hyper-parameter.

**ICH Application** The proposed anomaly localization method is then applied on the specific case of ICH segmentation using the inpainter described in section 4.5.2. The input image is inpainted with grids element of size \( g_h \times g_w = 32 \times 32 \). Between two inpainting grid, the grid elements are shifted by \( g_s = 8 \) pixels resulting in 16 error estimations per pixels. The initial distance map \( D_0 \) is thresholded with \( \alpha_{01} = 0.0 \) and \( \alpha_{02} = 3.0 \) in order to detect large error while still keeping nearby structure with smaller errors. \( M_0 \) is dilated by a disk of radius \( r_{d0} = 3 \) before correcting the image to ensure that the anomalies are well hidden. Note that the first correction is made by patch of \( 128 \times 128 \) pixels to ensure the availability of enough known pixel to the inpainter. Then \( M_0 \) is refined through maximum \( n_{iter} = 20 \) iteration steps during which, the thresholding of \( D_i \) is made with \( \alpha_1 = 1.0 \) and \( \alpha_2 = 1.5 \) to detect many normal pixel per iteration while still being conservative. The imporved mask \( M_i \) is then morphologically closed and opened respectively with disk of radius \( r_c = 1 \) and \( r_o = 1 \) with the assumption that ICH anomalies are usually bigger than three pixels wide. To speed up the computation,
the iterative process is stopped if the new anomaly segmentation $M_i$ does not change more than 25 pixels. The anomaly segmentation is repeated over 10 transformation of the image: the identity, the horizontally-flipped image, the rotation of the image by $-15^\circ$, $-7.5^\circ$, $7.5^\circ$, or $15^\circ$ combined or not with horizontal flipping. A hysteresis threshold is applied to the resulting anomaly head-map with a lower threshold of $t_{lr} = 0.45$ and an upper threshold of $t_{hr} = 0.75$.

However, the proposed anomaly detection approach is extremely computationally intensive. To reduce the computation time on our dataset of 2814 images we chose in the first time to apply our method only on the CT slices containing an ICH (315 images) which assumes that slices can be perfectly classified. As it may not be the case in a real world situation, we also chose to train a classifier on the RSNA dataset to decide which images of the dataset should be processed by our method. We trained a ResNet-18 [55] on the RSNA dataset for the binary classification of ICH. We use 277'139 images in the test set among which 97'139 present an ICH, and 30'794 images to validate the classification threshold. The ResNet-18 is trained for 50 epoch with a batch-size of 96. The weights are optimized with the binary cross-entropy loss, and the Adam optimizer with an initial learning rate of 5e-4 exponentially decayed with a base of 0.96 after each epoch. Following the validation analysis (see Fig. 17 and table 4), we chose a classification threshold of $t = 0.25$ to have a large recall and process the maximum number of slices with ICH. This threshold results also in more false positives, but it should not be a problem since we just want to ease the computation time and the anomaly detection method should be able to not detect anomalies on those.

4.5.4 Auto-Encoder Baseline

In order to gauge the capabilities of our method on the ICH localization we train an Auto-Encoder (AE) as baseline. The AE trained to reconstruct normal images should have trouble in reconstructing new features, such as an ICH, resulting in a higher reconstruction error. We decided to use the architecture proposed by Baur et al. [34] to detect white matter lesion in MRI scans. The encoder consists of four convolutions layers with kernel size of 5 with Batch-Normalization layer and ReLU. The three later each reduce the feature map size by a factor two while increasing the number of channels by a factor two. Then the bottleneck feature map is obtained with a convolution with a kernel size of three that reduces the feature map by a factor
two but also reduce the number of channels to 32. The encoder’s bottleneck represents thus a 8-fold compression of the image. The decoder is a symmetry of the encoder using transposed convolutions. The reconstruction is then passed through an hyperbolic tangent activation function. An overview is presented on Fig.10. For a fair comparison with our method, the AE is also trained on 20k healthy CT-slices of the RSNA dataset. We train the AE on the sum of the L1 and L2 reconstruction loss for 75 epochs followed by 25 epoch on the the equally weighted sum of L1, L2 and gradient difference loss (GDL) similarly as in [34]. The GDL allows to get crisper reconstruction by enforcing the image gradient to be similar. We used a batch size of 32 with an initial learning rate of 0.005 exponentially decayed with a base of 0.96 every epoch. The gradient descent is preformed with the Adam optimizer with the default parameters. The network weights are regularized with an L2 penalty weight of 1e-6. We use the pixel-wise absolute value of the reconstruction as anomaly map $A$ and apply an hysteresis threshold with $t_{low} = q_{75}(A) + 1.5 \cdot IQR$ and $t_{high} = 2q_{75}(A) + 3.0 \cdot IQR$ where $IQR = q_{75}(D_0) - q_{25}(D_0)$. This threshold startegy assumes that most pixels are normal. It should thus extract every pixels that are considered as outlier in a boxplot of the errors. Noteworthy that the we also use the ResNet-18 classifier upstream of the anomalies detection to let the AE benefits from the same advantages as our approach.

4.5.5 FCDD Baseline

In the field of anomaly detection, Ruff et al proposed an approach specifically designed for the task [56, 57]. The rational is to train an encoder $\phi$ to learn a representation where normal samples are enclosed in a compact hyper-sphere. The model should thus fail to map anomalous sample in the sphere and the distance to the center can be used as detection score. However this approach does not allow to localize anomalies. That is why Liznersky et al [38] proposed an extension: Fully-Convolutional-Data-Descriptor (FCDD). It follows the same principle but makes use of a fully convolutional network $\phi: \mathbb{R}^{h \times w \times c} \rightarrow \mathbb{R}^{u \times v}$ that yield a feature map of size $u \times v$ with a similar topology as the input image $X \in \mathbb{R}^{h \times w \times c}$. The network is trained on the hyper-sphere-classifier loss to minimize the values in this feature map for normal samples and maximize it for anomalous samples:

$$L_{HSC} = \frac{1}{n} \sum_{i=1}^{n} (1 - y_i) \frac{1}{u \cdot v} \|A(X_i)\|_1 - y_i \log \left( 1 - \exp \left( - \frac{1}{u \cdot v} \|A(X_i)\|_1 \right) \right)$$  

(9)
where $A(X_i) = \sqrt{\phi(X_i)^2 + 1} - 1$ and $y_i$ equals 1 for abnormal samples and 0 for normal ones. $\|x\|_1$ is the sum of the entries in $x$. Because the topology is preserved by the network, the presence of an anomaly in the input should induce a large value in the feature map $A(X)$ only for output pixels with a receptive covering the anomaly. The feature map is thus a down-scaled anomaly heat-map. Up-sampling $A(X)$ gives a full resolution anomaly heat-map. Liznersky et al. advocate the use of a transpose convolution with a fixed Gaussian kernel (with a size dependant on the receptive field) to up-sample $A(X)$. The network $\phi$ can be optimized in a semi-supervised way using real anomalies and unknown images assumed normal, but it can also be trained in an unsupervised way using artificial anomalies which allow a proper training as they argue. The artificial anomalies can be anything, for example they use random drawing of stain on the MV-tec dataset.

In our experiment, we adopt a similar architecture as Liznersky: a VGG-like network composed of seven convolution layers combined with batch-normalization layers and ReLU. The feature map size reduction is performed using Max-Pooling layers. The up-sampling is performed via a transposed convolution with a Gaussian kernel of size $K \times K$ (depending on receptive field) with a fixed standard deviation of 8.0. A detailed scheme of the architecture and of the method is provided on Fig.11. Generated anomaly heat-maps need to be scaled between 0.0 and 1.0. With a given evaluation dataset, heat-maps are scaled using a min-max normalization where the minimum and maximum values are picked respectively as the minimum quantile 2.5% over the batch and the maximum 99.9% quantile over the batch. Then the rescaled heat-map is clamped between in $[0; 1]$. We then get a segmentation by thresholding the heat-map with a fixed threshold $t = 0.5$. In all experiment, $\phi$ is optimized for 100 epochs with a batch-size of 64 and an initial learning rate of 0.005 exponentially decayed with a based of 0.96 every epoch. Adam is used with the default parameters to perform the gradient descent and the network’s parameters are L2-regularized with a weight of 1e-6. The difference between experiments resides in different data settings. First, to compare FCDD with our inpainting method, we chose to optimize the network $\phi$ using only 20k normal CT slices and generate artificial anomalies on 50% of them. Since we are targeting ICH, artificial anomalies are generated from healthy slices to mimic ICH by drawing some ellipses of random sizes with a bright pixel intensity combined with Gaussian noise to add some texture. A samples of generated anomalies is presented in Fig.23 in the annexes. Note that for a fair comparison we also use the ResNet-18 classifier upstream of the FCDD when applied to the segmentation dataset. Second, since we have access to explicit examples of hemorrhages labelled for classification, we can use them as true anomalies to drive the learning exactly in the direction we are interested in. We explore two different ratios of ICH in the training dataset: 2% or 5% with a total of 50k training slices. We also exploit the FCDD method with a similar data settings as in the binary classification pre-training where we have access to 200k samples with 50% of ICH.

4.6 Semi-Supervised Inpainting-attention U-Net 2.5D

With the anomaly heat-map generated with inpainting, we further explore another semi-supervised approach for ICH segmentation. We suppose that feeding a standard U-Net with the input image together with the anomaly heat-map as additional channel may help the U-Net in better extracting ICH. This heat-map channel can be seen as an attention channel pinpointing to the U-Net regions of interest. We train a U-Net 2.5D in a similar fashion as in section 4.3 with the only difference that the input has two channels: the original image stacked with the anomaly heat-map generated with the inpainting anomaly detection module.
5 Results & Discussion

5.1 Brain Extraction

In the first place, we briefly comment the results from the brain extraction procedure. The mean dice, precision and recall scores over the 5 cross-validation folds are presented on table 1 for the models trained on the bone and the tissue window, as well as for the resulting union of the two segmentation. At first glance, the tissue model appears to perform slightly better in term of dice (99.39 ± 0.26%) and precision (99.53 ± 0.46%) while the union presents the highest recall (99.48 ± 0.56%). However, those values does not completely reflect the models’ capabilities and flaws as highlighted by the segmentation results of a CT-scan from the ICH segmentation dataset in Fig.12. Indeed, the bone window model shows reduced detection on boundary slices (i.e. top and bottom of brain) but it is not bothered by the hemorrhage. On the other hand, the tissue model provides a better segmentation on the boundary slices but it is hindered by the presence of the hemorrhage and fails to detect it as part of the brain. As a results, the union of both model takes the best of both and yield a satisfying segmentation of the brain even in the presence of hemorrhage.

<table>
<thead>
<tr>
<th></th>
<th>Dice</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bone window</td>
<td>99.08 ± 0.33%</td>
<td>99.20 ± 0.43%</td>
<td>98.97 ± 0.95%</td>
</tr>
<tr>
<td>Tissue window</td>
<td>99.39 ± 0.26%</td>
<td>99.53 ± 0.46%</td>
<td>99.26 ± 0.78%</td>
</tr>
<tr>
<td>Union</td>
<td>99.25 ± 0.15%</td>
<td>99.03 ± 0.44%</td>
<td>99.48 ± 0.56%</td>
</tr>
</tbody>
</table>

Table 1: Brain segmentation Performances in term of Dice, Precision and recall at the slice level. Values are given as the mean ± 1.96std over the 5-Folds.
| Textures       | IoU | AUC | | IoU | AUC | | IoU | AUC | | IoU | AUC | | IoU | AUC |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Carpet        | 0.69 | 0.38 | 0.34 | 0.27 | - | 0.16 |
| Grid          | 0.88 | 0.83 | 0.04 | 0.84 | - | 0.27 |
| Leather       | 0.71 | 0.67 | 0.34 | 0.44 | - | 0.38 |
| Tile          | 0.04 | 0.23 | 0.08 | 0.14 | - | 0.20 |
| Wood          | 0.36 | 0.29 | 0.14 | 0.54 | - | 0.23 |
| Bottle        | 0.15 | 0.22 | 0.05 | 0.26 | - | 0.45 |
| Cable         | 0.01 | 0.05 | 0.01 | 0.04 | - | 0.11 |
| Capsule       | 0.09 | 0.11 | 0.04 | 0.39 | - | 0.25 |
| Hazelnut      | 0.00 | 0.41 | 0.02 | 0.52 | - | 0.37 |
| Metal nut     | 0.01 | 0.26 | 0.00 | 0.07 | - | 0.14 |
| Pill          | 0.07 | 0.25 | 0.17 | 0.24 | - | 0.17 |
| Screw         | 0.03 | 0.34 | 0.01 | 0.57 | - | 0.31 |
| Toothbrush    | 0.08 | 0.51 | 0.07 | 0.61 | - | 0.26 |
| Transistor    | 0.01 | 0.22 | 0.08 | 0.06 | - | 0.08 |
| Zipper        | 0.10 | 0.13 | 0.01 | 0.46 | - | 0.29 |
| Mean          | 0.22 | 0.33 | 0.09 | 0.36 | - | 0.24 |

<table>
<thead>
<tr>
<th></th>
<th>AE (sim) [54]</th>
<th>AE (L2) [54]</th>
<th>AnoGAN [54]</th>
<th>SMAI [37]</th>
<th>FCDD [38]</th>
<th>ADI (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carpet</td>
<td>0.69</td>
<td>0.38</td>
<td>0.34</td>
<td>0.27</td>
<td>-</td>
<td>0.16</td>
</tr>
<tr>
<td>Grid</td>
<td>0.88</td>
<td>0.83</td>
<td>0.04</td>
<td>0.84</td>
<td>-</td>
<td>0.27</td>
</tr>
<tr>
<td>Leather</td>
<td>0.71</td>
<td>0.67</td>
<td>0.34</td>
<td>0.44</td>
<td>-</td>
<td>0.38</td>
</tr>
<tr>
<td>Tile</td>
<td>0.04</td>
<td>0.23</td>
<td>0.08</td>
<td>0.14</td>
<td>-</td>
<td>0.20</td>
</tr>
<tr>
<td>Wood</td>
<td>0.36</td>
<td>0.29</td>
<td>0.14</td>
<td>0.54</td>
<td>-</td>
<td>0.23</td>
</tr>
<tr>
<td>Bottle</td>
<td>0.15</td>
<td>0.22</td>
<td>0.05</td>
<td>0.26</td>
<td>-</td>
<td>0.45</td>
</tr>
<tr>
<td>Cable</td>
<td>0.01</td>
<td>0.05</td>
<td>0.01</td>
<td>0.04</td>
<td>-</td>
<td>0.11</td>
</tr>
<tr>
<td>Capsule</td>
<td>0.09</td>
<td>0.11</td>
<td>0.04</td>
<td>0.39</td>
<td>-</td>
<td>0.25</td>
</tr>
<tr>
<td>Hazelnut</td>
<td>0.00</td>
<td>0.41</td>
<td>0.02</td>
<td>0.52</td>
<td>-</td>
<td>0.37</td>
</tr>
<tr>
<td>Metal nut</td>
<td>0.01</td>
<td>0.26</td>
<td>0.00</td>
<td>0.07</td>
<td>-</td>
<td>0.14</td>
</tr>
<tr>
<td>Pill</td>
<td>0.07</td>
<td>0.25</td>
<td>0.17</td>
<td>0.24</td>
<td>-</td>
<td>0.17</td>
</tr>
<tr>
<td>Screw</td>
<td>0.03</td>
<td>0.34</td>
<td>0.01</td>
<td>0.57</td>
<td>-</td>
<td>0.31</td>
</tr>
<tr>
<td>Toothbrush</td>
<td>0.08</td>
<td>0.51</td>
<td>0.07</td>
<td>0.61</td>
<td>-</td>
<td>0.26</td>
</tr>
<tr>
<td>Transistor</td>
<td>0.01</td>
<td>0.22</td>
<td>0.08</td>
<td>0.06</td>
<td>-</td>
<td>0.08</td>
</tr>
<tr>
<td>Zipper</td>
<td>0.10</td>
<td>0.13</td>
<td>0.01</td>
<td>0.46</td>
<td>-</td>
<td>0.29</td>
</tr>
<tr>
<td>Mean</td>
<td>0.22</td>
<td>0.33</td>
<td>0.09</td>
<td>0.36</td>
<td>-</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Table 2: Anomaly localization performances over the 15 categories of the MV-Tec dataset. Results reported in the original paper [54] are reported here for comparison as well as the resulted reported by [37] for SMAI and by [38] for FCDD.

To ensure a proper segmentation of the hemorrhages as part of the brain, out of the 10 volumes used for training, we selected 5 that were labelled as having an hemorrhage. Nonetheless, this precaution did not suffice to properly train the tissue model robustly. It highlights the variety present in hemorrhage and the difficulty to train a robust model with few examples only. The tissue model may certainly be sufficient for the brain extraction in presence of hemorrhages if it can be trained with enough samples displaying hemorrhage. It would represent a huge labelling work and the use of a bone window seems to work fine as a work around.

5.2 Inpainting Anomaly Detection Validation

First of all, the SN-PatchGANs trained for inpainting the MV-Tec categories yield satisfying generation capabilities for all the categories. A sample of inpainted images for each category in provided on Fig.22 in the annexes. In most cases the generated regions are realistic in term of shape and texture. There are sometimes some little flaws in the generated regions mostly at the border of the image. Beside, the SN-PatchGANs is trained to generated realistic images, but it does not imply that the resulting image will be exactly similar to the original one. For example, the red dots on the pill are realistic but likely to be different from the dots on the original image.

The overall anomaly localization performances by categories are presented on table 2 together
Figure 13: MV-Tec Overview by Type. The segmentation power of the generated anomaly heat-map on the MV-Tec dataset is presented by the AUC per image. The AUC distribution is presented by category as well as by anomaly type. The dot line represents the threshold of randomness.

with the performances reported in the publications of other methods as the mean AUC per images and the mean IoU per image. Independently of other methods’ performances, it appears that our approach yield meaningful detection capabilities with a mean AUC over the categories of 81% and a mean IoU of 24%. The generated anomaly heat-maps yield the best performances for the hazelnut with a large AUC of 92%, while the worst performances are obtained for the tile with an AUC of 62%. Therefore our method allows to generate heat-maps able to extract anomalies meaningfully. It thus demonstrates that our approach can successfully detect and localize anomalies. However, when compared to other methods (AE, AnoGAN, SMAI and FCDD), our approach does not give the best performances and is often surpassed by SMAI and FCDD. Nonetheless, our methods presents competing performances on some categories and often surpass AnoGAN, especially on texture categories. For example, on the tile category, only FCDD yields a higher AUC while our approach surpasses or equates all others.

There are several types of anomalies per categories and each of them present different challenges. In order to get a better insight on what type of anomalies our method can detect, we decided to observe the image AUC distribution per anomaly types. The visualization through boxplots by anomaly types are presented in Fig.13. From this detailed view, it appears that our methods fails mostly on contextual anomalies and close-intensity anomalies. Contextual anomalies refers to anomaly involving a problem of context such as a misplacement of object or an absence of object or a wrong combination of elements. For example, our approach fails on the misplaced transistor because the inpainter can reconstruct the background and the transistor at different
location on the images and it leads to nearly no inpainting error while the ground truth define the whole transistor as anomalous. The same goes for swap in cable colors: the inpainter is not troubled by the presence of two time the same color and manage to inpaint them properly, thus the low performance on that type of anomalies. Close intensity anomalies refers to anomalies that presents a relatively similar pixel intensity as what would be expected in a normal image such as a liquid drop that let the texture appear through it. This configuration usually leads to a small error and thus a more difficult detection. For example, some of the cracks on the hazelnut are very large and let the nut appear, and the nut presents a roughly similar color as the shell. As a results, when the inpainter fill those region, the pixels values are roughly similar and are thus not detected as anomalous. The same occurs with the liquid drop on the wood or the fabric interior on the zipper. It would also explain the low performances on some anomaly types on the tile. In contrast, our approach yield very good localization performances on anomalies with stronger pixel intensity differences such as the prints on the hazelnut or the broken grid. The methods work well at different scales and manage to extract anomalies that cover larger regions such as the cracks on the tiles than spans the whole image, but also smaller anomalies such as the holes in the wood or the small anomalies on the screw. A sample of the generated anomaly heat-maps is presented on Fig.14 and highlight the visual quality of the localization with a pixel-wise resolution.

Now that we have a clearer understanding of the method’s behavior, we discuss different possible reasons of failure and the pitfalls of our approach. First of all, any method detecting anomalies based on the pixel-intensity difference (e.g. AE, AnoGAN, SMAI) will make its decision based on the reconstruction error. This measure is, by its nature, biased toward high contrast anomalies where the anomalous pixel displays a large gray scale difference to the normal image (such as the print on the hazelnut). Whereas anomalies with reduced contrast becomes more tedious to detect due to the naturally smaller reconstruction error. This bias results in trouble shooting of those methods on lower contrast anomalies (such as liquid drop, glue, etc) and makes the definition of an overall segmentation threshold difficult. Opposed to those approaches, FCDD is designed for anomaly detection and the anomaly heat map is nothing else than the final feature map of the deep convolutional encoder which is trained to yield low values on normal images. It can thus provide high activation on any anomalies regardless of the pixel intensity since the neural network is a learned non-linear mapping. This observation contributes to explain the better performances and better consistency of FCDD compared to other methods.

Specifically to our inpaint-based approach, the inpainter is trained to generated seamless results by using the context and by borrowing textures and features from the available part of the image. This behavior is in general beneficial to generate fine details such as printing on the capsule or hairs of the toothbrush, but it can sometimes be troublesome when the inpainter successfully reconstruct a partially hidden anomalies such as a crack on the wood or a cut on the hazelnut. Indeed, the SN-PatchGAN understand that the part of the crack available has to be prolonged similarly as the wood seam it has been trained to complete. As a results the ability of the inpainter to integrate cues for realistic inpaint is both beneficial and potentially problematic.

In addition, the anomalies segmentation ground truth are often delineating the anomalous area as region where the anomaly is present. For example, in the grid when there is a bend the anomaly is defined as the ‘S’ of the bend, but, the intersection between the bend and a normal grid present roughly the same pixel intensity. As a result in a pixel wise detection, intersecting
Figure 14: MV-Tec Anomaly Map Samples

Six example of generated anomaly heat-map are presented for each categories. Each sample is presented as the original image on the left, and the overlay of the anomaly heat-map on the right. Bright yellow highlight a high detection of anomaly. The sample's AUC is displayed on the upper left corner of the pair. The first three samples on the left are the ones yielding the best AUC of the categories while the three samples on the right are randomly sampled from all the other samples.
part would not be detected as anomalous since the intensity is correct but not the context. This is maybe why SMAI yield better performances: it generates a coarser heat-map at the super-pixel level which is likely to better match the ground truth. However, the coarse heat map generated is qualitatively less attractive for it only give a broad idea of where an anomaly is. Therefore, the detection at the pixel resolution has advantage and inconvenient.

Moreover, it happens that some features of the object are detected as anomaly in the first phase and are completely hidden away during the correction of the image. As a results, the inpainter may not generate them again since it has no cue or incentive in drawing the hidden structure. This leads to a persistent anomaly over the iterative process and thus more false positives. For example, the red stains on the pills or the colors on the tile are a random pattern and the inpainter can generate realistic texture but it is likely not matching the original image. Another example would be the brand name on the capsule: as it does not appear on every image, the model will not generate it if the brand name is completely hidden by the mask.

In the opposite direction, if the inpainter happens to generate a pixel value close to the original one, the pixel may be removed from anomaly mask in our iterative cleaning process and will thus not be detected leading to false negatives.

To summarize, our method does work to localize anomalies and allow to generate pixel-wise predictions. As our approach is based on the reconstruction error, it performs better on anomalies presenting a large contrast compare to a normal image while it often fails to properly localize anomalies with closer pixel intensity or contextual anomalies such as misplaced objects. Nonetheless, this approach yields better overall performances than AnoGAN and compete with the Auto-Encoder. Even though, it does not perform better than SMAI the generated anomaly heat-map are visually better than SMAI’s ones because of the pixel resolution instead of the super-pixel ones.

We have assessed the relevance of our approach on the MV-tec dataset that contains several types of anomalies. However, the application that interests us is the localization of ICH in brain CT-scans, and ICH often presents an intensity different from the healthy brain tissues which makes our approach suited for it.

5.3 ICH Segmentation

Firstly, all the methods’ segmentation performances are presented on table 3 as the mean over volumes when in the test fold of the 10-fold cross-validation. The segmentation performances are further presented on Fig.15 as the volume dice, volume precision and volume recall distributions to better appreciate and compare the true performances. In both case, the dice, precision and recall are presented in two fashions: either over all the 75 volumes (with or without ICH) or only over the 36 volume containing an ICH. We make this distinction to avoid possible bias toward methods that does not detect very well ICH. Indeed, in a volume without hemorrhage (i.e. without true positives), the dice, precision and recall are either 1.0 or decay rapidly toward zero with few false positives. As a results, a method with a low detection rate will see its performances greatly increased when considering the volume without ICH, but it does not reflect the hemorrhage detection capability which is the main objective.

The straightforward implementation of the U-Net 2.5D trained only with slices containing an ICH does not yield a satisfying mean dice score over volumes (15.73% on all volumes and 32.64% on ICH volumes). This low dice can be mainly imputed to a low precision of the model (respectively 14.79% and 30.67%) resulting from a large number of false positives. On the
Figure 15: Performance Overview. Segmentation performance of all models are presented for each volume (when in validation) as the dice, precision and recall over every volumes (Left) or only over the volume containing an ICH (Right). Each measure is presented as a boxplot hovered by a histogram highlighting the underlying distribution. The methods are ordered by need of labelled data: the higher on the figure, the more labels are required.
global contrastive 60

multi-label classification (ce) 62

ad inpainting 19

fcdd (5%) iCH 54

fcdd (50% iCH) 59

U-Net 2.5D mixt + BM 38

U-Net 2.5D iCH-only + BM 18

Attention U-Net 42

context restoration 47

41.42 56.30 41.29 72.87 36.46 44.36 38.00 63.51 29.14 43.47 33.28

attention U-Net 42.31 41.26 46.36 42.11 72.22 36.94 40.33 30.45 44.77 32.13 33.20

Unet 2.5D mixt + BM 38.39 39.17 46.29 41.28 72.18 36.30 42.22 29.20 58.68 30.66 42.05 31.60

Unet 2.5D mixt 38.36 39.16 46.24 41.28 72.18 36.30 42.17 29.19 58.58 30.70 42.04 31.59

Unet 2.5D ICH-only + BM 18.51 27.42 19.22 27.73 74.49 33.61 38.31 26.50 28.78 28.09 44.77 32.50

Unet 2.5D ICH-only 15.74 25.06 14.79 23.97 73.49 35.61 32.64 27.53 30.67 26.67 44.77 32.49

FCDD (50% ICH) 54.72 44.48 61.06 42.57 64.87 41.23 16.66 13.28 29.87 30.46 44.61 27.49

FCDD (5% ICH) 59.37 42.94 72.38 37.75 62.95 42.21 18.12 17.30 45.24 35.82 22.84 24.78

FCDD (2% ICH) 60.84 42.58 77.19 34.06 61.71 42.76 18.41 17.76 52.47 35.24 20.23 22.36

AD inpainting + BM 20.00 31.28 21.11 34.36 60.33 43.36 16.29 19.38 18.60 23.26 17.36 19.10

AD inpainting 19.13 33.04 19.45 33.51 60.35 43.35 16.40 19.77 15.26 19.87 17.39 19.13

FCDD artificial 26.65 43.20 38.73 47.67 53.88 48.97 2.62 9.16 27.79 42.53 9.91 13.92

AE 12.95 32.23 12.53 32.33 62.43 41.48 1.97 3.18 1.10 1.90 21.73 20.00

Table 3: Methods’ performance are presented as the mean µ and standard deviation σ of the dice score, precision and recall over volumes when in the evaluation fold. All Volume means that the performances are computed on all the volume of the dataset (36 with an ICH and 39 without). ICH Volume means that the performances are computed only on the volumes that contain an ICH (36 volumes). The highest performance by metric is highlighted in bold. The methods are ordered by need of labelled data: the lowest number of labelled data are required to perform the best. Noteworthy that the method respectively uses or not the associated type of data. (✓) means that the labels were used indirectly to ease the computation time but are not required by the method. BM: Brain Mask.

other hand, the detection rate of ICH is good enough to compete with other methods with respectively 73.49% and 44.77%. This model has thus a large sensitivity but a low specificity and tends to detect hemorrhages everywhere which is not a desired behavior. Subsequently, by simply using the generated brain mask to remove meaningless prediction output of the brain, the performances are increased due to a large reduction in false positives and it enables to reach a dice of 18.51% on all volumes and 38.31% on ICH volumes. Even with this improvement, the precision still remains quite low reflecting a large number of false positives inside the brain.

Interestingly, the simple adaptation of the U-Net 2.5D training consisting of adding slices without ICH and giving them a lower weight on the loss computation, allows to easily improves the segmentation capabilities of the U-Net. We obtained a mean dice score of 38.36% on all volume and 42.17% on ICH volume which represents a gain of respectively 22.63% and 9.53% compared to the U-Net trained only on ICH slices. The improvement arises from a large increase in precision by two to three folds while conserving a rather similar recall. It suggests that presenting the model with more images (even without the targeted hemorrhage) helps in learning better and more robust features. We hypothesize that the addition of non-ICH images in the mini-batch allows the model to better compare the healthy structures from the ICH’s ones. Noteworthy that this simple improvement relies solely on the benchmark dataset itself and does not require any external data (such as the RSNA dataset). It is therefore a data-efficient approach. In this case, the adaptation of the brain mask does not further increase the scores which highlights that the adaptation allows the U-Net to focus implicitly on the brain region.

In our case, we do have access to additional data non-labelled for segmentation: the RSNA dataset. Its harnessing through the self-supervised context restoration task enables to further improves the segmentation performances on all fronts. Indeed, a dice of 47.17% on all volumes and 44.36% on ICH volumes is obtained by increasing both the precision and recall compared to the supervised mixt model. Additionally, the initialization of the U-Net encoder with the
global contrastive task results in a mean dice on all volumes of 60.41% and of 44.98% on volume with ICH. The large performance increase on all volume is mostly due to a better precision of the model that greatly reduce the amount of false positives on healthy volumes. Indeed the precision is increased by around 20% compared to the supervised approach (mixt). The addition of the local contrastive training of the partial decoder results in a better precision of the segmentation model both over all the volumes (69.31%) and over only the ICH volumes (67.71%). However, this precision increase does not seems to transpose in a better dice on all volumes (58.62%) compared to the use of only the global contrastive task even though the recalls are roughly similar. Nonetheless the dice is increased to 45.42% on ICH volumes only. Note that the dice can be lower even though the precision and recall are increased due to the averaging over folds. Therefore, based on the precision and recall, the global-local contrastive task seems to improves the segmentation performances of the downstream U-Net. The performance increase is not so large and may be due to the to challenging contrastive task on the local scale: even though we tried to use transformation that does not hinder much the spatial organisation, the contrastive objective may become counter intuitive for the network on some cases. There is thus room for improvement in this direction to better define the local contrastive objective.

Additionally, with access to classification labels, pre-training the U-Net weights on the binary classification task (i.e. weakly supervised approach) enables to explicitly learn what we aimed to learn with the self-supervised: a representation that discriminate hemorrhage slices from the other and thus features that are tailored for ICH. With this pre-training scheme, the recall is greatly improved to 76.14% on all volumes and 50.29% on ICH volumes and highlight the transfer to features well suited for ICH. On the other hand the precision is reduced compared to the contrastive pre-training resulting in a slightly lower dice on all volumes (56.00%) but not on ICH volumes (48.32%). Small false positives in a volume without hemorrhage will strongly impact the dice value and lower it quickly to zero. Reporting the performances on ICH volume only is important to avoid the small bias of considering non-ICH volume in the competition. In the case of binary classification pre-training, a higher dice is obtained on ICH volumes which attest the relevance of this approach for improved segmentation.

Furthermore, with access to more detailed labels such as the type of hemorrhages, pre-training on the multi-label classification task appears to yield even better results. Indeed, compared to the binary-classification pre-training, both the precision and recall are increased either on all volumes or on ICH volumes. For example, the recall is increased by 1% on all volumes and by 2% on ICH volumes together with a high precision. Note that using a weighted cross-entropy as objective seems to yield higher detection rate while using the dice loss as objective seems to yield more precise segmentation models. Multi-label classification pre-training results in the largest mean dice obtained, namely 63.85% on all volumes with the dice loss and 50.06% on ICH volumes with the cross entropy loss.

With a first insight on the methods’ performances we can now compare them. Compared to the supervised approach, the self-supervised pre-training enables to greatly increase the model precision. Nevertheless the recall is not increased much and may highlight the difficulty of the detection task on some cases. Indeed, some of the available labelled ICH can be very subtle and the self-supervised tasks (context restoration and contrastive) tends to learn lower level features that are not necessarily as specific and tailored for ICH detection. For example, the context restoration task imposes a reconstruction loss, and in the case of a small ICH the model
Figure 16: Pre-training t-SNE. Bottleneck representation learned by the different pre-training task (self-supervised or weakly-supervised) on the validation set. The large embedding space is represented in 2D thanks to a t-SNE transformation [58]. Each sample is colored depending on whether it represents an image with or without ICH. Below the main visualization, the same t-SNE is presented for the five ICH types in which orange highlights samples of the given type and gray represents the absence of the type. EDH: epidural hemorrhage, IPH: intraparenchymal hemorrhage, IVH: intraventricular hemorrhage, SAH: subarachnoid hemorrhage, SDH: subdural hemorrhage.

may perform really well in reconstructing the majority of the image but not a small ICH which does not impair much the reconstruction loss. Therefore the model has a lower incentive in focusing on this specific small structure. A similar issue may arise with the contrastive task that is trained to recognize two versions of an image among the images in the mini-batch. Since, the model is trained on any slices from CT-scans, the mini-batch is likely to contains images with large different structures: for example, a slice on the upper part of the skull contains different structures than a slice at the level of the eyes. As a results, the model can easily focuses of lower level features such as the presence of eye-balls or bone structures to differentiate images in the mini-batch. The model may thus not need to rely on small ICH to distinguish two images apart. Oppositely to self-supervised pre-training, weakly supervised pre-training enables to force the model in focusing on the hemorrhage to differentiate images. Pre-training on the binary-classification task enables to initialize the segmentation encoder with features truly tailored for ICH detection which results in a higher detection rate as well as a higher precision. To further compare the different initialization schemes, we observe the 512-dimensional representation (the feature map transferred to the segmentation model averaged over height
and width) of the validation set using a t-SNE transformation \[58\], colored by ICH presence. t-SNE visualizations can be found on Fig.16. Note that for a fair comparison, the different models where trained on the same amount of data and evaluated with similar validation sets (composed of around 5000 slices among which around half contains an ICH). At first glance, the two self-supervised tasks yield bottleneck representations that looks roughly similar: ICH slices tend to be mapped in the same cluster and are slightly blended with some healthy slices, while some non-ICH clusters clearly sticks out. Those healthy slices are likely part of the CT-scans where the brain is not present and thus where no ICH could be found, such as the neck. Even though the representations look similar, with the contrastive task, there exist an area containing exclusively ICH slices whereas it is not the case with the context restoration task where positive and negatives slices are almost always blended. More precisely, the contrastive task seems to isolate intraparenchymal and intraventricular hemorrhages (IPH and IVH) while it is not the case in context restoration. Even though those ICH type are the easy case of ICH, the self-supervised contrastive method provide the incentive to rely on those structures to differentiate images. The fact that the contrastive approach can rely on those structure and not the context restoration highlights the better ability of the contrastive task to learn salient features. In brief, the contrastive tasks has learned features that better discriminates ICH slices form the others. Those better features latter transpose in better segmentation performances compared to the context restoration task. Note that the context restoration has the advantage of providing initialized features for the whole U-Net while the contrastive task can only pre-train an encoder. Consequently, with less information transferred, the contrastive task enables to reach better performances.

With the access to classification label, the representation learned on the binary classification tasks presents a clean separation of ICH and non-ICH slices with a blending area containing challenging cases. As a results, the features learned in this weakly supervised way are tailored specifically for ICH and naturally lead to better segmentation performances especially in term of recall. Beside, based on the representation by ICH type on Fig.16 it again appears that IPH and IVH are the two types that are well pushed away from healthy slices and thus away from the blending area (characterizing challenging cases). Nonetheless, as shown on the t-SNE representation by ICH type, features learned on the binary classification tasks are optimized to differentiate ICH as a whole and the model is not explicitly encouraged to learned more detailed features about those hemorrhage (i.e. what differentiate the ICH types). On the other hand, the pre-training on the multi-label classification task (DL and CE) provides such an incentive to learn higher level features as demonstrated by the t-SNE representations on Fig.16. Indeed, not only the ICH and non-ICH slices are nicely separated similarly as in the binary classification case, but the different ICH types also form clusters highlighting that the model has learned to discriminate them. The clusters appears to be more separated when using the dice loss as objective. Note that the cluster boundaries are fuzzy because of the multi-label settings: one sample can contains ICH of various types. An interesting aspect of the learned representation compared to the others is the formation of a cluster of EDH which is one of the main hemorrhage type in the segmentation dataset but not in the RSNA dataset (see Fig.1A and Fig.2A). The multi-label classification pre-training task enables thus to learn efficiently EDH specific features for the segmentation dataset from a limited number of EDH example, which may explain why the multi-label classification pre-training yields the best segmentation performances.
The performances of the ResNet-18 trained to select the slices are presented on Fig. 17 and table 4 for the various classification thresholds considered. It shows really good classification capabilities since the softmax outputs yield a bootstrapped mean AUC of 97.85%. Therefore, with a large dataset, one can train a model that efficiently detect hemorrhages, and this simple classifier could easily be used in any diagnostic pipeline to filter out relevant slices from a CT-scan. Different thresholds of the softmax outputs naturally yield different classifications, and from our validation set a threshold of 0.35 or 0.25 appears to yield the best F1-scores. The choice between those two values depends on the application, \( t = 0.25 \) gives a higher recall while \( t = 0.35 \) gives a better precision. For our use prior to the anomaly localization methods, we opt for a higher recall as we want to detect as many hemorrhage as possible and the classifier is just used to ease the computation. In brief, with this threshold we can expect a good classification of the slices and this step should not impair much the performance measurement of the anomaly localization method.

The confusion matrix of the classification of the segmentation dataset is presented on table 7 in the annexes. Noteworthy that we did not rely on those results to choose the threshold. With the chosen threshold, there are 40 false negatives over the 318 positives. This slightly higher false positive rate compared to the one reported on the classification validation set suggests that the segmentation set contains more challenging cases.

The SN-PatchGAN inpainter network capabilities are highlighted on Fig. 18 with a sample of validation images. The inpainting quality is impressive and it is hard to discriminate gener-
Figure 18: SNPatchGAN Sample. Example of the inpainting capabilities of the SNPatchGAN on few validation head CT-scan. The top row is the original image, the middle row is the masked image and the bottom row is the inpainted image. Note that the masks are displayed in a pale orange color for improved visualization, the image fed to SNPatchGAN has the masked pixels set to zero (i.e. black).

ated and original regions. Even though the images look realistic, it can be observed that the inpainter do not reconstruct exactly the original input but rather provide a realistic alternative (which is what it has been trained to do). For our anomaly localization, such differences are sometimes wanted, as shown on the sixth image where the inpainter has no reason to reconstruct the hemorrhage, but they are sometimes just natural variation of realistic structures that we need to filter out. Overall, the inpainter appears to be efficient to reconstruct realistic images without apparent flaws.

One could argue that using the classifier prior to the anomaly detection makes it a weakly-supervised approach. The methods as such do benefits from the classification labels, yet it does not use any segmentation labels and is thus still more thrifty than the other label-efficient methods explored. In addition, we performed validation on the MV-tec dataset to assess the relevance of the method in a pure unsupervised setting. Eventually, we measure the method’s performances when processing only the slices with an ICH, to compare the performances with other studies working only on this set of slices. In this set up we obtained a mean slice dice of 21.13%. In the study where they released the segmentation dataset, Hssayeni et al. [14] reported a mean slice dice of 31.5% over the ICH slices only using a U-Net processing patches of $160 \times 160$ pixels. Therefore, considering that the anomaly detection based approach is unsupervised, it performs quite honourably compared to the supervised U-Net.

At first glance, our method yield meaningful anomaly heat maps as presented on Fig.19a. After thresholding the heat-map, the resulting segmentation seems to work on the most obvious cases but also extract some structure other than hemorrhages. Details of the measured performances are presented on the lower part of Fig.15. In term of mean volume dice, the combination ResNet-18 and anomaly detection yield 19.13% with all volume and 14.60% with only ICH volume. This is considerably lower than the other supervised. Those lower performances were expected since the approach is unsupervised and is designed for anomaly detection. Indeed, hemorrhages are not the only anomalies potentially present in the CT-scans as there are fractures, additional element in the scan such as hands holding the heads, hematoma (swollen tissues around skull) or even air bubbles in the brain. All those will likely be picked up by the model as anomaly, increasing the amount of false positive and as a results lowering greatly the
Figure 19: Anomaly Map Results. Anomaly maps and the derived segmentation are presented for a sample of slices. The top row is the input image, the middle row shows the computed anomaly heat-map (yellow representing stronger anomaly). The bottom row presents the classification obtained from thresholding the anomaly map. The resulting slice dice is presented on the upper left corner of the segmentation mask. (a) Anomaly localization using our Anomaly inpainting approach. (b) Anomaly localization using FCDD trained with 5% of ICH. Note that the same sample is displayed for both method.

precision and the dice. On the other hand, the recall is lower than with the other methods, meaning that the method also fails to detect some hemorrhages (i.e. many false negatives). It may partially be due to the dataset challenging cases in which hemorrhages are hardly visible as they are small and appears in a texture closer to the brain’s one. And since the anomalies are detected by inpainting reconstruction, hemorrhages with a texture close to the brain will likely be reconstructed flawlessly and therefore not be detected as anomaly. Furthermore, the delineated hemorrhage in the ground truth often includes some features within it that present a similar color as the gray matter: when the model inpaint those regions, the resulting error is not so high and those pixels are not detected as anomalous. It results in incomplete hemorrhage extraction contributing to increasing the false positive rate and thus lowering the recall (see volume 94 slice 25 on Fig.19a). Similarly as with the MV-tec dataset, the localization quality can be reported through the AUC of the anomaly heat-map which allow a quantification
Figure 20: Anomaly Map AUC Results. The segmentation potential of the anomaly heat-map is reported through the AUC of the ICH slices and of the ICH volumes. To avoid bias toward method rejecting only the background, we also present the AUC computed only on the pixels belonging to the brain using the generated brain mask. Each measure is presented as a boxplot hovered by a histogram highlighting the underlying distribution. The mean value is reported below below the boxplot in gray.

independent of the arbitrary choice of threshold. However, the AUC can only be computed on volumes/slices containing ICH. We thus measure the mean AUC per ICH slices and the mean AUC per ICH volumes\(^2\) and we obtained respectively 73.86% and 73.54%. Those values represent the goodness of the anomaly heat-map generated and it thus appears that the heat-map has the ability to provide a meaningful localization of hemorrhages. In addition, the distribution of slices and volumes AUC presented on the right of Fig.20 shows that the majority of the heat-maps (slice) yield a good AUC with some allowing a nearly perfect detection (\(i.e.\) AUC close to 1.0). Moreover, the dice, precision and AUC can be easily increased with the use of the generated brain masks that allow to remove any anomalies detected outside the brain such as fracture, hands, element of the scanner, etc. Using the brain masks, the dice is increased to reach 20.00% on all volumes and 16.29 on ICH volumes. In addition, the heat-maps classification capabilities are also increase to reach an AUC of 75.44% and 74.15% respectively on ICH slices and ICH volumes. In this case, using the brain mask is thus an easy way to improves the performances by few percents. The discrepancy between the AUC measure and the dice/precision/recall highlight the difficulty of choosing proper decision thresholds for the whole dataset.

Comparing the anomaly inpainting approach’s performances with methods that use more labels does not allow a fair comparison. We can therefore compare it with the AE and the FCDD trained on artificial anomalies that are both optimized using the same amount of data and the

\(^2\)Noteworthy that the AUC is computed for each slice or volume independently at the pixel level, providing a AUC value for each volume/slice. Then the mean of those AUC is measured.
same label access (i.e. unlabelled data and a ResNet-18 classifier). We decided to compare the method based on the mean slice AUC and the mean volumes AUC. At first sight, the AE’s heat-maps seems to yield higher AUC when considering the whole image, however this value can be mis-leading since the positives pixels (i.e. the ICH) only occurs in the brain and a model that tend to extract the whole brain tissues from the background will be rewarded with a good precision and a good recall since it likely picked all the hemorrhages. The computation of the AUC is thus biased. To cope with this bias, we decided to measure the AUC using only the pixels of the brain (using the brain mask). This way, we restrict the AUC measure exclusively to pixels that can contains an hemorrhage. Without this biased, the performances of the AE drops to a mean AUC over slices of 55.61% while the AUC with our method is barely affected with a mean slice AUC of 72.98%. In addition, FCDD trained with artificial anomalies does not yield competing AUC when considering the brain pixels only: 38.29%, which means the heat-maps tend to pick the inverse of the hemorrhages. The same goes with the volume AUC on the brain where a mean of 66.36% is obtained with the AE, 58.27% with the unsupervised FCDD, and 72.01% with our approach. To summarize, using only unlabelled data (and the ResNet-18 classifier) the anomaly heat-maps generated using our approach are more meaningful than the one generated using the AE or FCDD trained with artificial anomalies and allow a better localization of ICH.

Even though the training of FCDD in an unsupervised way does not allow a proper anomaly localization, using real ICH slices enables to localize hemorrhage better than our approach in term of AUC. Indeed, using as few as 2% ICH examples (i.e. 1000 slices) yields a mean slice AUC of 74.83% and a mean volume AUC of 77.55% (considering only the brain). Using more ICH in the training of FCDD enables to generated more potent anomaly heat-maps: using 5% anomalies increase the mean slice AUC and mean volume AUC to respectively 79.80% and 80.96% while using 50% of ICH enables to obtain a mean slice AUC and mean volume AUC to respectively 87.21% and 87.68%. Therefore, using only and few classification labels, the FCDD approach can generate anomaly heat-map with a good localization potential. A sample of heat-maps generated with the FCDD trained on 5% anomalies is presented Fig.19b. It shows that the heat-maps obtained with FCDD are less sharp than the one obtained with our approach due to the Gaussian kernel up-sampling, and, visually, our approach yield more precise and crisper heat-maps. FCDD may perform better thanks to the Gaussian up-sampling that can extract regions within hemorrhages that have a similar intensity as the brain while our approach cannot (see volume 94 slice 26 on Fig.19).

To summaries, even though our inpainting based anomaly localization method does not compete with supervised, semi-supervised and weakly supervised method, it yields more meaningful anomaly heat-maps than the AE or FCDD trained on artificial anomalies using the same data settings. Nonetheless, with access to ICH, even few of them, FCDD can be trained to localize efficiently ICH as highlighted on Fig.24 in the annexes. Moreover, FCDD is extremely lighter in term of computational cost and requires less hyper-parameters tuning than our approach.

Additionally, adding anomaly heat-maps generated with our method to the CT-slice as a second channels to the U-Net 2.5D input does not seems to improves the performances: compared to the supervised mixt approach, a better dice is obtained on all volumes (42.37% vs 38.30%) while the inverse is observed when considering only the ICH volumes (39.96% vs 41.83%). This
approach may benefit more from the empty heat-maps generated using the ResNet-18 leading to a better non-detection on healthy scans. But the presence of the heat-map may confuse the network on ICH volumes. As a results, this approach may be biased and the results should be considered with caution.

Finally, we briefly discuss the toughness of the benchmark dataset used in this study. First, The benchmark segmentation dataset made available by Hssayeni et al. [14] contains only few volumes containing an hemorrhage (36) with a small depth resolution (around 5mm) which transposes in a low number of positive slices. This low number of available examples makes the training of robust deep neural network challenging. Moreover, ICH comes is a variety of flavour with a broad spectrum of shapes and localization in the brain: some are extremely straight forward and easy to distinguish with the characteristic white texture on the CT-scans, but on the other hand some ICH are really small and subtle (see third sample on Fig.1C). Additionally, one of the 36 volumes belongs to a patient admitted two weeks after the injury leading to a completely different ICH texture (darker). As a results, whenever this volumes is on the validation set, the trained model will never have seen such a case in its optimization and will most likely not detect it and therefore affect the performance measures. The toughness of this dataset can be further observed through the classification performances with the ResNet-18 trained on the RSNA dataset. Indeed, based on the validation of the classifier (see table 4), we were expecting a recall of 90.62%, a precision of 89.81% and a F1-score of 90.20%. However, on the benchmark dataset, the ResNet-18 (with \( t = 0.25 \)) yields a recall of 87.42%, a precision of 51.01% and a F1-score of 64.43% (see table 7 in the annexes). It thus highlights that the benchmark dataset contains more challenging cases with potentially confusing structures. Finally, the representation of different ICH types is uneven and contains mostly epidural hemorrhages and a rather low number of subarachnoid hemorrhages. Subarachnoid hemorrhages may be more challenging to segment as their boundaries are less sharp and they are often smaller. As a results with less example of those challenging hemorrhages, the model will be trained on less example and will thus have a reduced incentives to detect them. All the aforementioned challenges of the benchmark dataset may contributes to the observed lower dice obtained in this study compared to other research on ICH segmentation based on private datasets.

6 Conclusion & Future Works

To conclude, in this work we assessed the relevance of label-efficient methods for volumetric segmentation of intracranial hemorrhages on a small dataset of publicly available CT-scans. We first showed that using only slices with hemorrhage in a supervised way does not yield a satisfying results and confirm the need to use additional data. We then showed that volumetric segmentation performances can be greatly improved through the use of additional unlabelled and weakly labelled data in a transfer learning setting. Most importantly, even though self-supervised pre-training allow performances improvements, the use of classification labels, and especially multi-classification labels, yield the best feature pre-training explored in this work: a mean volume dice of 50.06%. Although we explored multiple methods, they are mostly based on transfer learning. For a better exploration, other approach should be evaluated such as methods that process labelled and unlabelled data in parallel.

We highlighted that the ICH segmentation is more challenging than what it looks like in the
first place because of the variety of shapes, sizes and locations of the hemorrhages. Some of them can be very subtle and represent only few pixels in a large volume. This variety emphasizes the important need for benchmark datasets for ICH segmentation in order to faithfully evaluate a model and its capacity on a representative sample and compare different methods against one another. Even though the benchmark dataset used in this work enables a fair comparison of methods, it lacks a good resolution between slices which prohibits the exploration of 3D architectures. Moreover, the low number of scans \(i.e n=75\) with 36 positives may not provide a representative sample of the ICH. In consequence, there is a need to develop a more complete and more versatile benchmark dataset for ICH segmentation. A possible direction would be to have the CQ500 dataset labelled by trained radiologist.

Moreover, in absence of available segmentation labels we showed that ICH can be localized meaningfully through the anomaly detection framework. We additionally proposed an unsupervised anomaly localization methods based on image inpainting that allow to localize anomalies at a pixel resolution. Our method performs well for the ICH segmentation compared to other unsupervised anomaly localization approaches. We further evaluated our method on the MV-tec dataset and obtained competing results. Nonetheless our approach is computationally intensive and more research work should be invested to make it lighter. We therefore proved the potential of this approach, highlighted its benefits and flaws, but further research is needed to improve it.
Acknowledgements

I would like to thanks the Zeno-Karl-Schindler Foundation for their trust in me as well as in this project, and for offering me the opportunity to complete it in the best environment possible through their financial support. I am also thankful to the Movetia Foundation for their support to exchange students. Additionally, I would like to thanks the laboratory of biomedical imaging of KTH for their continuous support, supervision, and resources, essential to the completion of this project. Finally, I am grateful to Prof. Jean-Philippe Thiran for supervising this project on behalf of the EPFL.

References


[37] Zhenyu Li, Ning Li, Kaitao Jiang, Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Superpixel Masking and Inpainting for Self-Supervised Anomaly Detection. page 12.


[40] DenOtter Td and Schubert J. Hounsfield Unit. October 2019.


[42] Hounsfield units - scale of HU, CT numbers | Classifications, online calculators, and tables in radiology.


Annexes

This section contains additional material to support and complete the presented work.

![U-Net Architecture Diagram](image)

**Figure 21: Detail of U-Net Architecture.** The U-Net architecture presented here is the one used in all the experiments.

<table>
<thead>
<tr>
<th>Textures</th>
<th>( n_{\text{epoch}} )</th>
<th>batch size</th>
<th>( b_y )</th>
<th>( b_d )</th>
<th>( \lambda_{L_1} )</th>
<th>( \lambda_{GAN} )</th>
<th>( \gamma )</th>
<th>Translation</th>
<th>Rotation</th>
<th>H-Flip</th>
<th>V-Flip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carpet</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>reflect</td>
<td>reflect</td>
</tr>
<tr>
<td>Grid</td>
<td>1250</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>reflect</td>
<td>reflect</td>
</tr>
<tr>
<td>Leather</td>
<td>3000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>reflect</td>
<td>reflect</td>
</tr>
<tr>
<td>Tile</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>reflect</td>
<td>reflect</td>
</tr>
<tr>
<td>Wood</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>reflect</td>
<td>reflect</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objects</th>
<th>( n_{\text{epoch}} )</th>
<th>batch size</th>
<th>( b_y )</th>
<th>( b_d )</th>
<th>( \lambda_{L_1} )</th>
<th>( \lambda_{GAN} )</th>
<th>( \gamma )</th>
<th>Translation</th>
<th>Rotation</th>
<th>H-Flip</th>
<th>V-Flip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottle</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.05</td>
<td>0.05</td>
<td>nearest</td>
<td>nearest</td>
</tr>
<tr>
<td>Cable</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.05</td>
<td>0.05</td>
<td>constant</td>
<td>constant</td>
</tr>
<tr>
<td>Capsule</td>
<td>4000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.075</td>
<td>0.075</td>
<td>nearest</td>
<td>nearest</td>
</tr>
<tr>
<td>Hazelnut</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>nearest</td>
<td>nearest</td>
</tr>
<tr>
<td>Metal nut</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.05</td>
<td>0.05</td>
<td>nearest</td>
<td>nearest</td>
</tr>
<tr>
<td>Pill</td>
<td>2000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.075</td>
<td>0.075</td>
<td>nearest</td>
<td>nearest</td>
</tr>
<tr>
<td>Screw</td>
<td>2000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>nearest</td>
<td>nearest</td>
</tr>
<tr>
<td>Toothbrush</td>
<td>6000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.1</td>
<td>0.1</td>
<td>nearest</td>
<td>nearest</td>
</tr>
<tr>
<td>Transistor</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.075</td>
<td>0.075</td>
<td>constant</td>
<td>constant</td>
</tr>
<tr>
<td>Zipper</td>
<td>1000</td>
<td>7</td>
<td>1e-4</td>
<td>4e-4</td>
<td>1.0</td>
<td>0.05</td>
<td>0.995</td>
<td>-0.05</td>
<td>0.05</td>
<td>constant</td>
<td>constant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Draws number ( N )</th>
<th>Vertexes Number ( N_v )</th>
<th>Brush Width ( w )</th>
<th>Segment Length ( l_i )</th>
<th>Vertexes angles ( \alpha_i )</th>
<th>Sphere Number ( N_s )</th>
<th>Sphere Radius ( r_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>10</td>
<td>30</td>
<td>15</td>
<td>25</td>
<td>15</td>
</tr>
</tbody>
</table>

**Table 5:** SN-PatchGANs’ training parameters for each categories of the MV-tec dataset. The lower table contains the parameters used to generated the free-form mask.
Table 6: Parameters of the inpainting anomaly detection module for each categories of the MV-tec dataset. Note that in the rotation angle entry, some values are given as $[\text{start} : \text{step} : \text{stop}]$.

Table 7: Confusion Matrix for the detection of ICH by the ResNet-18 on the segmentation dataset [14]. The results for the five thresholds explored are presented.

Figure 22: MV-Tec SNPatchGAN Sample Example of SN-PatchGANs performances on the MV-Tec dataset categories. On the left is shown the masked image and the inpainted one is shown on the right. Note that the masks are displayed in a pale orange color for improved visualization, the image fed to SN-PatchGAN has the masked pixels set to zero (i.e. black).
Figure 23: Artificial Anomalies Samples
Sample of artificial anomalies used to train FCDD in an unsupervised way. Anomalies consist of \( n \) ellipses with major axis \( a_{\text{major}} \) and minor axis \( a_{\text{minor}} \) rotated by an angle \( \alpha \) and colored with a pixel intensity of \( I \) combined with Gaussian noise of standard deviation \( \sigma \). \( n \) is sampled uniformly in \([1, 5]\); \( a_{\text{major}} \) is sampled in \([25, 50]\); \( a_{\text{minor}} \) is sampled in \([5, 25]\); \( \alpha \) is sampled in \([0, 2\pi]\); \( I \) is sampled in \([0.5, 1]\); and \( \sigma \) is 0.025.

Figure 24: FCDD heat-map sample
Sample of anomaly heat-maps generated by the FCDD trained with 5\% of ICH slices on validation images of the RSNA containing hemorrhage.