Using self-supervised learning to decrease the need for labeled data in medical image object detection*

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*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

A. Motivation

One of the largest problems in medical image processing is the lack of annotated data. To function robustly and to show their generalizability potential, deep learning networks require a large amount of annotated images. However,

annotating medical imaging is often time-consuming, precise work. There is a need to improve the data-efficiency and robustness of deep learning networks trained on smaller datasets. Furthermore, there is a need to make the labeling process faster to save experts' time. This paper represents a step towards both of those goals. First, we present a method which uses self-supervised learning to extract salient information from *unlabeled* images which can then be used to more easily train a deep learning network on a more limited dataset of *labeled* images.

Secondly, we evaluate this on an object detection dataset. In terms of labeling complexity, it is simpler and quicker to label a dataset with bounding boxes than to label each instance as is needed for semantic segmentation. In a lot of medical imaging tasks, a precise semantic segmentation map is not required, and a bounding box communicates sufficient information for further diagnosis, treatment or research on a given image. By improving data efficiency for bounding box labels, we hope to increase their usefulness and thus save time by allowing experts to use bounding box labels in place of semantic segmentation maps.

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 [1]. 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.

We chose this dataset as a good indicator of the generalizability of the findings in this paper due to several reasons. Firstly, to our knowledge, this is the largest radiograph dataset with bounding box labels for each finding. Secondly, having multiple labels for each image allows us to perform a large variety of statistical evaluation methods, such as comparing the inter-observer variability. Having multiple labels also allows us to use automatic methods to find a concensus between them, possibly leading to less noisy annotations.

A. Data preparation

Each DICOM image from the dataset was resized to a resolution of 512 $\times$ 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%, 3075 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 original dataset differentiates 14 different classes, one of which is the "no finding" class. We discard this class, resulting in 13 possible class labels. The class distribution of the full dataset is shown in Figure **??**.

Each image can have one or more labels from multiple experts, and these labels often overlap. To produce the least noisy labels, we fused overlapping labels from multiple experts into one label by finding an average rectangle of several overlapping rectanges. To determine if two rectangles are overlapping, an intersection-over-union (IoU) threshold of 20% is used. A rectangle R_i is defined by its top-left corner (X_{i1}, Y_{i1}) and its bottom-right corner (X_{i2}, Y_{i2}). An average rectangle $\bar{R}$ defined by coordinates $(\bar{X_1}, \bar{Y_1})$ and $(\bar{X_2}, \bar{Y_2})$ is calculated as follows:

This work was supported in part by the Croatian Science Foundation under Project UIP-2017-05-4968.

$$\bar{X_1} = \frac{\sum_{i=1}^{N} X_{i1}}{N}, \bar{Y_1} = \frac{\sum_{i=1}^{N} Y_{i1}}{N}, \quad (1)$$
$$\bar{X_2} = \frac{\sum_{i=1}^{N} X_{i2}}{N}, \bar{Y_2} = \frac{\sum_{i=1}^{N} Y_{i2}}{N}$$

Where N is the number of rectangles to average. This approach is based on weighted boxes fusion described in [2], 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 **??**.

III. Methods

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

To evaluate the pre-training, we randomly split the traning dataset into two separate datasets, a pre-traning and fine-tuning dataset. For the pre-traning dataset we discard all class labels, as this dataset will be used to pre-train 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 summary of our approach is shown in 1.

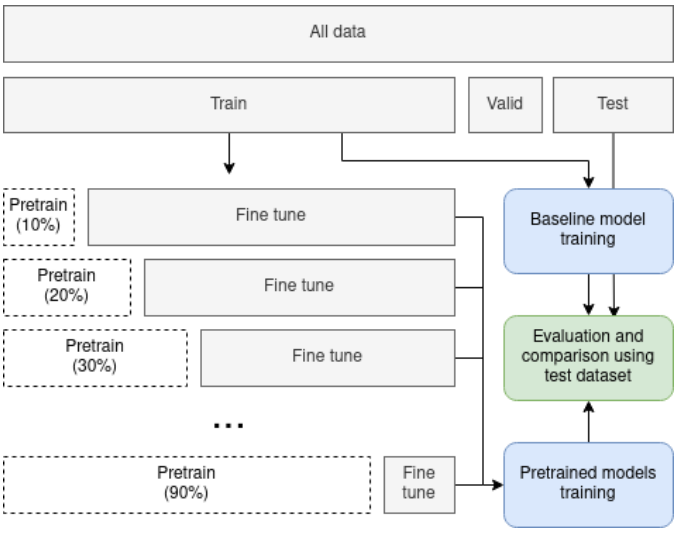

Fig. 1. A summary of our experiments. A percentage of the training dataset is moved to the pre-training dataset and models are pre-trained using the pre-training datasets and then fine tuned with the rest of the training data. The pre-training datasets are unlabeled. A separate baseline model is trained using the full labeled dataset.

A. Balancing the dataset

As shown in Figure **??** the dataset is highly imbalanced. To improve class-balance during training we oversample examples with less-represented classes. We use the oversampling approach described in [3]. The general approach is as follows:

1) For each class c, compute it's frequency $f(c)$ in the training set.
2) Compute a class repeat factor $r(c) = max(1, \sqrt{t/f(c)})$, where t is a threshold value empericially set to 0.4 in our experiments. The value of 0.4 produced the maximum validation mean average precision for the baseline model.
3) For each image I, compute an image repeat factor $r(I) = C * r(c)$ where $C = max(r(c_i))$ for each class c_i in image I.

This balancing is performed on-line during training and only on the training set. We found that balancing the dataset in this way drastically 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 [4] with a ResNet50 encoder [5]. The model is initialized using pre-trained 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, momentum of 0.9 and a weight decay of 0.0001.

C. Pretraining model details

The pretraining model we use is a SimCLR-based model [6] 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 pre-training 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.

SimCLR uses contrastive learning where an example image is augmented randomly twice and each augmentation is fed into a separate encoder branch, where the branches use shared weights. The network outputs two feature maps, one for each augmentation. The loss function measures the difference between these two feature maps. The closer the two feature maps are, the lower the loss. This ensures that two augmentations from the same example will produce similar feature maps, thus making the network learn salient features which are invariant to the chosen augmentations.

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 and 2, and a kernel size of 21.

4) A random amount of Gaussian noise with σ being a random number between 12.5% and 25% of the mean image pixel value.

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

IV. Results

A summary of the results of our experiments is shown on Table I. 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 dataset. These metrics are also calculated class-wise. In addition, we calculate the average recall given 100 detections per image (AR@100).

TABLE I
A SUMMARY OF THE RESULTS OF OUR EXPERIMENTS. MAP IS THE MEAN AVERAGE PRECISION AT IOU VALUES FROM 0.5 TP 0.95 AT 0.05 (MAP@[.5, .95]) INCREMENTS ACROSS ALL CLASSES, THE STANDARD METRIC FOR THE COCO BENCHMARK. MAP50 IS THE MAP AT IOU = 50% (MAP@0.5), THE STANDARD METRIC FOR THE PASCAL VOC BENCHMARK. MAP SMALL IS THE MAP@[.5, .95] FOR OBJECTS WITH AN AREA SMALLER THAN 32 PIXELS^2. AR IS THE AVERAGE RECALL GIVEN 100 DETECTIONS PER IMAGE (AR@100). AR SMALL IS THE SAME AS AR BUT ONLY FOR OBJECTS WITH AN AREA SMALLER THAN 32 PIXELS^2. 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

The baseline model, trained on all of the labeled data, achieves an mAP of 0.129. By adding 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, only slightly smaller than the baseline model. In terms of percentages, with only 60% of the labels we still achieve more than 95% of the performance of the baseline model in terms of mAP. Similar results can be see in terms of recall — the baseline model achieves an AR 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 0.412.

V. Discussion

Discussions should be brief and focused. In some disciplines use of Discussion or 'Conclusion' is interchangeable. It is not mandatory to use both. Some journals prefer a section 'Results and Discussion' followed by a section 'Conclusion'. Please refer to Journal-level guidance for any specific requirements.

VI. Conclusion

Conclusions may be used to restate your hypothesis or research question, restate your major findings, explain the relevance and the added value of your work, highlight any limitations of your study, describe future directions for research and recommendations.

In some disciplines use of Discussion or 'Conclusion' is interchangeable. It is not mandatory to use both. Please refer to Journal-level guidance for any specific requirements.

References

[1] 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.

[2] R. Solovyev, 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.