

CONSTRAINED CONTRASTIVE REPRESENTATION: CLASSIFICATION ON CHEST X-RAYS WITH LIMITED DATA

WeiQi Zhang¹, Hongbo Wang^{1,2,3,*}, Zhiping Lai¹, Chao Hou¹

¹Academy for Engineering and Technology, Fudan University, China;

²Shanghai Engineering Research Center of AI and Robotics, Fudan University, China;

³Engineering Research Center of AI and Robotics, Ministry of Education, China.

{weiqizhang19,Wanghongbo,zplai19,chou18}@fudan.edu.cn

ABSTRACT

One of the challenges in the field of medical image classification is the expensiveness of labeled data. Most of the previous computer-aided diagnostic methods are based on a paradigm of object detection. Such ways need tons of labeled sample images with positioning annotations, which always need practicing radiologists to process data manually. We focus on Chest X-ray(CXR) images classification and propose an effective framework for lung disease diagnosis based on a self-supervised feature extracting mechanism trained in a constrained contrastive method. Our proposed framework can train on a relatively small dataset in a semi-supervised way and without any positioning annotation. We experiment with the proposed framework on several lung disease diagnosis tasks, including pneumonia and tuberculosis diagnosis, and obtain state-of-the-art results even outperform previous supervised transfer-learning methods.

Index Terms— Computer-aided diagnosis, limited data, chest X-ray, semi-supervised learning, contrastive learning

1. INTRODUCTION

The lack of high-quality medical resources is one of the most prominent problems in the medical field, especially in the developing world. Chest X-ray(CXR) imaging is one of the most common and easy-to-use diagnostics. The detection of specific diseases with CXR images can be abstracted to an image classification task, which has aroused the interest of many researchers [1–4].

In most cases, the lesion can only appear in a relatively small area and a large amount of information is noisy. To extract highly abstract features, we can make use of a deep convolutional network [5]. Here comes a barrier that the training tends to overfit when we don't have enough samples.

The label is necessary for modern deep-learning-based methods. Due to the reason that it needs to collect a bunch

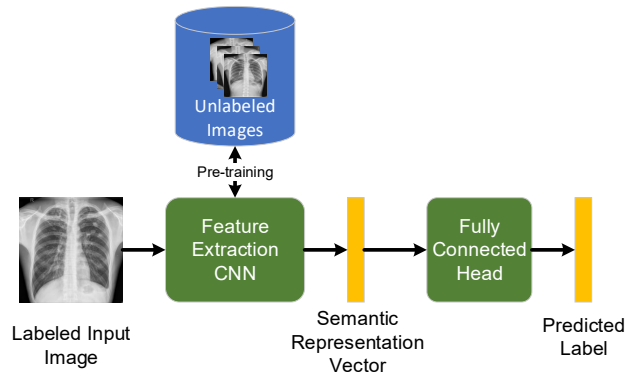


Fig. 1. The feature extraction CNN built to produce semantic representations is pre-trained on unlabeled task-agnostic images at the first stage. And then, the fully connected head is trained with task-specific labeled images.

of structured labeled data, it is expensive to train a computer-aided diagnostic system that follows the supervised object-detection paradigm. Recently, many efforts have been devoted to training a classification model in a semi-supervised way. Inspired by modern language representation models such as BERT [6], a variety of semi-supervised learning algorithms have been proposed [7–10], assuming that the training dataset comprises a relatively small number of labeled data and a large number of unlabeled data. Most of these methods establish the entire framework based on a two-stage paradigm(see Fig. 1). A feature extraction CNN is trained through a bunch of unlabeled task-agnostic samples followed by training a task-specific classification head. These methods can obtain good performance on the public benchmark dataset, such as ImageNet. But the premise is that the task-specific data or labeled data is similar in statistical distribution with most of the unlabeled data.

Here we consider this two-stage paradigm mentioned above as a solution of overfitting, noting that we can train our model with a small labeled dataset. We just need our feature extraction model to learn from task-agnostic CXR images instead of ImageNet. We find that a relatively small unlabeled CXR dataset which doesn't as large as ImageNet can work

This work was supported by the Key Technology R&D Program of Hebei, China (20371801D).

*Hongbo Wang is the corresponding author.

well, due to the reason that CXR image samples have their distinct characteristics that there is a high degree of similarities among the input images. Our feature extraction model is trained in a contrastive self-supervised way, which learns knowledge by comparing two images and finds the critical information. We choose as our starting point the task-agnostic part of the SimCLRv2 [10] model which is already trained on ImageNet. It establishes a task-agnostic CNN for semantic representation, see Section 3.2 for more detail.

During task-agnostic training, various transformations (data augmentations) will be applied to images for self-supervised learning, such as cropping, flipping, gaussian blur. To learn proper representations, the way of image cropping is one of the crucial factors. We introduce a schema called center-specific cropping constraint(C3), which lets the model learn semantic representations easier, and a center-specific rotation transformation. Our empirical results show it brings better performance for downstream classification tasks.

The main contributions of our work can be summarized as follow:

- We demonstrate that the contrastive self-supervised model can be trained on a small dataset when it comes to medical images among which there is a high degree of invariance.
- Our empirical data indicates that our proposed C3 schema and the corresponding center-specific rotation together can prompt our feature extractor to learn more useful semantic representations for downstream classification tasks.
- We apply our proposed framework on two CXR images classification tasks, comprising pneumonia and tuberculosis diagnosis, with limited datasets and achieve state-of-the-art results. The regions of lesions can also be visualized for analysis.

The rest of the paper is organized as follows. Section 2 presents some related work. Section 3 depicts the detail of our framework. Section 4 describes the setting of our experiments and the results. Section 5 concludes.

2. RELATED WORK

Deep-learning-based CXR Analysis. Many researchers have been dedicated to automated diagnosis with medical images. Rajpurkar et al. [1] proposed CheXNet for pneumonia detection, which is a 121-layer convolutional neural network. Wang et al. [11] trained a Deep CNN for weakly-supervised multi-label image classification. Liu et al. [2] used the information captured by contrasting positive and negative images to localize the pathologies and made predictions on those regions. Kermany et al. [12] followed a framework of transfer-learning to classify CXR images for pneumonia diagnosis. Filho et al. [3] trained various CNN architectures and attempted to diagnose tuberculosis by CXR classification.

Unsupervised Pre-training. The unsupervised pre-training paradigm has been prospering in the field of natural language processing [6]. Unsupervised pre-training is critical for various downstream supervised tasks for the reason that predictable representations may benefit classification tasks in terms of data efficiency [13]. Self-supervised learning methods have been widely used for representation pre-training [7–10]. Among them, the auto-regressive model is a common way of self-supervised learning, which predicts a masked part of pixels given other visible parts. Recent auto-regressive methods [7, 8] have been proposed and achieve state-of-the-art performance in downstream supervised tasks. Another approach of self-supervised representation learning is training in a contrastive way [9, 10], i.e. applying various augmentations on images and forcing the model to predict whether two augmented images come from the same original image.

3. METHODS

3.1. Motivation

There are mainly two statistical properties of modern CXR images: 1. The high similarity among CXR images; 2. The low signal to noise ratio of significant information for diagnosis. Inspired by the recent prosperity of contrastive self-supervised learning [7, 9, 10, 13], we can exploit the two properties and learn representations without most of the insignificant information similar among images. The previous frameworks [9, 10] always randomly crop two patches out of the original image and resize them to the original size for contrastive training. However, the random cropping may result in two different areas cropped out from the same image which indicate completely different information and not have much value to be contrasted. For avoiding this problem, we propose the C3 schema so that the two cropped patches are guaranteed to overlap each other. What’s more, considering that the rotation around the center of the image may affect the cropped area, we adjust the rotation to match the cropping operation.

3.2. Overall Pre-training Framework

The overall framework we propose follows the canonical two-stage paradigm illustrated in Fig. 1, in which the crucial part is the pre-training stage, which can be zoomed in and illustrated in Fig. 2. In this pipeline, we introduce the C3 schema and a corresponding rotation when we train the feature extractor. Multiple unlabeled images will be input into the pipeline and each image will be transformed into two independent branches.

If the batch size is n , we will eventually get $2n$ patches after all transformations. All these $2n$ patches will be sent into the network in pairs. This prediction process follows the way of SimCLRv2 [10]. The feature extraction CNN is a ResNet-50 [14] network variant according to ResNet-D [15] and selective kernels [16] are introduced. The CNN is followed with

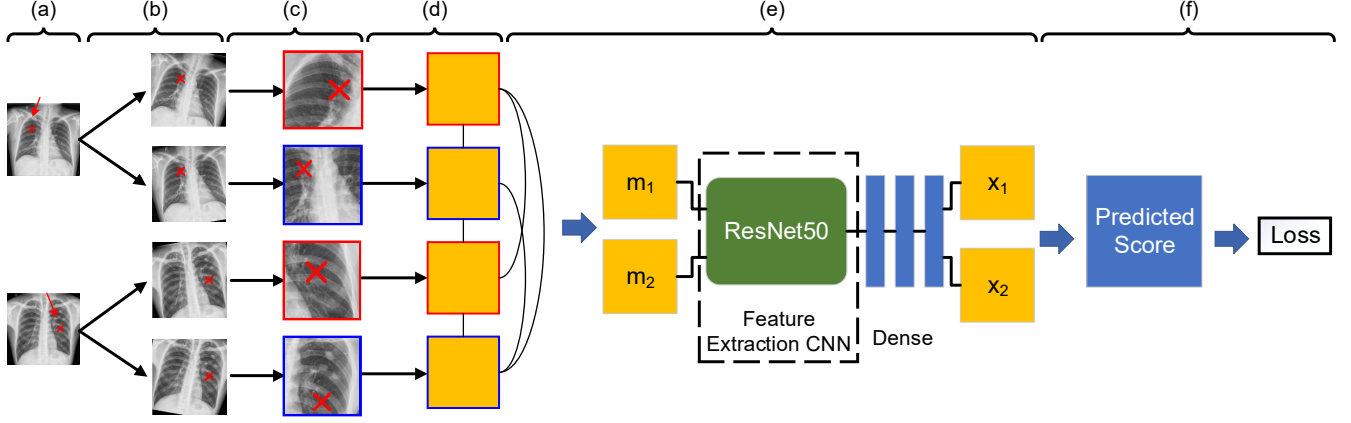


Fig. 2. Take the input batch size of 2 as an example. (a) For any input image, a particular operating point will be chosen randomly; (b) The image will rotate around the operating point; (c) An area around the operating point will be cut out as a patch; (d) Other augmentations will be applied sequentially; (e) The resulting patches will be sent into the network in pairs and their respective projected vectors will be output; (f) Training loss will be calculated according to the predicted scores which depict the relative similarities between projected vectors.

3 dense layers to project the semantic representation vectors into a lower dimension. Let m_i denote the i th patch sent into the network and x_i be the corresponding projected vector. Let x_{2h-1} and x_{2h} be the two vectors from the h th input image. The corresponding predicted score of pairs composed of x_i and x_j is calculated by a softmax function and given by:

$$\mathcal{P}_{i,j} = \frac{\exp(x_i \cdot x_j / \tau)}{\sum_{\substack{k=1 \\ k \neq i}}^{2n} \exp(x_i \cdot x_k / \tau)}, \quad i \neq j \quad (1)$$

where τ is a scalar indicating the temperature of the softmax function and the inner product can be treated as the similarity between two vectors. The training loss can then be represented by:

$$\mathcal{L} = - \sum_{\substack{i,j=1 \\ i \neq j}}^{2n} \mathbb{1}_{\{[i/2]=[j/2]\}} \log \mathcal{P}_{i,j} \quad (2)$$

where $\mathbb{1}_{\{\cdot\}}$ represents the indicator function. Note that during supervised fine-tuning, the three dense layers in the pre-training process are rejected and the weights of the feature extraction CNN are fixed.

3.3. Center-specific Cropping Constraint(C3)

During pre-training, we need to constrain the random cropping to guarantee the two cropped areas from the same image overlap each other. Fig. 2(c) shows an example of this operation. We decide the cropped area by a bounding box denoted by its upper-left corner P_{ul} and bottom-right corner P_{br} . Assuming that we have an operational center point $O(x_c, y_c)$ and that the size of images is (W, H) , we randomly choose P_{ul} from the upper-left side of O , i.e. $\{(x, y) | 0 < x < x_c, 0 < y < y_c\}$, and P_{br} from the bottom-right side of O ,

i.e. $\{(x, y) | x_c < x < W, y_c < y < H\}$. In this way, if we generate multiple cropped patches, it can be guaranteed that a certain area around the center point is shared among them.

The choosing of O is random in the 2-D space of the image, but the distribution is critical. If the point O is near the edge of the image, the common information between two cropped areas is most likely from the margin of the image, which always doesn't have much value in our tasks. Therefore, the uniform distribution may not be a good choice, so we use a diagonal gaussian distribution for sampling point O .

The coordinates of O can be obtained by sampling scale factors with respect to the size of the images. The probability distribution of the factors s of point O is defined by $s \sim \mathcal{N}(\mu, \Sigma)$ where

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \quad (3)$$

and

$$O = s \circ \begin{bmatrix} W \\ H \end{bmatrix} = \begin{bmatrix} s_x W \\ s_y H \end{bmatrix} \quad (4)$$

The value of μ is fixed at $[0.5, 0.5]^T$. The scale factors s will be clipped into the range $[0, 1]$ and of course, the sampled point O will be round into the integer grid.

In addition, unlike during pre-training, we use a different constraint during supervised fine-tuning. The overall information of an image should be observed when we train the classification head during fine-tuning. So the bounding box for cropping is determined by randomly choosing P_{ul} from $\{(x, y) | 0 < x < 0.25W, 0 < y < 0.25H\}$ and P_{br} from $\{(x, y) | 0.75W < x < W, 0.75H < y < H\}$ to let our feature extractor pay more attention to the overall information. As a more specific control, a lower bound of the area of the cropped patches, denoted by A_l , is introduced.

3.4. Center-specific Rotation

The angle changes may be introduced when taking X-ray films, so rotation should be considered as one of the data augmentations. The center of rotation is $O(x_c, y_c)$. The angle of rotation ϕ is a randomly chosen value between $[-\theta, \theta]$ and the direction is counterclockwise. Note that the bilinear interpolation should be used to prevent sawtooth artifacts.

To dodge the issue that applying rotation may result in black edges at four corners of the image, we should: 1. apply rotation before the corresponding cropping; 2. check the cropped patch whether one of its corners is within a produced black edge. If the randomly cropped result doesn't meet the requirement that no corner within a black edge exists, another attempt will be made. Sometimes, even a tiny rotation will inevitably introduce black edges (say, when the operational center point is near the edge of an image), so the rotation will be canceled in this case.

If the point of the corner to be checked is (x, y) , we can imaginarily rotate (x, y) around the center (x_c, y_c) in the opposite direction to the previous rotation and obtain (x', y') , which is given by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (5)$$

Then we examine whether $0 \leq x' \leq W$ and $0 \leq y' \leq H$. The point will be accepted if the result is true and rejected otherwise. If all of the four corners of the cropped area are accepted, the cropped result will be accepted. Note that the rotation is canceled during fine-tuning.

3.5. Lesion Localization

A heatmap can be generated by Grad-CAM [17], which is trying to find the weights of a series of feature maps in a specific convolutional layer. The gradients of the score for a specific class c with respect to the activations of the feature map A_k are calculated firstly. Next, the gradients of each feature map are global-average-pooled to obtain the corresponding weight w_k of this specific feature map. The weighted sum of the feature maps followed by a ReLU function, i.e. $ReLU(\sum_k w_k A_k)$, will be treated as our final heatmap.

Our interested class c is the positive class and we choose the output of the last convolutional layer in our feature extractor as our feature maps to generate the heatmap.

4. EXPERIMENTAL RESULTS

4.1. Datasets and Settings

For our pneumonia diagnostic task, we conducted the experiments on the dataset ChestXRay2017 [18] includes 5,856 CXR images in total. 624 images are separated as the test set and another 16 images as the validation set. For the tuberculosis task, we mixed two public datasets made by the

Table 1. Comparison Between Different Models

Task	Model	AUC	#Labels
Pneumonia	Kermany et al. [12]	0.968	5232
	ours	0.980	1536
Tuberculosis	AlexNet [3]	0.814	892
	GoogleNet [3]	0.812	892
	ResNet [3]	0.785	892
	ours	0.923	448

U.S. National Library of Medicine [19] contain 662 and 138 labeled images respectively. 160 images are separated as the test set and another 16 images as the validation set. In each of the tasks, we shuffled the training set and randomly extracted subsets from it with different sizes for the subsequent supervised fine-tuning.

All of the CXR images except for those in test sets and validation sets had also been mixed without labels for unsupervised pre-training. All the input images used in the experiments are resized into the size of 224×224 .

During pre-training, we chose $\sigma = 0.2$, $\theta = 10^\circ$ and $\tau = 0.1$ and trained the network with a batch size of 16. Other augmentations include flip, color jitter, and gaussian blur. Weights of the feature extraction CNN and three dense layers were initialized with weights pre-trained on ImageNet.

The weights of the classification head are initialized randomly. During fine-tuning, the lower bound of the cropped area, i.e. A_l , is one of the hyper-parameters that vary with different tasks. As for our pneumonia dataset, we chose $A_l = 0.50$; And for the tuberculosis dataset, we chose $A_l = 0.65$. The training process was conducted with a batch size of 32.

The Adam optimizer was utilized for all of the training processes and one NVIDIA 2080TI GPU is used.

For measuring the performance of our model and excluding the influence of different thresholds of binary classification, we used three metrics: 1. AUC score [20], which is the proportion of the area under the ROC curve; 2. Max F1 score, which is the maximum achievable value of F1 score; 3. the number of labeled samples used for training.

Some more detail about the model training can be found in supplemental materials.

4.2. Comparison with State-of-the-art Methods

The pre-trained feature extraction CNN was embedded in the task-specific classification pipeline and two different tasks were conducted. The results shown in TABLE 1 compared our models with the baseline methods comprising the pneumonia classification model trained by Kermany et al. [12] and the tuberculosis models trained by Filho et al [3].

As shown in Table 1, our framework performs better on both tasks compared to the previous methods which use com-

Table 2. Ablation Studies

Task	Schema	AUC	Max F1
Pneumonia	Random	0.977	0.946
	C3(Uniform)&Rotation	0.976	0.946
	C3(Gaussian)	0.975	0.944
	C3(Gaussian)&Rotation	0.980	0.950
Tuberculosis	Random	0.918	0.845
	C3(Uniform)&Rotation	0.913	0.843
	C3(Gaussian)	0.922	0.834
	C3(Gaussian)&Rotation	0.923	0.855

pletely supervised learning or transfer learning based on CNN initialized with weights trained on ImageNet. We show that the model can achieve good results although only trained on a relatively small dataset. About 30% of the labels in the pneumonia task and 50% of the labels in the tuberculosis task are used compared to the previous works [3, 12].

4.3. Ablation Studies

We conducted experiments with several different ways of cropping during pre-training: 1. completely random cropping; 2. our proposed C3 and rotation with center chosen through uniform distribution; 3. C3 with center chosen through Gaussian distribution and without rotation; 4. our final schema, using Gaussian distribution and rotation. The sizes of the labeled training subsets of two tasks are 1536 and 448 respectively.

We can see that both Gaussian sampling and center-specific rotation are necessary. The two changed feature extractors have suffered varying degrees of performance degradation. The performance of our proposed cropping schema performs significantly better than the random case in both tasks, either from the aspect of the AUC score or the F1 score, while the other two degraded schemas haven't shown obvious advantages comparing to the random situation.

4.4. Influence of the Number of Labeled Samples

In this section, 6 training subsets per task with different numbers of images were used for supervised fine-tuning. The results are shown in Fig. 3.

We can see the trend that the overall performance tends to enhance when the number of labels increasing. The figures illustrate that the model can achieve an acceptable consequence although there are only hundreds of labeled images. In the case of pneumonia diagnosis, it is clear that the overall performance tends to keep steady after the number of labels is over 1536. As for the tuberculosis task, the AUC score is increasing continuously due to the total number of labels is

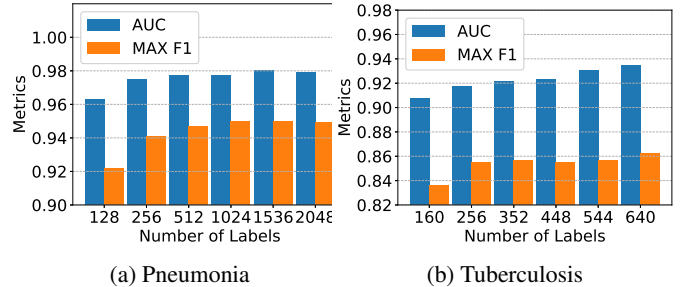


Fig. 3. The performance of the model when the number of labels increasing.

relatively small and the Max F1 score also improves notably when the number of labels reaches 640.

4.5. Visualization of Lesion Areas

The heatmaps generated by Grad-CAM [17] are shown in Fig. 4. In both pneumonia and tuberculosis task, the model indeed locates the lesions and recognizes the critical information.

5. CONCLUSION

Our empirical data shows that in the case of CXR images, contrastive self-supervised training is effective although the number of training samples is relatively small. To enhance the effect of the previous semi-supervised approach and adapt it to CXR classification tasks, we propose our C3 schema with center-specific rotation. The knowledge learned by the feature extractor comes from the comparable information of two different patches, which lets the model know what the semantically synonymous texture is. The resulting framework can achieve an acceptable performance by training on datasets that may only comprise thousands of shared task-agnostic unlabeled images and hundreds of task-specific labels. We can also visualize the location of lesions by generating gradient-based heatmaps, which can annotate the approximate region of lesions so that some guidance information can be provided to radiologists in real clinical practice. Our future work will concentrate on transferring this framework to other diseases other than pulmonary ones. Other modalities of medical images will also be considered, such as MRI and CT.

6. REFERENCES

- [1] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpan-skaya, et al., “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” *arXiv preprint arXiv:1711.05225*, 2017.
- [2] J. Liu, G. Zhao, Y. Fei, M. Zhang, Y. Wang, and Y. Yu, “Align, attend and locate: Chest x-ray diagnosis via

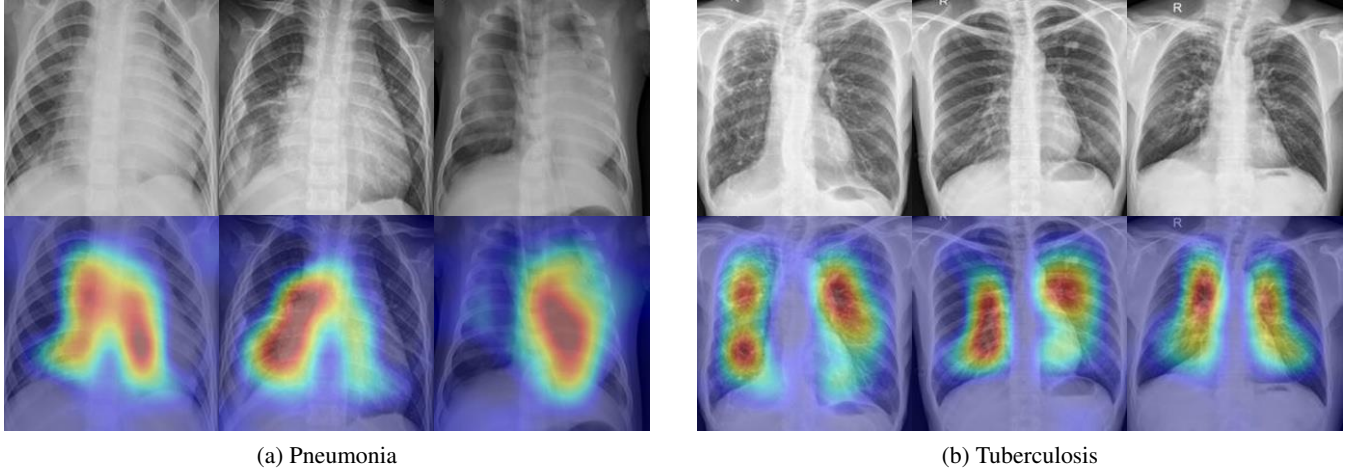


Fig. 4. The heatmaps marking the lesion areas. For each task, we choose 3 examples. In both (a) and (b), the upper row shows the input CXR images and the bottom row shows the corresponding heatmaps overlapping the original input images.

- contrast induced attention network with limited supervision,” in *ICCV*, 2019, pp. 10632–10641.
- [3] M. E. Colombo Filho, R. M. Galliez, F. A. Bernardi, L. L. de Oliveira, A. Kritski, M. K. Santos, and D. Alves, “Preliminary results on pulmonary tuberculosis detection in chest x-ray using convolutional neural networks,” in *ICCS*. Springer, 2020, pp. 563–576.
 - [4] A. Narin, C. Kaya, and Z. Pamuk, “Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks,” *arXiv preprint arXiv:2003.10849*, 2020.
 - [5] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *ECCV*. Springer, 2014, pp. 818–833.
 - [6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
 - [7] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
 - [8] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, P. Dhariwal, D. Luan, and I. Sutskever, “Generative pretraining from pixels,” in *ICML*, 2020.
 - [9] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al., “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv preprint arXiv:2006.07733*, 2020.
 - [10] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, “Big self-supervised models are strong semi-supervised learners,” *arXiv preprint arXiv:2006.10029*, 2020.
 - [11] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *CVPR*, 2017, pp. 2097–2106.
 - [12] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, et al., “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.
 - [13] O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord, “Data-efficient image recognition with contrastive predictive coding,” *arXiv preprint arXiv:1905.09272*, 2019.
 - [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.
 - [15] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in *CVPR*, 2019, pp. 558–567.
 - [16] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” in *CVPR*, 2019, pp. 510–519.
 - [17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *ICCV*, 2017, pp. 618–626.
 - [18] D. Kermany, K. Zhang, and M. Goldbaum, “Labeled optical coherence tomography (oct) and chest x-ray images for classification,” *Mendeley data*, vol. 2, 2018.
 - [19] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wang, P.-X. Lu, and G. Thoma, “Two public chest x-ray datasets for computer-aided screening of pulmonary diseases,” *Quantitative imaging in medicine and surgery*, vol. 4, no. 6, pp. 475, 2014.
 - [20] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.