Computers in Biology and Medicine 183 (2024) 109242

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Less is More: Selective reduction of CT data for self-supervised pre-training of deep learning models with contrastive learning improves downstream classification performance

Daniel Wolf [a,b,*], Tristan Payer [a], Catharina Silvia Lisson [b], Christoph Gerhard Lisson [b], Meinrad Beer [b], Michael Götz [b,1], Timo Ropinski [a,1]

[a] Visual Computing Research Group, Institute of Media Informatics, Ulm University, James-Franck-Ring, Ulm, 89081, Germany
[b] Experimental Radiology Research Group, Department for Diagnostic and Interventional Radiology, Ulm University Medical Center, Albert Einstein Allee, Ulm, 89081, Germany

ARTICLE INFO

Dataset link: https://github.com/Wolfda95/Less_is_More

Keywords:
Deep learning
Medical imaging
Computed Tomography (CT)
Self-supervised pre-training
Contrastive learning
Transfer learning

ABSTRACT

Background: Self-supervised pre-training of deep learning models with contrastive learning is a widely used technique in image analysis. Current findings indicate a strong potential for contrastive pre-training on medical images. However, further research is necessary to incorporate the particular characteristics of these images.
Method: We hypothesize that the similarity of medical images hinders the success of contrastive learning in the medical imaging domain. To this end, we investigate different strategies based on deep embedding, information theory, and hashing in order to identify and reduce redundancy in medical pre-training datasets. The effect of these different reduction strategies on contrastive learning is evaluated on two pre-training datasets and several downstream classification tasks.
Results: In all of our experiments, dataset reduction leads to a considerable performance gain in downstream tasks, e.g., an AUC score improvement from 0.78 to 0.83 for the COVID CT Classification Grand Challenge, 0.97 to 0.98 for the OrganSMNIST Classification Challenge and 0.73 to 0.83 for a brain hemorrhage classification task. Furthermore, pre-training is up to nine times faster due to the dataset reduction.
Conclusions: In conclusion, the proposed approach highlights the importance of dataset quality and provides a transferable approach to improve contrastive pre-training for classification downstream tasks on medical images.

1. Introduction

Supervised training of a deep learning model requires large and accurate datasets. Annotations are necessary for all training samples. In the medical imaging domain, annotated datasets for specific tasks are often limited due to factors such as the rarity of diseases, limited access, or the high complexity of annotations [1,2]. To overcome this challenge, deep learning models can be pre-trained on large medical image datasets without annotations, using self-supervised learning techniques [3]. These techniques train the models to create meaningful representations from unlabeled datasets, allowing them to learn general high-level features of the images. To fine-tune the models for specific tasks, the so-called "downstream tasks", small annotated datasets are sufficient after pre-training. Contrastive learning is a state-of-the-art approach for self-supervised pre-training on unannotated images [4] and according to Huang et al.'s study [3], currently the most popular approach in the medical imaging domain. Several works show remarkable performance gains on medical downstream tasks when the models are pre-trained with contrastive learning on large unannotated medical image datasets compared to training the models from scratch [5–10]. Despite the great potential of contrastive pre-training techniques in the medical domain, the special characteristics of volumetric radiological images, consisting of many consecutive slices, such as CT, MRI or PET, have not been sufficiently exploited. In our work, we evaluate the composition of the pre-training datasets for contrastive learning on CT scans.

When it comes to deep learning on CT scans in general, there are two widely used approaches, both of which show excellent results on clinically relevant imaging tasks. The first approach is to train on the

* Corresponding author at: Experimental Radiology Research Group, Department for Diagnostic and Interventional Radiology, Ulm University Medical Center, Albert Einstein Allee, Ulm, 89081, Germany.
E-mail address: daniel.wolf@uni-ulm.de (D. Wolf).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.compbiomed.2024.109242
Received 16 January 2024; Received in revised form 30 September 2024; Accepted 1 October 2024
Available online 9 October 2024

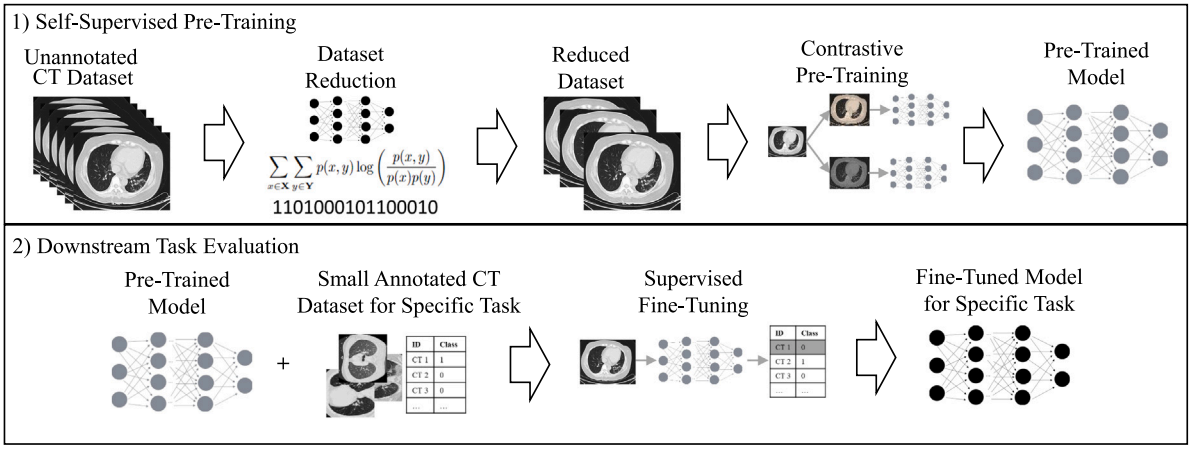

Fig. 1. This figure gives an overview of our approach to investigate the hypothesis that using all CT slices for contrastive pre-training may lead to performance degradation due to the high similarity of the slices. The first step is to pre-train a deep learning model. Therefore, we start with a dataset of unannotated CT slices, select slices in such a way that we obtain a reduced dataset with increased variation, and pre-train the model with contrastive learning on the reduced dataset. The second step is to evaluate the pre-training on downstream tasks. Therefore, the pre-trained model is fine-tuned with supervised learning on small datasets with annotations for the specific task. Our work compares different strategies to reduce CT image pre-training datasets.

whole CT volumes using a 3D model [11,12], and the second approach is to train on individual slices of the volumes using a 2D model [13–16]. Each approach has its own advantages. While training a 3D model on volumes enables the model to better capture the 3D properties of the images [17], training a 2D model by using each slice of a volume separately can improve performance on small datasets due to the increased sample size [18–20], and reduces the computational cost of training and inference as smaller GPUs are sufficient [17–19,21,22]. For both approaches, there are several publications that investigate self-supervised pre-training with contrastive learning, achieving significant performance gains on several CT image downstream tasks. Tang et al. [6] and Dufumier et al. [10] pre-train 3D models on CT volumes, while Wolf et al. [23], Ghesu et al. [5] and Chen et al. [7] pre-train 2D models on CT slices. In this study, we chose to conduct our experiments with 2D models because we see that it is critical for deep learning to be globally accessible without the need for powerful GPUs, and the advantages of 2D models for sparse data remain significant even when using pre-trained models, as small annotated datasets are a major challenge in the field of medical imaging.

Contrastive learning involves the following steps: A dataset of unlabeled images is used as a starting point. Random augmentations are applied to generate multiple randomly varied samples of each original image. These are fed into a deep learning model to obtain latent-space representations for each sample. The model always compares two representations and is trained to discriminate whether these are derived from the same original image (referred to as positive pairs) or derived from different original images (referred to as negative pairs). Previous works on contrastive pre-training with CT slices have included as many slices as possible, following the traditional approach of maximizing the pre-training dataset [5,7,23]. In this paper, we hypothesize that using all slices of each CT volume in a dataset for contrastive pre-training may lead to performance degradation. We derive our hypothesis from the fact that CT datasets have very low variance compared to natural image datasets due to the high similarity of the CT slices. This may result in the model being unable to discriminate between positive and negative pairs since the similarity between two augmented versions of a slice might be lower than the similarity between two different slices. Our hypothesis is supported by recent work that provides preliminary evidence that this may be a challenge in contrastive learning. Using ImageNet data, Jing et al. [24] show that a lower variance of the data distribution than the variance caused by the data augmentation of contrastive learning leads to performance degradation in downstream tasks. Conrad and Narayan [25] show on electron microscopy images that low variance in the pre-training dataset affects downstream task results.

To investigate our hypothesis that using all CT slices for contrastive pre-training may lead to performance degradation, we explore various strategies based on deep embedding, information theory, and hashing to identify and reduce redundancy in pre-training datasets. Fig. 1 illustrates our general approach. Starting with a dataset of unannotated CT slices, we perform different reduction strategies and pre-train the models with contrastive learning on the reduced datasets. The pre-trainings are evaluated on downstream tasks by fine-tuning the pre-trained models with supervised learning. We choose two pre-training datasets and three downstream classification tasks, the benchmark task for evaluating self-supervised pre-training [3]. The outcomes support our hypothesis, as the downstream results improve with our dataset reduction strategies. Furthermore, we investigate which reduction strategy is best suited for CT datasets and what is the optimal threshold that represents the best trade-off between high variation but also a sufficiently large number of samples in the pre-training dataset to achieve the best downstream results. Finally, our work provides a ready-to-use model for improving self-supervised pre-training on CT datasets for classification downstream tasks. These findings have the potential to improve the handling of small annotated CT datasets while maintaining low computational costs. The pre-trained models, as well as the ready-to-use code, are available on GitHub: https://github.com/Wolfda95/Less_is_More

2. Materials and methods

In this section, we explain in detail the methods for investigating our hypothesis that using all slices of each CT volume for contrastive pre-training may lead to performance degradation due to the high similarity of the slices. We first present strategies for selecting slices of CT volumes to obtain a reduced pre-training dataset with increased variation. This is followed by describing the contrastive pre-training methods and datasets. Finally, we introduce the downstream tasks to evaluate the impact of the reduction strategies on contrastive pre-training.

2.1. Dataset reduction

We investigate our hypothesis by comparing six approaches for slice selection: two baseline approaches and four similarity-based approaches. The similarity-based approaches perform a pairwise comparison between all slices in a volume. A similarity score is calculated

for each slice pair. The pairs are sorted from most similar to most dissimilar. Starting from the most similar pair, one slice is removed from the pairs until either all pairs have similarities below a given threshold or until a given number of slices is left. We incorporate commonly used similarity computation methods from different fields, such as information theory, deep embedding, and hashing, without claiming completeness. The methods we selected are well-established for image comparison and computationally fast, which is necessary due to the large number of pairwise comparisons.

ALL: The first baseline approach follows the current state of the art [5,7]. All slices are included in the training.

EveryN: The second baseline approach is our baseline reduction method. Here, CT datasets are reduced by using every nth slice of a volume.

SSIM: As our first similarity-based approach, we use the Structural Similarity Index (SSIM) [26] from information theory, which is a common similarity measure for images [27]. It compares the luminance, contrast, and structure of two given images $\mathbf{x}$ and $\mathbf{y}$ by the equation

$$\mathrm{d}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + (K_1L)^2)(2\sigma_{xy} + (K_2L)^2)}{\left(\mu_x^2 + \mu_y^2 + (K_1L)^2\right)\left(\sigma_x^2 + \sigma_y^2 + (K_2L)^2\right)}, \tag{1}$$

where μ_x, μ_y and σ_x, σ_x are the mean and standard deviation and σ_{xy} the covariance of all pixel values of two images. To avoid instability, $(K_1L)^2$ and $(K_2L)^2$ are added, where L is the dynamic range of the pixel values and $K_1 = 0.01$ and $K_2 = 0.03$ are small constants. SSIM is computed as the average result of a moving 11×11 kernel with a Gaussian weighting function. The parameters were chosen as suggested by Wang et al. [26].

MI: We use Mutual Information (MI) as the second similarity-based approach from information theory. MI is a widely used technique for similarity calculation and registration of medical images [28,29] and measures the dependence between two images $\mathbf{x}$ and $\mathbf{y}$ by calculating the Kullback–Leibler divergence

$$\mathrm{KL}(\mathbf{X} \parallel \mathbf{Y}) = \sum_{x\in\mathbf{X}}\sum_{y\in\mathbf{Y}} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right), \tag{2}$$

between the joint distribution $p(x,y)$ and the independent distributions $p(x)p(y)$ of the pixel values. We use the normalized Mutual Information as introduced by Studholme et al. [30].

DeepNet: Motivated by the success of the Perceptual Similarity Metric [31] for image comparison, which is computationally expensive, we introduce DeepNet similarity, which reduces the complexity so that pairwise comparisons can be performed in a reasonable amount of time. Like perceptual similarity, DeepNet similarity compares two images by running them through a pre-trained deep learning model. Instead of computing the cosine similarity in the channel dimension, DeepNet similarity computes the cosine similarity between the output vectors. Using PyTorch's ResNet50 [32] pre-trained on ImageNet [33] to compute the output vectors, we get the following equation

$$\mathrm{d}(\mathbf{x},\mathbf{y}) = \frac{\mathrm{ResNet}(\mathbf{x}) \cdot \mathrm{ResNet}(\mathbf{y})}{\|\mathrm{ResNet}(\mathbf{x})\|_2 \, \|\mathrm{ResNet}(\mathbf{y})\|_2}, \tag{3}$$

to compare two images $\mathbf{x}$ and $\mathbf{y}$.

HASH: The HASH similarity is based on the comparison of hash values derived from each image. It is motivated by Conrad and Narayan [25], who used it to extract dissimilar images from an electron microscopy dataset for contrastive pre-training. The procedure is as follows: Each image is compressed to the size of 9×8 pixel and encoded into a 64-bit hash. The compression is performed with the Antialias function from Pillow [34]. The hash is computed by looping through each row of the compressed image, comparing each pixel with its right neighbor, and selecting one if the neighbor is larger and zero if the neighbor is smaller. For each row of nine pixels, this results in a hash of eight bits, leading to a 64-bit hash in total. The Hamming distance

$$\mathrm{d}(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} |\mathrm{Hash}(\mathbf{x})_i - \mathrm{Hash}(\mathbf{y})_i|, \tag{4}$$

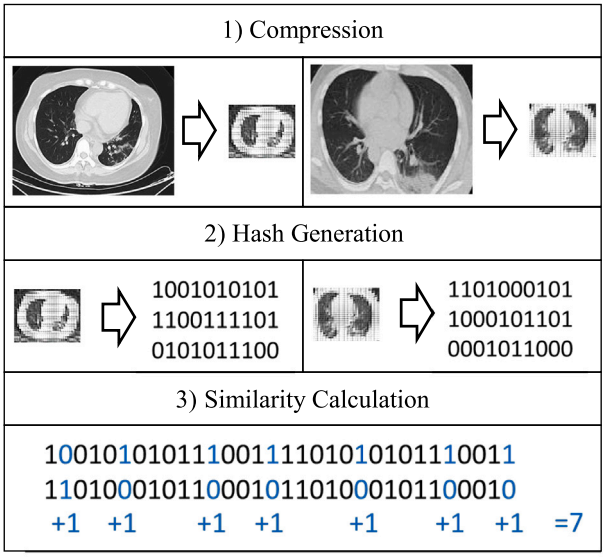

Fig. 2. This figure explains the similarity calculation between two images using the HASH method. First, the images are compressed to the size of 9×8. In the second step, a 64-bit hash is computed by looping through each row of the compressed images, comparing each pixel with its right neighbor, and choosing one if the neighbor is larger and zero if the neighbor is smaller. To calculate the similarity, the hashes for the two images are compared with the Hamming distance, which is the sum of the different bits.

between the two hashes of images $\mathbf{x}$ and $\mathbf{y}$ measures the similarity, where $n = 64$ is the length of the hash. All parameters are chosen following Conrad and Narayan [25]. Fig. 2 illustrates the similarity calculation between two images with the HASH method.

2.2. Pre-training

Following Huang et al.'s. [3] study, popular contrastive learning methods from natural image processing that are widely utilized for medical pre-training are SimCLR [35], MoCo [36], BYOL [37], and SwAV [38]. SimCLR follows the basic contrastive learning strategy, where the model is trained to distinguish between positive and negative pairs within a mini-batch. This requires a large batch size in order to obtain a sufficient number of negative samples within one mini-batch. MoCo adds a queue for storing negative samples, which reduces batch size requirements but increases storage demands. BYOL introduces two competing models to decrease batch size demands. SwAV adds online feature clustering to the latent space representations, which lowers the batch-size constraints.

For our evaluations, we chose the method SwAV [38], since it outperformed the other state-of-the-art contrastive learning methods with convolutional models on several natural imaging benchmark tasks [39], is more computational efficient [38], and was already successfully applied for pre-training on CT slices [5,23]. Due to the identical basic concept of all methods, our findings are expected to be generalizable to other contrastive pre-training methods. A detailed explanation of the SwAV pre-training method can be found in Appendix A. In order to create positive pairs of one original image, SwAV uses the transforms color jitter, Gaussian blur, and a multi-crop strategy, where two transformed images are obtained by cropping a part of the original image with a larger crop size, and several additional samples are cropped with a smaller crop size. For our evaluations, we use exactly the transform settings of the original paper, as they have been shown to be the most appropriate for this pre-training method. Details can be found in Appendix A.

The pre-training is performed on the CT slices of two publicly available image datasets, summarized in Table 1:

Table 1
Pre-training datasets.

	PET-CT	LIDC
Modality	CT	CT
Body part	Whole body	Lung
Volumes	900	1010
Slices	541,439	244,527
Availability	Public	Public

Table 2
Downstream tasks to evaluate the pre-trainings.

	COVID-19	OrgMNIST	Brain
Modality	CT	CT	CT
Body part	Lung	Abdomen	Brain
Classification	Binary	Multi-class	Binary
Slices	746	25,221	200
Availability	Public	Public	Internal

PET-CT: The FDG-PET-CT [40,41] dataset, which was part of the MICCAI 2022 AutoPET challenge [42], consists of whole-body PET/CT scans of 900 patients, from which we extract 541,439 CT slices.

LIDC: The Lung Image Database Consortium Image Collection (LIDC-IDRI) [43,44] dataset consists of lung CT volumes of 1010 patients acquired from seven academic centers, initiated by the National Cancer Institute (NCI). We extracted 244,527 CT slices of the dataset.

We used only the CT slices of the datasets; all other information or labels were excluded. The pre-training is performed separately on the two datasets for better generalizability of findings. Implementations are done in PyTorch Lightning [45]. We choose a ResNet50 [32] as our model due to its popularity in medical image analysis [46] and its widespread use as a baseline for comparisons in vision studies [47]. We pre-train the model for 800 epochs on an Nvidia GeForce RTX 3090 GPU and perform a downstream task evaluation every 50 epoch to find the best-performing epoch. All pre-training hyperparameters can be found in Appendix A.

2.3. Downstream evaluation

As Huang et al. [3] shows, classification tasks are commonly used as a benchmark for evaluating self-supervised pre-training. Usually, only a single linear layer is added to the pre-trained encoder to adjust the model to the correct output size, resulting in only the weights of one layer not being pre-trained. In contrast, segmentation tasks require the addition of a large decoder to the pre-trained encoder, such as in a U-Net [48], resulting in a more significant proportion of untrained model weights. This increases the dependency on the dataset of the downstream task. Therefore, we focus on classification downstream tasks to evaluate pre-training performance, although our results are expected to apply to other tasks as well.

We selected three classification tasks on CT slices, ensuring that the images do not overlap with those in the pre-training datasets. These tasks include two public challenges and an internal task as part of a clinical study. For the two publicly available challenges, we perform five downstream runs with the given train/validation/test split of the challenge, to ensure the comparability with other challenge participants. For the internal task, a five-fold stratified cross-validation is performed. For each fold, four parts of the data are used for training and validation (90% training, 10% validation), and the remaining part that has not been used for training and validation is used for testing. This ensures, that the model works on different data splits. The mean and standard deviation of accuracy, AUC score, and F1-sore over the five runs are reported for all three tasks. The tasks include CT scans from different hospitals, scanners, and body parts to prove the generalizability of our findings. The three tasks, summarized in Table 2, are the following:

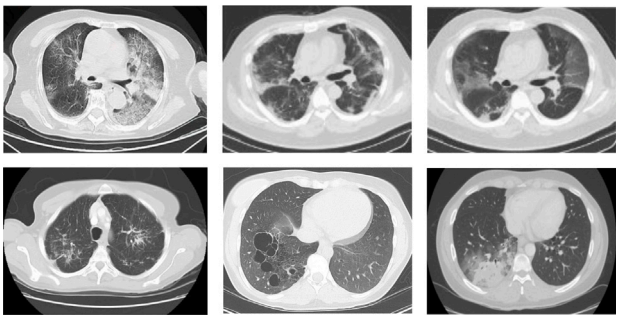

Fig. 3. Example slices of the COVID-19 classification downstream task from Grand Challenge [49]. Upper Row: COVID-19 findings; Lower Row: No COVID-19 findings.

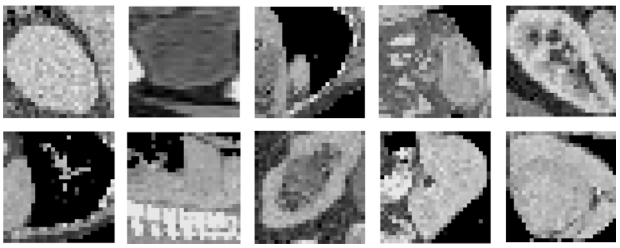

Fig. 4. Example patches for the OrgMNIST multi-class classification downstream task of the OrganSMNIST Challenges [50].

COVID-19: The COVID-19 CT Classification Grand Challenge [49] dataset consists of 349 CT slices (216 patients) and 397 CT slices (171 patients) with and without clinical findings of COVID-19, respectively. The task is to classify between COVID-19 findings and no COVID-19 findings. Fig. 3 shows an example slice for both classes.

OrgMNIST: The OrganSMNIST Challenge from MedMNIST [50] consists of 25,221 image patches of the size 28×28, cropped around organs from abdominal CT scans of 201 patients. The challenge is a multi-class classification of 11 body organs. Fig. 4 shows some example images of cropped patches.

Brain: An internal dataset with CT slices from 100 patients with and 100 patients without brain hemorrhage is used for the third downstream task. All CT examinations were part of the routine clinical practice at the University Hospital of Ulm. Representative slices were selected by Dr. Ch. G. Lisson and Dr. Ca. S. Lisson, two well-trained senior radiologists. This study aims to determine whether brain hemorrhages can be detected automatically on CT scans, which could help physicians in their diagnosis. All patients provided written consent for the use of their anonymized data for research purposes upon signing the treatment contract between the University Hospital of Ulm and the patient. Ethical approval was given by the Ethics Committee of Ulm University under ID 302/17. More details about the collected slices can be found in Appendix D. The task is to classify between brain hemorrhage and no brain hemorrhage, with pre-training being essential due to the small dataset size. Fig. 5 shows some example images of cropped patches.

Our implementations are done in PyTorch Lightning [45] with MONAI [51]. We resize the slices of all tasks to 224×224 in a preprocessing step and train on an Nvidia GeForce RTX 3090 GPU using the Adam optimizer with learning rate 10^{-4} and batch-size 64. We add one linear layer to the pre-trained encoder. Only the linear layer is trained during the first ten epochs before the complete model is fine-tuned.

3. Experiments and results

Our experiments are designed to investigate our hypothesis by answering whether dataset reduction leads to performance gains, which of

Table 3
This table shows the results of the three downstream tasks COVID-19, OrgMNIST, and Brain without using any pre-training. The weights of the model are initialized with PyTorch's standard random initialization (Accuracy can be found in Table E.12 in Appendix E).

Pre-training		Downstream results					
Dataset	Method	COVID-19		OrgMNIST		Brain	
		AUC	F1	AUC	F1	AUC	F1
–	–	0.737 ± 0.033	0.679 ± 0.033	0.971 ± 0.001	0.755 ± 0.003	0.678 ± 0.037	0.447 ± 0.157

Table 4
Evaluation A: This table compares the baseline pre-training method ALL, the current state-of-the-art, which uses all slices of a CT dataset for contrastive pre-training, with the baseline reduction pre-training method EveryN. Pre-training with SwAV is performed on the datasets PET-CT and LIDC with all slices, with 20% of the slices by using every fifth slice, and with 10% of the slices, by using every tenth slice. The different pre-trainings are evaluated on the three downstream tasks COVID-19, OrgMNIST, and Brain (Accuracy can be found in Table E.13 in Appendix E).

Pre-training		Downstream results					
Dataset	Method	COVID-19		OrgMNIST		Brain	
		AUC	F1	AUC	F1	AUC	F1
PET-CT	ALL	0.775 ± 0.009	0.719 ± 0.010	0.968 ± 0.003	0.752 ± 0.003	0.727 ± 0.042	0.534 ± 0.073
	EveryN 20%	0.801 ± 0.006	0.735 ± 0.009	0.972 ± 0.003	0.782 ± 0.003	0.781 ± 0.035	0.665 ± 0.070
	EveryN 10%	**0.810 ± 0.007**	**0.740 ± 0.016**	**0.973 ± 0.002**	**0.793 ± 0.002**	**0.798 ± 0.031**	**0.674 ± 0.074**
LIDC	ALL	0.807 ± 0.006	0.744 ± 0.013	0.972 ± 0.003	0.769 ± 0.003	0.734 ± 0.046	0.609 ± 0.072
	EveryN 20%	0.810 ± 0.004	0.751 ± 0.010	0.977 ± 0.005	0.792 ± 0.003	0.739 ± 0.044	0.610 ± 0.046
	EveryN 10%	**0.812 ± 0.006**	**0.756 ± 0.010**	**0.979 ± 0.002**	**0.800 ± 0.003**	**0.740 ± 0.041**	**0.614 ± 0.046**

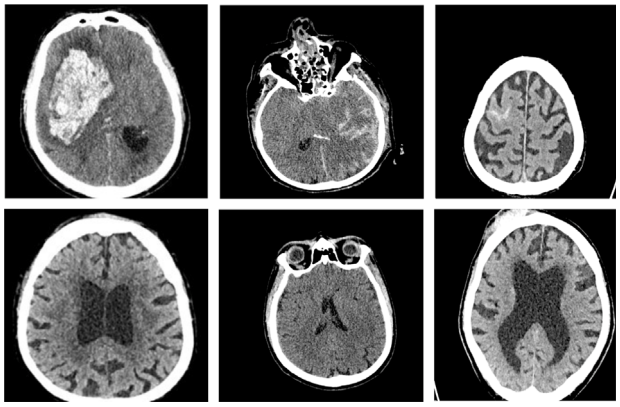

Fig. 5. Example slices of the internal Brain classification downstream task. Upper Row: With brain hemorrhage; Lower Row: Without brain hemorrhage.

our selected reduction methods performs best, what is the optimal similarity threshold, and how much performance gain can be achieved. All experiments are performed separately on the two pre-training datasets PET-CT and LIDC. In total, we conducted 24 different pre-trainings, resulting in over 2000 pre-training hours. The pre-trainings are evaluated on the three downstream tasks COVID-19, OrgMNIST, and Brain. For all results, we report the mean and standard deviation of AUC and F1 scores over five fine-tuning runs on the downstream tasks. Table 3 shows the downstream task results without any pre-training as a reference.

3.1. Evaluation A: Does reduction lead to performance gains?

The first experiment evaluates whether reducing CT datasets for contrastive pre-training leads to performance gains in downstream tasks. To answer this question, we compare the baseline method ALL with the baseline reduction method EveryN. Pre-training is performed on both pre-training datasets with all slices (ALL), with every tenth slice, and with every fifth slice (EveryN). The reduction numbers are chosen randomly. Table 4 shows the downstream task results. Performance gains are achieved in all three downstream tasks by reducing the pre-training dataset to 20%, and 10% with the EveryN method. The performance gains are slightly higher for the 10% reduction.

3.2. Evaluation B: Which reduction method performs best?

Having found that CT data reduction for contrastive pre-training leads to considerable performance gains in downstream tasks, the second experiment investigates which of our selected reduction methods is the best option. We compare the baseline reduction method EveryN with the similarity-based approaches SSIM, MI, DeepNet, and HASH. For an accurate comparison, the reduced datasets should contain the same number of slices for each method. We chose to reduce the pre-training datasets to 10%, since we found a considerable performance gain for reducing the datasets to 10% with the baseline method. The similarity-based approaches reduce the dataset by performing a pairwise comparison of all slices in a volume and removing one slice from pairs with a high similarity, starting with the highest similarity until only 10% of the volume is left. This results in ten pre-training datasets, the reduced PET-CT and LIDC datasets with the approaches EveryN, SSIM, MI, DeepNet, HASH.

Table 5 shows the downstream task results. The HASH method outperforms the baseline reduction method EveryN and all other similarity based approaches. We examined the remaining slices after reduction. Fig. 6 shows the first five remaining slices for an example volume for each of the two pre-training datasets PET-CT and LIDC. To examine how alike the datasets are after the different reduction methods, we compare each dataset with all other datasets and count how many of the remaining slices are equal. The percentage of equal slices across the reduction methods ranges from 9% to 30%, with SSIM and MI having the highest equality and the equality between EveryN and the other approaches being the lowest, between 9% and 11%. We further evaluated the execution time for dataset reduction. The EveryN approach has the shortest execution time with less than one minute, followed by HASH with less than 30 min, both executed on an AMD Ryzen 9 5900X. SSIM, MI, and DeepNet are computed on an Nvidia GeForce RTX 3090 GPU with execution times of 421 h, 312 h, 6 h for the PET-CT dataset and 53 h, 48 h, 2 h for the LIDC dataset.

3.3. Evaluation C: What is the optimal threshold?

When comparing five approaches for reducing CT datasets to 10%, we found that the HASH approach performs best. However, the percentage of a CT dataset volume that leads to the best results can vary from dataset to dataset, depending on the variation of the datasets. Datasets with high variation, for example, due to higher slice thickness, may require less reduction than datasets with lower variation.

Table 5
Evaluation B: This table compares different methods for reducing the pre-training datasets to 10% of the slices. The first method is the baseline reduction method EveryN, which reduces the pre-training dataset by using every tenth slice, followed by the similarity based methods, which perform a pairwise comparison of all slices in a CT volume and remove one slice from pairs with high similarity (Accuracy can be found in Table E.14 in Appendix E).

Pre-Training		Downstream results					
Dataset	Method	COVID-19		OrgMNIST		Brain	
		AUC	F1	AUC	F1	AUC	F1
PET-CT	EveryN	0.810 ± 0.007	0.740 ± 0.016	0.973 ± 0.002	0.793 ± 0.002	0.798 ± 0.031	0.674 ± 0.074
	SSIM	0.811 ± 0.005	0.741 ± 0.010	0.974 ± 0.001	0.794 ± 0.002	0.801 ± 0.309	0.701 ± 0.309
	MI	0.810 ± 0.006	0.748 ± 0.005	0.974 ± 0.001	0.794 ± 0.002	0.819 ± 0.020	0.720 ± 0.024
	DeepNet	0.791 ± 0.008	0.734 ± 0.005	0.973 ± 0.002	0.795 ± 0.002	0.814 ± 0.020	0.721 ± 0.011
	HASH	**0.825 ± 0.004**	**0.755 ± 0.009**	**0.975 ± 0.001**	**0.800 ± 0.002**	**0.821 ± 0.009**	**0.725 ± 0.009**
LIDC	EveryN	0.812 ± 0.006	**0.756 ± 0.010**	0.979 ± 0.002	0.800 ± 0.003	0.740 ± 0.041	0.614 ± 0.046
	SSIM	0.820 ± 0.005	0.751 ± 0.008	0.980 ± 0.001	0.799 ± 0.003	0.813 ± 0.031	0.740 ± 0.027
	MI	0.820 ± 0.007	0.752 ± 0.010	0.980 ± 0.001	0.800 ± 0.002	0.803 ± 0.021	0.741 ± 0.025
	DeepNet	0.800 ± 0.005	0.744 ± 0.011	0.978 ± 0.002	0.793 ± 0.002	0.817 ± 0.028	0.742 ± 0.064
	HASH	**0.825 ± 0.007**	0.754 ± 0.013	**0.981 ± 0.002**	**0.802 ± 0.002**	**0.829 ± 0.020**	**0.744 ± 0.021**

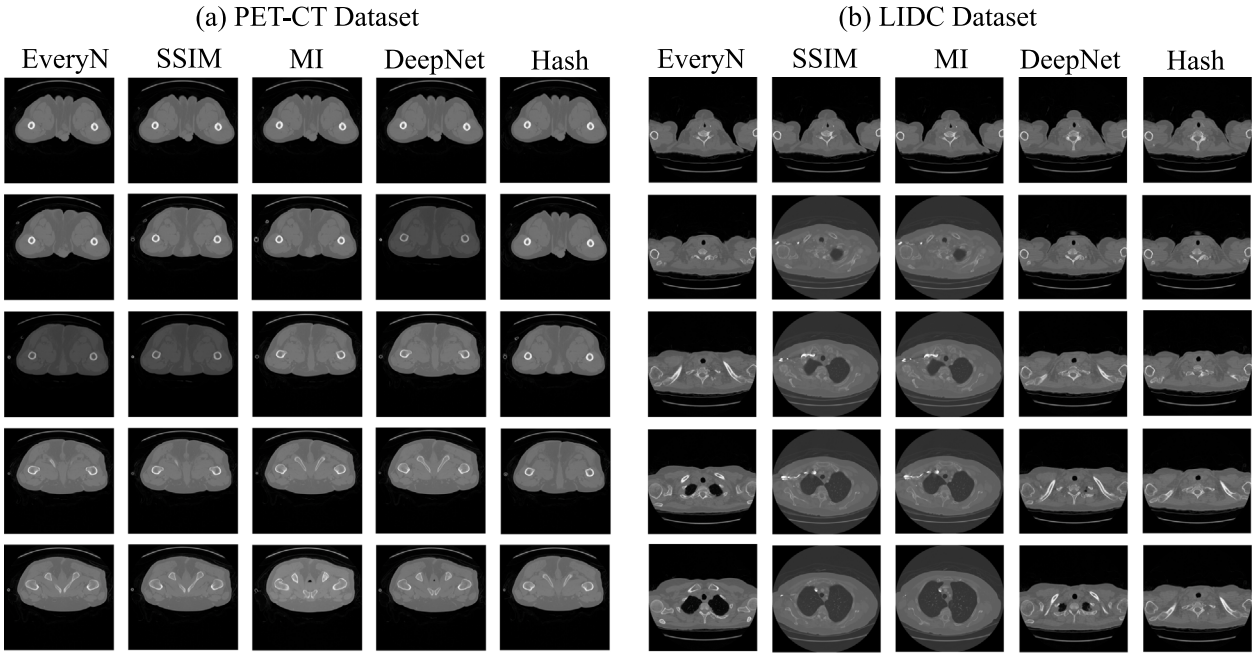

Fig. 6. This figure compares the selected images for pre-training with the PET-CT dataset (a) and the LIDC dataset (b) after the reduction using the EveryN, SSIM, MI, DeepNet, HASH methods. Five consecutive slices from an example volume are shown, starting from the first slice of the volume in the top row to the fifth remaining slice in the bottom row.

Therefore, in the third experiment, we attempt to find the optimal threshold for the degree of similarity between the slices that leads to the highest results in downstream tasks. We use the best-performing slice selection method HASH and test different similarity thresholds. The similarity score for comparing two slices using the HASH approach is the Hamming distance, which ranges from 0 (most similar) to 64 (most dissimilar). Reducing the dataset to a chosen similarity threshold of the Hamming distance leads to a dataset where, within each volume, no pairs of slices are more similar than the threshold. We compare three thresholds: Hamming distances three, six, and twelve. Number of slices: PET-CT: 120,750 (22.3%); 48,718 (9%); 19,497 (3.6%) and LIDC: 44,416 (18.2%); 22,672 (9.3%); 9828 (4%). Table 6 compares the downstream results of the different similarity thresholds. The best performance on all three downstream tasks is achieved with threshold hamming distance six. Higher and lower similarity thresholds, resulting in larger and smaller remaining portions of the pre-training datasets, lead to slightly degraded results.

3.4. Evaluation D: How much performance gain can be achieved?

Through several experiments, we found that the HASH approach with a Hamming distance threshold of six (HASH-6) performs best. In the last step, we compare the downstream task results of the best-performing approach with the baseline method ALL, the current state of the art, using all slices of the dataset for pre-training. Fig. 7 shows the pre-training duration and the AUC scores of the downstream task results. On the PET-CT pre-training dataset, we achieve performance gains in AUC values from 0.775 to 0.830, 0.968 to 0.978, and 0.727 to 0.831 for the COVID-19, OrgMNIST, and Brain downstream tasks, respectively. Performance gains from 0.807 to 0.823, 0.972 to 0.982, and 0.734 to 0.840 are achieved on the LIDC pre-training dataset. The pre-training time is reduced from 538 h to 62 h and from 280 h to 27 h on the PET-CT and LIDC datasets, respectively, with a slice selection time of less than 30 min. For a better interpretation of our results, in the following we further analyze the difference between pre-training

Table 6
Evaluation C: This table compares different similarity thresholds of the best performing reduction method HASH, in order to obtain the optimal degree of similarity between the slices for contrastive pre-training (Accuracy can be found in Table E.15 in Appendix E).

Pre-training		Downstream results					
Dataset	Method	COVID-19		OrgMNIST		Brain	
		AUC	F1	AUC	F1	AUC	F1
PET-CT	HASH - 3	0.821 ± 0.004	0.764 ± 0.004	0.976 ± 0.001	0.797 ± 0.003	0.799 ± 0.012	0.687 ± 0.018
	HASH - 6	**0.830 ± 0.006**	**0.777 ± 0.016**	**0.978 ± 0.001**	**0.800 ± 0.003**	**0.831 ± 0.021**	**0.765 ± 0.027**
	HASH - 12	0.822 ± 0.003	0.755 ± 0.006	0.976 ± 0.001	0.796 ± 0.002	0.770 ± 0.030	0.697 ± 0.039
LIDC	HASH - 3	0.813 ± 0.004	0.730 ± 0.009	0.981 ± 0.001	0.800 ± 0.002	0.790 ± 0.008	0.723 ± 0.041
	HASH - 6	**0.823 ± 0.005**	**0.768 ± 0.008**	**0.982 ± 0.001**	**0.802 ± 0.002**	**0.840 ± 0.016**	**0.800 ± 0.033**
	HASH - 12	0.811 ± 0.003	0.737 ± 0.013	0.980 ± 0.001	0.798 ± 0.002	0.798 ± 0.026	0.677 ± 0.026

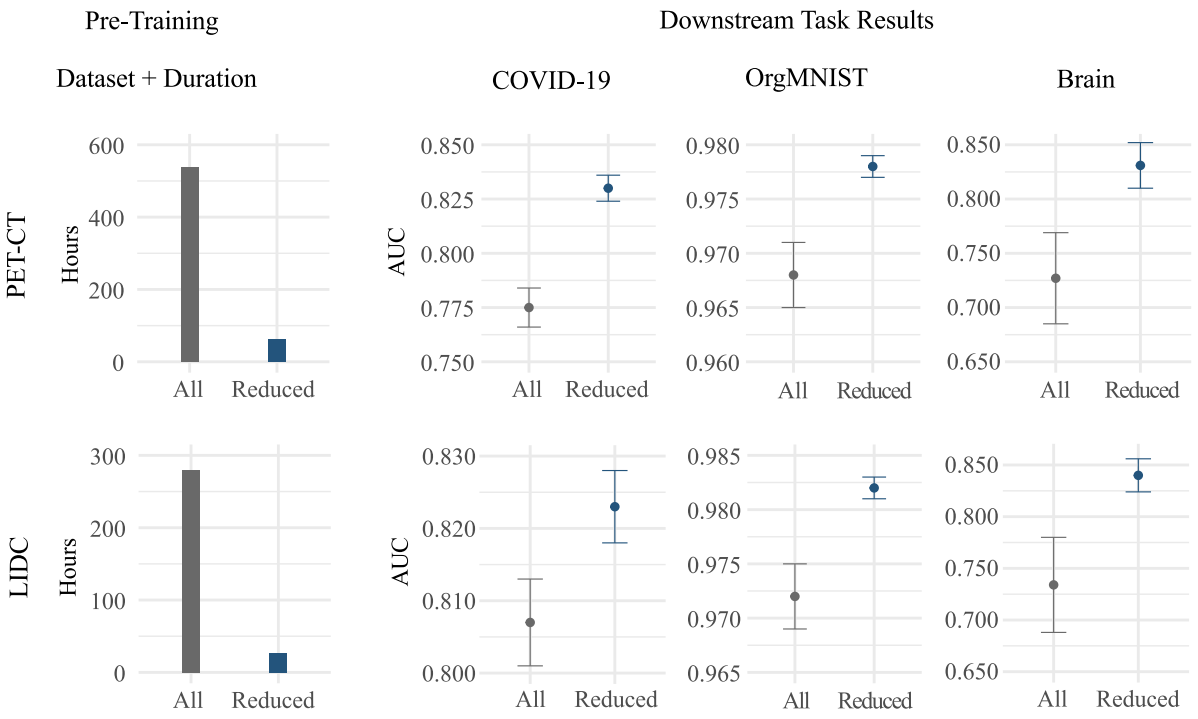

Fig. 7. Here, we show on two pre-training CT datasets (PET-CT, LIDC) and three downstream CT classification tasks to evaluate the pre-trainings (COVID-19, OrgMNIST, Brain) that our proposed hashing based dataset reduction method leads to shorter pre-training duration and improves downstream task performances, compared to the current state-of-the-art approach, which uses pre-training with all slices of the dataset.

with ALL data and pre-training with the best performing method HASH-6 by calculating the Centered Kernel Alignment CKA [52], visualizing the t-Distributed Stochastic Neighbor Embedding [53] and visualizing the model's attentions with Grad-Cam [54].

Centered Kernel Alignment CKA [52] measures the similarity of the representations from two models at the different layers of the models. In Fig. 8 we show the CKA between pre-training with ALL data and pre-training with HASH-6 reduced data. The calculation was done directly after pre-training, before fine-tuning the model for a specific downstream task. For both pre-training datasets PET-CT and LIDC, the plots show a relatively high similarity for early layers of the model, but a relatively low similarity for later layers. Thus, for both pre-training datasets, the reduction of the dataset mainly affects the later layers of the model. In Fig. 9 we show the CKA similarity between the model after pre-training and the model after fine-tuning. As can be seen in the plots, fine-tuning mainly affects the later layers of the model while the earlier layers keep similar representations to the stage after pre-training. Now we compare in Fig. 9 the CKA plots of pre-training with ALL data to the CKA plots of pre-training with HASH-6 reduced data. This comparison shows that with the HASH-6 reduced data, the representations of more layers have a high similarity between pre-training and fine-tuning, compared to pre-training with ALL data. So the Centered Kernel Alignment CKA between pre-training and fine-tuning is higher when the HASH-6 reduced dataset is used for pre-training.

T-Distributed Stochastic Neighbor Embedding (t-SNE) [53] is a technique for visualizing high-dimensional data by reducing the data to lower-dimensional spaces. We use this technique to visualize the fully connected classifier output at the end of our model after the convolutional layers. We propagate the images of the test datasets of the three downstream tasks COVID-19, OrgMNIST, and Brain through the model up to the fully connected layer and plot the values with t-SNE to visualize the distributions of the predictions and see how well the model can discriminate between the classes. This visualization was done once directly after pre-training, before fine-tuning the model, and once after fine-tuning. Fig. 10 shows the plots for the PET-CT pre-training dataset and Fig. 11 for the LIDC pre-training dataset. We again compare pre-training with ALL data and pre-training with HASH-6 reduced data. After pre-training, there is no clear separation of the different classes, neither for pre-training with ALL data nor for pre-training with the HASH-6 reduced data. After fine-tuning, especially for the COVID-19 and the OrgMNIST task, a clearer separation of the classes is visible when using HASH-6 reduced pre-training compared to ALL pre-training. As a quantitative measure, we calculated, for fine-tuning, the Pearson Correlation Coefficient (PCC) between the t-SNE values of the model and the target classes. We get between 2% and

16% higher PCC values after fine-tuning when pre-training with the HASH-6 reduced data compared to pre-training with ALL data. Thus, the distribution of predictions indicates that the model pre-trained with HASH-6 reduced data can better discriminate between classes after fine-tuning compared to ALL data pre-training.

We visualize the attention region of the model with the gradient-weighted class activation mapping Grad-Cam [54]. We generated the attention heatmaps on the test datasets of the three downstream tasks for both pre-training datasets. The attention heatmaps were generated once directly after pre-training, before fine-tuning the model, and once after fine-tuning. Fig. 12 shows three example images for each downstream task. The attention regions were qualitatively analyzed by two well-trained radiologists. As can be seen in the example images in Fig. 12 for the COVID-19 and Brain downstream task, the main attention after pre-training with ALL data is often not even in the lung or brain region and is far away from the model's final attention after fine-tuning. In contrast, when pre-training with the HASH-6 reduced, the model's main attention is already after pre-training mostly much closer to the actual target and the final attention. For example, for the COVID-19 task, in row 4, column 2, the Grad-cam image for ALL data after pre-training shows an attention that lies outside the body region and is far away from the final attention after fine-tuning (image row 4, column 3). In contrast, in the Grad-cam image for the HASH-6 reduced data after pre-training (row 4, column 4), the attention is clearly in the lung region and already close to the final attention after fine-tuning (row 4, column 5). And the final attention for the HASH-6 reduced data covers the area that the well-trained radiologists would look at much better. For the OrgMNIST task, for example in row 6, column 6, the attention after pre-training with ALL data is somewhere completely different from the final attention in column 7. Meanwhile, with the HASH-6 reduced data, the attention after pre-training is already relatively close to the final attention after fine-tuning (row 6, column 8 and 9). For the Brain task in the last row, column 2, the attention after pre-training with ALL data is widely distributed over the image and not close to the final attention after fine-tuning (last row, column 3). And even the final attention does not cover the bleeding perfectly. In contrast, after pre-training with HASH-6 reduced data (last row, column 4), the attention is in the brain area and already covers the bleeding almost perfectly. After fine-tuning, the attention becomes only slightly more precise (last row, column 5). The same pattern can be seen for most of the Grad-cam images on both pre-training datasets and all three downstream tasks. For a quantitative analysis, we computed the Intersection over Union (IoU) between the heatmap after pre-training and the heatmap after fine-tuning, to see how close the model's attention after pre-training is already to the model's final attention after fine-tuning. Again, we compared pre-training with ALL data and pre-training with HASH-6 reduced data. For all three downstream tasks, on both pre-training datasets, we get between 7% and 9% higher IoU values with the reduced pre-training dataset, as with all pre-training data. Thus, with our HASH-6 reduction approach, the model's attention after pre-training is already closer to the final attention after fine-tuning.

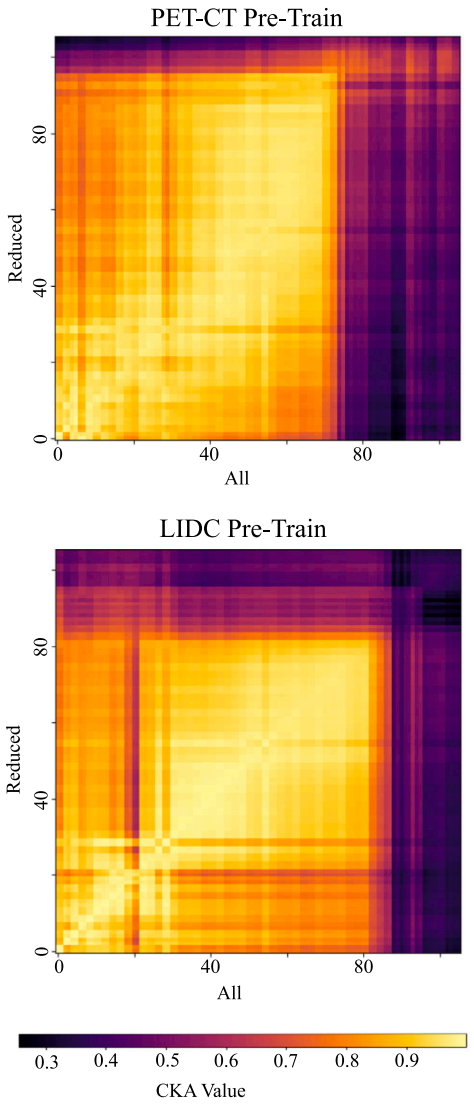

Fig. 8. Here we visualize the Centered Kernel Alignment CKA [52] between the model pre-trained with ALL data and the model pre-trained with HASH-6 reduced data, for both pre-training datasets PET-CT and LIDC. The calculations are done after pre-training, before fine-tuning the model. On the x- and y-axis are the layers of the models starting from zero as the first layer up to the last layer of the model. At the bottom is a scale of the CKA value. High values of the CKA mean that the representations of the two models are similar. The calculations are done by CKA.pytorch [55].

3.5. Evaluation E: Other self-supervised pre-training approaches

To show the generalizability of our results, we tested further self-supervised pre-training approaches with the best-performing reduction approach HASH. For this evaluation, we only use the LIDC dataset which is smaller and thus needs less pre-training time and has less computational effort. According to Huang et al.'s. [3] study, another popular contrastive learning approach on convolutional neural networks from natural image processing that is widely utilized for medical pre-training is MoCo (Momentum Contrast) [36]. MoCo has been slightly updated in MoCo version 2 [58] and has also been successfully applied to pre-training on CT slices [7,23]. A completely different approach for self-supervised pre-training is masked image modeling, which has gained much popularity in the imaging field in the last few years [59]. In a recent study, Tian et al. [39] show on ImageNet [33] data that masked autoencoders [59], that have been mainly used for self-supervised pre-training of transformers [3], can be adapted to convolutional models. Masked autoencoders divide the images into patches, mask part of the patches, and train the model to reconstruct the original images. Due to the moderate success of this method for convolutional models so far [3], they adapted it by using sparse convolutions instead of normal convolutions for the pre-training, where they achieved comparable results to contrastive learning. In [23], their method, called SparK, was applied to CT slices and shows similar performance to SwAV and MoCo for self-supervised pre-training and is particularly robust for small downstream datasets.

We compare the best-performing reduction method HASH with threshold six against the baseline method ALL on the contrastive learning approach MoCo Version 2 and the masked autoencoder approach SparK. To prove that our results are generalizable to other contrastive

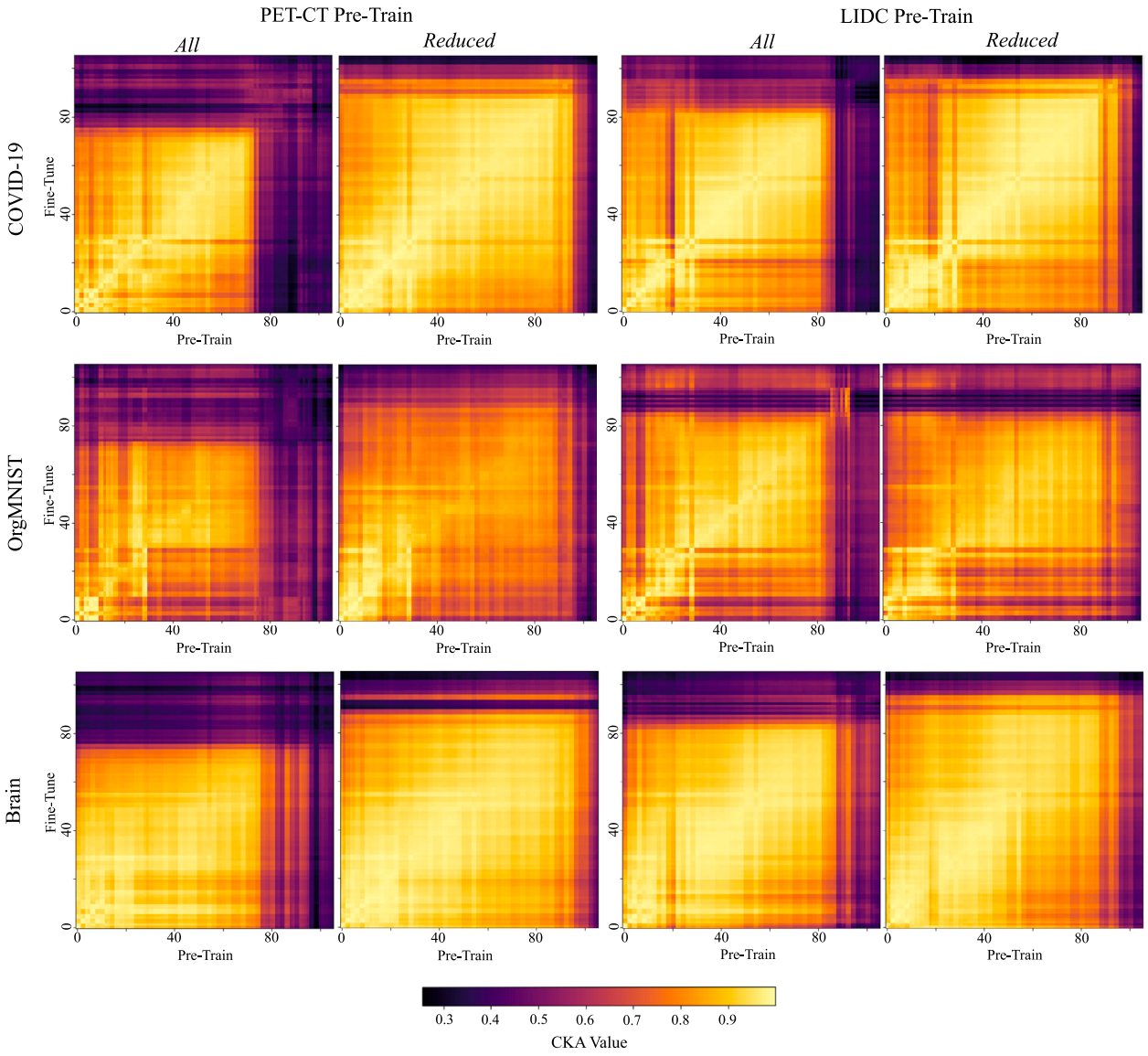

Fig. 9. Here we visualize the Centered Kernel Alignment CKA [52] between the model after pre-training and the model after fine-tuning. We show the plots for fine-tuning on the three downstream tasks COVID-19, OrgMNIST, and Brain and the two pre-training datasets PET-CT and LIDC, each once for pre-training with ALL data and once for pre-training with the HASH-6 reduced data. At the bottom is a scale of the CKA value. High values of the CKA mean that the representations of the two models are similar. The calculations are done by CKA.pytorch [55].

pre-training methods, we expect to see performance gains with our slice reduction method for MoCo, since, as in SwAV, less similar images should improve the model's ability to distinguish in latent space. On the other hand, since there is no distinguishing involved in masked autoencoder pre-training approaches, similar images should not be a problem there. Thus, we expect no performance gain or slightly reduced performance for the masked autoencoder method SparK, since less similar images should not bring any advantage and the model just has less training data. Detailed explanations of MoCo and Spark can be found in Appendices B and C. Table 7 shows the downstream task results. The contrastive learning method MoCo performs better with the reduced dataset, analogous to the contrastive learning method SwAV discussed earlier. For the masked autoencoder method SparK, we do not achieve any improvements with the reduction, using all slices achieves superior results.

4. Discussion

Self-supervised pre-training of deep learning models with contrastive learning on large unannotated datasets is a common and successful approach in medical imaging to cope with small annotated datasets [3]. The most popular contrastive learning methods were initially developed for natural image processing and transferred to the medical domain [3]. Many methods can be directly applied to the medical domain without adaptation; however, not all methods show the same behavior because medical images have different structures and color schemes [60]. In this work, we investigate the composition of the pre-training datasets for contrastive learning on CT slices. We perform our investigations on two large pre-training datasets separately to ensure generalizability and evaluate the pre-trainings on three classification downstream tasks, the benchmark task for evaluating self-supervised pre-training [3]. Table 3 shows the results without pre-training and Table 4, row PET-CT ALL and row LIDC ALL, show the results when contrastive pre-training on all slices of the pre-training dataset is applied, the current state-of-the-art [5,7]. Contrastive pre-training improves the downstream results for the COVID-19 and the Brain tasks with both pre-training datasets. However, for the OrgMNIST task, we only achieve performance gains when pre-training with the

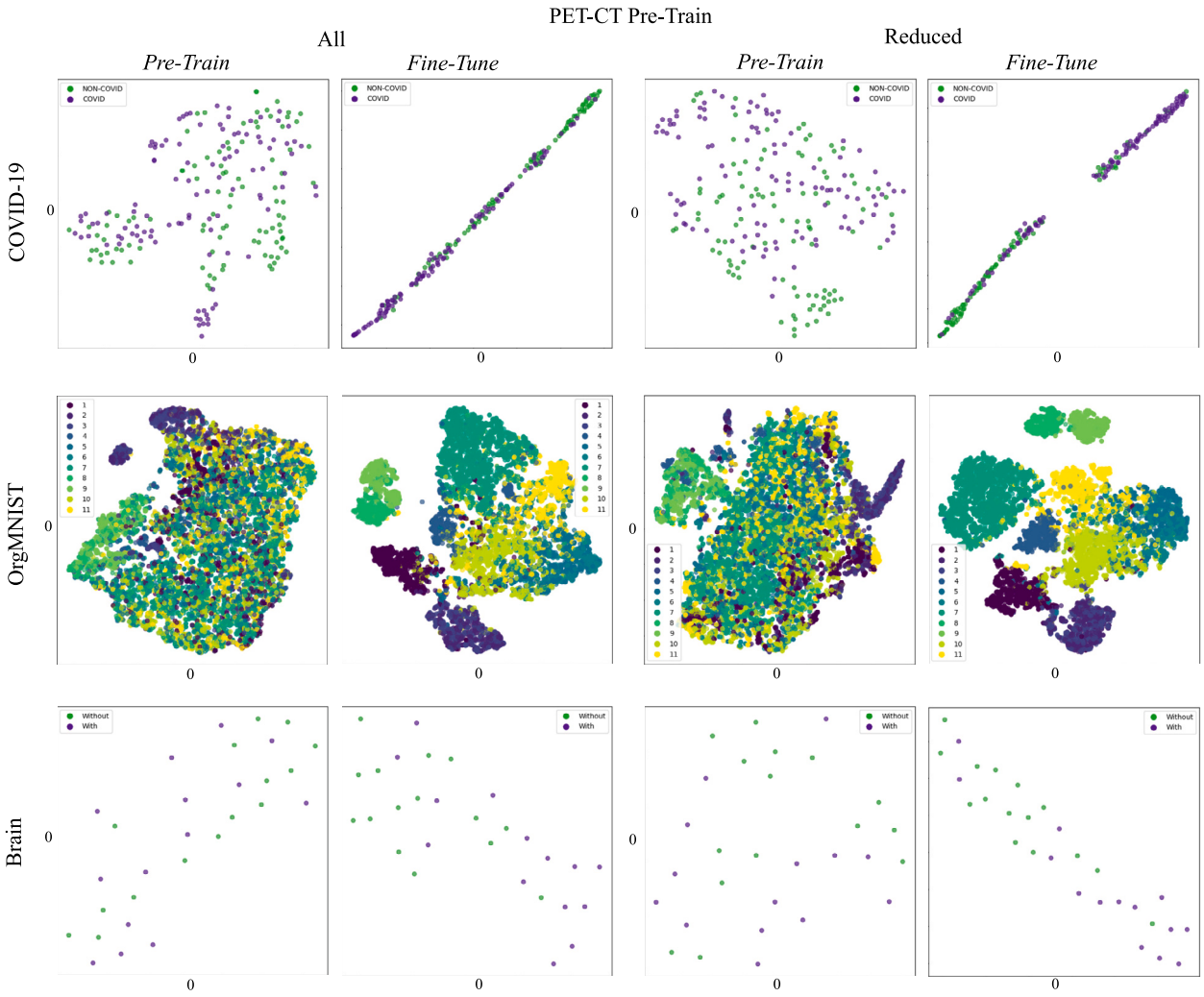

Fig. 10. Here we visualize the T-Distributed Stochastic Neighbor Embedding (t-SNE) [53] at the fully connected classifier output after the convolutional layers of our model when the test dataset images of the downstream tasks COVID-19, OrgMNIST, Brain are propagated through the model. The plots were generated once after pre-training and once after fine-tuning. On the left, the model was pre-trained with ALL data, and on the right with HASH-6 reduced data. The colors of the dots indicate the target classes of the images. COVID-19 and Brain are binary and OrgMNIST is multi-class classification tasks. The calculations are done with scikit-learn [56].

Table 7
Evaluation E: This table shows the downstream task results for pre-training with the contrastive learning (CL) approaches SwAV and MoCo Version 2 and the masked autoencoder (MAE) approach SparK. For all three approaches, we compare pre-training with all data (All) to pre-training with the reduced dataset using the hash reduction method and threshold 6 (Reduced) on the LIDC dataset. (Accuracy can be found in Table E.16 in Appendix E).

Pre-training		Downstream results					
Approach	Data	COVID-19		OrgMNIST		Brain	
		AUC	F1	AUC	F1	AUC	F1
SwAV (CL)	All	0.807 ± 0.006	0.744 ± 0.013	0.972 ± 0.003	0.769 ± 0.003	0.734 ± 0.046	0.609 ± 0.072
	Reduced	**0.823 ± 0.005**	**0.768 ± 0.008**	**0.982 ± 0.001**	**0.802 ± 0.002**	**0.840 ± 0.016**	**0.800 ± 0.033**
MoCoV2 (CL)	All	0.824 ± 0.005	0.780 ± 0.009	0.981 ± 0.001	0.817 ± 0.001	0.825 ± 0.010	0.770 ± 0.064
	Reduced	**0.830 ± 0.005**	**0.781 ± 0.005**	**0.982 ± 0.003**	**0.820 ± 0.004**	**0.897 ± 0.015**	**0.791 ± 0.032**
SparK (MAE)	All	**0.828 ± 0.006**	**0.776 ± 0.009**	**0.981 ± 0.001**	**0.808 ± 0.003**	**0.919 ± 0.015**	**0.812 ± 0.080**
	Reduced	0.810 ± 0.006	0.761 ± 0.020	0.978 ± 0.001	0.782 ± 0.002	0.882 ± 0.024	0.809 ± 0.051

LIDC dataset. Contrastive pre-training on all slices of the PET-CT dataset slightly decreases the results of the OrgMNIST task by 0.003 AUC score. This shows that pre-training with contrastive learning on CT scans does not improve downstream performance in all cases, which is also confirmed in Huang et al.'s. [3] study.

In contrastive learning, the model is trained to distinguish between latent space representations of positive pairs coming from two augmented views of the same original image and latent space representations of negative pairs coming from two different original images. We hypothesized that using each slice of a CT volume for contrastive pre-training might lead to a model that is unable to discriminate between positive and negative pairs since the similarity between two augmented versions of a slice might be lower than the similarity between two different slices. In the first experiment, we reduced the pre-training datasets by using only every *n*th slice of a volume. The results listed in Table 4 support our hypotheses, as performance improves on the downstream tasks. Using only every *n*th slice of a volume increases the variation between slices in the pre-training dataset, potentially allowing

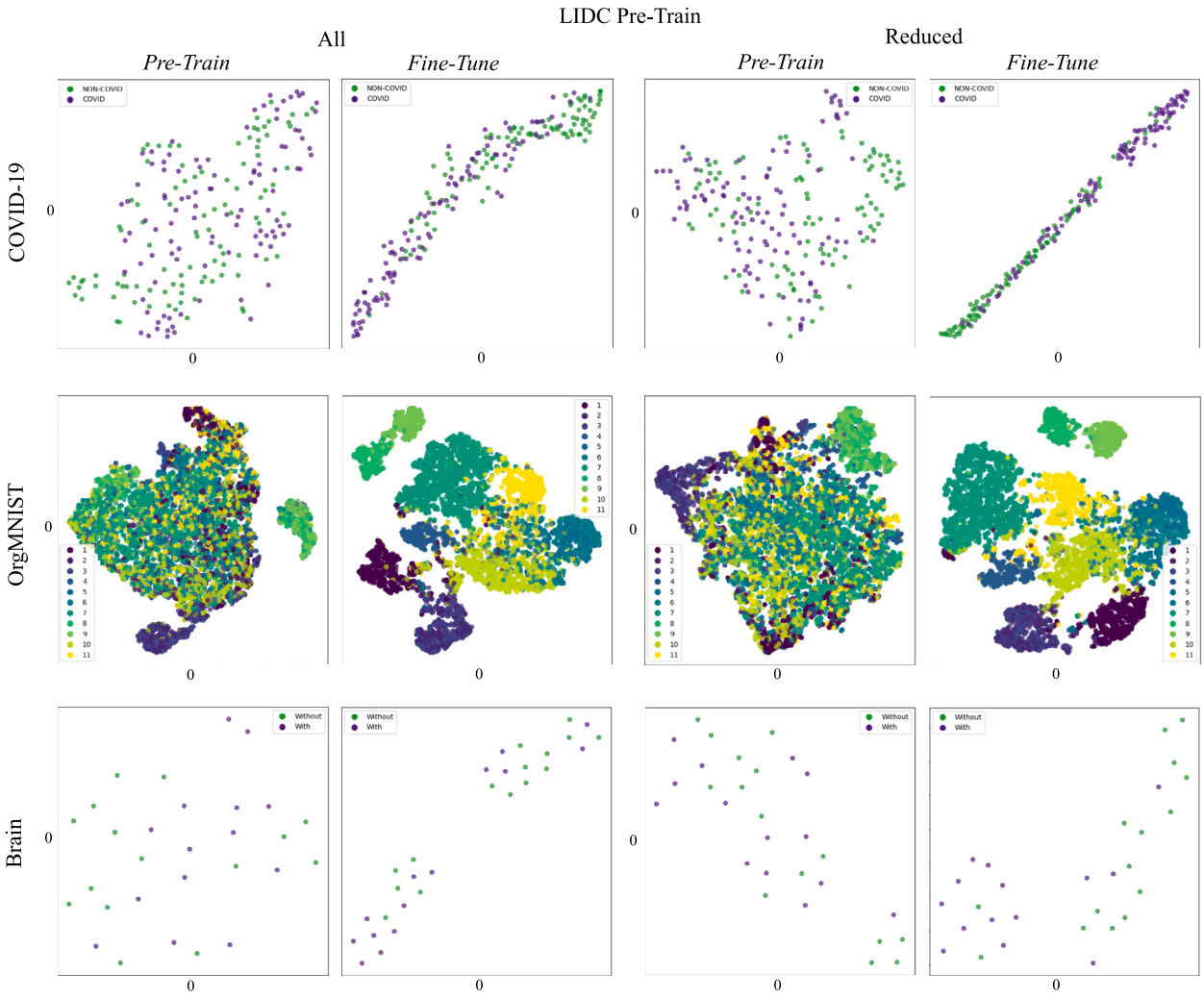

Fig. 11. Here we visualize the T-Distributed Stochastic Neighbor Embedding (t-SNE) [53] at the fully connected classifier output after the convolutional layers of our model when the test dataset images of the downstream tasks COVID-19, OrgMNIST, Brain are propagated through the model. The plots were generated once after pre-training and once after fine-tuning. On the left, the model was pre-trained with ALL data, and on the right with HASH-6 reduced data. The colors of the dots indicate the target classes of the images. COVID-19 and Brain are binary and OrgMNIST is multi-class classification tasks. The calculations are done with scikit-learn [56].

the model to better distinguish between positive and negative pairs. After the reduction, contrastive pre-training with SwAV outperforms no pre-training for all tasks and pre-training datasets, including the OrgMNIST task and pre-training on the PET-CT dataset, where we had performance losses when pre-training with all slices.

In a second experiment, we aimed to evaluate whether there are dataset reduction methods that are more suitable than the baseline reduction. Further, we aimed to find the optimal threshold for the degree of similarity between the slices that leads to the highest results in downstream tasks. As shown in Table 5, of all evaluated methods, the HASH method performs best on the downstream tasks, and as shown in Table 6, a similarity threshold of six for the HASH approach seems to ensure the best degree of similarity. These experiments lead to the assumption that the HASH reduction approach with threshold six ensures the best compromise between a high variation and a sufficiently large number of samples in the pre-training dataset. Furthermore, with an execution time of less than 30 min, the HASH dataset reduction is computed faster than all other similarly based reduction methods evaluated.

In a further experiment, we tested the HASH reduction method on two other self-supervised pre-training approaches. We expected that our results would generalize to other contrastive learning approaches due to the identical basic concept, but that reduction would not improve the results for other self-supervised pre-training methods. As shown in Table 7, pre-training with the contrastive learning approach MoCo is improved with the HASH reduction method. This proves the generalizability of our results to other contrastive pre-training methods, that are trained by distinguishing between latent space representations of augmented views and original images and thus have the problem of too similar images. As also shown in Table 7 the masked autoencoder pre-training method SparK performs best with all slices. Since masked autoencoders are trained to reconstruct masked patches of images where no distinguishing is involved, reducing the dataset does not bring any advantage and the model just has less training data. The results in Table 7 support our hypothesis that selective CT data reduction is beneficial for contrastive pre-training due to the distinguishing challenge, but pre-training methods that do not rely on distinguishing in latent space do not benefit from dataset reduction.

A major benefit of pre-training dataset reduction for contrastive learning is that we significantly reduce the pre-training time. With less time and thus less energy cost, better pre-training results on CT image classification downstream tasks can be achieved, as summarized in Fig. 7.

5. Limitations

As our work shows the great potential of CT dataset reduction for contrastive pre-training, it would be interesting to further investigate

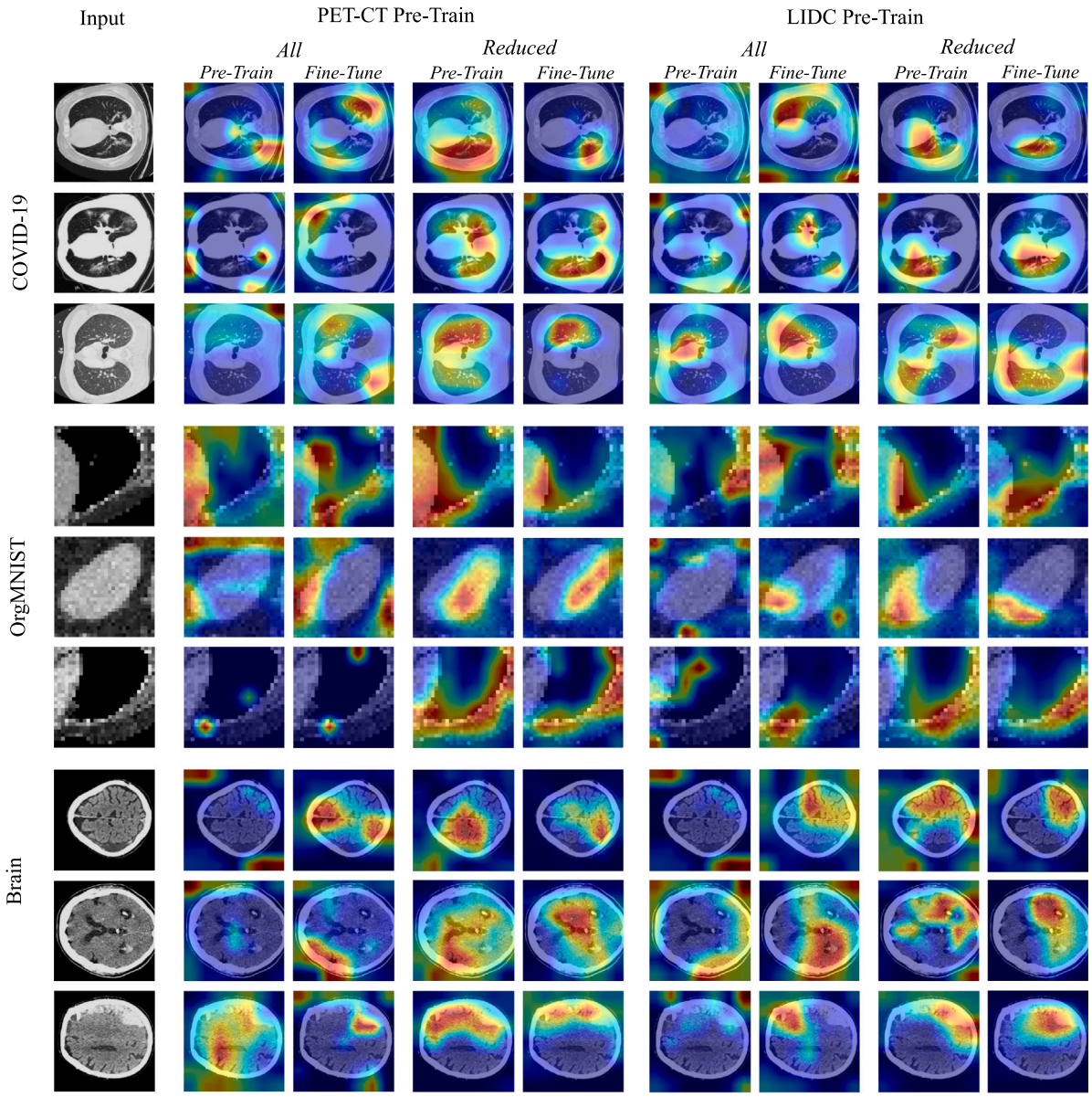

Fig. 12. Here we visualize the attention region of the network with Grad-Cam [54] for three example images of the COVID-19, OrgMNIST and Brain downstream tasks' test datasets. The first row shows the input image, the next four rows the attention heatmaps with the PET-CT pre-training dataset, and the last four rows the attention maps with the LIDC pre-training dataset. For both pre-training datasets, we visualize the heatmaps for pre-training with ALL data and for the best reduction approach HASH-6. We compare the attention heatmap after the self-supervised pre-training before fine-tuning the model with the attention heatmap after the fine-tuning for the specific downstream task. The plots are generated by pytorch-grad-cam [57].

these findings in future research. A limitation of our work is that we only chose well-established, computationally fast dataset reduction methods that are based on similarity calculations. For future work, it would be interesting to see if there are other dataset reduction techniques that could lead to even better results. For example, core-set selection such as SVP (Selection via Proxy) [61] or CRAIG (Coresets for Accelerating Incremental Gradient descent) [62] might be a promising idea. However, these methods require significantly more time and computational effort for the dataset reduction.

Another limitation of our work is that we did not investigate different augmentation strategies for the contrastive pre-training. The ability to distinguish between latent space representations coming from two augmented views of the same original image and latent space representations coming from two different original images also depends on the type and amount of augmentations used. We took exactly the augmentations of the original contrastive learning publications SwAV [38] and MoCo [36], since they have done an intensive evaluation on which augmentation techniques are best suited for their pre-training method. However, for future work, it would be interesting to investigate different types and amounts of augmentation to see if less data reduction is needed and if even better results can be achieved. Furthermore, it would be interesting to better understand why exactly the HASH method leads to the best results and if this is dependent on the used augmentation methods.

A further limitation is that our experiments were performed with only one deep learning model. Analogous to the original publications of the two self-supervised pre-training methods SwAV [38] and MoCo [36], we chose the ResNet50 [32] as our model, due to its widespread use as a baseline for comparisons in vision studies [47] that were later successfully transferred to other models and its popularity in medical image analysis [46]. For further research, it would be interesting to apply our findings to other deep learning models as well. Furthermore, applying our results to other modalities of volumetric

images consisting of consecutive slices like MRI or PET, would be possible for future research projects.

In addition to the very important task of CT image classification with a lot of ongoing research [3,63] CT image segmentation is another popular task in medical imaging. We have tested our self-supervised pre-trained classification model on several segmentation tasks by adding a U-Net [48] decoder to the pre-trained ResNet50. However, we did not achieve any performance gains with our pre-training, neither on ALL data nor on HASH-6 reduced data. Thus, a clear limitation of our work is that our results cannot be directly applied to downstream segmentation tasks. As shown in [64], pure contrastive pre-training with methods from the RGB imaging domain only on the encoder does not lead to significant performance gains for segmentation downstream tasks. Instead, other specific pre-training methods for segmentation can lead to improved performance. For future work, it would be interesting to further investigate our findings on such segmentation-optimized contrastive learning methods.

Another idea would be to combine our findings with the work of Joshua, et al. [65]. With our proposed HASH-6 approach, slices that are too similar to be distinguished by the model when using contrastive pre-training can be identified and removed from the dataset before the training. Joshua, et al. developed a method to improve the model's pre-training, by targeting the samples that turn out to be difficult for the model to distinguish during the training. This is done by analyzing in the latent space which images the model places close together, but should actually be far apart, as they are negative samples that should be pushed apart in the latent space. These samples, called hard negative samples, are pushed apart by a special loss function. So one possibility would be to first apply our approach to filter out images that are too similar to be meaningfully distinguished. Then, hard negative mining could be applied to further improve the pre-training by targeting cases that turn out to be still difficult for the model during the training. Hard negative mining by Joshua, et al. has originally been developed on RGB images. We see strong potential in adapting this method to CT slices and exploring the combination of the two approaches.

By performing pre-training and fine-tuning with a 2D model on the slices of CT scans instead of using a 3D model on volumes, we ensure low computational costs for inference on downstream tasks so our findings can be applied globally without requiring powerful GPUs. However, training 3D models on volumes and training 2D models on slices are both widely used approaches for deep learning on CT scans, with several recent publications demonstrating excellent results for clinically relevant CT imaging tasks on both 3D [11,12], and 2D [13–16] models. Both approaches have their advantages. After evaluating the pre-training dataset composition for contrastive pre-training of 2D models on CT slices, for future work, an evaluation of the properties of pre-training datasets with entire volumes would be interesting.

6. Conclusion

In our work, we investigate how to exploit the characteristics of CT datasets to improve contrastive pre-training. We hypothesized that using all slices in each CT volume of a pre-training dataset may lead to performance degradation due to the low variation in the data. The experiments, with over 2000 pre-training hours, support our hypothesis. In conclusion, we propose to reduce pre-training datasets using the HASH method and a threshold of six. This approach leads to considerable performance gains in classification downstream tasks in all our experiments and outperforms the other evaluated dataset reduction methods. The time to reduce the datasets using the HASH approach is negligibly short, with execution times of less than half an hour, while the pre-training duration is substantially reduced. Research on CT data with contrastive learning in the future can incorporate our findings to improve their performance on classification tasks and speed up learning by reducing their pre-training dataset with our proposed method.

CRediT authorship contribution statement

Daniel Wolf: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Tristan Payer:** Writing – review & editing, Validation, Software, Methodology. **Catharina Silvia Lisson:** Writing – review & editing, Validation, Data curation. **Christoph Gerhard Lisson:** Writing – review & editing, Validation, Data curation. **Meinrad Beer:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Michael Götz:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition. **Timo Ropinski:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Pre-Training: The LIDC-IDRI [43,44] dataset is available for public use under the license CC BY 3.0. The FDG-PET-CT [40,41] dataset is available for public use listed under request at TCIA [66].

Downstream: The COVID-19 CT Classification Grand Challenge [49] dataset is available at https://covid-ct.grand-challenge.org/ ; The OrganSMNIST dataset from MedMNIST [50] is available for public use under the license CC BY 4.0; The internal Brain dataset cannot be made publicly available due to strict data security restrictions.

Code: https://github.com/Wolfda95/Less_is_More.

Acknowledgments

The authors acknowledge the National Cancer Institute and the Foundation for the National Institutes of Health, and their critical role in the creation of the free publicly available LIDC/IDRI Database used in this study.

Funding

This work is funded by "NUM 2.0" (FKZ: 01KX2121) as part of the Racoon Project.

Ethics declarations

For the internal Brain task, ethical approval was granted by the Ethics Committee of Ulm University under ID 302/17. The procedure was in accordance with the ethical standards of the World Medical Association (Declaration of Helsinki).

Appendix A. Contrastive learning with SwAV

SwAV [38] starts with a large dataset of images $\underline{I} = \{I_1, I_2, I_3, \ldots\}$. For all images in one mine-batch with batch-size Bs, two random transformations are performed in order to obtain two randomly different images from each original image: $\underline{A} = \{A_1, A_2, A_3, \ldots, A_{Bs}\}$ and $\underline{B} = \{B_1, B_2, B_3, \ldots, B_{Bs}\}$. The transformed images are computed by a deep learning, which can be any convolutional encoder followed by an MLP, to a latent space representation: $\underline{AQ} = \{AQ_1, AQ_2, AQ_3, \ldots, AQ_{Bs}\}$ and $\underline{BQ} = \{BQ_1, BQ_2, BQ_3, \ldots, BQ_{Bs}\}$. The latent space representations are further computed by feature clustering with cluster prototypes $\underline{C} = \{C_1, C_2, C_3, \ldots, C_K\}$, which leads to the cluster codes

$$\underline{AQC} = \{\frac{1}{\tau} \cdot \underline{AQ}^T \cdot \underline{C_1}, \ldots, \frac{1}{\tau} \cdot \underline{AQ}^T \cdot \underline{C_K}\} \tag{A.1}$$

and

$$\underline{BQC} = \{\frac{1}{\tau} \cdot \underline{BQ}^T \cdot \underline{C_1}, \ldots, \frac{1}{\tau} \cdot \underline{BQ}^T \cdot \underline{C_K}\} \tag{A.2}$$

Table A.8
Hyperparameters for pre-training with SwAV.

Parameters	Values
Input size	512
Numb. of crops	2; 6
Size of crops	224; 96
Min scale crops	0.90; 0.10
Max scale crops	1.0; 0.33
Optimizer	Lars
Batch size	128
Learning rate	0.15
Weight decay	1e−6
Max epochs	800
Sinkhorn iterations	3
Number prototypes	500
Freeze prototypes	313
Size MLP	2048
Output dimension	128

with the temperature value τ and the number of prototypes K as hyperparameters. The model is trained to predict the cluster codes of transformed images $\underline{A}$ by the cluster codes of the transformed image $\underline{B}$ and the other way around within one min-batch by applying a cross-entropy loss with swapped predictions

$$L = -\sum_{k=1}^{K} \underline{BQC}_k \cdot \log(\underline{AQC}_k^*) - \sum_{k=1}^{K} \underline{BQC}_k \cdot \log(\underline{BQC}_k^*), \tag{A.3}$$

where the terms $\underline{AQC}_k^*$ and $\underline{BQC}_k^*$ are the softmax activation functions applied to the cluster codes.

As transformations, SwAV uses color jitter, Gaussian blur, and a multi-crop strategy, where two transformed images A and B are obtained by cropping a part of the original image with a larger crop size, and several additional samples are cropped with a smaller crop size. The transforms are implemented with torchvision with the following settings: two large crops of size 224, four small crops of the size 94, Gaussian blur with probability 0.5, and color jitter with probability 0.8 and channels $(0.4, 0.4, 0.2, 0.2)$. The cluster prototypes $\underline{C}$ are learned during training. The computed cluster codes $\underline{AQC}$ and $\underline{BQC}$ of one mini-batch should be equally partitioned by the prototypes. To ensure this equal partitioning and to avoid the trivial solution where all images collapse into the same code, the cluster codes are computed by maximizing the similarity between the latent space representations and the prototypes with the constraint

$$\max_{\underline{AQC}} \mathrm{Tr}(\underline{AQC}^T \underline{C}^T \underline{AQ}) + \epsilon H(\underline{AQ}), \tag{A.4}$$

were H is the entropy and ϵ a regularization parameter. The same constraint for transform B. The clustering is performed by using the iterative Sinkhorn–Knopp algorithm [67].

Table A.8, shows the hyperparameters for pre-training with SwAV. We choose the hyperparameters exactly as in the original SwAV paper.

Appendix B. Contrastive learning with MoCo

MoCo [36] starts with a large dataset of images $\{I_1, I_2, I_3, \ldots\}$, where two random transformations are performed to obtain two randomly different images from each original image: $\{A_1, A_2, A_3, \ldots\}$ and $\{B_1, B_2, B_3, \ldots\}$. Starting, for example, with the original image I_5, the transformed image A_5 is computed by an encoder to the latent space representation AQ_5, and the transformed image B_5 is computed by a momentum encoder to the latent space representation BQ_5. The encoders have the same architecture and can be any convolutional deep learning model. A dictionary is used to store the computed latent space representation of the momentum encoder BQ_5 together with the latent space representations of the momentum encoder from previous images $dict[\ldots, BQ_2, BQ_3, BQ_4, BQ_5]$. The samples in the dictionary are called keys. Inside the dictionary, there is now one key that comes from the same original image as the latent space representation of the encoder. In our example, this is BQ_5 and the pair $AQ_5 + BQ_5$ is called positive pair. The other keys in the directory come from different original images. The pairs $AQ_5 + BQ_4$, $AQ_5 + BQ_3$, $AQ_5 + BQ_2$, ... are called negative pairs. The model is trained to classify between positive and negative pairs by computing the InfoNCE loss

$$L_{10} = -\log \frac{\exp(AQ_5 \cdot BQ_k/\tau)}{\sum_{i=0}^{5} \exp(AQ_5 \cdot BQ_k/\tau)}, \tag{B.1}$$

which calculates a similarity score and where τ is a temperature hyperparameter.

Table B.9
Hyperparamters for pre-training with MoCo.

Parameters	Values
Input size	512
Number of crops	2
Size of crops	224
Optimizer	SGD
Batch size	64
Learning rate	1e−4
Momentum	0.9

MoCo Version 2 [58] is an updated version of MoCo that adds an MLP projection head to the encoder and additional data transformations. As transformations, MoCo V2 uses random crop, horizontal flip, and Gaussian blur. The transforms are implemented with torchvision with the following settings: two crops of size 224, Gaussian blur with probability 0.5, color jitter with probability 0.8 and channels $(0.4, 0.4, 0.2, 0.2)$, and horizontal flip with probability 0.5.

Table B.9, shows the hyperparameters for pre-training with MoCo V2. We choose the hyperparameters exactly as in the original paper.

Appendix C. Masked autoencoder with SparK

Inspired by natural language processing, where models are pre-trained by predicting missing words in a sentence, masked autoencoders pre-train vision models by dividing the images into patches, masking some of the patches, and training the model to reconstruct the original unmasked images [5]. SparK [39] is the first successful adaption of masked autoencoders to Convolutional neural networks.

Starting with a large dataset of images $\{I_1, I_2, I_3, \ldots\}$, each image is divided into non-overlapping square patches and each patch is masked independently with a given probability, called "mask ratio". The model consists of an encoder, which can be any convolutional model and a decoder. The encoder is adapted to perform submanifold sparse convolutions, which only compute when the center of a sliding window kernel is covered by a non-masked element. The decoder is built in a U-Net [48] design with three blocks of upsampling layers. The empty parts of the feature maps computed by the encoder are filled with learnable mask embeddings before being computed by the decoder. After the decoder, a head module is applied with two more upsampling layers to reach the original resolution of the input image. The model is trained with an L2 Loss between the predicted images of the model $\{I_1^*, I_2^*, I_3^*, \ldots\}$ and the original images $\{I_1, I_2, I_3, \ldots\}$, computed only on masked positions. For the downstream tasks, only the encoder is used.

Table C.10, shows the hyperparameters for pre-training with SparK. We choose the hyperparameters exactly as in the original paper.

Appendix D. Downstream task brain

Brain hemorrhage, also known as intracranial hemorrhage, is a condition characterized by bleeding inside the skull [68]. Rapid diagnosis is critical because of the potential complications it can cause, including brain swelling, brain infection, or death of brain matter. The etiology of this bleeding is the rupture of blood vessels within the skull, which can be caused by factors such as physical trauma or stroke [68].

Table C.10
Hyperparameters for pre-training with SparK.

Parameters	Values
Input size	512
Patch site	32 × 32
Mask ratio	60%
Augmentations	Horizontal flip, crop
Batch size	32
Optimizer	LAMB
Learning rate	Cosine Annealing (peak: 25e−6)

Table D.11
Downstream task Brain:.

Parameters	Values
Format	DICOM
Area	Brain
Window center	35/700 HU
Window width	80/3020 HU
Tube voltage	100–120 kV
Slice thickness	1 mm
CTDI	33–45
DLP	490-805 mgy·cm
Type	No Contrast-Enhanced
Size	512 × 512
Kernel	Soft Tissue
Scanners	PHILIPS Brilliance iCT 256 Siemens Somatom Definition AS+ Siemens Somatom Edge Plus
Gender	Unknown (anonymization)
Age	Unknown (anonymization)

Table E.12
This table shows the results of the three downstream tasks COVID-19, OrgMNIST, and Brain without using any pre-training. The weights of the model are initialized with PyTorch's standard random initialization.

Pre-training		Downstream results		
Dataset	Method	COVID-19 Acc	OrgMNIST Acc	Brain Acc
–	–	0.673 ± 0.026	0.755 ± 0.003	0.596 ± 0.034

Table E.13
Evaluation A: This table compares the baseline pre-training method ALL, the current state-of-the-art, which uses all slices of a CT dataset for contrastive pre-training, with the baseline reduction pre-training method EveryN. Pre-training with SwAV is performed on the datasets PET-CT and LIDC with all slices, with 20% of the slices by using every fifth slice, and with 10% of the slices, by using every tenth slice. The different pre-trainings are evaluated on the three downstream tasks COVID-19, OrgMNIST, and Brain.

Pre-training		Downstream results		
Dataset	Method	COVID-19 Acc	OrgMNIST Acc	Brain Acc
PET-CT	ALL	0.685 ± 0.012	0.752 ± 0.003	0.628 ± 0.100
	EveryN 20%	0.743 ± 0.015	0.789 ± 0.002	0.738 ± 0.047
	EveryN 10%	**0.755 ± 0.005**	**0.793 ± 0.020**	**0.772 ± 0.015**
LIDC	ALL	0.712 ± 0.015	0.769 ± 0.003	0.681 ± 0.058
	EveryN 20%	0.738 ± 0.009	0.801 ± 0.003	0.681 ± 0.034
	EveryN 10%	**0.746 ± 0.013**	**0.802 ± 0.002**	**0.683 ± 0.013**

An internal dataset with CT slices from 100 patients with and 100 patients without brain hemorrhage was selected by the two well-trained senior radiologists, Dr. Ch. G. Lisson and Dr. Ca. S. Lisson from the University Hospital of Ulm. Table D.11 shows details of the selected slices.

Table E.14
Evaluation B: This table compares different methods for reducing the pre-training datasets to 10% of the slices. The first method is the baseline reduction method EveryN, which reduces the pre-training dataset by using every tenth slice, followed by the similarity based methods, which perform a pairwise comparison of all slices in a CT volume and remove one slice from pairs with high similarity.

Pre-training		Downstream results		
Dataset	Method	COVID-19 Acc	OrgMNIST Acc	Brain Acc
PET-CT	EveryN	0.755 ± 0.005	0.793 ± 0.020	0.772 ± 0.015
	SSIM	0.752 ± 0.007	0.796 ± 0.002	0.770 ± 0.012
	MI	0.730 ± 0.006	0.798 ± 0.004	0.772 ± 0.015
	DeepNet	0.712 ± 0.002	0.799 ± 0.003	0.769 ± 0.013
	HASH	**0.758 ± 0.008**	**0.800 ± 0.003**	**0.775 ± 0.015**
LIDC	EveryN	0.746 ± 0.013	0.802 ± 0.002	0.683 ± 0.013
	SSIM	0.746 ± 0.006	0.802 ± 0.001	0.758 ± 0.028
	MI	0.748 ± 0.028	**0.803 ± 0.004**	0.734 ± 0.024
	DeepNet	0.727 ± 0.016	0.801 ± 0.006	0.706 ± 0.054
	HASH	**0.749 ± 0.009**	**0.803 ± 0.003**	**0.759 ± 0.019**

Table E.15
Evaluation C: This table compares different similarity thresholds of the best performing reduction method HASH, in order to obtain the optimal degree of similarity between the slices for contrastive pre-training.

Pre-training		Downstream results		
Dataset	Method	COVID-19 Acc	OrgMNIST Acc	Brain Acc
PET-CT	HASH - 3	0.749 ± 0.003	0.797 ± 0.002	0.724 ± 0.022
	HASH - 6	**0.764 ± 0.014**	**0.799 ± 0.033**	**0.793 ± 0.022**
	HASH - 12	0.739 ± 0.008	0.793 ± 0.002	0.703 ± 0.051
LIDC	HASH - 3	0.728 ± 0.008	0.803 ± 0.004	0.752 ± 0.040
	HASH - 6	**0.755 ± 0.008**	**0.804 ± 0.003**	**0.806 ± 0.035**
	HASH - 12	0.726 ± 0.011	0.798 ± 0.003	0.717 ± 0.033

Table E.16
Evaluation E: This table shows the downstream task results for pre-training with the contrastive learning (CL) approaches SwAV and MoCo Version 2 and the masked autoencoder (MAE) approach SparK. For all three approaches, we compare pre-training with all data (All) to pre-training with the reduced dataset using the hash reduction method and threshold 6 (Reduced) on the LIDC dataset.

Pre-training		Downstream results		
Approach	Data	COVID-19 Acc	OrgMNIST Acc	Brain Acc
SwAV (CL)	All	0.712 ± 0.015	0.769 ± 0.003	0.681 ± 0.058
	Reduced	**0.755 ± 0.008**	**0.804 ± 0.003**	**0.806 ± 0.035**
MoCoV2 (CL)	All	0.753 ± 0.014	0.817 ± 0.001	0.800 ± 0.003
	Reduced	**0.756 ± 0.005**	**0.819 ± 0.004**	**0.814 ± 0.032**
SparK (MAE)	All	**0.746 ± 0.005**	**0.783 ± 0.012**	**0.845 ± 0.003**
	Reduced	0.735 ± 0.013	0.782 ± 0.002	0.841 ± 0.042

Appendix E. Accuracy of all experiments

See Tables E.12–E.16.

References

[1] N. Kiryati, Y. Landau, Dataset growth in medical image analysis research, J. Imaging 7 (8) (2021) 155.
[2] L. Maier-Hein, M. Eisenmann, A. Reinke, S. Onogur, M. Stankovic, P. Scholz, T. Arbel, H. Bogunovic, A.P. Bradley, A. Carass, et al., Why rankings of biomedical image analysis competitions should be interpreted with care, Nat. Commun. 9 (1) (2018) 5217.
[3] S.-C. Huang, A. Pareek, M. Jensen, M.P. Lungren, S. Yeung, A.S. Chaudhari, Self-supervised learning for medical image classification: a systematic review and implementation guidelines, NPJ Digit. Med. 6 (1) (2023) 74.
[4] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, et al., A cookbook of self-supervised learning, 2023, arXiv preprint arXiv:2304.12210.

[5] F.C. Ghesu, B. Georgescu, A. Mansoor, Y. Yoo, D. Neumann, P. Patel, R.S. Vishwanath, J.M. Balter, Y. Cao, S. Grbic, et al., Contrastive self-supervised learning from 100 million medical images with optional supervision, J. Med. Imaging 9 (6) (2022) 064503.

[6] Y. Tang, D. Yang, W. Li, H.R. Roth, B. Landman, D. Xu, V. Nath, A. Hatamizadeh, Self-supervised pre-training of swin transformers for 3d medical image analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20730–20740.

[7] X. Chen, L. Yao, T. Zhou, J. Dong, Y. Zhang, Momentum contrastive learning for few-shot COVID-19 diagnosis from chest CT images, Pattern Recognit. 113 (2021) 107826.

[8] S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, et al., Big self-supervised models advance medical image classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3478–3488.

[9] N. Ewen, N. Khan, Targeted self supervision for classification on a small covid-19 ct scan dataset, in: 2021 IEEE 18th International Symposium on Biomedical Imaging, ISBI, IEEE, 2021, pp. 1481–1485.

[10] B. Dufumier, P. Gori, J. Victor, A. Grigis, M. Wessa, P. Brambilla, P. Favre, M. Polosan, C. McDonald, C.M. Piguet, et al., Contrastive learning with continuous proxy meta-data for 3d mri classification, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part II 24, Springer, 2021, pp. 58–68.

[11] C.S. Lisson, C.G. Lisson, M.F. Mezger, D. Wolf, S.A. Schmidt, W.M. Thaiss, E. Tausch, A.J. Beer, S. Stilgenbauer, M. Beer, et al., Deep neural networks and machine learning radiomics modelling for prediction of relapse in mantle cell lymphoma, Cancers 14 (8) (2022) 2008.

[12] V. Andrearczyk, V. Oreiller, M. Jreige, M. Vallieres, J. Castelli, H. Elhalawani, S. Boughdad, J.O. Prior, A. Depeursinge, Overview of the HECKTOR challenge at MICCAI 2020: automatic head and neck tumor segmentation in PET/CT, in: Head and Neck Tumor Segmentation: First Challenge, HECKTOR 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings 1, Springer, 2021, pp. 1–21.

[13] M. Jiang, H. Yang, X. Li, Q. Liu, P.-A. Heng, Q. Dou, Dynamic bank learning for semi-supervised federated image diagnosis with class imbalance, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 196–206.

[14] X. Xing, J. Huang, Y. Nan, Y. Wu, C. Wang, Z. Gao, S. Walsh, G. Yang, CS 2: A controllable and simultaneous synthesizer of images and annotations with minimal human intervention, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 3–12.

[15] N.A. Baghdadi, A. Malki, S.F. Abdelaliem, H.M. Balaha, M. Badawy, M. Elhoseini, An automated diagnosis and classification of COVID-19 from chest CT images using a transfer learning-based convolutional neural network, Comput. Biol. Med. 144 (2022) 105383.

[16] X. Wang, T. Shen, S. Yang, J. Lan, Y. Xu, M. Wang, J. Zhang, X. Han, A deep learning algorithm for automatic detection and classification of acute intracranial hemorrhages in head CT scans, NeuroImage: Clin. 32 (2021) 102785.

[17] A. Avesta, S. Hossain, M. Lin, M. Aboian, H.M. Krumholz, S. Aneja, Comparing 3D, 2.5 d, and 2D approaches to brain image auto-segmentation, Bioengineering 10 (2) (2023) 181.

[18] N. Zettler, A. Mastmeyer, Comparison of 2D vs. 3D U-Net Organ Segmentation in abdominal 3D CT images, in: International Conference on Computer Graphics, Visualization and Computer Vision 2021 - WSCG, 2021.

[19] D. Kern, U. Klauck, T. Ropinski, A. Mastmeyer, 2D vs. 3D U-net abdominal organ segmentation in CT data using organ bounds, in: Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications, Vol. 11601, SPIE, 2021, pp. 192–200.

[20] R. Bhattacharjee, L. Douglas, K. Drukker, Q. Hu, J. Fuhrman, D. Sheth, M. Giger, Comparison of 2D and 3D U-Net breast lesion segmentations on DCE-MRI, in: Medical Imaging 2021: Computer-Aided Diagnosis, Vol. 11597, SPIE, 2021, pp. 81–87.

[21] J. Yu, B. Yang, J. Wang, J. Leader, D. Wilson, J. Pu, 2D CNN versus 3D CNN for false-positive reduction in lung cancer screening, J. Med. Imaging 7 (5) (2020) 051202.

[22] T. Nemoto, N. Futakami, M. Yagi, A. Kumabe, A. Takeda, E. Kunieda, N. Shigematsu, Efficacy evaluation of 2d, 3D U-Net semantic segmentation and atlas-based segmentation of normal lungs excluding the trachea and main bronchi, J. Radiat. Res. 61 (2) (2020) 257–264.

[23] D. Wolf, T. Payer, C.S. Lisson, C.G. Lisson, M. Beer, M. Götz, T. Ropinski, Self-supervised pre-training with contrastive and masked autoencoder methods for dealing with small datasets in deep learning for medical imaging, Nat. Sci. Rep. 13 (1) (2023) 20260.

[24] L. Jing, P. Vincent, Y. LeCun, Y. Tian, Understanding dimensional collapse in contrastive self-supervised learning, in: International Conference on Learning Representations, 2022.

[25] R. Conrad, K. Narayan, CEM500K, a large-scale heterogeneous unlabeled cellular electron microscopy image dataset for deep learning, Elife 10 (2021) e65894.

[26] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[27] Z. Wang, A.C. Bovik, Mean squared error: Love it or leave it? A new look at signal fidelity measures, IEEE Signal Process. Mag. 26 (1) (2009) 98–117.

[28] J.P. Pluim, J.A. Maintz, M.A. Viergever, Mutual-information-based registration of medical images: a survey, IEEE Trans. Med. Imaging 22 (8) (2003) 986–1004.

[29] D.B. Russakoff, C. Tomasi, T. Rohlfing, C.R. Maurer, Image similarity using mutual information of regions, in: Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III 8, Springer, 2004, pp. 596–607.

[30] C. Studholme, D.J. Hawkes, D.L. Hill, Normalized entropy measure for multi-modality image alignment, in: Medical Imaging 1998: Image Processing, Vol. 3338, SPIE, 1998, pp. 132–143.

[31] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.

[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[34] A. Clark, Pillow (PIL Fork) documentation, 2015, URL https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf. (Acsession Date 01 July 2023).

[35] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.

[36] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.

[37] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., Bootstrap your own latent-a new approach to self-supervised learning, Adv. Neural Inf. Process. Syst. 33 (2020) 21271–21284.

[38] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, Adv. Neural Inf. Process. Syst. 33 (2020) 9912–9924.

[39] K. Tian, Y. Jiang, qishuai diao, C. Lin, L. Wang, Z. Yuan, Designing BERT for convolutional networks: Sparse and hierarchical masked modeling, in: The Eleventh International Conference on Learning Representations, 2023.

[40] S. Gatidis, T. Küstner, A whole-body FDG-PET/CT dataset with manually annotated tumor lesions (FDG-PET-CT-Lesions) [Dataset], Cancer Imaging Arch. (2022).

[41] S. Gatidis, et al., A whole-body FDG-pet/CT dataset with manually annotated tumor lesions, Sci. Data 9 (1) (2022) 601.

[42] S. Gatidis, T. Küstner, M. Ingrisch, M. Fabritius, C. Cyran, Automated lesion segmentation in whole-body FDG- PET/CT, 2022, http://dx.doi.org/10.5281/zenodo.6362493, (Acsession Date 01 July 2023).

[43] S.G. Armato III, et al., The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans, Med. Phys. 38 (2) (2011) 915–931.

[44] S.G. Armato III, et al., Data from LIDC-IDRI (Data set), Cancer Imaging Arch. (2015).

[45] W. Falcon, J. Borovec, A. Wälchli, N. Eggert, J. Schock, J. Jordan, N. Skafte, V. Bereznyuk, E. Harris, T. Murrell, et al., PyTorchLightning/pytorch-lightning: 0.7.6 release, 2020, http://dx.doi.org/10.5281/zenodo.3828935, (Acsession Date 01 July 2023).

[46] P. Kora, C.P. Ooi, O. Faust, U. Raghavendra, A. Gudigar, W.Y. Chan, K. Meenakshi, K. Swaraja, P. Plawiak, U.R. Acharya, Transfer learning techniques for medical image analysis: A review, Biocybern. Biomed. Eng. 42 (1) (2022) 79–107.

[47] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986.

[48] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[49] J. Zhao, Y. Zhang, X. He, P. Xie, COVID-CT-dataset: a CT scan dataset about COVID-19, 2020, arXiv preprint arXiv:2003.13865.

[50] J. Yang, et al., MedMNIST v2-a large-scale lightweight benchmark for 2D and 3D biomedical image classification, Sci. Data 10 (1) (2023) 41.

[51] M. Consortium, MONAI: Medical open network for AI: 1.0.0 release, 2022, http://dx.doi.org/10.5281/zenodo.7086266, (Acsession Date 01 July 2023).

[52] S. Kornblith, M. Norouzi, H. Lee, G. Hinton, Similarity of neural network representations revisited, in: International Conference on Machine Learning, PMLR, 2019, pp. 3519–3529.

[53] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[54] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.
[55] D. Kim, B. Han, On the stability-plasticity dilemma of class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 20196–20204.
[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.
[57] J. Gildenblat, contributors, PyTorch library for CAM methods, 2021, URL https://github.com/jacobgil/pytorch-grad-cam. (Acsession Date 01 July 2023).
[58] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning, 2020, arXiv preprint arXiv:2003.04297.
[59] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.
[60] M. Raghu, C. Zhang, J. Kleinberg, S. Bengio, Transfusion: Understanding transfer learning for medical imaging, Adv. Neural Inf. Process. Syst. 32 (2019).
[61] C. Coleman, C. Yeh, S. Mussmann, B. Mirzasoleiman, P. Bailis, P. Liang, J. Leskovec, M. Zaharia, Selection via proxy: Efficient data selection for deep learning, in: International Conference on Learning Representations, 2020.
[62] B. Mirzasoleiman, J. Bilmes, J. Leskovec, Coresets for data-efficient training of machine learning models, in: International Conference on Machine Learning, PMLR, 2020, pp. 6950–6960.
[63] H.E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M.E. Maros, T. Ganslandt, Transfer learning for medical image classification: a literature review, BMC Med. Imaging 22 (1) (2022) 69.
[64] L. Wu, J. Zhuang, H. Chen, Voco: A simple-yet-effective volume contrastive learning framework for 3d medical image analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 22873–22882.
[65] J.D. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, in: International Conference on Learning Representations, 2021.
[66] K. Clark, et al., The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository, J. Digit. Imaging 26 (2013) 1045–1057.
[67] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, Adv. Neural Inf. Process. Syst. 26 (2013).
[68] A.I. Qureshi, S. Tuhrim, J.P. Broderick, H.H. Batjer, H. Hondo, D.F. Hanley, Spontaneous intracerebral hemorrhage, N. Engl. J. Med. 344 (19) (2001) 1450–1460.