

Self-supervised Learning as a Means to Reduce the Need for Labeled Data in Medical Image Analysis

Marin Benčević^{*†}, Marija Habijan^{*}, Irena Galić^{*}, Aleksandra Pizurica[‡]

^{*}Faculty of Electrical Engineering, Computer Science and Information Technology

J. J. Strossmayer University of Osijek, Osijek, Croatia

[†]Email: marin.bencevic@ferit.hr

[‡]TELIN-GAIM, Faculty of Engineering and Architecture
Ghent University, Ghent, Belgium

Abstract—

One of the largest problems in medical image processing is the lack of annotated data. Labeling medical images often requires highly trained experts and can be a time-consuming process. In this paper, we evaluate a method of reducing the need for labeled data in medical image object detection by using self-supervised neural network pretraining. We use a dataset of chest X-ray images with bounding box labels for 13 different classes of anomalies. The networks are pretrained on a percentage of the dataset without labels and then fine-tuned on the rest of the dataset. We show that it is possible to achieve similar performance to a fully supervised model in terms of mean average precision and accuracy with only 60% of the labeled data. We also show that it is possible to increase the maximum performance of a fully-supervised model by adding a self-supervised pretraining step, and this effect can be observed with even a small amount of unlabeled data for pretraining.

Index Terms—contrastive learning, deep learning, medical image processing, object detection, self-supervised learning

I. INTRODUCTION

In medical image processing, the lack of annotated data is a common obstacle for deep learning models. To function robustly and to show their generalizability potential, deep neural networks require a large amount of annotated images [1]. However, annotating medical images often requires time-consuming, tedious work from trained clinicians. Hence, there is a huge need to improve the data efficiency and robustness of deep learning networks for medical image processing trained on smaller datasets. Furthermore, there is a need to make the labeling process faster to save experts' time. This paper presents a step towards both of these goals. The primary contribution of the paper is the evaluation of a method that uses self-supervised learning to extract salient information from *unlabeled* images which can then be used to more easily train an object detection model on a more limited dataset of *labeled* images.

We show that it is possible to improve the performance of an object detection model for medical images by utilizing

self-supervised learning in a pretraining phase. In addition, we also show that it is possible to achieve similar performance to a fully supervised model with only 60% of the labeled data. The code for all of the experiments in this paper is available at github.com/marinbenc/ssl-for-medical-object-detection.

A. Related work

There are very few papers published that present methods for object detection in chest X-ray images. One notable paper similar to our work is the one by Luo et al. [2]. They use the same dataset as the experiments presented in this paper and train a YOLOv5 model with a ResNet50-based backbone. They achieve results similar to the ones in our experiments, however, they do not employ any kind of unsupervised learning and train on the full labeled dataset. The goal of this paper is not to beat the state of the art performance, but rather to show if similar results could be achieved with much less labeled data.

1) *Improving data efficiency of neural nets*: There are several ways to try to utilize unlabeled data or data prepared for other tasks to improve the data efficiency of medical image processing, including semi-supervised learning, transfer learning and, more recently, self-supervised learning. While transfer learning on large image datasets such as ImageNet is very common in medical image processing, Raghu et al. [3] found that this provides little benefit in terms of model performance, but does improve convergence speed during training.

Another way to utilize an unlabeled dataset to improve performance of a model is semi-supervised learning, where the unlabeled dataset is mined for soft signals to nudge the model towards better overall performance. One such example is the one by Liu et al. [4] where they present a semi-supervised learning method for medical image classification using a combination of unlabeled and labeled data from the same domain.

2) *Self-supervised learning*: Recently more and more papers use self-supervised learning (SSL) to pre-train neural networks on unlabeled data, and then fine-tune the networks on the available labeled data. SSL is a method of unsupervised neural network training where the goal is to train an encoder that will understand useful features for a downstream task such

as object detection, classification, or similar. SSL has been shown to improve data efficiency [5] as well as robustness to dataset imbalance [6].

There are several approaches to training the encoder such that it learns useful features. One approach is to use a constructed *pretext* task for which one can automatically obtain the correct solution from unlabeled data so that the correct solution can be used for supervised training. An example of this approach is presented by Noroozi and Favaro [7] where the neural network is trained to solve a jigsaw constructed from an unlabeled dataset.

Recently, a more common approach to SSL is contrastive learning. In contrastive learning, the encoder is trained to minimize the distance between feature vectors of positive examples and maximize the distance between negative examples. The positive examples are constructed in an unsupervised manner by e.g. randomly augmenting an image twice, thus producing two examples for which the feature vectors should be similar. Among others, notable examples of such approaches are SimCLR [5] and MoCo [8].

3) *Self-supervised learning in medical images:* Due to its potential to improve data efficiency, SSL is a promising approach for medical image processing. Several recent papers proposed SSL methods both using the pretext task approach [9], [10] as well as contrastive learning [11]. Several papers evaluate self-supervised pretraining on an unlabeled subset of the data for medical imaging tasks. Taleb et al. [12] evaluate a variety of self-supervised pretraining approaches for medical image segmentation in 3D MRI and CT images as well as classification on 2D fundus photography images. They use unlabeled data of the same modality but from a different corpus. Azizi et al. [13] introduce a novel contrastive learning method and evaluate it at various percentages of used dataset labels on dermatology and chest X-ray classification. However, to our knowledge there are currently no papers that evaluate self-supervised pretraining for object detection tasks in medical images.

II. DATASET DESCRIPTION AND DEMOGRAPHICS

The dataset used in this paper is a dataset of 15,000 labeled chest radiographs called VinDr-CXR, described in more detail in [14]. While the original dataset contains 3,000 additional test images, we were not able to obtain the labels for these images, and they were not used in this paper. Each scan of the dataset was labeled by three separate radiologists. The dataset was collected from two major Vietnamese hospitals and is to our knowledge the largest dataset for radiograph object detection. Therefore, it is a good indicator for the generalizability of our experiments on other radiograph datasets.

A. Data preparation

Each DICOM image from the dataset was resized to a resolution of 512×512 pixels. We discard all examples for which there is no anomaly found (examples labeled as “no finding”). After discarding, we are left with a total of 4,394 images. We randomly split this dataset into a training set (70%,

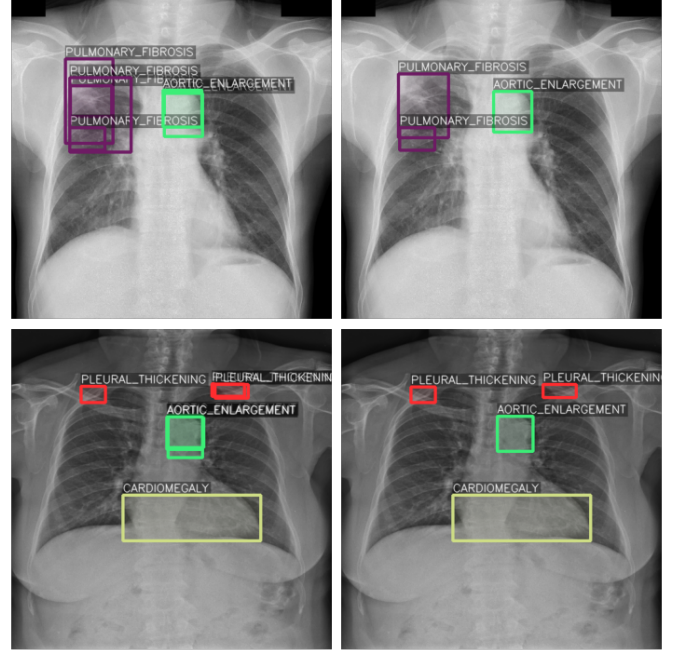


Fig. 1. Examples of bounding box averaging. The images on the left show the original bounding boxes, while the images on the right show fused bounding boxes which are used in our experiments.

3,075 images), validation set (10%, 440 images) and a test set (20%, 878 images). The training set was used to train the models, the validation set was used to tune the model hyperparameters and determine when to stop training, and the test set is used for final evaluation. The model did not have access to the test set during training. The class distribution of the full dataset is shown in Fig. 3. During training, each image had a 50% chance of being flipped horizontally.

Each image can have one or more labels from multiple experts, and these labels often overlap. To produce the least noisy labels, we average overlapping bounding boxes of the same class into one bounding box. To determine if two boxes are overlapping, an intersection-over-union (IoU) threshold of 20% is used. This approach is based on weighted boxes fusion described in [15] but modified such that each bounding box has equal weight and confidence since they were manually labeled by an expert. An example of fused bounding boxes is shown in Fig. 1.

III. METHODS

A summary of our approach is shown in Fig. 2. The main goal of this paper is to analyze how self-supervised pretraining affects data efficiency for object detection in medical images. Therefore, we first train a *baseline* deep learning model with no pretraining and on the full labeled training dataset following a standard approach for this type of problem, described later in this section. This model will be used as a point of comparison to more objectively evaluate the pre-trained models.

To evaluate the pretraining, we randomly split the training dataset into two separate datasets, a pretraining and fine-

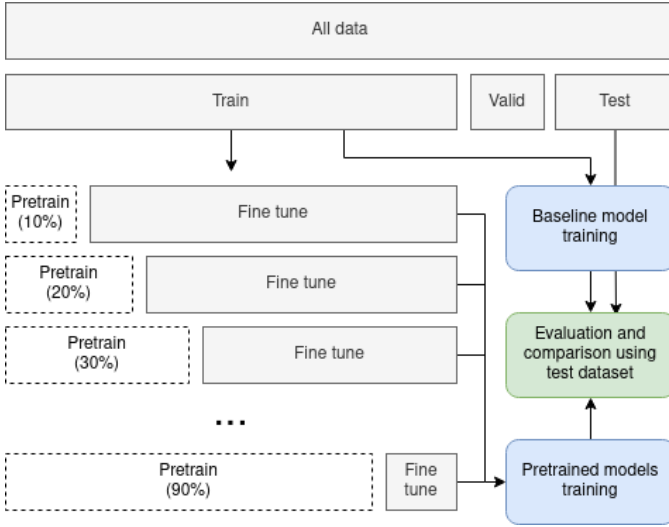


Fig. 2. A summary of our approach. A percentage of the training dataset is moved to the pretraining dataset and used to pretrain the model, which is then fine-tuned with the rest of the training data. The pretraining datasets are unlabeled. A separate baseline model is trained using the full labeled dataset.

tuning dataset. For the pretraining dataset, we discard all class labels, as this dataset will be used to pretrain the model using self-supervised learning on unlabeled data. The fine-tuning dataset will then be used to fine-tune the pre-trained model using standard supervised learning. We train nine different pre-trained models in total, ranging from 10% to 90% of the total training set in the unlabeled pre-train dataset, in increments of 10%.

A. Balancing the dataset

As shown in Fig. 3, the dataset is highly imbalanced. To improve class balance during training we use augmentation to oversample examples with less-represented classes. We use the oversampling approach described in [16] with an oversampling threshold of 0.4. This balancing is performed online during pre-training and fine-tuning and only on the training set. We found that this approach significantly improved the mean average precision when averaged across all classes.

B. Baseline model details

For an objective and fair comparison, we train a baseline model using a standard deep learning-based approach for object detection. We use a Faster R-CNN-based model [17] with a ResNet-50 encoder [18]. The baseline model is initialized using pretrained weights trained on the COCO dataset for 12 epochs, a batch size of 2, and using the SGD optimizer with a learning rate of 0.0002, the momentum of 0.9, and a weight decay of 0.0001. Training is done by using the SGD optimizer with a learning rate of 0.001, a momentum of 0.9, a weight decay of 0.0001, and a batch size of 8. The baseline model converged in 15 epochs. We use the cross-entropy loss for the classes and the L1 loss for the bounding boxes.

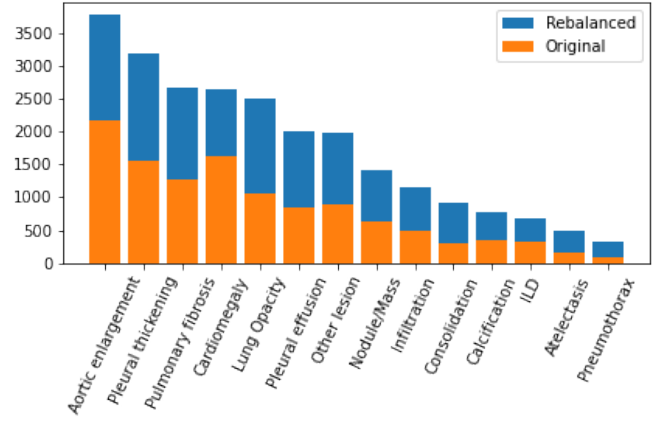


Fig. 3. A histogram of the class balance of the original dataset (by number of images containing the class), and the histogram of the oversampled dataset used for training.

C. Pretraining model details

The pretraining model we use is a SimCLR-based model [5] to pre-train a ResNet-50 backbone, the same backbone used in the baseline model. We train the SimCLR model (described later in this section) using the unlabeled pretraining dataset. We then use the pre-trained backbone in the same model as our baseline model and fine-tune the final model on the fine-tuning dataset. The result is a model similar to our backbone model but trained on fewer labeled data.

In our experiments, we use the following augmentations for SimCLR training:

- 1) A random crop and resize of the original image by a factor of 0.2 to 1.
- 2) A random horizontal flip (with a 50% chance).
- 3) A random Gaussian blur with σ between 0.1 and 2, and a kernel size of 21.
- 4) A random amount of Gaussian noise with σ between 12.5% and 25% of the mean image pixel value.

In addition, each training and validation image has histogram normalization applied.

We pretrain the models using the SGD optimizer with a learning rate of 0.001, a momentum of 0.9, and a weight decay of $5 \cdot 10^{-4}$. We also use a cosine annealing learning rate as described in [19]. For pretraining we use the NTXent loss as described in [5] with a temperature of 0.5. The pretraining was stopped after 30 epochs in all experiments.

IV. RESULTS

A summary of the results of our experiments is shown in Table I. All of the results in this section are calculated on the testing dataset. Our main evaluation metrics are the mean average precision (mAP), averaged across all classes and IoU thresholds from 0.5 to 0.95 in 0.05 intervals (mAP@[.5, .95]), as is standard for benchmarking the COCO dataset. We also calculate the mAP at a fixed IoU of 0.5 (mAP@0.5), which is standard for evaluating models on the PASCAL VOC

TABLE I

A SUMMARY OF THE RESULTS OF OUR EXPERIMENTS. THE DETAILS OF THE METRICS ARE DESCRIBED IN IV. TRAINING IMAGES IS THE TOTAL NUMBER OF LABELED TRAINING EXAMPLES AVAILABLE TO THE MODEL.

| | mAP | mAP50 | mAP small | AR | AR small | Training images |
|----------|--------------|--------------|--------------|--------------|--------------|-----------------|
| Baseline | 0.129 | 0.278 | 0.021 | 0.412 | 0.154 | 3075 |
| SSL 10% | 0.142 | 0.284 | 0.026 | 0.413 | 0.156 | 2767 |
| SSL 20% | 0.139 | 0.292 | 0.020 | 0.412 | 0.158 | 2460 |
| SSL 30% | 0.130 | 0.268 | 0.014 | 0.402 | 0.146 | 2152 |
| SSL 40% | 0.123 | 0.272 | 0.014 | 0.394 | 0.131 | 1845 |
| SSL 50% | 0.109 | 0.248 | 0.011 | 0.387 | 0.131 | 1537 |
| SSL 60% | 0.104 | 0.230 | 0.011 | 0.378 | 0.120 | 1230 |
| SSL 70% | 0.095 | 0.202 | 0.007 | 0.363 | 0.108 | 922 |
| SSL 80% | 0.081 | 0.178 | 0.006 | 0.338 | 0.089 | 615 |
| SSL 90% | 0.060 | 0.135 | 0.003 | 0.303 | 0.064 | 307 |

dataset. In addition, we calculate the average recall given 100 detections per image (AR@100). We separately calculate the mAP[.5, .95] as well as the AR@1000 for small objects (less than 32 pixels²).

The baseline model, trained on all of the labeled data, achieves an mAP of 0.129. In addition, by adding pretraining on a small percentage of the dataset (20% or fewer) the performance improves for all of the metrics we calculated. For instance, when pretraining on 10% of the data (i.e. removing 10% of the labels) the mAP increases to 0.142. This is the best-performing model in our experiments. However, even with just 60% of the labeled data, we achieve an mAP of 0.123, more than 95% of the performance of the baseline model in terms of mAP. Similar results can be seen in terms of recall — the baseline model achieves an average recall of 0.412 while with 60% of the labels we achieve an AR of 0.394, more than 95% of the base model’s AR.

However, the differences are larger when looking at small objects. While the baseline model achieves a 0.021 mAP for small objects, the model trained on 60% of the labeled images achieves a small objects mAP of 0.014, 66.66% of the baseline model. Similarly, the model trained on 60% of the labeled images achieves a small object AR of 0.131, 85% of the baseline model’s small object AR of 0.154.

A graphical summary of the class-wise results is shown in Fig. 4. The performance on most classes drops off linearly between the baseline model and the model trained on 60% of the labels data. The dropoff becomes more significant after training on less than 60% of the labels. In all cases, the mAP at 60% of the labeled data is highly correlated with the mAP across all classes as described earlier in the section, and the same conclusions apply to all classes.

The model performs the best at detecting cardiomegaly and aortic enlargement, two classes with a large number of examples and a large average object area size. All of the models perform significantly worse at detecting other anomalies. However, by adding pretraining on a small percentage of the

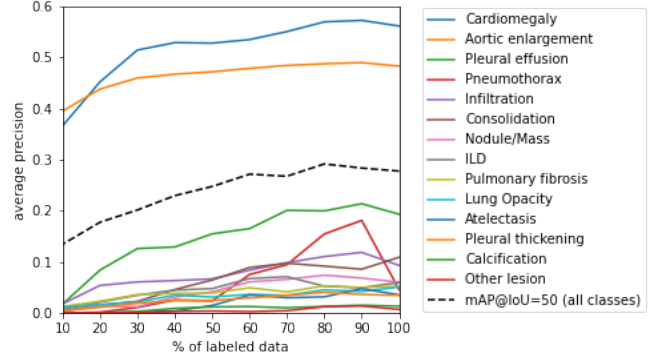


Fig. 4. The mean average precision (mAP@[.5, .95]) across different percentages of labeled data for different classes in the dataset. The model at 100% of labeled data is the baseline model.

data (20% or fewer) the mAP improves for almost all classes, most significantly for detecting other lesions and infiltration.

We also evaluated our experiments in terms of IoU, and compared the inter-observer IoU between multiple experts and our experiments, as shown in Table II. The original dataset was labeled by a group of three radiologists from a total of 17 radiologists identified as *R1* through *R17*. In our test dataset, 747 images (85% of the test dataset) were labeled by both radiologists *R9* and *R10*, while 750 images (85.4% of the test dataset) were labeled by both radiologists *R9* and *R11*. Therefore, we calculate the inter-observer IoU between radiologists *R9* and *R10* as well as *R9* and *R11*, since they represent the majority of the dataset. For calculating the inter-observer IoU, we only take into account labels of the same class where the IoU overlap is over 0.2, to avoid counting multiple instances of a class as the same prediction. We calculate the models’ IoU metric as the mean of IoUs between each fused ground truth annotation and the maximum-overlap model prediction of the annotation’s class.

The inter-observer IoU is between 0.58 and 0.67, while the IoU between ground truth and predicted labels of our baseline model is 0.57, slightly worse than the variability between two human experts. The IoU starts to drop after more than 20% of the labels are removed. At 60% of the labels, the IoU is 0.545, more than 96% of the baseline IoU. Note that the

TABLE II
THE INTER-OBSERVER IOU BETWEEN RADIOLOGIST *R9* AND RADIOLOGISTS *R10* AND *R11*, AS WELL AS THE MEAN IOU OF THE MODEL’S PREDICTIONS.

| | mean IoU | σ |
|------------|----------|----------|
| R09 vs R10 | 0.580 | 0.284 |
| R09 vs R11 | 0.674 | 0.149 |
| Baseline | 0.565 | 0.309 |
| SSL 10% | 0.566 | 0.309 |
| SSL 20% | 0.565 | 0.308 |
| SSL 30% | 0.555 | 0.312 |
| SSL 40% | 0.545 | 0.316 |

model's IoU is based on the best predictions coming from the model. Under real-world conditions, it is not known which predictions are best and some heuristic is needed which might not always choose the best prediction, so the presented IoU values should be interpreted as best-case scenario IoU values. Still, the relative IoU differences between the baseline and the pretrained models should remain the same.

V. DISCUSSION AND CONCLUSION

Our results on medical object detection experiments show that it is possible to use fewer labeled examples and still achieve similar performance by utilizing self-supervised pre-training. All of the global metrics we measured, including average precision and recall as well as IoU, remain within 95% of the baseline model's performance. The same is true for class-based metrics. In our experiments, using 60% of labeled data is a good compromise between using fewer labels while still maintaining good object detection performance. We hypothesize that the backbone trained under self-supervised contrastive learning produces more salient feature maps and thus overcomes the lack of labels when compared to the baseline model.

The biggest performance loss for the self-supervised models is in smaller objects, where the model trained on 60% of the labeled data achieves less than 65% of the small object mean average precision of the baseline model. A very common occurrence in deep learning models for object detection is a gap in performance between predicting small and large objects. Smaller objects are less frequent, and cover a smaller area of the image, making it difficult for models to predict them. This problem can be overcome at least in part using augmentation [20], and it's possible our results could be improved by adding oversampling and small object augmentation to both the training and pretraining steps.

However, at 60% of labeled data, the model still retains 85% of the small object recall of the baseline model. In medical image processing tasks, recall is often more important than precision as the cost of finding false positives is much lower than the cost of missing true positives. Therefore, the reduction in small object detection performance is less severe than it appears from the mAP.

We hope that the findings of this paper will lead to enabling more deep learning-based methods in medical object detection with fewer labeled data, as well as increase the performance of existing models trained on small datasets. This paper is also an indicator that self-supervised learning is a very promising avenue for future research in medical image processing.

REFERENCES

- [1] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, Dec. 2017.
- [2] Y. Luo, Y. Zhang, X. Sun, H. Dai, and X. Chen, "Intelligent Solutions in Chest Abnormality Detection Based on YOLOv5 and ResNet50," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–11, Oct. 2021.
- [3] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding Transfer Learning for Medical Imaging," *arXiv:1902.07208 [cs, stat]*, Oct. 2019.
- [4] Q. Liu, L. Yu, L. Luo, Q. Dou, and P. A. Heng, "Semi-supervised Medical Image Classification with Relation-driven Self-ensembling Model," *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, pp. 3429–3440, Nov. 2020.
- [5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," *arXiv:2002.05709 [cs, stat]*, Jun. 2020.
- [6] H. Liu, J. Z. HaoChen, A. Gaidon, and T. Ma, "Self-supervised Learning is More Robust to Dataset Imbalance," *arXiv:2110.05025 [cs, stat]*, Oct. 2021.
- [7] M. Noroozi and P. Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, vol. 9910, pp. 69–84.
- [8] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *arXiv preprint arXiv:1911.05722*, 2019.
- [9] W. Bai, C. Chen, G. Tarroni, J. Duan, F. Guitton, S. E. Petersen, Y. Guo, P. M. Matthews, and D. Rueckert, "Self-Supervised Learning for Cardiac MR Image Segmentation by Anatomical Position Prediction," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham: Springer International Publishing, 2019, vol. 11765, pp. 541–549.
- [10] J. Zhu, Y. Li, Y. Hu, K. Ma, S. K. Zhou, and Y. Zheng, "Rubik's Cube+: A self-supervised feature learning framework for 3D medical image analysis," *Medical Image Analysis*, vol. 64, p. 101746, 2020.
- [11] H.-Y. Zhou, S. Yu, C. Bian, Y. Hu, K. Ma, and Y. Zheng, "Comparing to Learn: Surpassing ImageNet Pretraining on Radiographs by Comparing Image Representations," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racocanu, and L. Joskowicz, Eds. Cham: Springer International Publishing, 2020, vol. 12261, pp. 398–407.
- [12] A. Taleb, W. Loetzsch, N. Danz, J. Severin, T. Gaertner, B. Bergner, and C. Lippert, "3D self-supervised methods for medical imaging," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 18 158–18 172.
- [13] S. Azizi, B. Mustafa, F. Ryan, A. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, V. Natarajan, and M. Norouzi, "Big Self-Supervised Models Advance Medical Image Classification," *arXiv:2101.05224 [cs, eess]*, Apr. 2021.
- [14] H. Q. Nguyen, K. Lam, L. T. Le, H. H. Pham, D. Q. Tran, D. B. Nguyen, D. D. Le, C. M. Pham, H. T. T. Tong, D. H. Dinh, C. D. Do, L. T. Doan, C. N. Nguyen, B. T. Nguyen, Q. V. Nguyen, A. D. Hoang, H. N. Phan, A. T. Nguyen, P. H. Ho, D. T. Ngo, N. T. Nguyen, N. T. Nguyen, M. Dao, and V. Vu, "VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations," *arXiv:2012.15029 [eess]*, Jan. 2021.
- [15] R. Solovyyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104117, Mar. 2021.
- [16] A. Gupta, P. Dollar, and R. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [17] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 91–99.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.
- [19] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," *arXiv:1608.03983 [cs, math]*, May 2017.
- [20] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," in *9th International Conference on Advances in Computing and Information Technology (ACITY 2019)*. Airc Publishing Corporation, Dec. 2019, pp. 119–133.