Chest X-ray Disease Diagnosis

https://mediaspace.illinois.edu/media/t/1_i6sl0f2n

Pun, Long
longpun2, CS598 LHO
University of Illinois, Urbana-Champaign
Macau, China
longpun2@illinois.edu

Deng, Jesse
jessedd2, CS598 LHO
University of Illinois, Urbana-Champaign
Chicago, USA
jessedd2@illinois.edu

Imtiaz, Asad Bin
Aimtiaz2, CS598 LHO
University of Illinois, Urbana-Champaign
Bern, Switzerland
aimtiaz2@illinois.edu

Yiming, Gu
yimingg7, CS598 LHO
University of Illinois, Urbana-Champaign
Shanghai, China
yimingg7@illinois.edu

***Abstract*— Lung diseases are common throughout the world, posing a huge risk especially to low- and middle-income countries. Timely detection is essential for medical treatment and prevention of death. Chest X-ray (CXR) is one of the dominant and non-invasive imaging methods for identifying lung diseases. However, it takes years of training and expertise for doctors to correctly diagnose and classify the scans. With the recent advances in deep learning, newer machine learning architectures such as convolutional neural networks (CNN) have successfully been applied to diagnose CXR with a high degree of accuracy in detecting lung diseases. In this paper, we propose an architecture which utilizes the state-of-the-art deep learning techniques to identify 14 common lung pathologies from chest X-ray images, with attention mechanism introduced to improve classification accuracy and enhancing interpretability and locality of the area of interest.**

Keywords—X-ray; medical imaging; deep learning; convolutional neural network; lung diseases

I. INTRODUCTION

Respiratory diseases impose an immense worldwide health burden. For example, about 65 million people suffer from chronic obstructive pulmonary disease (COPD) and 3 million die from it each year [1], making it the third leading cause of death worldwide. A lot of global effort has been exerted on their prevention and cure. Traditionally, the chest x-ray (CXR) is the most performed diagnostic examination for detecting lung diseases, such as pneumonia and emphysema. However, reading CXRs rely on professional knowledge and manual observations, which may take many years of training and practicing for a doctor to make correct judgement. Even so, it is not uncommon for doctors to make mistakes or disagree with each other's judgement due to the difficulty of the task, and do not scale well with the number of images. Therefore, it is beneficial to develop automatic, accurate and objective CXR classification methods.

In recent years, noticeable progress in deep learning has led to many success stories in medical image analysis, for example lesion detection and disease classification. In this proposal, we propose a deep learning framework which can accurately classify various lung diseases based on CXR images as well as able to localize the area of interest. The goal is to build a tool which serves as a robust and objective second opinion for medical practitioners.

II. LITERATURE REVIEW

A lot of previous research has been conducted on classification of lung diseases by CXR. For example, Wang *et al.* [2] presented a chest X-ray database (NIH Chest X-ray) of over 100K chest X ray images and evaluated four classic CNN architectures for detecting presence of multiple pathologies. The research provided accurate predictions for locality of disease areas on the images. On the other hand, Irvin & Rajpurkar *et al.* [3] presented another CXR dataset (CheXpert) with over 200,000 images and investigated different approaches to train CNN with reasonably high accuracy for prediction of the 14 observations such as the presence of diseases and support devices etc.

Based on the work of [2], a lot of research attempt to improve prediction accuracy by substituting the deep learning models with advance ones. Rajpurkar *et al.* [17] replace the model in [2] by a 121-layer Densely Connected Convolutional Networks (DenseNet) [11]. Bharati *et al.* [4] introduces *VDSNet*, an architecture composed of VGG16 [10] and capsule network (CapsNet) [5]. It also incorporates other information such as

patient's age and gender in classification and claims to overperform the benchmark classification models in [2]. Herlihy *et al.* [6] develop a multi-label CNN using the network architectures from deep learning community such as Residual Network (ResNet) [16] and DenseNet as baselines to detect selected diseases, yielding better performance compared to [4]. Since then, a lot of research was done on utilizing many state-of-the-art deep learning models as drop-in replacement in classification. A comprehensive comparison on the performance of different models and hyperparameters on classification can be found in [9].

On the other hand, several other research works focused on localizing the areas of interest in addition to classifying diseases on whole image. The rationale behind is that locating whereabouts of the disease would guide the classifier to pay better attention to areas of high interest instead of looking at the whole image, thus enhancing accuracy. In addition, the localization stage could provide insights in the form of heatmap which is desirable for the sake of explainability and interpretability. This idea was investigated in depth by [4] and [6]. One line of research is the Class Activation Map (CAM) [29], which can locate the region of interest which leads to a specific classification prediction. Some example algorithms in this line includes GradCAM [25] and ScoreCAM [26]. In the field of CXR classification, Preechakul *et. al.* [7] proposed an architecture named Pyramid Localization Network (PYLON), an extension of CAM, which claimed to produce a good quality heatmap to guide subsequent classification. Another line of research is based on the attention mechanism, which does not locate the area of interest for a specific prediction, rather questions on the whereabouts of the input image that a trained deep learning classification model considers to be more important for a prediction. Such attention-guided mechanism is illustrated in [27]. Based on this idea, Guan *et al.* [8] introduces an attention-guided CNN (AG-CNN) model which trains a model focusing on whole image along with another model which focuses on a only at a local portion, and finally ensembles these two models to give significantly accurate predictions.

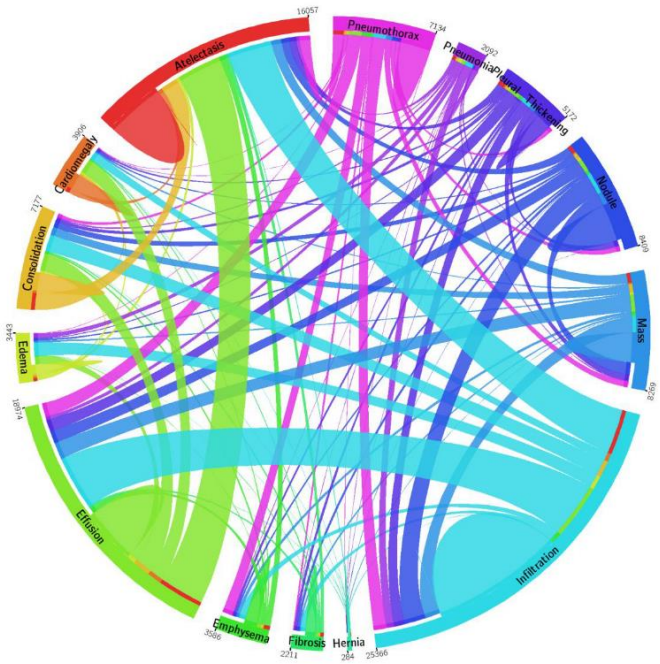

Figure 1. *The circular diagram shows the proportions of images with multi-labels in each of 14 pathology classes and the labels' co-occurrence statistics. Courtesy of [2].*

III. DATA

There are several open Chest X-ray (CXR) datasets available for academic and research use, such as NIH Chest X-ray Dataset [2], CheXpert Dataset [3] and JSRT Database [13]. The NIH Clinical Center CXR (ChestXray-14) Dataset consists of 112,120 frontal-view X-ray images of 30,805 unique patients, with text-mined fourteen disease image labels mined from the associated radiological reports using natural language processing. Each image can have multi-labels, i.e., none or multiple diseases may be spotted in one image. The text-mined disease labels are expected to have accuracy >90%. The fourteen diseases are: *Atelectasis*, *Consolidation*, *Infiltration*,

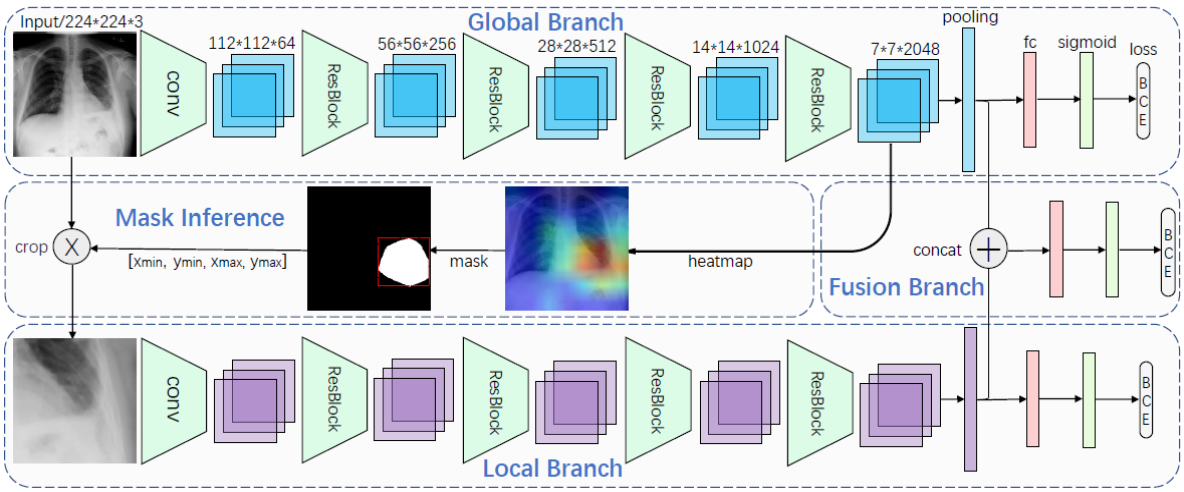

Figure 2. *Model architecture of [8] (Figure 2). Both the global and local branches use ResNet-50 as backbone network. Chest X-ray images are fed through the global branch. For each image and disease combination, a heatmap and the corresponding bounding box is produced from the feature map of the global branch, from which a local section of the original image is cropped. The local image serves as input to the local branch. Output vectors after pooling layers of the two branches are concatenated to form the fusion branch which consists of a fully connected classification layer to compute the final prediction. Image Courtesy of [8].*

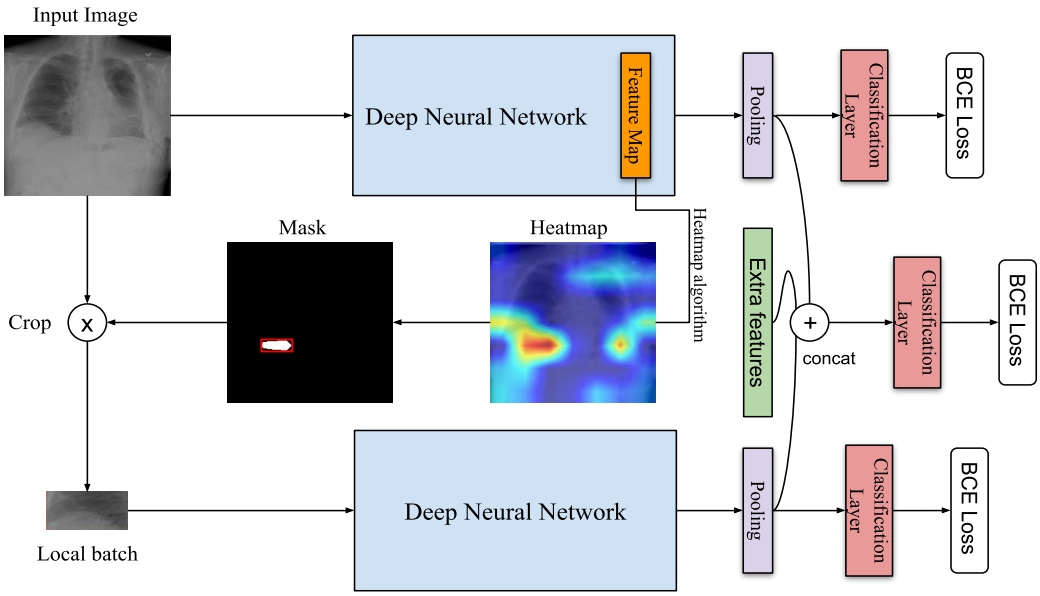

Figure 3. *Proposed model architecture. The global and local branches may be a standalone or cascaded models e.g., DenseNet + CapsNet etc.. In addition to the output vectors after pooling layers of the two branches, extra features such as the age of patient are concatenated to form the fusion branch which computes the final prediction.*

Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural Thickening, Cardiomegaly, Nodule, Mass, and Hernia. This dataset contains a reasonable number of good-quality CXR images and supplementary information and is therefore selected in this project. Figure 1 shows the distribution and co-occurrence statistics of the diseases in the dataset.

The NIH Chest dataset also comes with two metadata files. The *Data_Entry.csv* file contains information of each image, including the disease label, patient's ID, follow-up number, age and gender, image view position (Anterior-Posterior/Posterior-Anterior), original image width and height, and pixel spacing in x and y axis.[1]

The other metadata file is *BBox_List.csv* containing the ground truth box coordinate information that could also be used for locating the disease(s) in each image. However, it only covers eight out of the fourteen diseases and a total of 985 images, which is very modest compared to the total >100K images.

On the other hand, a *train_val.txt* and *test.txt* files are provided which split the images into train/validation and test set respectively. For the sake of reproducibility and alignment with

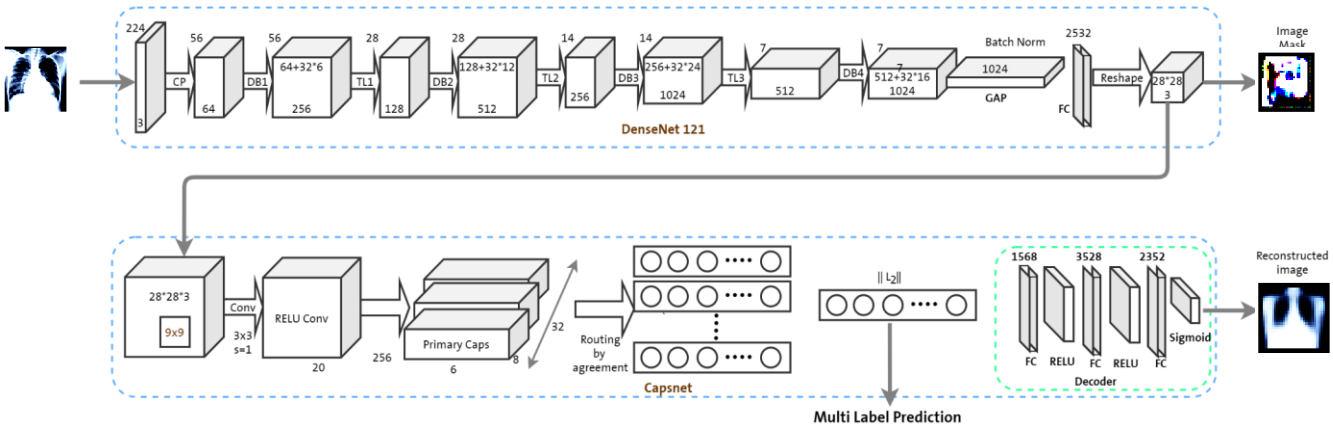

Figure 4. *Model Architecture of CapsNet augmented to DenseNet121 with Multi-Label prediction from CapsNet. Image masks are generated from DenseNet which are fed into CapsNet to regenerate the Original Image representation capturing the essential image features such as shape, contrast and orientation as well as making the prediction.*

[1] The *Data_Entry.csv* was updated in 2020 with some error corrections. We always refer to this update version in the paper.

previous research, we also split our data according to these files in this project.

IV. APPROACH

We formulate our problem as a multi-label classification task, with an aim to classify the 14 diseases from CXR images where each image may have no, one or more than one disease. Moreover, given that there is a class imbalance problem with all diseases (i.e., a specific disease usually exists in much less than a half of the images. For example, there are only 284 images labelled with Hernia in the whole dataset. See Figure 1), we use the Area Under Receiver Operating Curve (AUROC) as the evaluation metric. This choice is in alignment with most previous research works e.g., [2][8][17][19].

The overall model architecture is based on the works of [8], named AG-CNN. Fig. 1 shows this architecture. It consists of a "*global*", a "*local*" and a "*fusion*" branch. On the high level, the global branch looks at the whole CXR image, while the local model looks only at a segmented portion of the image. The fusion branch consolidates the two branches to give the final prediction.

The global branch consists of a base model (ResNet-50 in [8]) which take the CXR images as input and performs multilabel prediction. In addition, the intermediate output before pooling layer ('feature map') is extracted and fed through some attention mechanism (e.g. Grad-CAM [14], or attention-guided mask [27]) to produce a heatmap, indicating the location(s) on which the global branch concentrates while performing classification. **Error! Reference source not found.** gives an overview of how Grad-CAM works.

The attention guide is computed as follows: given a global image, let $f_k(x, y)$ represent the activation of spatial location (x, y) in the k-th channel of the output of the last convolutional layer, where $k \in 1,2, \dots, K$, for example, K = 2,048 in ResNet-50. We first take the absolute value of the activation values $f_k(x, y)$ at position (x, y). Then the attention heatmap H is generated by counting the maximum values along channels,

$$H(x, y) = \max_{\mathrm{k}}\left(\left|\mathrm{f}^{\mathrm{k}}(x, y)\right|\right), k \in \{1, \dots, K\} \qquad (1)$$

where the value of H indicates the importance of the activation in classification. The heatmap is then transformed into a binary mask via intensity thresholding.

For the local branch, the input images are batches cropped according to the bounding box. Another deep neural model with a classification layer would then classify the image to be having disease(s) or not.

Lastly, we ensemble the global and local models together to build a final neural network, which produces the final prediction. The prediction results can then be compared to other benchmarks such as [9].

With the baseline architecture in mind, we contribute to the following innovations:

1. In addition to the ResNet-50 and DenseNet-121 models in [8], we experiment other model architectures such as ResNeXt [15] with an aim to boost prediction accuracy. Moreover, a heterogeneous architecture, which utilizes different model types for the global and local branch, is experimented.
2. Instead of utilizing one standalone base model, we experiment a cascade of a base model and CapsNet, such as the architectures proposed in [5] and [21]. The way was to pre-train a CapsNet network on NIH Xray Images [9] with image reconstructions enabled to capture essential image features, and then to augment this capsule network with our base network.
3. Accurate heatmap is the key to the performance of the local branch. We evaluate multiple class activation algorithms which produces a distinct heatmap for each image and disease combination and compare to image-level attention-based heatmap.
4. Currently, the predictions are solely based on CXR images. In our model, additional information including patient's age, gender, and the view position (anteroposterior or periapical) of imaging are incorporated as input features to the fusion branch as well.

Layout of the proposed architecture is shown in Figure 3.

V. EXPERIMENT SETUP

In this section, we present the hardware and software utilized as well as the implementation details of this project.

A. Hardware

For this project, due to the large volume of data in the format of images, much computational power is required to achieve acceptable runtimes and timely results. We utilize a combination of local workstations and cloud computing resources (Google and Amazon cloud) for model training.

B. Software

Python 3 is the primary programming language used. Other major software packages used in this project include: PyTorch (v 1.8.1), Torchvision (v 0.9.1), Scikit-Learn (v 0.24.1), Pillow-SIMD (v 7.0.0), OpenCV (v 4.5.1.48) and Scikit-Image (v 0.18.1). Ray Tune is utilized for hyperparameter search, though no experiment was performed due to time and computational resources constraint.

C. Overall Architecture

In the training of the global branch, we tested several different models as backbone: ResNet-50, ResNeXt-50, ResNeXt-101 [15], DenseNet-121, CapsNet, and MobileNet [28]. The weights of each model (except CapsNet) are pre-trained with ImageNet [22] and the output size of the final classification layer is reshaped to match with the number of diseases (14). Out of the 112,120 CXR images available, 82,736 (74%) and 3,788 (3.4%) images are randomly chosen as training and validation sets where no patient exists in both sets.

The remaining 25,596 (23%) images (listed in *test.txt*) serve as test set.

All images are resized to 224x224 pixels to fit the input width and height of the pre-trained models. Data augmentation techniques are applied to enhance robustness of the model, for instance a random horizontal flip of probability of 0.5 and rotation of ± 15 degrees. After that, the pixel intensity in each image is scaled to a range between [0,1] before normalized by the mean and standard deviation of ImageNet dataset on which the models are pretrained on. To accommodate the input dimensions of pre-trained models, the originally single-channel grayscale images are broadcasted to 3-channel.

As **Error! Reference source not found.** indicated, the training images are fed into each of the global branch in batches of 32/16/8 images depending on the memory available in the machine. The feature map (the layer immediately before the last pooling layer) of the global branch is extracted and a heatmap is produced via either an attention-guided algorithm or an algorithm of the CAM family which gives each pixel in the image an "importance" between [0, 1]. An image mask is computed by thresholding the heatmap by 0.7 and keeping the largest connected area.

A local patch of the original image is cropped according to the mask and resized according to the expected input size of the local branch. The output after the pooling layers of global and local branches are concatenated together with the age, gender and view position of the patient/image and serve as input to the fusion branch, which calculates the final prediction labels.

The loss function of all three branches is binary cross-entropy (BCE Loss), weighed by the inverse of prevalence of positive samples within each disease. The optimizer chosen is AdamW [23] with an initial learning rate of 1e-4, betas = (0.9, 0.999) and weight decay of 0.01. In addition, a learning rate scheduler monitors the validation loss and multiplies the learning rate by 0.1 if there is no improvement for 3 consecutive epochs.

In the first 4 training epochs, only the classification layers of the global and local branches are trainable (the fusion branch is always trainable since it has only one fully connected layer). On epoch 5, all learning rates are multiplied by a factor of 0.2 and starting from this epoch, one last untrainable block in each model is made trainable ("unfreeze") every epoch until all parameters become trainable. The whole architecture is trained for a maximum of 100 epochs or until there is no improvement in the validation loss in the fusion branch for more than 10 epochs after the previous best validation loss was logged. We treat the models when the best validation loss in fusion branch was achieved as the final model and evaluate the test set on it.

VI. RESULTS

We perform extensive experiments on the choice of model architecture and algorithms etc. and the results are presented in this section.

A. *Overall performance*

We experiment a total of 12 configurations of baseline models, 6 of which with identical model type in both global and local branch and 6 of which are different. The results and comparison to previous research are summarized in TABLE I and **Error! Reference source not found.**.

TABLE I shows that we do not achieve a better result in terms of average AUROC compared to the best previous works (0.8067 vs. 0.8710). However, there are 4 diseases in which we outperform all previous works: *Infiltration*, *Nodule*, *Pneumonia*, and *Pneumothorax*. This suggests that we have reached different local minima to other works.

We also experiment settings in which different model types are used in global and local branch. The result is summarized in **Error! Reference source not found.**. Interestingly, the best average AUROC is achieved on the global branch and outperforms the same model in TABLE I. This may be a result of different batch sizes used and the epoch on which the training stopped early. Overall, a heterogeneous model type does not seem to lead to significant improvement.

Finally, we evaluate the performance of an unweighted soft-voting ensemble of three of our trained models with ResNet-50, ResNeXt-50 and ResNeXt-101 as backbone. This ensemble achieves an average AUROC of **0.8170**, which is the best among all our attempts. It shows that model ensemble is a viable approach to boost performance.

B. *CapsNet*

Capsule Networks can fetch spatial information to avoid losses with pooling operations. This is achieved by generating vectors in the direction of orientation of images and routing by agreement where the capsule with largest predictions gets its weightage increased, while the others get their impact reduced. This routing by agreement produces better results compared to Max-pooling operations. The original CapsNet paper [5] performed prediction on MNIST handwritten digits dataset [20]. Our implementation of CapsNet achieved >99% AUROC for the MNIST images. The MNIST handwritten digit images are however very small in size, with height and width of 28x28 pixels. If CapsNet is trained on large images (such as ChestX-Ray images which are 1024x1024 pixels in size), due to the

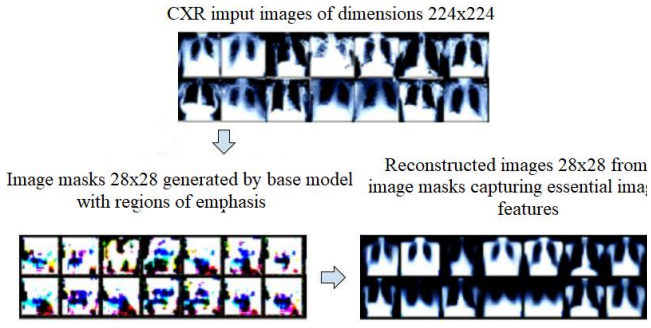

Figure 5. CXR images on dimensions 1024x1024 are downscaled to 224x224 and processed by DenseNet to produce image masks of 28x28. These image masks are in turn processed by CapsNet to reconstruct original images as well as to make the predictions.

routing by agreement, the memory requirements would increase exponentially, and the required computation time would make model training impractical. In our experiment, it was not feasible to finish training within a reasonable time even with a drastically downscaled version of input images of 56x56 pixels. Therefore, we downsized the CXR images to 28x28 and pretrained the CapsNet on them, in hope of concatenating it after one of our baseline models and getting better accuracy. The results of our pre-training were similar to that of [4], with CapsNet taking a lot of time to train and was slow to converge, giving a test ROC of **0.5910** on the input dataset.

The CapsNet alone did not achieve good accuracy on prediction as shown in Table I. Inspired by [21], it was decided to attach the Capsule Network succeeding our base neural network, where the output from the fully connected layer of the base model as in Figure 4. The image is reshaped to a dimension of 28x28, serving as input mask to the Capsule Network as depicted in **Error! Reference source not found.**. The image is reshaped to a dimension of 28x28, serving as input mask to the Capsule Network and used to reconstruct original image as depicted in **Error! Reference source not found.**.

In the first step, the CapsNet was trained on downsized NIH images of dimensions of 28x28, with reconstruction to capture essential image features. In the second step, this pre-trained CapsNet was attached after the base network to improve the classification result, similar to COVID-Caps implementation in [21]. In this case, we used a pre-trained DenseNet-121 as the base neural network. The intermediate output of DenseNet served as an image mask to visualize which specific areas of image the model is currently emphasizing on. The CapsNet then tried to reproduce the original image from this mask while making a prediction. Specifically, CapsNet concatenated after DenseNet-121 as in Figure 4 resulted in best test ROC of **0.6942** and that after Resnet-50 **of 0.6844.** The results were no better than DenseNet-121 or ResNet50 alone.

C. Disease Localization

To localize the diseases, we generate heatmap using three CAM algorithms, namely Grad-CAM [14], Grad-CAM++[25] and XGrad-CAM [30]. The intensity of heatmap is scaled to a range of [0,1] and a threshold of 0.7 is applied to generate the binary mask.

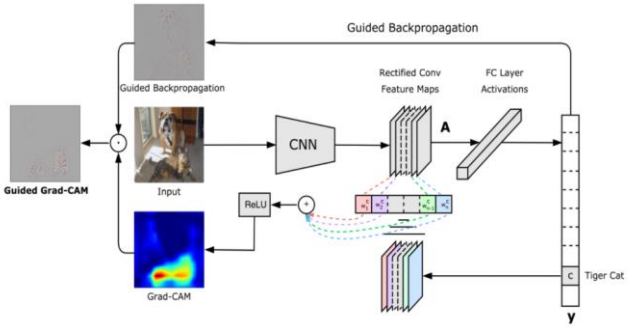

Figure 6. O*verview of Grad-CAM. Courtesy of gradcam.cloudcv.org.*

To evaluate the generated heatmap, we used the Intersection over Union (*IoU*) as the evaluation metric. *IoU*, also known as Jaccard index, is the most used metric to compare two arbitrary shapes. Specifically, the *IoU* evaluates the accuracy of a prediction object against the ground truth object, which in this case are two bounding boxes, the ground truth box provided and predicted bounding box generated according to the heatmap, by using Formula (2):

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \qquad (2)$$

The Area of Overlap and Area of Union are computations based on the bounding boxes, which are illustrated as in **Error! Reference source not found.**. In our experiment, we compute the IoU between the bounding boxes generated by our heatmap and the ground truth. Table III shows the results.

VII. DISCUSSION

A. Overall performance

Comparing our results with those of the state-of-the-art, there is significant space for improvement. Due to memory constraint, we are only able to train on a relatively small batch size of maximum 32, where [9] concluded that a batch size of 256 of above gives the best performance. In addition, we set a relatively high initial learning rate to speed up training, which may lead to suboptimal results. In future work, we can perform hyperparameter search to seek for the best parameter setting, test out more combinations of data augmentation techniques, and alleviate the memory constraint with gradient accumulation trick.

Another interesting point is that the performance of global branch by itself sometimes outperforms that of the fusion branch. This suggests that a single fully connected layer may not be appropriate for the task. Adding additional layers such as dropout may help.

B. CapsNet.

CapsNet alone did not perform well as expected due to huge down-sampling of images. In our experiments, merging

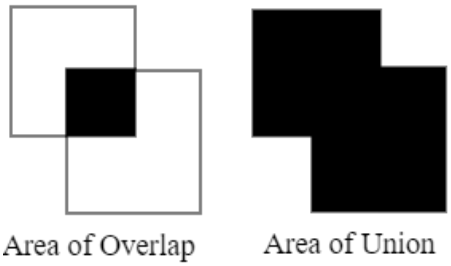

Figure 7. *Area of Overlap and Area of Union.*

CapsNet with a trained baseline neural network as depicted in Fig. 4 also did not produce promising results, with AUROC no better than that of baseline neural network alone. CapsNet memory requirements increase exponentially as the image size is increased. Our experimentation with CXRs resized to 28x28 was too small for any good prediction. With such small images, it was also observed that with higher epochs, the validation loss started to increase, suggesting overfitting. Training with larger images was not feasible due to huge memory requirements and training times.

In out merged model as depicted in Figure 4, the results from CapsNet did improve but were far from ideal. One possible explanation was the fact that the integration of two separately trained models was confusing the entire neural network with the introduction of a random middle integration layer. In subsequent steps, we integrated the two models such that the earlier model is kept frozen for all layers but the last, for the earlier few epochs, until the validation loss converges. After that, the frozen layers were stepwise released backwards. Moreover, we disabled image reconstructions for the merged model and used BCE Loss to calculate the combined model loss instead of the margin loss of CapsNet. We did see a slight increase in the accuracy with this approach. CapsNet however worked better as local branch in our architecture as illustrated in **Error! Reference source not found.** where it only saw a small portion of the image.

C. Disease Localization

Regarding the localization of diseases, we have applied several variations of CAM algorithms on different models. TABLE III shows the results. The best mean *IoU* was **0.111** from DenseNet-121 with Grad-CAM. This is not as ideal as the result of [24], which achieves an overall mean *IoU* of **0.201** with Grad-CAM++. One of the possible reasons is that our model in global branch is not well-trained, which deteriorated the accuracy of heatmap. On the other hand, we can search for the best value of the binary threshold. On the other hand, the long computation time of CAM-family algorithms make it infeasible for training our models in reasonable time.

VIII. CONTRIBUTIONS

All members of the team participated in planning and discussions on the approach, as well as the writing and maintaining the documentation for the project. Project code and documentation can be found on the git repository: https://github.com/comsaint/dlh_project. A video presentation of our work is uploaded on University Of Illinois MediaSpace: https://mediaspace.illinois.edu/media/t/1_i6sl0f2n as well as on YouTube: https://youtu.be/oa1-U0SvwP0.

Specifically, Long Pun built the overall structure of the project and implemented a significant part of the model architecture. He also contributed significantly towards managing the structure of the code repository.

Asad Bin Imtiaz primarily focused on our implementation and use of the CapsNet model and its integration in the base architectures. Besides this, he contributed in designing the data processing and training modules, as well as training several model combinations and compiling results.

Jesse Deng focused on the generation of heatmap for evaluation of the trained models with different methods. He also trained a set of different combination of models.

Gu Yiming contributed towards generation of masks for the heatmaps and parameters for the bounding boxes. He trained several neural network combinations with heatmap models and compiled their results.

IX. CONCLUSIONS

We have applied several state-of-the-art approaches presented in literature to process and classify NIH X-Ray image dataset. Although our model performance fell slightly short of the results obtained in previous works in terms of average AUROC, we were able to outperform them on some of the diseases.

Our experiments on disease localization with the family of CAM algorithms shows that while they may work in theory, in practice they require significant computation resources to finish in reasonable time. Thus attention-guided approach is more appropriate in our case. Our experimentation of CXR images resized to 28x28 with CapsNet did not gave good results, mainly because of small image sizes. Due to enormous memory and long training requirements, training of CapsNet on larger images was not feasible. The CapsNet however performed reasonably well as local branch in our architecture as in Figure 3.

We believe that the performance of our models can be significantly improved with more data augmentation techniques, thorough hyperparameter searching and tuning, and increasing the number of images in the training dataset. On the other hand, other methods of heatmap generation and model ensembles can be tested as well.

REFERENCES

[1] Forum of International Respiratory Societies. The Global Impact of Respiratory Disease – Second Edition. Sheffield, European Respiratory Society, 2017, pp.6.

[2] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, Ronald Summers, ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases, IEEE CVPR, pp. 3462-3471, 2017.

[3] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, Andrew Y. Ng, CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison, AAAI 2019.

[4] Subrato Bharati, Prajoy Podder, M. Rubaiyat Hossain Mondal, Hybrid deep learning for detecting lung diseases from X-ray images, Informatics in Medicine Unlocked, Volume 20, 2020.

[5] Sabour, Sara; Frosst, Nicholas; Hinton, Geoffrey E., "Dynamic Routing Between Capsules". arXiv:1710.09829v2 [cs.CV], Oct 2017.

[6] Christine Herlihy, Charity Hilton, Kausar Mukadam, "Chest X-ray Disease Diagnosis with Deep Convolutional Neural Networks", arXiv:1801.09927v1 [cs.CV], Aug 2018.

[7] Konpat Preechakul, Sira Sriswasdi, Boonserm Kijsirikul, Ekapol Chuangsuwanich, "High resolution weakly supervised localization architectures for medical images", arXiv:2010.11475 [cs.CV], Oct 2020.

[8] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng and Yi Yang, "Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification", arXiv:1801.09927 [cs.CV], Jan 2018.

[9] Spyros Garyfallos, Brent Biseda, and Mumin Khan, NIH-Chest-X-rays-Classification setup, UC Berkeley. Accessed on: March 29, 2021. [Online]. Available: https://github.com/paloukari/NIH-Chest-X-rays-Classification

[10] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition". arXiv:1409.1556 [cs.CV], Sep 2014.

[11] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, "Densely Connected Convolutional Networks", arXiv:1608.06993 [cs.CV], Aug 2016.

[12] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv:1704.04861 [cs.CV], Apr 2017.

[13] Shiraishi J, Katsuragawa S, Ikezoe J, Matsumoto T, Kobayashi T, Komatsu K, Matsui M, Fujita H, Kodera Y, and Doi K.: Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. AJR 174; 71-74, 2000.

[14] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", arXiv:1610.02391 [cs.CV], Oct 2016.

[15] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, "Aggregated Residual Transformations for Deep Neural Networks", arXiv:1611.05431 [cs.CV], Nov 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", arXiv:1512.03385 [cs.CV], Dec 2015.

[17] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya et al., "CheXNet: Radiologist Level pneumonia detection on chest x-rays with deep learning," arXiv preprint arXiv:1711.05225, 2017.

[18] len001 (2020) AG-CNN [Source code]. https://github.com/Ien001/AG-CNN.

[19] Li Yao, Eric Poblenz, Dmitry Dagunts, Ben Covington, Devon Bernard, Kevin Lyman, "Learning to diagnose from scratch by exploiting dependencies among labels", arXiv:1710.10501 [cs.CV], Oct 2017.

[20] LeCun, Yann, et al. "MNIST Database of handwritten digits." The MNIST Database, http://yann.lecun.com/exdb/mnist/

[21] Afshar, Parnian, et al. COVID-CAPS: A Capsule Network-based Framework for Identification of COVID-19 cases from X-ray Images. 2020.

[22] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[23] Ilya Loshchilov, Frank Hutter, "Decoupled Weight Decay Regularization", arXiv:1711.05101 [cs.LG], Nov 2017.

[24] Hugo Kitano, "Classification and Localization of Disease with Bounding Boxes from Chest X-Ray Images", Mar 2020. Accessed on: Apr. 20, 2021. [Online]. Available: http://cs230.stanford.edu/projects_winter_2020/reports/32135302.pdf

[25] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, Vineeth N Balasubramanian, "Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks", arXiv:1710.11063, Oct 2017.

[26] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, Xia Hu, "Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks", arXiv:1910.0127, Oct 2019.

[27] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Andrea Vedaldi, "Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks", arXiv:1810.12348 [cs.CV], Oct 2018.

[28] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv:1704.04861 [cs.CV], Apr 2017.

[29] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, "Learning Deep Features for Discriminative Localization", arXiv:1512.04150 [cs.CV], Dec 2015.

[30] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, Biao Li, "Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs", arXiv:2008.02312 [cs.CV], Aug 2020.

TABLE I. MODEL PERFORMANCE COMPARISON

Method	Model	Branch	Diseases													Mean
			Atel	*Card*	*Effu*	*Infi*	*Mass*	*Nodu*	*Pne1*	*Pne2*	*Cons*	*Edem*	*Emph*	*Fibr*	*PT*	*Hern*	
Wang et al. [2]	R-50	N/A	0.7003	0.81	0.7585	0.6614	0.6933	0.6687	0.658	0.7993	0.7032	0.8052	0.833	0.7859	0.6835	0.8717	**0.7451**
Yao et al. [19]	D-/		0.772	0.904	0.859	0.695	0.792	0.717	0.713	0.841	0.788	0.882	0.829	0.767	0.765	0.914	**0.7980**
**Rajpurkar et al.* [17]	D-121		0.8094	0.9248	0.8638	0.7345	0.8676	0.7802	0.768	0.8887	0.7901	0.8878	0.9371	0.8047	0.8062	0.9164	**0.8414**
Guan et al. [8]	D-121	Global	0.832	0.906	0.887	0.717	0.87	0.791	0.732	0.891	0.808	0.905	0.912	0.823	0.802	0.883	**0.840**
		Local	0.797	0.865	0.851	0.704	0.829	0.733	0.710	0.850	0.802	0.882	0.874	0.801	0.769	0.872	**0.810**
		Fusion	0.853	0.939	0.903	0.754	0.902	0.828	0.774	0.921	0.842	0.924	0.932	0.864	0.837	0.921	**0.8710**
	R-50	Global	0.818	0.904	0.881	0.728	0.863	0.78	0.783	0.897	0.807	0.892	0.918	0.815	0.800	0.889	**0.8410**
		Local	0.798	0.881	0.862	0.707	0.826	0.736	0.716	0.872	0.805	0.874	0.898	0.808	0.770	0.887	**0.8170**
		Fusion	0.844	0.937	0.904	0.753	0.893	0.827	0.776	0.919	0.842	0.919	0.941	0.857	0.836	0.903	**0.8680**
*Our work	D-121	Global	0.7337	0.8545	0.7078	0.8257	0.7946	0.8724	0.8021	0.925	0.6872	0.7427	0.6997	0.7337	0.6816	0.8348	**0.7783**
		Local	0.6783	0.7834	0.6780	0.7859	0.7322	0.8011	0.7318	0.8535	0.6572	0.6396	0.6324	0.6780	0.6387	0.7788	**0.7192**
		Fusion	0.6686	0.8007	0.6694	0.7865	0.7499	0.8471	0.7468	0.8841	0.639	0.6742	0.6202	0.6873	0.6247	0.7522	**0.7251**
	R-50	Global	0.7518	0.8782	0.7298	0.8367	0.8128	0.9128	0.8029	0.9178	0.6956	0.7930	0.7383	0.7581	0.6860	0.8442	**0.7970**
		Local	0.7040	0.8186	0.7022	0.8003	0.7610	0.8455	0.7455	0.8779	0.6728	0.7037	0.6502	0.6947	0.6503	0.8051	**0.7451**
		Fusion	0.7526	0.8795	0.7303	0.8405	0.8139	0.9115	0.8039	0.9265	0.6956	0.7917	0.7378	0.7572	0.6951	0.8471	**0.7988**
	Rx-50	Global	0.7454	0.8605	0.7228	0.8379	0.8102	0.8943	0.7937	0.9302	0.6936	0.7842	0.7383	0.7488	0.6918	0.8508	**0.7930**
		Local	0.7042	0.7941	0.688	0.7993	0.7479	0.8138	0.748	0.9061	0.6594	0.6876	0.649	0.6908	0.649	0.7936	**0.7379**
		Fusion	0.7451	0.8573	0.7228	0.8396	0.8102	0.8928	0.801	0.9392	0.6913	0.7844	0.737	0.7476	0.6944	0.8514	**0.7939**
	Rx-101	Global	0.765	0.8829	0.7474	0.8302	0.823	0.9174	0.8201	0.9103	0.6974	0.8149	0.7581	0.7786	0.6972	0.8513	**0.8067**
		Local	0.5815	0.6792	0.6281	0.7112	0.6382	0.7334	0.658	0.7684	0.6237	0.5693	0.5818	0.6196	0.5866	0.6823	**0.6472**
		Fusion	0.7269	0.855	0.7103	0.8327	0.7858	0.8433	0.7901	0.9348	0.6861	0.8124	0.6911	0.7499	0.6833	0.8168	**0.7799**
	MobileNet	Global	0.7253	0.8465	0.7025	0.8313	0.7914	0.8501	0.7961	0.8948	0.6831	0.7298	0.6969	0.7395	0.6913	0.8404	**0.7728**
		Local	0.6735	0.7591	0.6685	0.7684	0.7151	0.7317	0.7294	0.8607	0.6431	0.6739	0.6371	0.6792	0.636	0.7609	**0.7098**
		Fusion	0.7311	0.8461	0.7021	0.8295	0.7894	0.8405	0.7943	0.9198	0.6823	0.7352	0.6994	0.7344	0.6879	0.8389	**0.7736**
	CapsNet	Global	0.6673	0.7022	0.6874	0.7814	0.7181	0.5966	0.6997	0.6424	0.659	0.5976	0.5574	0.6674	0.6085	0.6883	**0.6624**
		Local	0.5252	0.5247	0.5679	0.6238	0.5762	0.493	0.5507	0.4029	0.5645	0.5197	0.5045	0.5216	0.5553	0.5442	**0.5339**
		Fusion	0.6606	0.6606	0.6805	0.7694	0.717	0.5385	0.7192	0.8478	0.6558	0.5894	0.5627	0.6677	0.6032	0.6371	**0.6652**

AUROC of each disease and the average across the 14 diseases. * denotes that a different train/test split is used as stated in the paper. All the other methods split the dataset with 70% for training, 10% for validation and 20% for testing. Each pathology is denoted with its first four characteristics, e.g., Atelectasis with Atel. Pneumonia and Pneumothorax are denoted as Pneu1 and Pneu2, respectively. PT represents Pleural Thickening. For each column, the best and second-best results are highlighted in red and blue, respectively. For models, 'R', "Rx' and 'D' refers to ResNet, ResNeXt and DenseNet respectively, with the number after dash denoting the number of layers.

TABLE II. PERFORMANCE OF HETEROGENEOUS MODELS

Global Branch	Local Branch	Branch	Disease														Mean
			Atel	*Card*	*Effu*	*Infi*	*Mass*	*Nodu*	*Pne1*	*Pne2*	*Cons*	*Edem*	*Emph*	*Fibr*	*PT*	*Hern*	
D-121	CapsNet	Global	0.7604	0.8881	0.7491	0.8536	0.8233	0.9247	0.8308	0.9225	0.6949	0.8146	0.7552	0.7761	0.7157	0.8561	**0.8118**
		Local	0.6289	0.6596	0.6515	0.7002	0.7055	0.7512	0.5620	0.5743	0.5993	0.6182	0.5897	0.6080	0.5613	0.7142	**0.6374**
		Fusion	0.7400	0.8748	0.6951	0.8266	0.8064	0.9078	0.8073	0.9134	0.6620	0.7880	0.7262	0.7518	0.6691	0.8303	**0.7856**
R-50	CapsNet	Global	0.7480	0.8810	0.7319	0.8436	0.8124	0.9102	0.8090	0.9180	0.6943	0.7887	0.7218	0.7549	0.7055	0.8508	**0.7979**
		Local	0.6104	0.6307	0.6409	0.6546	0.6758	0.7096	0.5268	0.5097	0.5809	0.6342	0.5985	0.5925	0.5414	0.6984	**0.6146**
		Fusion	0.7470	0.8821	0.7306	0.8443	0.8121	0.9086	0.8107	0.9216	0.6908	0.7893	0.7203	0.7543	0.7079	0.8513	**0.7979**
R-50	D-121	Global	0.7621	0.8826	0.7366	0.8418	0.8205	0.9198	0.8230	0.9298	0.6864	0.8184	0.7464	0.7748	0.6846	0.8519	**0.8056**
		Local	0.7119	0.8099	0.7174	0.7978	0.7746	0.8617	0.7498	0.8797	0.6720	0.7200	0.6703	0.7030	0.6712	0.8076	**0.7533**
		Fusion	0.7291	0.8560	0.6984	0.8056	0.8035	0.8969	0.7689	0.8618	0.6665	0.7903	0.6933	0.7282	0.6489	0.8380	**0.7704**
D-121	R-50	Global	0.7430	0.8619	0.7205	0.8339	0.8058	0.8977	0.8139	0.9432	0.6890	0.7677	0.7246	0.7529	0.6948	0.8453	**0.7924**
		Local	0.7015	0.8145	0.6962	0.7943	0.7529	0.8546	0.7640	0.8599	0.6672	0.6920	0.6440	0.7072	0.6464	0.8140	**0.7435**
		Fusion	0.7183	0.8243	0.6840	0.7997	0.7867	0.8884	0.7810	0.9323	0.6553	0.7393	0.6728	0.7238	0.6280	0.8196	**0.7610**
R-50	MobileNet	Global	0.7470	0.8725	0.7176	0.8288	0.7986	0.8907	0.8108	0.9196	0.6821	0.7660	0.7220	0.7404	0.6777	0.8458	**0.7871**
		Local	0.6898	0.7655	0.6746	0.7791	0.7209	0.8013	0.7441	0.8874	0.6442	0.6538	0.6384	0.6727	0.6253	0.7811	**0.7199**
		Fusion	0.7480	0.8709	0.7085	0.8291	0.7942	0.8893	0.8144	0.9286	0.6800	0.7669	0.7179	0.7398	0.6857	0.8462	**0.7871**
Rx-101	Rx-50	Global	0.7504	0.8766	0.7345	0.8425	0.8132	0.9104	0.8128	0.9291	0.6871	0.7927	0.7376	0.7574	0.6959	0.8572	**0.7998**
		Local	0.6532	0.7671	0.6537	0.7635	0.7073	0.8231	0.7296	0.8273	0.6106	0.6441	0.6304	0.6475	0.6172	0.7660	**0.7029**
		Fusion	0.7080	0.8284	0.6681	0.8368	0.7541	0.8487	0.7820	0.9346	0.6311	0.7920	0.6652	0.7497	0.6510	0.8136	**0.7616**

AUROC of each disease and the average across the 14 diseases. Each pathology is denoted with its first four characteristics, e.g., Atelectasis with Atel. Pneumonia and Pneumothorax are denoted as Pneu1 and Pneu2, respectively. PT represents Pleural Thickening. For each column, the best and second-best results are highlighted in red and blue, respectively. For models, 'R', "Rx' and 'D' refers to ResNet, ResNeXt and DenseNet respectively, with the number after dash denoting the number of layers.

TABLE III HEATMAP PERFORMANCE

Algorithm	**Model**	**Disease**								**Mean**
		Atel.	***Card.***	***Effu.***	***Infi.***	***Mass***	***Nodu.***	***Pneu1***	***Pneu2***	
Grad-CAM	Rx-101	0.02647	0.13113	0.03656	0.03866	0.00045	0.00137	0.00400	0.00344	0.02274
Grad-CAM++		0.04216	0.29663	0.07101	0.13081	0.03493	0.00715	0.10157	0.02788	0.09588
XGrad-CAM		0.02647	0.13113	0.03656	0.03866	0.00045	0.00137	0.00400	0.00344	0.02274
Grad-CAM	D-121	0.07271	0.24658	0.04785	0.16482	0.09078	0.01916	0.13149	0.10033	0.11334
Grad-CAM++		0.03844	0.03963	0.01873	0.03909	0.05333	0.01446	0.01300	0.04567	0.03261
XGrad-CAM		0.03665	0.20341	0.06394	0.12766	0.05190	0.00649	0.09182	0.06270	0.08523
Grad-CAM	Rx-50	0.06289	0.21056	0.03668	0.14907	0.05626	0.00532	0.09431	0.02461	0.08632
Grad-CAM++		0.07628	0.29474	0.04059	0.15055	0.04818	0.00300	0.14048	0.01733	0.10607
XGrad-CAM		0.06289	0.21056	0.03668	0.14907	0.05626	0.00532	0.09431	0.02461	0.08632

*IoU of each disease and the overall mean. Each pathology is denoted with its first four characteristics, e.g., Atelectasis with Atel. Pneumonia and Pneumothorax are denoted as Pneu1 and Pneu2, respectively. The best and second-best results are highlighted in red and blue, respectively.