COVID-MobileXpert: On-Device COVID-19 Patient Triage and Follow-up using Chest X-rays

Xin Li
Department of Computer Science
Wayne State University
Detroit, MI 48202
Email: xinlee@wayne.edu

Chengyin Li
Department of Computer Science
Wayne State University
Detroit, MI 48202
Email: cyli@wayne.edu

Dongxiao Zhu
Department of Computer Science
Wayne State University
Detroit, MI 48202
Email: dzhu@wayne.edu

2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 978-1-7281-6215-7/20/$31.00 ©2020 IEEE DOI: 10.1109/BIBM49941.2020.9313217

***Abstract*—During the COVID-19 pandemic, there has been an emerging need for rapid, dedicated, and point-of-care COVID-19 patient disposition techniques to optimize resource utilization and clinical workflow. In view of this need, we present COVID-MobileXpert: a lightweight deep neural network (DNN) based mobile app that can use chest X-ray (CXR) for COVID-19 case screening and radiological trajectory prediction. We design and implement a novel three-player knowledge transfer and distillation (KTD) framework including a pre-trained attending physician (AP) network that extracts CXR imaging features from a large scale of lung disease CXR images, a fine-tuned resident fellow (RF) network that learns the essential CXR imaging features to discriminate COVID-19 from pneumonia and/or normal cases with a small amount of COVID-19 cases, and a trained lightweight medical student (MS) network to perform on-device COVID-19 patient triage and follow-up. To tackle the challenge of vastly similar and dominant fore- and background in medical images, we employ novel loss functions and training schemes for the MS network to learn the robust features. We demonstrate the significant potential of COVID-MobileXpert for rapid deployment via extensive experiments with diverse MS architecture and tuning parameter settings. The source codes for cloud and mobile based implementations are available from the following url: https://github.com/xinli0928/COVID-Xray.**

***Index Terms*—COVID-19, SARS-CoV-2, On-device Machine Learning, Trajectory Prediction, Chest X-Ray (CXR)**

I. Introduction

Due to its flu-like symptoms and potentially serious outcomes, a dramatic increase of suspected COVID-19 cases are expected to overwhelm the healthcare system during the flu season. Health systems still largely allocate facilities and resources such as Emergency Department (ED) and Intensive Care Unit (ICU) on a reactive manner facing significant labor and economic restrictions. To optimize resource utilization and clinical workflow, a rapid, automated, and point-of-care COVID-19 patient management technology that can triage (COVID-19 case screening) and follow up (radiological trajectory prediction) patients is urgently needed.

Chest X-ray (CXR), though less accurate than a PCR diagnostic, chest Computed Tomography (CT) or serological test, became an attractive option for patient management due to its impressive portability, availability and scalability [1]. Convolutional neural networks (CNNs) have achieved significant advancement in various tasks based on CXR [2]–[4]. During the pandemic, CNNs have also been successfully employed to assist with COVID-19 CXR interpretation, where three major tasks have been performed: diagnosis, severity evaluation, and trajectory prediction. The majority of related previous works concentrate on diagnosis. They directly borrowed or adopted well-known CNN architectures such as ResNet [5], [6], InceptionV3 [5], DenseNet [7], and VGG [6] for COVID-19 case screening. For severity evaluation, Cohen et al. [8] and Zhu et al. [9] predicted lung disease severity scores using a linear regression model based on features extracted from CNNs. To associate each score with a confidence value, Signoroni et al. [10] treated this task as a joint multi-class classification and regression problem using a compound loss function. Based on the severity assessment, the trajectory prediction can be achieved by calculating severity score difference between two adjacent CXR images. Other than score level interpretation, Duchesne et al. [11] built their trajectory prediction model based on feature level. They used logistic regression to classify the trajectory based on the feature extracted from a single CXR. However, the feature from a single CXR may not be sufficient to predict radiological trajectory. To tackle these challenges, we propose to forecast trajectory using feature extracted from a series of longitudinal CXR images, where subtle changes that are invisible to human, can be detected.

There is a growing interest to deploy machine learning models on the device to minimize latency and maximize the protection of privacy. Currently, most models for COVID-19 interpretation are full DNNs and not suitable to deploy on resource-constrained mobile devices. As there is no existing on-device medical image interpretation research, most of the recent work [12] focuses on comparing the performance of different lightweight DNNs using natural image datasets. To improve the performance of the lightweight models, knowledge distillation [13], [14] is usually used where a full size teacher model is trained first, and a lightweight student model is then trained with the 'knowledge' distilled via the soft labels from the teacher model. Knowledge distillation yields compact student models that outperform the compact models trained from scratch [15]. Unlike the natural images, on-device classification of medical images remain largely an uncharted territory due to the following unique challenges: 1) label scarcity significantly limits the generalizability of the system; 2) vastly similar and dominant fore- and background make it

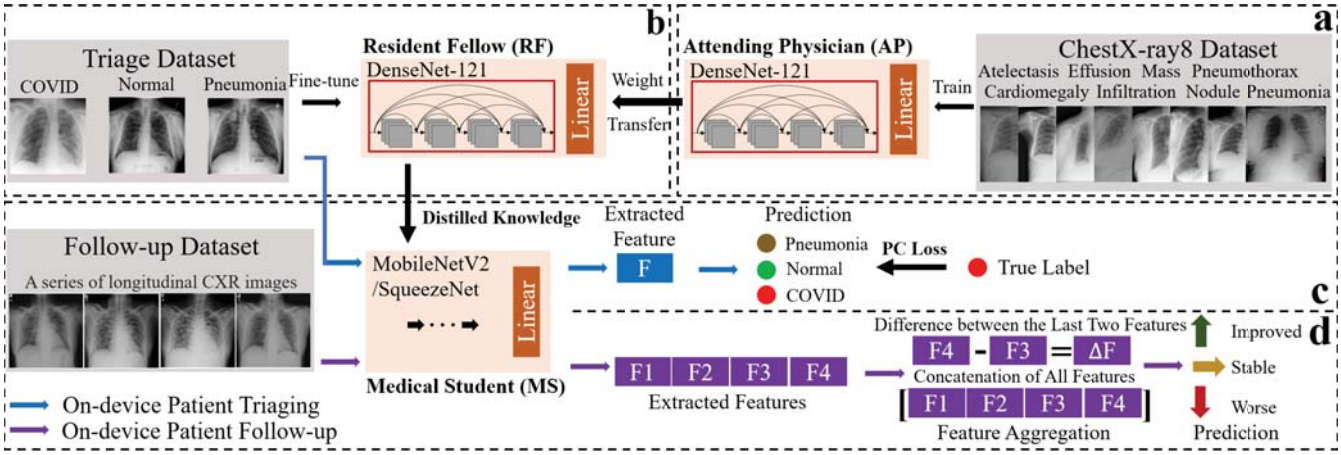

Fig. 1. Overview of the three-player KTD training architecture demonstrating the **knowledge transfer** from AP to RF and the **knowledge distillation** from RF to MS. The blue and purple arrows demonstrate the training for two tasks: patient triage and follow-up respectively.

hard samples for learning the discriminating features between different classes. To tackle these challenges, we propose a novel three-player knowledge transfer and distillation (KTD) framework composed of an Attending Physician (AP) network, a Resident Fellow (RF) network, and a Medical Student (MS) network for on-device COVID-19 patient triage and follow-up. We pre-train a full AP network using a large scale of lung disease CXR images [16], followed by fine-tuning a RF network via knowledge transfer using COVID-19 dataset, then we train a lightweight MS network for on-device COVID-19 patient triage and follow-up via knowledge distillation. The unique features of the KTD framework are knowledge transfer from large-scale lung disease images to enhance expressiveness of learned representation and novel loss functions to increase robustness of knowledge distillation to the MS network.

To the best of our knowledge, currently, there is no mobile AI system for on-device COVID-19 patient triage and follow-up using CXR images. In this work, we present an AI-powered system, COVID-MobileXpert, to triage and follow up COVID-19 patients using portable X-rays at the patient's location. At the ED, COVID-MobileXpert calculates COVID-19 probabilistic risk to assist automated triage of COVID-19 patients. At the ICU or general ward (GW), it uses a series of longitudinal CXR images to determine whether there is an impending deterioration in the health condition of COVID-19 patients. Therefore, COVID-MobileXpert is essential to fully realize the potential of CXR to exert both immediate and long-term positive impacts on US healthcare systems. The experiments demonstrate the effectiveness of our proposed framework and a strong potential of on-device patient management using COVID-MobileXpert.

II. Method

A. Model Architecture and KTD Training Scheme

We employ DenseNet-121 architecture as the template to pre-train and fine-tune the AP and RF networks. In addition, among well-studied lightweight CNNs [12], we select the most well-applied network MobileNetV2, and the most lightweighted network SqueezeNet as the candidate MS networks for on-device COVID-19 case screening and radiological trajectory prediction. Fig. 1 illustrates the three-player KTD training framework where the knowledge of abnormal CXR images is transferred from AP network to RF network and knowledge of discriminating COVID-19, non-COVID-19, and pneumonia is distilled from the RF to the MS network.

We pre-train the AP network as the *source task* (Fig. 1a), i.e., lung disease classification, and fine-tune the RF network as the *destination task* (Fig. 1b). Different from recent studies [17] that pre-train the models with natural image datasets such as ImageNet, we pre-train the DenseNet-121 based AP network using the more related ChestX-ray8 dataset [16] of 108,948 lung disease cases to extract the CXR imaging features of lung diseases instead of generic natural imaging features. After that we fine-tune the RF network using a smaller compiled dataset of 3 classes of CXR images, i.e., COVID-19, normal and pneumonia. The RF network is then used to train the lightweight MS network, e.g., MobileNetV2, or SqueezeNet, via knowledge distillation.

As shown in the MS section in Fig. 1c, after knowledge distillation, the trained MS network can triage patients by screening COVID-19 cases following the blue arrow. Then a radiological trajectory prediction model is further developed based on the trained MS network (Fig. 1d). Following the purple arrow, given a series of longitudinal CXR images from one patient, all images are fed into the pre-trained MS network for extracting disease-specific features. These features are then aggregated using different schemes before prediction. Here we investigate two different schemes: 1) calculating the difference between the last two CXR images' features; 2) chronologically concatenating all features. After feature aggregation, two fully connected layers are randomly initialized and trained with softmax loss function for the trajectory prediction.

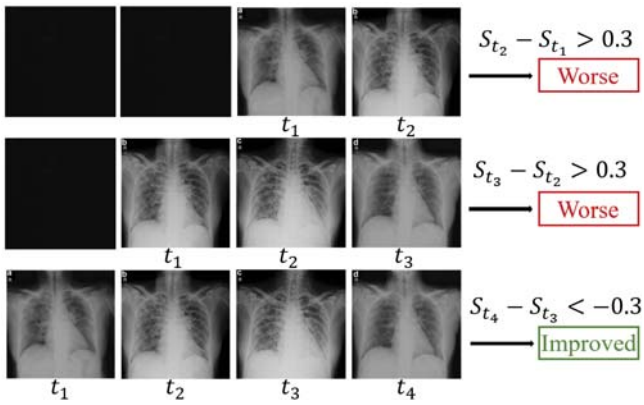

Fig. 2. An example of data preparation for a series of longitudinal CXR images with radiological trajectory labels. The patient is in critical condition on t_3 then recovered afterward.

B. Loss Functions

As stated above, a unique challenge in medical imaging classification is the so-called "hard sample problem" [18], i.e., a subtle difference on the ROI across the images with a large amount of shared fore- and backgrounds. Motivated by this, we use an in-house developed loss function, i.e., Probabilistically Compact (PC) loss, for training the MS model and compared with ArcFace [19], the additive angular margin loss for deep face recognition, using the classical softmax loss as the baseline. Both PC and ArcFace losses are designed for improving classification performance on hard samples. PC loss is to encourage the maximized margin between the most probable label (predictive probability) and the first several most probable labels whereas ArcFac loss is to encourage widening the geodesic distance gap between the closest labels. In terms of predicted probabilities, DNN robustness is beneficial from the large gap between $f_y(\boldsymbol{x})$ and $f_k(\boldsymbol{x})$ $(k \neq y)$, where $f_y(\boldsymbol{x})$ represents the true class and $f_k(\boldsymbol{x})$ $(k \neq y)$ represents the most probable class. Indeed, the theoretical study [20] in deep learning shows that the gap $f_y(\boldsymbol{x}) - \max_k f_k(\boldsymbol{x})$ can be used to measure the generalizability of DNNs.

The PC loss to improve CNN's robustness is as follows:

$$L_{pc}(\theta) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i_k \in S_k} \sum_{j=1, j \neq k}^{K} \max\{0, f_j(\boldsymbol{x}_{i_k}) + \xi - f_k(\boldsymbol{x}_{i_k})\}, \tag{1}$$

where N is the number of training samples, $\xi > 0$ is the probability margin treated as a hyperparameter. Here, we include all non-target classes in the formulation and penalize any classes for each training sample that violate the margin requirement for two reasons: (1) by maintaining the margin requirement for all classes, it provides us convenience in implementation as the first several most probable classes can change during the training process; and (2) if one of the most probable classes satisfies the margin requirement, all less probable classes will automatically satisfy this requirement and hence have no effect on the PC loss. Compared with previous works that explicitly learn features with large inter-class separability and intra-class compactness, the PC loss avoids assumptions on the feature space, instead, it only encourages the feature learning that leads to probabilistic intra-class compactness by imposing a probability margin ξ.

III. Experiment and Results

A. Datasets

The CXR image dataset for COVID-19 patient triage is composed of 179 CXR images from normal class [21], 179 from pneumonia class [21] and 179 from COVID-19 class containing both PA (posterior anterior) and AP (anterior posterior) positions [22]. All images are resized to 256×256 pixels and then center-cropped to 224×224 pixels before being fed into networks [22]. We split the dataset into training/validation/testing sets with 125/18/36 cases (7:1:2) in each class. Since some patients have multiple images, we sample images per patient for each split to avoid images from the same patient being included in both training and testing sets.

For the radiological trajectory dataset, we assign a opacity score S for each COVID-19 positive CXR image in [22] using the scoring system provided by [8]. Fig. 2 shows an example of how we generate CXR image sequences and assign corresponding radiological trajectory labels (i.e., "Worse", "Stable", "Improved"). Given a COVID-19 patient's CXR images over four time points (the maximum length is set to four time points), we can create three CXR image sequences with zero-padding. For each sequence, we calculate the difference in the score of the last two CXR images. If the difference is larger than 0.3 the sequence is categorized as "Worse", if the difference is less than -0.3, it is labeled as "Improved", otherwise, the category is "Stable". We collect a total of 159 CXR image sequences from 100 patients in [22] and the dataset contains 76 "Worse" samples, 38 "Stable" samples, and 45 "Improved" samples. Similarly, we split it into training/validation/testing sets with 111/16/32 samples (7:1:2).

B. Implementation Details

We implement our model on a GeForce GTX 1080ti GPU platform using PyTorch. The network is trained with the Adam optimizer for 50 epochs with a mini-batch size of 32 (triage task) and 10 (follow-up task). The parameter values that give rise to the best performance on the validation dataset are used for testing. Similar to [11], when training the radiological trajectory prediction model, we employ the pre-trained MS network as a feature extractor (fixed weights). To overcome the overfitting problem, we also apply a dropout regularization with a rate of 0.5.

C. Tunning Parameters

ξ: in the PC loss formula (Eq. 1), a larger value of ξ will encourage the probabilistic intra-class compactness. α: in knowledge distillation [13], [14], it regularizes the 'strength' of knowledge distillation by specifying the relative contributions of the distillation loss. T: it represents temperature in distillation loss. As T increases, the probability distribution generated by the softmax loss becomes softer, providing more information from RF model.

MobileNetV2/SqueezeNet (T=5)				
α	PC(ξ = 0.8)	PC(ξ = 0.995)	ArcFace	SM
0.2	0.870/0.798	0.833/0.777	0.870/0.750	0.861/0.777
0.4	**0.880**/0.777	0.870/0.815	0.861/0.796	0.833/0.759
0.6	0.851/0.796	0.851/0.787	0.851/0.805	0.861/0.796
0.8	**0.880**/0.824	0.870/0.796	0.851/0.796	0.833/0.787
MobileNetV2/SqueezeNet ($\alpha = 0.8$)				
T	PC(ξ = 0.8)	PC(ξ = 0.995)	ArcFace	SM
1	0.851/0.750	**0.880**/0.814	0.870/0.796	0.870/0.796
5	**0.880**/0.824	0.870/0.796	0.851/0.796	0.833/0.787
10	**0.880**/0.796	0.842/0.750	0.861/0.787	0.870/0.824

TABLE I
CLASSIFICATION PERFORMANCE OF MS NETWORKS, THE VALUES IN ./. INDICATE MOBILENETV2 VS. SQUEEZENET.

MobileNetV2/SqueezeNet (DenseNet-121)		
Classifiers	Difference	Concatenation
Logistic Regression	0.560/0.640 (0.720)	0.760/0.720 (0.800)
Gradient Boosting	0.680/0.640 (0.680)	0.680/0.680 (0.680)
Random Forest	0.680/0.600 (0.680)	0.720/0.680 (0.720)
Our FC-classifier	0.720/0.680 (0.720)	**0.800**/0.760 (0.800)

TABLE II
PERFORMANCE COMPARISON OF TWO FEATURE AGGREGATION SCHEMES (DIFFERENCE VS. CONCATENATION) WITH FOUR DIFFERENT CLASSIFIERS USING TWO MS NETWORKS (MOBILENETV2 AND SQUEEZENET) AS THE FEATURE EXTRACTOR. VALUES IN PARENTHESES INDICATE THE UPPER BOUND OF ACCURACY YIELDED BY RF NETWORK (DENSENET-121).

D. Evaluation of COVID-19 Patient Triage Performance

We first report the classification accuracy to select the best MS model under different values of hyperparameters, followed by evaluation of the best model's discriminating power of COVID-19 using AUROC values. With the knowledge transfer from the AP network pre-trained with a number of lung disease cases, the RF network demonstrates a remarkably high accuracy of 0.935 in the classification. Distilling knowledge from the RF network to the lightweight MS network, we observe an impressive performance that a vast majority of accuracy values are well above 0.850. Table I shows the results of both MobileNetV2 and SqueezneNet architecture with different settings. It is clear that the knowledge distillation is essential to train the lightweight MS network since the MS network alone, without knowledge distillation, achieves a baseline accuracy of 0.843 (MobileNetV2) and 0.732 (SqueezeNet), which are lower than the performance shown in Table I. In addition, we note that the performance of MobileNetV2 and SqueezeNet are not sensitive to the choice of Parameters T and α, but sensitive to the choice of loss functions. Overall, the PC loss performs the best across all settings, indicating the quality of knowledge distilled to the MS network plays a pivotal role to ensure an accurate on-device performance. In order to evaluate the triage performance under the different decision thresholds, we use the AUROC value to assess how well the model is capable of discriminating COVID-19 cases. Both MobileNetV2 and SqueezeNet achieve high AUROC values of 0.970 and 0.964 when discriminating COVID-19 cases against mixed pneumonia and normal cases demonstrating strong potential for on-device triage.

E. Evaluation of COVID-19 Patient Follow-up Performance

Similar to [11], we first report the classification accuracy of discriminating "Worse" versus "Improved" cases to select the best combination of classifiers and feature aggregation schemes, followed by evaluation of the best model's performance using AUROC values. Based on the extracted features, four classifiers are trained for radiological trajectory prediction: 1) logistic regression; 2) gradient boosting; 3) random forest and 4) MS networks followed by fully connected layers (our FC-classifier). As shown in Table II, we observe the classifiers trained based on the feature extracted from both compact MS networks, achieve a very similar level of performance to large scale RF network. This again demonstrates that KTD training architecture with PC loss performs a high-quality knowledge distillation. When doing comparison between the feature aggregation schemes, we can see a significant improvement from using a series of longitudinal features over using only the difference between the last two sets of features. As for classifier selection, compared with these conventional classifiers, our FC-classifier is able to learn a series of subtle changes related to radiological features from CXR images, thus achieving a better performance. As a result, the best on-device performance is obtained by our FC-classifier with feature concatenation using MobileNetV2, which attains the upper bound of accuracy (0.800) yielded by DenseNet-121. Duchesne et al. [11] also report a high accuracy (0.827) of predicting the "Worse" category based on the feature extracted from a single CXR with their highly imbalanced testing dataset, which contains over 84.6% samples labeled as "Worse". However, the reported accuracy is lower than a simple baseline: a dummy classifier that always predicts the most frequent label "Worse" would yield a higher accuracy of 0.846. To make a comparison, we reimplement their model [11] on our more balanced dataset and record a result of 0.600, which implies that using the feature from a single CXR may not be sufficient to predict radiological trajectory. In order to systematically evaluate the performance of the MS networks under the different decision thresholds, we again use the AUROC value to assess how capable the model is in discriminating "Worse" cases from "Improved" cases. It is important to note that MobileNetV2 networks can achieve a high AUROC value of 0.883 enabling it to identify "Worse" cases and show a significant potential of on-device follow-up.

Mobile Systems	Nexus One		Pixel		Pixel 2 XL	
The MS Network	CPU	Memory	CPU	Memory	CPU	Memory
MobileNetV2	69.3 %	69.4 MB	67.2 %	70.5 MB	68.7 %	72.8 MB
SqueezeNet	37.7 %	67.5 MB	29.0 %	29.0 MB	26.7 %	68.6 MB
	Nexus S		Pixel 2		Pixel 3 XL	
MobileNetV2	67.7 %	88.8 MB	66.2 %	69.4 MB	63.6 %	76.5 MB
SqueezeNet	32.7 %	64.4 MB	28.8 %	70.1 MB	25.8 %	66.1 MB

TABLE III
COMPARISON OF RESOURCE CONSUMPTION OF THE TWO MS NETWORKS DEPLOYED TO THE SIX ANDROID BASED MOBILE DEVICES. THE ENERGY CONSUMPTION IS HEAVY FOR MOBILENETV2 BASED APP BUT MEDIUM FOR SQUEEZENET BASED APP ON ALL DEVICES.

IV. PERFORMANCE EVALUATION ON MOBILE DEVICES

For on-device COVID-19 patient triage and follow-up with resource constraints, resource consumption is also an important consideration for performance evaluation in addition to accuracy. In order to systematically assess the performance of our COVID-19 on-device app, we select six mobile systems

released following a chronic order, i.e., Nexus One / Nexus S (low-end); Pixel/ Pixel 2 (mid-range) and Pixel 2 XL/ Pixel 3 XL (high-end). Using the Pytorch Mobile framework, we deploy the three MS networks to the six Android based mobile systems and compare the resource consumption with regard to CPU, memory and energy usages.

In Table III, it is clear that the MobileNetV2 based COVID-19 app is resource-hungry, demonstrated by much higher resource consumption than SqueezeNet. Thus, the high accuracy achieved by MobileNetV2 is at the cost of high resource consumption. Within each app, we observe a downward trend in resource consumption following the chronic order, reflecting a continuous improvement of mobile device hardware. Overall, MobileNetV2 based COVID-19 apps are more suitable for high-performing mobile devices due to the high accuracy achieved with a higher resource consumption. On the other hand, SqueezeNet is more suitable for low-end mobile devices with both lower accuracy and resource consumption.

V. Conclusions

The classical two-player knowledge distillation framework [13] has been widely used to train a compact network that is hardware-friendly with ample applications [12]. In the related task of on-device natural image classification, the teacher network is pre-trained with ImageNet and distill the knowledge to a lightweight student network (e.g., MobileNetV2). This two-player framework, although seemingly successful, can be problematic for on-device medical imaging based COVID-19 case screening and radiological trajectory prediction described herein. The large gap between natural images and the medical images of a specific disease such as COVID-19 makes the knowledge distillation less effective as it is supposed to be. The small number of labeled COVID-19 images for training further aggravates the situation.

In our three-player KTD framework, knowledge transfer from the AP network to the RF network can be viewed as a more effective regularization as they are built on the same network architecture, which in turn, make the knowledge distillation more effective since the RF network and MS network share the same training set. Different from what has extensively investigated focusing on the impact of distillation strength and temperature, we uncover a pivotal role of employing novel loss functions in refining the quality of knowledge to be distilled. Hence our three-player framework provides a more effective way to train the compact on-device model using a smaller dataset while preserving performance.

From a more broad perspective, the three-player KTD framework is generally applicable to train other on-device medical imaging classification and segmentation apps for point-of-care screening of other human diseases such as lung [16] and musculoskeletal [23] abnormalities.

References

[1] H. Y. F. Wong, H. Y. S. Lam, A. H.-T. Fong, S. T. Leung, T. W.-Y. Chin, C. S. Y. Lo, M. M.-S. Lui, J. C. Y. Lee, K. W.-H. Chiu, T. W.-H. Chung, E. Y. P. Lee, E. Y. F. Wan, I. F. N. Hung, T. P. W. Lam, M. D. Kuo, and M.-Y. Ng, "Frequency and distribution of chest radiographic findings in patients positive for covid-19," *Radiology*, vol. 296, no. 2, pp. E72–E78, 2020, pMID: 32216717.

[2] X. Li and D. Zhu, "Robust detection of adversarial attacks on medical images," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 1154–1158.

[3] X. Li, R. Cao, and D. Zhu, "Vispi: Automatic visual perception and interpretation of chest x-rays," *arXiv preprint arXiv:1906.05190*, 2019.

[4] X. Li, D. Pan, and D. Zhu, "Defending against adversarial attacks on medical imaging ai system, classification or detection?" *arXiv preprint arXiv:2006.13555*, 2020.

[5] M. E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. Al-Emadi *et al.*, "Can ai help in screening viral and covid-19 pneumonia?" *arXiv preprint arXiv:2003.13145*, 2020.

[6] L. O. Hall, R. Paul, D. B. Goldgof, and G. M. Goldgof, "Finding covid-19 from chest x-rays using deep learning on a small dataset," *arXiv preprint arXiv:2004.02060*, 2020.

[7] D. Lv, W. Qi, Y. Li, L. Sun, and Y. Wang, "A cascade network for detecting covid-19 using chest x-rays," *arXiv preprint arXiv:2005.01468*, 2020.

[8] J. P. Cohen, L. Dao, P. Morrison, K. Roth, Y. Bengio, B. Shen, A. Abbasi, M. Hoshmand-Kochi, M. Ghassemi, H. Li *et al.*, "Predicting covid-19 pneumonia severity on chest x-ray with deep learning," *arXiv preprint arXiv:2005.11856*, 2020.

[9] J. Zhu, B. Shen, A. Abbasi, M. Hoshmand-Kochi, H. Li, and T. Q. Duong, "Deep transfer learning artificial intelligence accurately stages covid-19 lung disease severity on portable chest radiographs," *PloS one*, vol. 15, no. 7, p. e0236621, 2020.

[10] A. Signoroni, M. Savardi, S. Benini, N. Adami, R. Leonardi, P. Gibellini, F. Vaccher, M. Ravanelli, A. Borghesi, R. Maroldi *et al.*, "End-to-end learning for semiquantitative rating of covid-19 severity on chest x-rays," *arXiv preprint arXiv:2006.04603*, 2020.

[11] S. Duchesne, D. Gourdeau, P. Archambault, C. Chartrand-Lefebvre, L. Dieumegarde, R. Forghani, C. Gagne, A. Hains, D. Hornstein, H. Le *et al.*, "Tracking and predicting covid-19 radiological trajectory using deep learning on chest x-rays: Initial accuracy testing," *medRxiv*, 2020.

[12] S. Dhar, J. Guo, J. Liu, S. Tripathi, U. Kurup, and M. Shah, "On-device machine learning: An algorithms and learning theory perspective," *arXiv preprint arXiv:1911.00623*, 2019.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[14] M. Goldblum, L. Fowl, S. Feizi, and T. Goldstein, "Adversarially robust distillation," *arXiv preprint arXiv:1905.09747*, 2019.

[15] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 5142–5151.

[16] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3462–3471.

[17] L. Wang and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images," *arXiv preprint arXiv:2003.09871*, 2020.

[18] X. Li, X. Li, D. Pan, and D. Zhu, "On the learning property of logistic and softmax losses for deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 4739–4746.

[19] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[20] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 5949–5958.

[21] R. S. of North America, "RSNA pneumonia detection challenge," https://www.kaggle.com/c/rsna-pneumonia-detection-challenge, 2018.

[22] J. P. Cohen, P. Morrison, and L. Dao, "Covid-19 image data collection," *arXiv preprint arXiv:2003.11597*, 2020.

[23] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball *et al.*, "Mura: Large dataset for abnormality detection in musculoskeletal radiographs," *arXiv preprint arXiv:1712.06957*, 2017.