AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF COMPUTER SCIENCE, ELECTRONICS AND TELECOMMUNICATIONS

INSTITUTE OF COMPUTER SCIENCE

Master's Thesis

Self-supervised learning for medical diagnosis and imaging

Uczenie częściowo nadzorowane na potrzeby diagnostyki medycznej

Author:	*Wojciech Konieczkowicz*
Degree programme:	*Data Science*
Supervisor:	*prof. dr hab. inż. Bogdan Kwolek*

Kraków, 2023

I gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016544.

Contents

1. Introduction

The recent development in the field of Artificial Intelligence (AI) has brought a new paradigm of solving problems and enabled humans to progress in various domains. One such area is the medical imaging field, where deep learning models based on convolutional neural networks and vision transformers are aimed at processing images to extract information helpful in various domains such as pathology and radiology.

The models are mostly trained with *supervised learning*, an approach that relies on annotated images. However, creating a sufficiently large dataset for each new problem is expensive and requires time of expert personnel. Recently, *self-supervised learning* (SSL), a new training paradigm that creates a training signal without human annotations, has been developed. It helps to adjust models' parameters during a phase known as *pretraining*. Then, during a follow-up phase referred to as *fine-tuning*, the pretrained model is fine-tuned to solve the target problem (known as *downstream task*) in a supervised way, obtaining better results, while requiring less annotated data. A significant part of the most recent SSL techniques [1–4] rely on generating the training signal by feeding random crops of the same image into a model and forcing it to return similar latent representations (such crops are further referred to as *positive crops*, *positive views* or *positives*). The techniques mentioned above were primarily developed and evaluated using the ImageNet [5] dataset, containing mostly images with single, dominant objects. While random cropping works well when pretraining is performed on object-centric datasets, it yields worse results in other cases [6]. Random cropping seems to have the potential to underperform also when pretraining occurs on computed tomography (CT) images, where multiple objects (such as organs) are contained within a single input image. That leads to enforcing similar latent representations for crops containing distinct organs, that may constitute separate classes in a downstream task, making the pretraining more likely to underperform. Several effective methods have been proposed to enhance the performance of SSL pretraining on medical images; however, they all come with additional technical assumptions or limitations.

This thesis aims to propose and evaluate a novel, simple strategy for generating positive image crops for visual self-supervised learning. Contrary to previous works, it does not require multiple images of the same patient or any kind of metadata [7, 8], generalizes to both 2D and

3D data while still using cropping to produce a rich set of pretraining data [9], requires no external models and works out-of-the-box with heterogeneous datasets [10]. The strategy consists in utilizing only the crops with overlap from a certain Intersection over Union (IoU) interval. The intuition behind such an approach is that maximizing similarity between image crops is beneficial only if they are semantically related and that for some of the domains, such crops appear mostly within a single area (and therefore crops with overlap > 0 are more likely to constitute a semantically related pair), while crops from remote areas (overlap = 0) might depict separate downstream classes. On the other hand, too high overlap (≈ 1) might present trivially similar examples and not create a strong learning signal. To find the optimal IoU interval, several non-overlapping IoU ranges were evaluated by pretraining with SimSiam [2] on 2,000 unlabelled cases and fine-tuning on organ segmentation task with Swin UNETR [11], using 50 labelled cases. All the images and labels used in this thesis come from the FLARE 2022 challenge [12]. Due to computational restrictions, 2D input images and square crops of equal sizes were considered only, although the method generalizes easily to 3D data as well as rectangular crops of any size. Under the aforementioned setting, when compared to random parameters initialization, pretraining with the best performing IoU interval $[0.3, 0.6]$ allowed to obtain an average improvement of $+0.01$ Dice score (DSC), whereas pretraining with the default random cropping strategy harmed the performance by -0.02 DSC. In the context of self-supervised learning in the medical imaging domain, the conducted experiments[1] demonstrate the advantage of the proposed method over the widely used random cropping strategy. They also highlight a non-trivial nature of applying recent SSL methods directly on medical images. However, to fully confirm the effectiveness of the method, broader experiments (including various datasets, multiple SSL algorithms and comparison against alternative random cropping strategies) are needed.

The thesis is structured as follows: Chapter 2 provides a theoretical introduction to readers who are less familiar with applications of deep learning and medical imaging, as well as an overview of related works. Chapter 3 provides details of the proposed method. Chapter 4 describes the experimental setup in depth, presents the results and offers a follow-up discussion. Chapter 5 summarizes and concludes the thesis.

[1] The code is publicly available at *https://github.com/dittohed/exploring-ssl-for-ct*.

2. Theoretical Background

The following chapter aims to provide a theoretical foundation on which the present work has been built. Firstly, it outlines the current general approach to solving computer vision problems with the use of deep learning. Next, some ways of minimizing the manual labelling effort are presented. Afterwards, the characteristics of computed tomography are introduced. The chapter concludes with a review of related works, while also identifying potential avenues for improvement.

2.1. Computer Vision Models

Computer vision is a field of study focused on processing, analyzing and understanding digital images, aimed at solving tasks such as image classification or object detection.

The vast majority of the current top-performing computer vision methods is based upon *deep learning* [5], which mostly consists in applying a composition of multiple matrix multiplications and various non-linear functions (constituting a *deep neural network*), to process input data (e.g. an image) into the desired output (e.g. a label indicating whether the image contains a cat). Such an approach results in a new paradigm of solving problems, that allows for tackling complex tasks without defining all the steps explicitly. To find optimal parameters (also known as *weights*) of the network, a given loss function is minimized by iteratively computing the gradient of the function with respect to all the weights in the network and updating them.

2.1.1. Convolutional Neural Networks

Historically, one of the first successful groups of neural networks for analyzing images were the *convolutional neural networks* (CNNs) [13]. Their unprecedented ability to solve visual tasks was manifested throughout successive editions of The ImageNet Large Scale Visual Recognition Challenge [5]. Starting from AlexNet [14], they have continued to outperform handcrafted features approaches and remained a standard approach, also in the medical images domain [15].

In the context of deep learning, *convolution* stands for a sliding window operation involving 2 matrices, an image I and a kernel K of width and height $2k+1$, where the kernel K contains learnable parameters, typically expressed with a cross-correlation term [16]:

$$S(i,j) = (K * I)(i,j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} K(m,n)I(i+m, j+n), \tag{2.1}$$

where $S(i,j)$ stands for an element in i-th row and j-th column of the resulting matrix S, and elements of K are indexed from $-k$ to k in each axis.

Intuitively, during training, parametrized kernels are trained to detect various features within a local, sliding window, resulting in new pseudo images (*feature maps*), within which high values correspond to presence of a certain feature and low values – lack of the feature. In practice, a convolutional neural network is a composition of multiple, stacked convolutions with additional intermediate layers (such as dropout [17] or batch normalization [18]).

In this thesis, CNN is utilized as part of a deep learning model for *organ segmentation*, a task where each pixel in the image is assigned a label corresponding to either one of the organs or the background.

2.1.2. Vision Transformers

Recently, alternative computer vision approaches based on the *transformer architecture* [19] emerged. The transformer architecture was originally designed to tackle machine translation tasks (e.g. translating texts between English and French). When compared to previous approaches, it obtained noticeably better results at a small fraction of the training costs and has quickly dominated the natural language processing domain by becoming a crucial part of multiple seminal methods such as BERT [20] and GPT [21]. The main idea behind transformers is to process a sequence of input tokens at once and use the attention mechanism to dynamically model inter-token dependencies. Concretely, three seperate linear layers are used to compute Q (*queries*), K (*keys*) and V (*values*) matrices respectively, using vector representations (*embeddings*) of the input tokens. Then, the attention function updates each token embedding by calculating a weighted sum of the embeddings of all tokens stored in V, where the weights are determined using vectors in Q and K:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.2}$$

where d_k is the dimensionality of the key vectors.

In 2020, [22] presented the first pure transformer-based architecture, the *vision transformer* (ViT), and since then, many follow-up works have been published [23, 24].

Adapting transformer to image domain mostly consists in reshaping the image $x \in \mathbb{R}^{h \times w \times c}$ into a sequence of n 2D crops (tokens) $x_c \in \mathbb{R}^{p^2 \cdot c}$, where h and w denote height and width of the input image, c stands for a number of input channels and p is the height and width of each crop (usually, $p = 16$).

One significant step to make the vision transformers perform on par with CNNs on a range of computer vision tasks was proposed by [24]. To improve the performance of the original ViT [22] on dense prediction tasks (e.g. object detection and semantic segmentation), decreasing input crop size is desirable. However, such modification alone would lead to a noticeable memory cost increase due to attention computation. To this end, the authors of [24] proposed to limit attention to a predefined window size (*local attention*) only and created a local window shifting strategy to allow cross-window connections and thereby greater modeling power. The method also includes merging neighboring tokens to model global dependencies. Applying all these modifications resulted in an architecture dubbed Swin (short for shifted windows), a well-performing hierarchical general-purpose transformer backbone that surpassed the previous state-of-the-art on object detection and semantic segmentation benchmarks.

In this thesis, Swin [24] is utilized as a foundation of all the models.

2.2. Pretraining

To solve a particular task of interest, deep learning models usually require a large number of labelled examples to obtain satisfactory performance. However, acquiring a big, annotated dataset might be expensive and time-consuming for many real-world applications, especially when expert personnel is required to accomplish the labelling process.

One common technique for addressing the problem of limited training data is *transfer learning* [25]. It consists in training a model on a large dataset before training the model on the goal task. This reduces the amount of training data needed to achieve good performance. In transfer learning, the pretrained model is typically a deep neural network that has been trained in a supervised way on a large annotated dataset, such as ImageNet [5]. However, applying transfer learning is not always suitable, e.g. when the target domain differs from the pretraining dataset domain or when the target modality is 3D instead of 2D. Such discrepancy occurs for some of the medical imaging types such as computed tomography.

An alternative method of pretraining deep neural networks is *self-supervised learning* [26]. In contrast to typical supervised pretraining, it does not require any human annotations and can learn useful representations from unlabeled data thanks to various *pretext tasks*. Self-supervised learning has become a popular approach in deep learning, particularly in natural language processing [20, 21], where large amounts of unlabeled data are readily available. Recently, it has also started being applied successfully in computer vision, beating supervised pretraining on various tasks [1, 3, 4, 27].

2.3. Computed Tomography

This thesis considers problems stemming from applying recent self-supervised methods on computed tomography. *Computed tomography* (CT) is a non-invasive medical imaging technique that uses X-rays and computer algorithms to produce detailed 3D image of the body.

The image is obtained by passing X-ray beams through the body at various angles while detecting the rays on the opposite side, and then reconstructing the image using computer algorithms [28]. Each of the voxels, i.e. unit cubes of a three-dimensional image, takes values from the *Hounsfield scale*, which determines radiodensity of substances and allows them to be distinguished from each other, especially bones from the surrounding soft tissues [29]. Values reported for various substances are determined empirically (Table 2.1).

Table 2.1. Approximated HU values for various substances [30]

Substance	HU
air	-1000 to -600
fat	-100
muscles	50
water	0
bones	> 250

Computed tomography scan results in a series of 2D grayscale images with pixel values according to the Hounsfield scale [29], where successive images display successive cross sections of the body (Figure 2.1). Each scan might be different in terms of slice thickness (a fixed distance between successive 2D images, expressed in millimeters), pixel spacing (a fixed distance between the centers of adjacent pixels along X, Y and Z axes, expressed in millimeters) and the scope of the scan (the examined part of the patient's body).

2.4. Related Work

Recent years have witnessed a rapid development of self-supervised visual feature learning approaches. The vast majority of the new methods have been proposed and evaluated based on natural images. Often, such approaches would be adapted to other domains only after a certain time. The following section will present the reader with an overview of some of the leading methods and identify possible improvements in applying SSL to computed tomography.

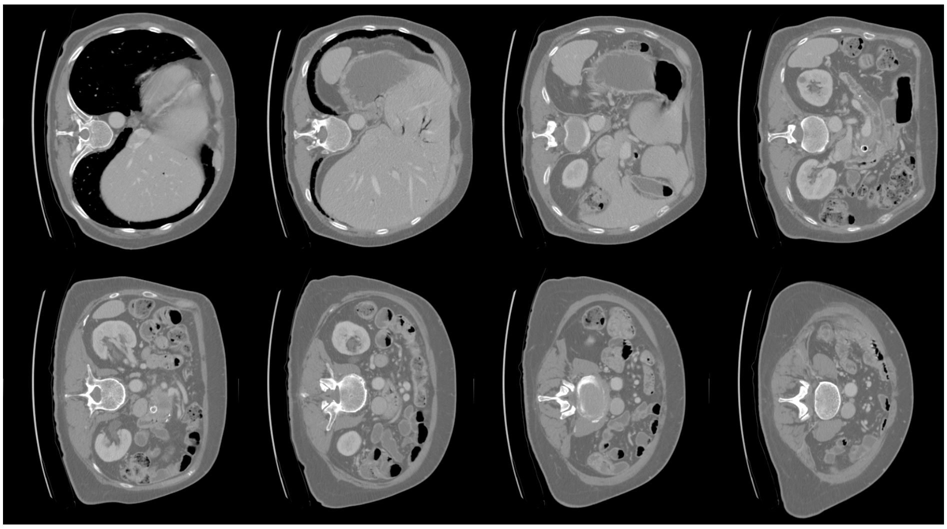


Figure 2.1. A series of successive cross-sections making up a 3D image, ordered in rows, from left to right; the cross-section in the upper left corner is at the level of esophagus, the subsequent cross-sections are images of progressively lower parts of the body (original CT source: FLARE 2022 dataset [12]).

2.4.1. Self-supervised Learning for Natural Images

A noticeable increase of interest in the subject could be observed for the first time around the year 2015. For several years, predictive and generative methods were dominating.

Predictive methods. The first branch of early research focused on framing the pretext task as a predictive (classification or regression) problem [31]. In *predictive self-supervised learning*, each image is assigned a pseudo label derived from the image itself. Some of the prominent methods following this approach include training the models to: predict an index of the original image given its heavily augmented version [32], output relative positions of 2 neighboring patches from the same image [33], solve jigsaw puzzle [34] or predict image rotation (0, 90, 180 or 270 degrees) applied on input [35]. However, at some point, it was found out that the predictive pretext tasks approach lacked generality – the authors of [36] noticed that such strategies encourage models to learn representations that are covariant to the applied transformations and proved that this leads to overfitting to particular pretext tasks.

Generative methods. An alternative branch of early research took advantage of the generative abilities of deep neural networks [31]. Namely, in *generative self-supervised learning*, models learn useful latent representations by being trained to regenerate original images using their modified versions or to generate new examples from a given distribution. Examples of such methods include training the models to: restore masked parts of images [37], regenerate the original colorized image given its gray-scale version [38] or generate images similar to training ones [39]. Soon, however, contrastive learning approaches started outperforming the generative ones.

Contrastive methods. *Contrastive learning* [31] aims at learning useful representations by forcing the model to return similar vector representations for similar (*positive*) inputs and (optionally) dissimilar representations for dissimilar (*negative*) inputs. Positive pairs correspond to semantically similar examples and are usually generated by creating 2 randomly transformed views of the same image, whereas negative pairs correspond to semantically dissimilar examples and are usually created by taking 2 distinct images, although typically a greater number of negative samples is used due to improved performance [40, 41]. Initial contrastive learning approaches utilize both positive and negative pairs for constructing the loss. MoCo [42] maintains a queue (*dictionary* or *memory bank*) of previously encoded samples to calculate InfoNCE loss [40], where many negative samples are needed. For every step, 2 perturbations of each image in a batch are generated. The first one enters the goal encoder (*query encoder*), while the second one enters another encoder (*momentum encoder*) with weights updated using the goal encoder by exponential moving average (EMA), and is added to the queue in a first-in-first-out manner. MoCo [42] was one of the first self-supervised methods to outperform supervised pretraining counterparts on several detection and segmentation tasks. Sometime later, SimCLR [41] showed that a proper composition of heavy data augmentations for generating positive views is crucial in improving performance, namely that simple random cropping combined with color jitter is essential. Since then, a significant part of more recent studies [1–4] would follow SimCLR and include random cropping along with other strong transformations (e.g. color distortions, Gaussian blur and flipping) as a default approach for generating correlated pairs. Additionally, the authors of SimCLR [41] showed that using large batch sizes, instead of maintaining a memory bank with negative samples, also allows for effective pretraining.

Contrastive methods with positives only. Recent advancements in contrastive learning have demonstrated that negative pairs may not be necessary to achieve state-of-the-art performance [1–4]. However, removing negative samples from pretraining loss leads to reducing the constraints on feature learning. This increases the risk of a feature collapse, a solution where model returns the same representation for every input while still achieving a small loss [2]. All the positives-only methods may be therefore considered as various proposals for counteracting the aforementioned issue. They typically share the overall structure, where positive views are processed in parallel by 2 networks, where the first one includes an encoder while the second one either includes the same encoder [1, 2] or a moving average of the first encoder [3, 43], updated periodically. DeepCluster [44] and its extended version, SwAV [1], use a cluster-discrimination-based method to learn useful representations. The methods boil down to training a model to assign correlated crops to the same cluster. SwAV [1] uses an MLP head that calculates the probabilities of assigning an image to each cluster. Each training step updates both the encoder's weights as well as the cluster centers. To avoid feature collapse, the authors use the Sinkhorn-Knopp method [45] that ensures equal distribution between the clusters. However, of particular significance, [1] proposed MultiCrop, a strategy to increase the number of positive views per image by using multiple global (covering larger image areas) and local (covering smaller image areas) views within a single training step. The approach was shown to improve performance of several self-supervised methods and has gained widespread adoption in recent literature. BYOL [43] learns feature representations by minimizing mean squared error between l_2-normalized vector representations of positive crops. It avoids trivial solutions by applying a momentum encoder strategy and appending an additional MLP on top of the student network. SimSiam [2] utilizes neither large batches nor momentum encoder strategy. Instead, it obtains good results thanks to applying the stop-gradient operation on one branch of the model and an additional MLP on top of the second branch. DINO [3] extends BYOL [43] – instead of appending predictor on top of the student, it utilizes centering and sharpening operations to process teacher output before calculating softmax and cross-entropy loss. *Centering* refers to the process of calculating the mean feature vector across a batch of samples, then subtracting it from logits before calculating softmax. As each component is normalized independently, centering prevents any single dimension from dominating, but at the same time, encourages convergence to a uniform distribution. *Sharpening*, on the other hand, involves dividing logits by a constant which increases the distance between highest and lowest values, encouraging a collapse where a single dimension always dominates. Applying both centering and sharpening balances their effects and thereby counteracts trivial solutions.

Masked-image methods. The significant success of masked language modeling in NLP [20, 21] and the growing presence of vision transformers has recently prompted researchers to revisit self-supervised methods based on masking (e.g. [37]). This gave rise to the most recent branch of research called *masked image modeling* (MIM). One of the first works in this area, Masked AutoEncoder (MAE) [46], uses an encoder-decoder ViT to reconstruct randomly masked patches from the input image. To prevent recovering missing patches with low-level understanding only, a high portion of 75% input patches is masked. A more recent work, image BERT pretraining with Online Tokenizer (iBOT) [4], combines MIM with self-distillation. The framework takes random crops of the same image. The student network takes the views with randomly masked patches, while the teacher network, being the EMA version of the student, takes the views without any masking. The goal is to train the student to output hidden representations for the masked patches so that they are equal to the ones from the teacher. This way, the trained ViT is forced to restore only a high-level representation of the masked patches rather than original pixels. This prevents wasting the model's capacity to model high-frequency, negligible details [4]. Additionally, global representations of the views are forced to be similar between the 2 models to make sure the global semantics are also captured.

2.4.2. Self-supervised Learning in Medicine

Although SSL techniques applied to natural images give measurable benefits, it is not always straightforward to adapt such methods to the medical domain due to inevitable differences such as input dimensionality (3D instead of 2D) or smaller variance of background and object appearance across multiple examples in the case of medical images. Therefore, while some works [47–50] just utilize or combine SSL techniques developed on natural images, other works [51–56] propose novel methods tailored for the medical domain. The following section outlines the general direction of SSL in medicine while providing several examples.

Just as in the context of natural images, predictive methods were developed for medicine, including predicting spatial order of 2 successive CT or MRI slices [51], predicting the correct, original order and rotation of randomly shuffled 3D subcubes that the input image is split into (interpreted as Rubik's cube recovery) [52] and predicting whether input CT or MRI slice was corrupted by replacing several patches between neighboring slices, performed by predicting the neighbor's offset with respect to the input image [53].

Generative SSL methods designed for medical images encompass: colorizing endoscopy video [47], reconstructing an input image corrupted by swapping disconnected crops multiple times [54] and recovering Rubik's cube using GAN-based architecture [55].

Just as for natural images, the newest methods in medicine mostly utilize contrastive learning and MIM. Prior-Guided Local (PGL) self-supervised learning [56] extends BYOL [43] by replacing the loss with a local consistency function, aimed at minimizing the difference between the aligned feature maps. Such alignment is possible by knowing the spatial transformations, which produce augmented views of the same image. In [48], the authors successfully pretrained a model for analyzing X-ray images using MoCo [42]. In [49], the authors proposed to combine contrastive learning, image inpainting and image rotation prediction. In [50], the authors perform successful pretraining with MAE [46] and SimMIM [57] on MRI and CT datasets.

2.4.3. Positives Selection

A significant part of the recent self-supervised methods mentioned in Section 2.4.1 utilize various random cropping strategies for generating positive views. However, most of these methods are usually developed using the object-centric ImageNet dataset, where images mostly contain single objects. That is why it might remain unnoticed that such a cropping strategy might underperform for datasets of different kinds, e.g. for medical images where random views could depict unrelated organs, constituting separate classes in a downstream segmentation task (Figure 2.2).

Alternative strategies for natural images. One of the first works to take up the problem of suboptimal random cropping effects was [6], where the authors pretrained 2 models, one on

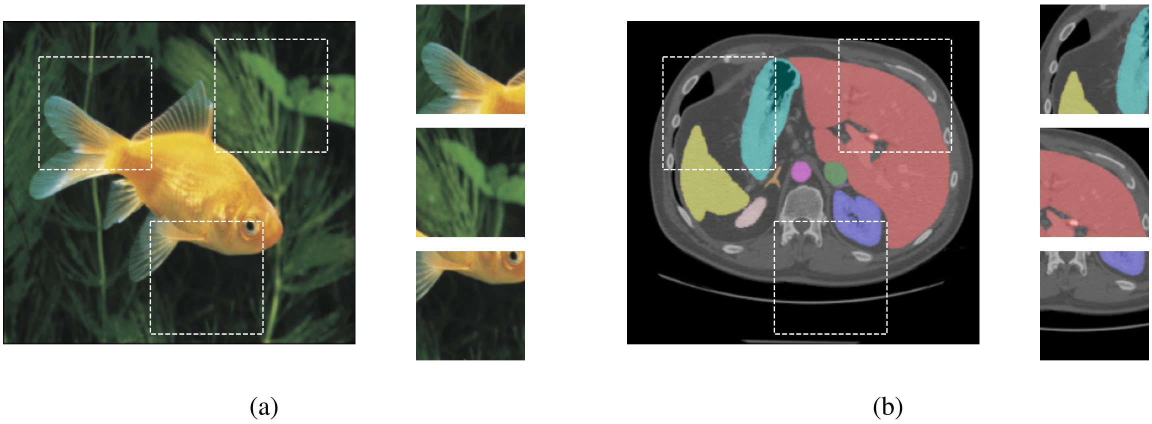


Figure 2.2. Visualization of random crops (denoted with dashed lines) for images of object-centric and medical datasets: (a) crops of object-centric images either point to the only object of interest or contain a correlated background (original image source: ImageNet dataset); (b) crops of medical images might contain distinct downstream objects of interest such as various organs, here denoted with distinct colors (original image source: FLARE 2022 dataset).

MSCOCO [58] dataset, typically containing multiple objects, and one on an ImageNet subset of comparable size. The results showed that the model pretrained on MSCOCO obtained noticeably worse discrimination power when fine-tuned for image classification. It has shown that aggressive cropping negatively affects fine-tuned model performance if not pretrained on an object-centric dataset. The authors of ContrastiveCrop technique [59] discuss the same issue. Having observed that, during training, a heatmap created from activations of the last convolutional layer roughly indicates the object on image, they propose to generate crop centers only from the heatmap areas. This way, falsely correlated crops are avoided. However, since that strategy then leads to many trivial positives, the authors additionally use a beta distribution to reduce the probability of gathering crops from around the heatmap centers. All in all, [59] proposes a method that generates positives while avoiding both distant false positives as well as very close easy positives and yields improvements across multiple pretraining algorithms and datasets. However, such a method cannot be directly adapted to CT, as even a single crop might contain multiple objects of interest making it impossible to distinguish between the central object and background. On the other hand, the method proposed in [59] can be simplified to a strategy of taking crops so that their overlap is neither too big (to avoid easy positives) nor too small (to avoid false positives), resulting in a simple and intuitive approach that is further studied in the present paper.

Alternative strategies for medical images. Strategies for enhancing default random cropping have been also explored specifically in the medical images domain. PGL [56] aims at improving downstream CT segmentation task performance by enforcing at least 10% overlap between positive views and imposing consistency of output feature maps of 2 positive views, aligned by using information about applied transformations. With respect to pure BYOL that PGL is built on, it improves performance on organ segmentation by a small margin. Multi-Instance Contrastive Learning (MICLe) [7] and MedAug [8] utilize metadata for sampling positive pairs. Views are considered positive only if they correspond to the same patient. The methods are used for pretraining on 2D images such as chest X-rays. The drawback of such an approach is a lack of technical generalizability – two full 3D CT images belonging to the same patient can't be fed into a neural network due to memory limitations, while any cropping could lead to generating false positive views. The authors of Positional Contrastive Learning strategy (PCL) [9] focus on SSL approaches including negative pairs and note that the direct application of such methods on CT and MRI problems inevitably introduces a lot of false negative pairs due to the existence of the same tissue or organ across the dataset. To this end, they propose a method that utilizes 2 observations to decide whether 2D slices can be considered positive or not: 1) adjacent 2D slices contain similar content; 2) assuming that different CT volumes are perfectly aligned, the corresponding slices often show similar structures. Although this method shows that leveraging the position information in volumetric medical images helps improving downstream performance, it is restricted to using whole slices (no spatial cropping is used). Finally, Alice [10] generates semantically consistent positive crops by using a pretrained SAM [60] model which locates the same body parts in different volumetric medical images, producing 2 highly related, yet different views. The method outperformed other techniques on organ segmentation benchmarks. However, it requires an additional, pretrained model and is suited for datasets with images depicting similar body parts.

2.4.4. Summary

To sum up, a significant part of self-supervised learning techniques, including those applied in the medical field, utilize random cropping to generate positive views. While such a strategy allows to obtain impressive results when used for object-centric datasets, it proves to be less effective for datasets that might contain multiple downstream classes within a single image. To counteract this problem, multiple effective techniques have been recently studied. However, all of them entail some additional assumptions. ContrastiveCrop [59] is a strategy for object-centric datasets with more background. Although this method cannot be applied directly to CT (as it is not object-centric), it can be interpreted as a strategy for avoiding distant false positives and close easy positives. MICLe [7] and MedAug [8] require multiple images of the same patient and applying the techniques for 3D data might not be possible due to memory requirements. PCL [9] utilizes whole slices only and Alice uses an external pretrained model and works best with homogenous data.

The thesis aims to address the aforementioned limitations by introducing a novel strategy of sampling positive crops based on the Intersection over Union (IoU) metric. The method is inspired by ContrastiveCrop [59], it does not require multiple images of the same patient or any metadata, generalizes to both 2D and 3D data, requires no external models and works out-of-the-box with heterogeneous datasets.

3. Method

In this chapter, *IoU-based positives*, a technique for sampling semantically related positive pairs using Intersection over Union (IoU) is proposed. First, the IoU metric is defined and the general idea behind the method is described, then a strategy for finding optimal values of the method's hyperparameters is proposed. Finally, implementation aspects are discussed.

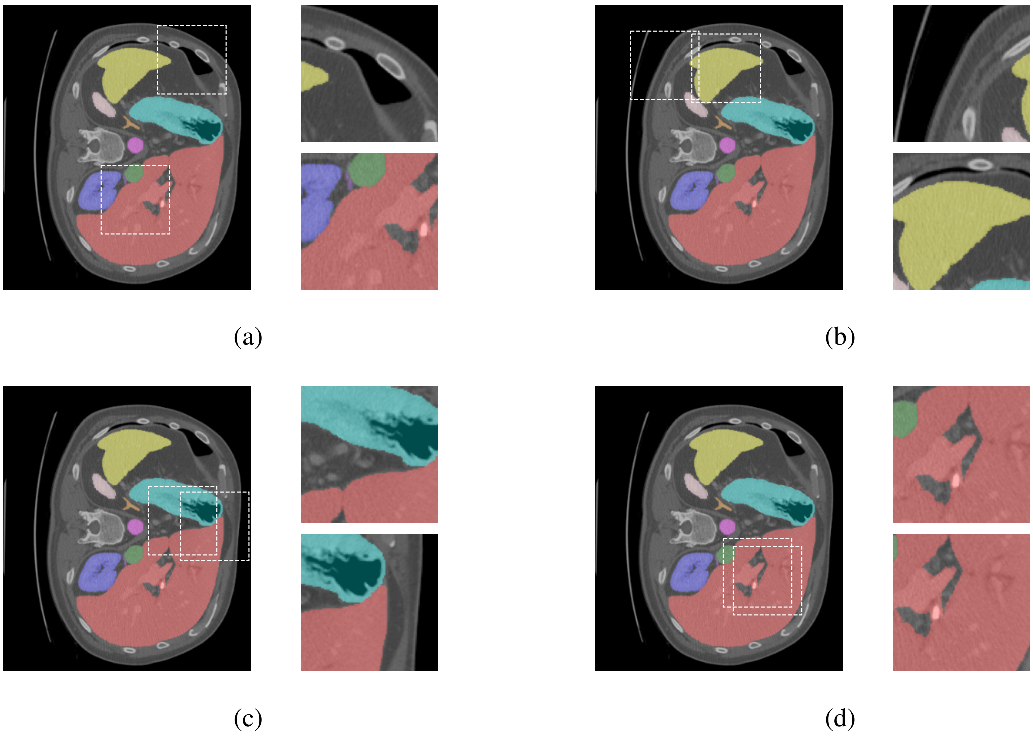
Intersection over union. *Jaccard index*, also known as *Intersection over Union* (IoU) measures similarity between 2 finite sets A and B:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{3.1}$$

where, if A and B are image crops, $|A|$ denotes number of pixels belonging to crop A, $|B|$ denotes number of pixels belonging to crop B and $|A \cap B|$ denotes number of pixels belonging to both crops at the same time. $IoU(A, B) = 0$ indicates no overlap between the crops, whereas $IoU(A, B) = 1$ represents a perfect overlap. In practice, knowing the crops' coordinates with respect to the whole image, IoU is calculated using only the coordinates of the upper-left and bottom-right corners that span the crops.

General idea. The proposed method works as follows. Given a 2D or 3D input image x, sample 2 positive crops x_1, x_2 so that $IoU(x_1, x_2) \in [i_{min}, i_{max}]$, where i_{min} and i_{max} correspond to minimum and maximum IoU allowed for the crops to be considered positive. The intuition behind such an approach is that for some of the domains, semantically consistent crops appear mostly within a single area, while crops from remote areas might depict various downstream classes. On the other hand, too high overlap might introduce easy positives that do not create a training signal. Due to computational restrictions, the present thesis considers 2D input images and square crops of equal sizes only, although the method generalizes easily to 3D data as well as rectangular crops of any size.

Finding optimal values. The values of i_{min} and i_{max} must be determined empirically. To assess optimal i_{min} and i_{max} for medical imaging pretraining, a few non-overlapping IoU intervals were first evaluated, only then the final interval $[i_{min}, i_{max}]$ was determined. The non-overlapping intervals were chosen so that they would reflect various levels of semantic and visual similarity between the positive crops: $[0, 0]$, $[0.0001, 0.3]$, $[0.3, 0.6]$ and $[0.6, 1.0]$ (Figure


(a)

(b)

(c)

(d)

Figure 3.1. Examples of crops with IoU from the chosen IoU intervals: (a) $[0, 0]$, the crops show no semantic similarity; (b) $[0.0001, 0.3]$, the crops have low visual similarity although both contain parts of the same organ; (c) $[0.3, 0.6]$, the crops have moderate visual similarity and both contain significant parts of the same organs; (d) $[0.6, 1]$, the crops have high visual similarity and both contain significant parts of the same organ.

3.1). The expected influence of the specific intervals was the following: 1) weak learning signal for $[0, 0]$ and $[0.6, 1.0]$ intervals due to either rare semantic similarity or trivial significant spatial proximity, 2) strong learning signal for $[0.0001, 0.3]$ and $[0.3, 0.6]$ intervals generating semantically related, yet non-trivial, pairs. To verify these expectations, over 25K positive pairs (corresponding to 5 iterations over all the 2D labeled images, more details on the dataset in Section 4.3) per IoU interval were randomly generated and analyzed in terms of semantic similarity (Figure 3.2).

Implementation aspects. To generate pairs with IoU from a predefined interval $[i_{min}, i_{max}]$ during training, a simple sampling strategy is utilized (see Algorithm 1 for pseudo-code implementation). Once the original input image x is loaded, x_1 coordinates are randomly sampled (Line 2), then x_2 coordinates are sampled up to r_1 times (Lines 3-8). If no x_2 coordinates are found so that $IoU(x_1, x_2) \in [i_{min}, i_{max}]$ within r_1 retries, x_1 coordinates are sampled again and the whole process is repeated up to r_2 times (Lines 1-9). To avoid costly sampling and infinite loops, a random pair is returned in case of no success (Line 10). The reasoning for resampling

x_1 coordinates only after r_1 unsuccessful retries instead of generating x_1 and x_2 coordinates jointly every sampling step is to eliminate a potential bias of avoiding hard x_1 crops around the image center for which finding x_2 (especially if $i_{min} = i_{max} = 0$ and input image x is relatively small) might require multiple retries.

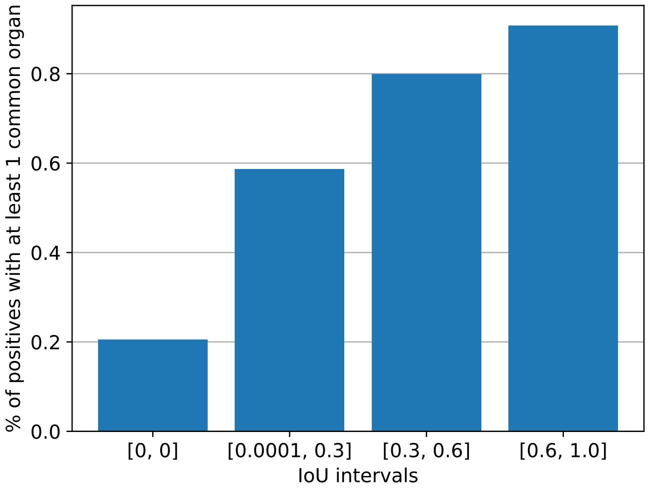


Figure 3.2. Approximate fraction of pairs with at least 1 common organ across various IoU intervals (cases where both views contained background only were excluded). $[0, 0]$ interval provides few semantically related samples when compared to others.

Algorithm 1 Generating IoU-based positives

```
Require: $i_{min}, i_{max} \in [0, 1]$ where $i_{min} \leq i_{max}$ and $r_1, r_2 \in \mathbb{N}^+$
Ensure: $x_1, x_2$
for $p$ = 1 to $r_2$ do
    Sample $x_1$ crop from $x$
    for $q$ = 1 to $r_1$ do
        Sample $x_2$ crop from $x$
        if $IoU(x_1, x_2) \in [i_{min}, i_{max}]$ then
            return $x_1, x_2$
        end if
    end for
end for
return $x_1, x_2$
```

4. Experiments

The following chapter presents the details of the experiments conducted to evaluate the proposed method. Firstly, the pretraining algorithm and the computer vision backbone are described. Next, the dataset, evaluation metrics and implementation details are presented. Finally, the results are discussed.

4.1. Pretraining Algorithm

SimSiam [2] was picked as a representative self-supervised learning method. Like many other state-of-the-art techniques (e.g. [3, 4, 10]), it learns by maximizing similarity between crops of the same image, while being simple and easy to interpret compared to other methods.

SimSiam (Figure 4.1) takes a pair (x_1, x_2) of randomly augmented views of input image x and firstly feeds it into a neural network f consisting of a backbone and a *projection MLP head* which yields hidden representations $z_1 = f(x_1)$ and $z_2 = f(x_2)$. To avoid representation collapse, it further processes one of the views with a *prediction MLP head* h, giving $p_i = h(z_i)$. The process is symmetrized for x_1 and x_2, resulting in the following loss function to minimize:

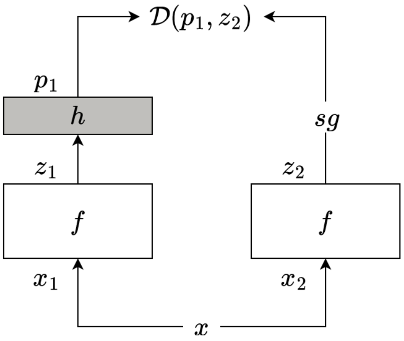


Figure 4.1. SimSiam architecture (depicted for a non-symmetrized loss)

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_1, sg(z_2)) + \frac{1}{2}\mathcal{D}(p_2, sg(z_1)), \tag{4.1}$$

where $\mathcal{D}$ denotes negative cosine similarity between two, l_2-normalized vectors and sg denotes the stop-gradient operation (which prevents using gradients with respect to the argument during backpropagation). Both sg operation and prediction head h are vital to preventing feature collapse [2].

The present work follows the original SimSiam architecture setting. The projection head consists of 3 layers (2 hidden + an output layer) with $d = 2048$ dimensions each, batch normalization applied to each layer and ReLU activation applied to each layer apart from the output one. Also, learnable affine transformation in batch normalization is disabled for the output layer. The prediction head is composed of 2 layers (bottleneck + output layer) with bottleneck dimension $d_b = 512$ and output dimension $d_o = 2048$, where batch normalization and ReLU are applied only to the bottleneck layer.

4.2. Backbone

Swin UNETR [11] was picked as the backbone. As most "U-shaped" [61] architectures that are used for segmentation in medicine, it consists of: 1) an encoder that progressively reduces the spatial dimensions of the input image while capturing semantic-rich features; 2) a decoder that consists of upsampling layers that gradually increase the spatial dimensions back to the original size; 3) skip connections between the encoder and decoder blocks that allow obtaining both high-resolution and semantic-rich representations. The backbone uses a slightly modified version of the Shifted windows transformer (Swin) [24] as the encoder. Swin utilizes shifting local attention windows, enabling effective processing of large images with small input tokens. This innovation allows Swin to serve as a general-purpose backbone, strengthening the growing advantage of vision transformers over CNNs. The present work follows the model architecture from [11], apart from using 2D input instead of 3D. The details of the utilized model are described below.

Encoder. Input image $x \in \mathbb{R}^{H \times W}$ is first split into non-overlapping 2×2 patches, that are further flattened into vectors and projected into a C-dimensional space using a linear embedding layer. Such projected patches are further processed in 4 stages, where each stage corresponds to a block consisting of 2 swin transformer blocks and a patch merging layer. Swin transformer block is a slightly modified typical transformer block [19] with multi-head self-attention (MSA) substituted with a module based on shifted windows (Figure 4.2). Each stage uses 2 swin transformer blocks with local window size M, where the first one uses a default window partitioning strategy (W-MSA, top-left window starts from the top-left element in feature map), while the second one (SW-MSA) uses windows shifted to bottom-right corner by $\lfloor \frac{M}{2} \rfloor$ elements. Restricting calculation of attention only to local windows decreases computational complexity while shifting windows in the second block prevents a lack of connections between windows that would lead to decreased modeling power [24]. The patch merging layer assigns patches neighboring in the feature map to 2×2 patch groups, then concatenates the patches and applies a linear layer with an output dimension of $2C$. This effects in decreasing spatial resolution by 2

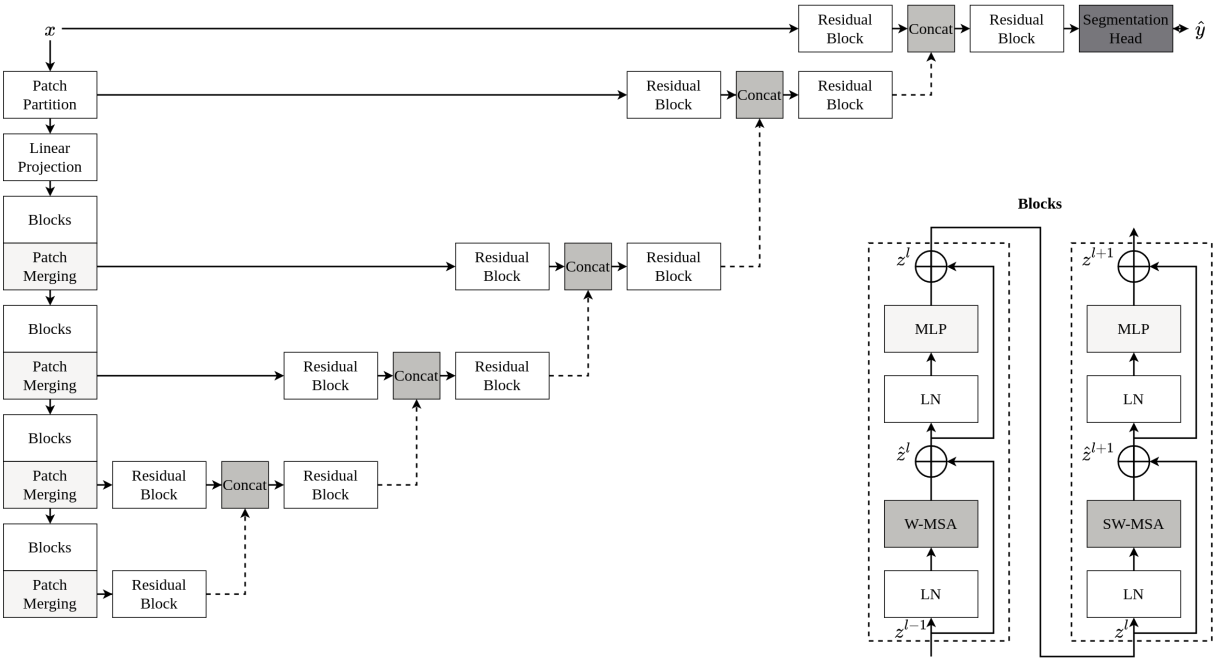


Figure 4.2. Swin UNETR architecture (left) and Swin transformer blocks (right). Dashed lines in the left diagram denote upsampling with transposed convolution. Each transformer block consists of a window-based MSA module, followed by an MLP module with GeLU activation in between. Each module is preceded by Layer Norm (LN) and followed by a residual connection.

while doubling hidden representation dimensionality. Contrary to the original Swin, Swin UNETR uses one extra patch merging layer at the very end. As in [11], C is set to 48, M is set to 7, whereas the number of attention heads is set to 3 for the first stage and is doubled in every follow-up stage.

Decoder. Output features of each encoder stage are reshaped to a corresponding spatial feature map and processed using convolutional layers. Each feature map is fed into a residual block, then concatenated with the also processed and upsampled feature map from the stage below, then fed into another residual block, then finally upsampled and concatenated with a processed feature map from the stage above. Every residual block consists of two 3×3 convolutional layers with instance normalization. Upsampling is performed with transposed convolutions. Outputs from the patch split module and the original input are processed analogously. To calculate segmentation predictions, a 1×1 convolutional layer with softmax activation function is applied to the final feature map.

4.3. Dataset & Preprocessing

FLARE 2022 challenge [12] dataset is used both for pretraining and fine-tuning. The dataset is collected from more than 20 medical groups. The training set includes 50 CT scans with voxel-level labels of 13 abdominal organs and 2,000 unlabeled CT scans. The validation set includes 50 visible unlabeled cases. The testing set includes 200 hidden cases.

The present work utilizes the challenge's training set only. For pretraining of the backbone, all the 2,000 unlabeled CT scans are used without any metadata. For fine-tuning the whole Swin UNETR, 50 labeled cases are utilized. To evaluate the IoU intervals thoroughly, a majority of cases are assigned to a validation subset: 35 cases are included in the fine-tuning validation subset and 15 cases are included in the fine-tuning training subset.

The training set images are normalized before training. Namely, all the scans are reoriented to the RAS axes order (where X and Y axes span back-front and right-left respectively, while Z-axis represents the direction from bottom to top, see Figure 2.1) and interpolated into the isotropic voxel spacing of $[1, 1, 2.5]$ mm. Then voxel values are further clipped to $[-500, 500]$ window and normalized to $[0, 1]$. Finally, each scan is cropped to the smallest possible cube containing all non-zero voxels, then X and Y dimensions are padded symmetrically to have at least 96 voxels. Such processed scans are saved slice by slice along the Z-axis, which yields 237,449 normalized, unlabeled 2D images for pretraining and 5,050 normalized, labeled 2D images for organ segmentation fine-tuning.

4.4. Evaluation Metrics

The Dice similarity coefficient (DSC) measured on the validation subset is used to evaluate organ segmentation fine-tuning:

$$\frac{2\sum_{i=1}^{I} Y_i\hat{Y}_i}{\sum_{i=1}^{I} Y_i + \sum_{i=1}^{I} \hat{Y}_i}, \tag{4.2}$$

where I denotes number of pixels in the image, Y and $\hat{Y}$ denote ground truth and prediction segmentation mask respectively. It measures overlap between two sets and is widely employed to evaluate the performance of segmentation algorithms. 0 indicates no overlap between the predicted and the ground truth segmentation, whereas 1 represents a perfect overlap.

Additionally, as in [2], a collapse coefficient is tracked during pretraining with SimSiam:

$$max(0, 1 - \sqrt{d} \cdot s), \tag{4.3}$$

where d denotes number of output dimensions and s denotes mean per-channel standard deviation of l_2-normalized prediction head output. It measures the extent to which the output collapses to a constant vector (in such case, standard deviation equals 0 for each output channel when averaged across the input batch). As derived in [2], a non-degenerated output scattered on a unit hypersphere corresponds to a standard deviation of $\frac{1}{\sqrt{d}}$.

4.5. Implementation Details

Data augmentation. To allow successful pretraining and improve generalization of the fine-tuned models, random augmentations are used in the following order. For pretraining (independently for each crop): $[0.8, 1.2]$ zoom with $p = 0.75$, X-axis flip with $p = 0.5$, Y-axis flip with $p = 0.5$, rotation by one of $\{90°, 180°, 270°\}$ with $p = 0.5$, intensity scaling[1] with factor sampled from $[-0.1, 0.1]$ and $p = 0.75$, intensity shift[2] with offset sampled from $[-0.1, 0.1]$ and $p = 0.75$, gaussian blur with σ_x sampled from $[0.25, 1.5]$, σ_y sampled from $[0.25, 1.5]$ and $p = 0.25$, gaussian noise with $\mu = 0$, $\sigma = 0.01$ and $p = 0.25$. For fine-tuning, 2 crops are first extracted per each image in batch, so that their central pixels correspond to a labeled organ with $p = 0.5$. Then, the following augmentations are used: X-axis flip with $p = 0.25$, Y-axis flip with $p = 0.25$, rotation by one of $\{90°, 180°, 270°\}$ with $p = 0.25$, gaussian blur with σ_x sampled from $[0.25, 1.5]$, σ_y sampled from $[0.25, 1.5]$ and $p = 0.15$, intensity scaling with factor sampled from $[-0.1, 0.1]$ and $p = 0.15$, intensity shift with factor sampled from $[-0.1, 0.1]$ and $p = 0.15$, gaussian noise with $\mu = 0$, $\sigma = 0.01$ and $p = 0.15$.

Trainings. Settings used for pretraining and fine-tuning experiments are presented in Table 4.1. During pretraining, weight decay is not applied to batch normalization parameters and biases, cosine decay is not applied to the prediction head. Additionally, drop path rate of 0.1 is used for the Swin UNETR backbone. During fine-tuning, weight decay is not applied to batch normalization parameters and biases. Validation DSC is evaluated every 10 epochs from the beginning and every epoch starting from 90th epoch. Training is stopped whenever there is no improvement in validation DSC for 20 epochs. A checkpoint corresponding to the best validation DSC is saved and referred to as the model's score. Inference is performed using a sliding window approach with a window size of 96×96 and an overlap of 0.25 between the windows. All computation during pretraining and fine-tuning is performed under automatic mixed precision.

IoU-based cropping. For all pretraining runs, r_1 was set to 30 and r_2 was set to 4. With such a setting, crops satisfying predefined IoU intervals could not be found occasionally (below 0.5% of all pairs used during pretraining) for $[0, 0]$ and $[0.6, 1.0]$ intervals only. Python's `time.process_time()` function was used for the assessment of extra CPU time required by the proposed method.

[1] Modifies intensity of input image x given factor α with $x = x \cdot (1 + \alpha)$

[2] Modifies intensity of input image x given offset β with $x = x + \beta$

Table 4.1. Training settings used for pretraining and fine-tuning experiments.

	Pretraining	Fine-tuning
No. of epochs	100	225
Batch size	128	32
Optimizer	SGD (momentum $= 0.9$)	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate	initial 0.025 + cosine decay to 1e-6	10-epoch linear warmup to 0.001 + cosine decay to 5e-5
Weight decay	1e-5	1e-5

Technical setup. All the training code is implemented in PyTorch [62] and MONAI [63] and run on a single NVIDIA V100 GPU (with 9.1GB of maximum memory allocated) and an Intel Xeon Gold 6242 CPU (10 cores for pretraining, 4 cores for fine-tuning) available thanks to the Polish high-performance computing infrastructure PLGrid (ACK Cyfronet AGH).

4.6. Results

IoU intervals evaluation. To evaluate the IoU intervals, two pretrainings were executed per each interval. Subsequently, each pretrained model served as a starting point for two downstream organ segmentation trainings, resulting in four Dice scores per IoU interval. The $[0, 1]$ interval, corresponding to random cropping, was evaluated in the same way to compare IoU-based cropping to the default strategy. To make the results as comparable as possible, the same set of random seeds was used for both random weights initialization and data loading across all the IoU intervals. Additionally, four organ segmentation trainings without any form of pretraining were run. Table 4.2 presents a comparison between the intervals and training from scratch.

Compared to training from scratch, only the models pretrained with the $[0.3, 0.6]$ interval obtained better results ($+0.01$ DSC), while maintaining low variance between the runs. It is also worth noticing that the models pretrained with the widely used random cropping, yielded the worst results on average.

Feature collapse. It was observed that the lowest fine-tuning scores were related to high collapse coefficient values tracked during pretraining. As depicted in Figure 4.3, pretraining with intervals $[0, 0]$ and $[0, 1]$ quickly leads to degenerated solutions.

Time costs. To assess the additional CPU cost required for finding IoU-based positives, the time needed for basic loading of 1,000 batches of size 128 was measured for 2 IoU intervals: 1) $[0, 1]$ (random cropping); 2) $[0.3, 0.6]$ (optimal). For this experiment, any further data augmentation was excluded. Loading a batch of size 128 with plain random cropping took $0.489s \pm 0.086$

Table 4.2. Results of the non-overlapping IoU intervals evaluation. All the scores are measured on 35 validation CT scans. The first row corresponds to fine-tuning with no previous pretraining. The last row corresponds to the widely used random cropping strategy ($[0, 1]$ interval). Apart from the individual results, mean and standard deviation are reported.

IoU interval	Mean DSC	DSC 1	DSC 2	DSC 3	DSC 4
-	0.843 ± 0.004	0.842	0.848	0.846	0.836
$[0, 0]$	0.838 ± 0.007	0.836	0.831	0.834	0.850
$[0.0001, 0.3]$	0.839 ± 0.012	0.847	0.831	0.855	0.824
$[0.3, 0.6]$	$\mathbf{0.853 \pm 0.002}$	0.854	0.851	0.851	0.857
$[0.6, 1]$	0.843 ± 0.013	0.850	0.855	0.848	0.822
Random crop.	0.824 ± 0.011	0.815	0.841	0.828	0.812

on average, while $0.508s \pm 0.084$ was needed for $[0.3, 0.6]$ interval, which approximately yields a 4% increase of time required for basic processing before further augmentation.

Under the technical setup described in Section 4.5, pretraining and fine-tuning took on average 14.9 and 4.3 GPU hours, respectively. All the experiments together (10 pretrainings and 24 fine-tunings, see Table 4.2) consumed 252 GPU hours.

4.7. Discussion

When compared to both 1) pretraining with random cropping and 2) training from scratch, the proposed method allows to obtain better results at negligible extra CPU time cost. Since only the $[0.3, 0.6]$ interval improved the score when compared to no pretraining, the approximately optimal i_{min} and i_{max} values should be set to 0.3 and 0.6 correspondingly.

As hypothesized in Chapter 3, generating positive views with IoU in $[0, 0]$ (mostly remote, semantically unrelated views) and $[0.6, 1]$ (very high visual and semantic similarity) intervals don’t create a strong learning signal for the CT domain, which results in unsuccessful pretraining. Contrary to the expectations, the $[0.0001, 0.3]$ interval also fails to generate proper positive pairs.

Surprisingly, the worst results are produced by the $[0, 1]$ interval, which corresponds to the default random cropping strategy. This is most likely due to the model being forced to solve multiple conflicting tasks simultaneously. Specifically, at the same time, the model is trained to return similar representations for 1) unrelated views ($[0, 0]$ interval); 2) adjacent and weakly related views ($[0.0001, 0.3]$ interval); 3) moderately and strongly related views ($[0.3, 0.6]$ and

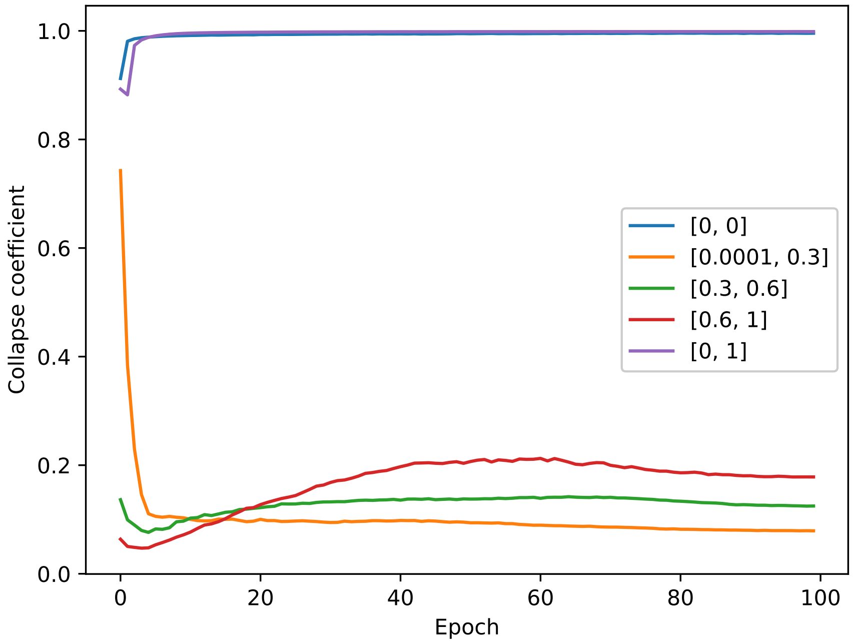


Figure 4.3. Collapse coefficient during pretraining with various IoU intervals. $[0, 0]$ and $[0, 1]$ quickly lead to feature collapse, while other ranges remain more stable. The values in the plot correspond to single runs, however, the same dynamics could be observed for other random initializations.

$[0.6, 1]$ intervals). To test this hypothesis, an experiment consisting of pretraining with alternating IoU intervals within a single run could be conducted for both object-centric and CT datasets. However, such an experiment falls outside the scope of the present work.

The obtained mean DSC scores (see Table 4.2) are within the range of the results reported in the FLARE 2022 leaderboard[3], that were measured on the 50 visible unlabeled cases (the best submission obtained 0.906 DSC). The thesis aimed to compare various cropping strategies rather than attaining highest results possible, therefore the part of enhancing the model and evaluating it on the 50 visible unlabeled cases was skipped. To increase the absolute results, one could use 3D input instead of 2D [64], utilize a more recent self-supervised method such as iBOT [4], and perform cross-validation with a smaller validation portion (when compared to the 15/35 train-validation split used in the thesis).

Currently, the proposed method has only been applied to 2D CT images. Therefore the effectiveness of sampling positive crops with IoU from the $[0.3, 0.6]$ interval for 3D images (also from other medical imaging modalities like MRI) and 2D natural scene images (that also may contain multiple objects of interest within a single image) remains unknown. This, together

[3] *https://flare22.grand-challenge.org/evaluation/challenge/leaderboard/*

with utilizing a broader set of SSL techniques for evaluation, outlines the potential direction of the follow-up studies. Additionally, to confirm the effectiveness when more labels are available, larger downstream training subsets should be included.

5. Summary

This thesis aimed to introduce and evaluate *IoU-based positives*, a novel strategy of generating semantically aligned positive pairs for visual self-supervised learning. The method consists in utilizing only crops with an overlap from a specific Intersection over Union (IoU) interval and can be easily employed with multiple SSL algorithms. The intuition behind creating the strategy was that maximizing similarity between crops is beneficial only if they are 1) semantically related (e.g. presenting parts of the same organs in medical imaging) and 2) sufficiently different in order to create a strong learning signal. *When compared to the random cropping strategy used in many SSL techniques, in the context of medical imaging, the proposed method improves downstream performance at negligible extra CPU time. Contrary to previous works presenting alternative strategies for generating positive crops, it does not require multiple images of the same patient or metadata, generalizes to both 2D and 3D data, uses no external models and works seamlessly with heterogeneous datasets.*

The proposed strategy was evaluated by pretraining with 2,000 CT cases using SimSiam [2] and fine-tuning with 50 CT cases on organ segmentation task with Swin UNETR [11]. All the images and labels used in this work were obtained from the FLARE 2022 challenge [12]. Due to computational restrictions, this study considered 2D input images and square crops of equal sizes only, although the method generalizes easily to 3D data as well as rectangular crops of any size. Various, non-overlapping IoU intervals were extensively tested. The only interval that improved the downstream performance when compared to no pretraining was $[0.3, 0.6]$, providing $+0.01$ DSC improvement. Surprisingly, the default random cropping method yielded the lowest score among all the variants, with a -0.02 DSC decrease. Overall, the results demonstrate the advantage of the proposed method over the default strategy of generating positive pairs in the context of self-supervised learning in the medical imaging domain. They also highlight that utilizing SSL for medical images is not trivial and that not all the methods developed with natural images can be directly transferred while maintaining optimal performance.

In the future, IoU-based positives could be evaluated on tasks involving 3D medical images and natural scene images, including a broader set of self-supervised learning techniques and bigger downstream training subsets. Additionally, the approach could be compared against MultiCrop [1] strategy.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] Mathilde Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9912–9924.

[2] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15745–15753. DOI: *10.1109/CVPR46437.2021.01549*.

[3] Mathilde Caron et al. "Emerging Properties in Self-Supervised Vision Transformers". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9630–9640. DOI: *10.1109/ICCV48922.2021.00951*.

[4] Jinghao Zhou et al. "Image BERT Pre-training with Online Tokenizer". In: *International Conference on Learning Representations*. 2022.

[5] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Apr. 2015), pp. 211–252. DOI: *10.1007/s11263-015-0816-y*.

[6] Senthil Purushwalkam and Abhinav Gupta. "Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 3407–3418.

[7] Shekoofeh Azizi et al. "Big Self-Supervised Models Advance Medical Image Classification". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 3458–3468. DOI: *10.1109/ICCV48922.2021.00346*.

[8] Yen Nhi Truong Vu et al. "MedAug: Contrastive learning leveraging patient metadata improves representations for chest X-ray interpretation". In: *Proceedings of the 6th Machine Learning for Healthcare Conference*. Ed. by Ken Jung et al. Vol. 149. Proceedings of Machine Learning Research. PMLR, June 2021, pp. 755–769.

[9] Dewen Zeng et al. "Positional Contrastive Learning for Volumetric Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*. Springer International Publishing, 2021, pp. 221–230. DOI: *10.1007/978-3-030-87196-3_21*.

[10] Yankai Jiang et al. *Anatomical Invariance Modeling and Semantic Alignment for Self-supervised Learning in 3D Medical Image Segmentation*. 2023. arXiv: *2302.05615* [cs.CV].

[11] Ali Hatamizadeh et al. "Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Ed. by Alessandro Crimi and Spyridon Bakas. Cham: Springer International Publishing, 2022, pp. 272–284.

[12] Jun Ma and Bo Wang, eds. *Fast and Low-Resource Semi-supervised Abdominal Organ Segmentation*. Springer Nature Switzerland, 2022. DOI: *10.1007/978-3-031-23911-3*.

[13] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: *10.1162/neco.1989.1.4.541*.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012.

[15] D. R. Sarvamangala and Raghavendra V. Kulkarni. "Convolutional neural networks in medical image understanding: a survey". In: *Evolutionary Intelligence* 15.1 (Jan. 2021), pp. 1–22. DOI: *10.1007/s12065-020-00540-3*.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. *http://www.deeplearningbook.org*. MIT Press, 2016.

[17] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[18] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456.

[19] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[20] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: *10.18653/v1/N19-1423*.

[21] Tom Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[22] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021.

[23] Hugo Touvron et al. “Training data-efficient image transformers & distillation through attention”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 10347–10357.

[24] Z. Liu et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2021, pp. 9992–10002. DOI: *10.1109/ICCV48922.2021.00986*.

[25] Chuanqi Tan et al. “A Survey on Deep Transfer Learning”. In: *Artificial Neural Networks and Machine Learning – ICANN 2018*. Ed. by Věra Kůrková et al. Cham: Springer International Publishing, 2018, pp. 270–279.

[26] Linus Ericsson et al. “Self-Supervised Representation Learning: Introduction, advances, and challenges”. In: *IEEE Signal Processing Magazine* 39.3 (2022), pp. 42–62. DOI: *10.1109/MSP.2021.3134634*.

[27] Yucheng Tang et al. “Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 20698–20708. DOI: *10.1109/CVPR52688.2022.02007*.

[28] Haijo Jung. “Basic Physical Principles and Clinical Applications of Computed Tomography”. In: *Progress in Medical Physics* 32.1 (Mar. 2021), pp. 1–17. DOI: *10.14316/pmp.2021.32.1.1*.

[29] Richard Bibb, Dominic Eggbeer, and Abby Paterson. “2 - Medical imaging”. In: *Medical Modelling (Second Edition)*. Ed. by Richard Bibb, Dominic Eggbeer, and Abby Paterson. Second Edition. Oxford: Woodhead Publishing, 2015, pp. 7–34. ISBN: 978-1-78242-300-3. DOI: *https://doi.org/10.1016/B978-1-78242-300-3.00002-0*.

[30] Francis H. Glorieux, John M. Pettifor, and Harald Jüppner. *Pediatric Bone*. Elsevier, 2012. DOI: *10.1016/c2009-0-63398-5*.

[31] Saeed Shurrab and Rehab Duwairi. "Self-supervised learning methods and applications in medical imaging analysis: a survey". In: *PeerJ Computer Science* 8 (July 2022), e1045. DOI: *10.7717/peerj-cs.1045*.

[32] Alexey Dosovitskiy2014 et al. "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.

[33] C. Doersch, A. Gupta, and A. A. Efros. "Unsupervised Visual Representation Learning by Context Prediction". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Dec. 2015, pp. 1422–1430. DOI: *10.1109/ICCV.2015.167*.

[34] Mehdi Noroozi and Paolo Favaro. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Nicu Matas Jiriand Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 69–84. ISBN: 978-3-319-46466-4.

[35] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations". In: *International Conference on Learning Representations*. 2018.

[36] I. Misra and L. van der Maaten. "Self-Supervised Learning of Pretext-Invariant Representations". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2020, pp. 6706–6716. DOI: *10.1109/CVPR42600.2020.00674*.

[37] D. Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2016, pp. 2536–2544. DOI: *10.1109/CVPR.2016.278*.

[38] Richard Zhang, Phillip Isola, and Alexei A. Efros. "Colorful Image Colorization". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 649–666. ISBN: 978-3-319-46487-9.

[39] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *International Conference on Learning Representations*. Nov. 2016.

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2019. arXiv: *1807.03748* `[cs.LG]`.

[41] Ting Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 1597–1607.

[42] Kaiming He et al. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. DOI: *10.1109/cvpr42600.2020.00975*.

[43] Jean-Bastien Grill et al. “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21271–21284.

[44] Mathilde Caron et al. “Deep Clustering for Unsupervised Learning of Visual Features”. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 139–156. ISBN: 978-3-030-01264-9.

[45] Marco Cuturi. “Sinkhorn Distances: Lightspeed Computation of Optimal Transport”. In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc., 2013.

[46] Kaiming He et al. “Masked Autoencoders Are Scalable Vision Learners”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 16000–16009.

[47] Tobias Ross et al. “Exploiting the potential of unlabeled endoscopic video data with self-supervised learning”. In: *International Journal of Computer Assisted Radiology and Surgery* 13.6 (Apr. 2018), pp. 925–933. DOI: *10.1007/s11548-018-1772-0*.

[48] Hari Sowrirajan et al. “MoCo Pretraining Improves Representation and Transferability of Chest X-ray Models”. In: *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*. Ed. by Mattias Heinrich et al. Vol. 143. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 728–744.

[49] Yucheng Tang et al. “Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 20698–20708. DOI: *10.1109/CVPR52688.2022.02007*.

[50] Zekai Chen et al. “Masked Image Modeling Advances 3D Medical Image Analysis”. In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 1969–1979. DOI: *10.1109/WACV56688.2023.00201*.

[51] Pengyue Zhang, Fusheng Wang, and Yefeng Zheng. "Self supervised deep representation learning for fine-grained body part recognition". In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. 2017, pp. 578–582. DOI: *10.1109/ISBI.2017.7950587*.

[52] Xinrui Zhuang et al. "Self-supervised Feature Learning for 3D Medical Images by Playing a Rubik's Cube". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Ed. by Dinggang Shen et al. Cham: Springer International Publishing, 2019, pp. 420–428.

[53] Xuan-Bac Nguyen et al. "Self-Supervised Learning Based on Spatial Awareness for Medical Image Analysis". In: *IEEE Access* 8 (2020), pp. 162973–162981. DOI: *10.1109/ACCESS.2020.3021469*.

[54] Liang Chen et al. "Self-supervised learning for medical image analysis using image context restoration". In: *Medical Image Analysis* 58 (2019), p. 101539. ISSN: 1361-8415. DOI: *https://doi.org/10.1016/j.media.2019.101539*.

[55] Xing Tao et al. "Revisiting Rubik's Cube: Self-supervised Learning with Volume-Wise Transformation for 3D Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Ed. by Anne L. Martel et al. Cham: Springer International Publishing, 2020, pp. 238–248. ISBN: 978-3-030-59719-1.

[56] Yutong Xie et al. *PGL: Prior-Guided Local Self-supervised Learning for 3D Medical Image Segmentation*. 2020. arXiv: *2011.12640* `[cs.CV]`.

[57] Zhenda Xie et al. "SimMIM: a Simple Framework for Masked Image Modeling". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 9643–9653. DOI: *10.1109/CVPR52688.2022.00943*.

[58] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[59] Xiangyu Peng et al. "Crafting Better Contrastive Views for Siamese Representation Learning". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 16010–16019. DOI: *10.1109/CVPR52688.2022.01556*.

[60] Ke Yan et al. "SAM: Self-Supervised Learning of Pixel-Wise Anatomical Embeddings in Radiological Images". In: *IEEE Transactions on Medical Imaging* 41 (2020), pp. 2658–2669.

[61] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241.

[62] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[63] M. Jorge Cardoso et al. “MONAI: An open-source framework for deep learning in healthcare”. In: (Nov. 2022). DOI: *https://doi.org/10.48550/arXiv.2211.02701*.

[64] Xiangrong Zhou et al. “Performance evaluation of 2D and 3D deep learning approaches for automatic segmentation of multiple organs on CT images”. In: *Medical Imaging 2018: Computer-Aided Diagnosis*. Ed. by Kensaku Mori and Nicholas Petrick. SPIE, Feb. 2018. DOI: *10.1117/12.2295178*.