Multi-Stage Deep Learning with Explainability for Chest X-Ray Pathology Detection using DenseNet-121

Mrs. T. Kranthi[1], K. Nitish V Venkatesh[2], Marrapu Tarun[3], Karanam Swapna Preethi[4], Gorli Praveena[5], Shaik Mohammed Marshuk[5]
[1]*Assistant Professor, Dept. of Computer Science & Engineering*
Anil Neerukonda Institute of Technology and Sciences Visakhapatnam, Andhra Pradesh, India
[2,3,4,5] *Dept. of Computer Science & Engineering, Anil Neerukonda Institute of Technology and Sciences Visakhapatnam, Andhra Pradesh, India*

Abstract*—Chest radiography remains one of the most widely deployed and economically accessible modalities in clinical di- agnostics, yet its interpretation is inherently limited by radiologist availability, cumulative fatigue, and measurable inter-observer variability—particularly under high patient-load conditions. This paper presents an intelligent Chest X-Ray Di- agnostic Assistant that addresses these systemic constraints through a cascaded, multi-stage inference architecture. Prior to pathology analysis, each uploaded image traverses a three- level binary validation cascade: MobileNetV2 filters non-medical content, a fine-tuned ResNet18 confirms thoracic anatomy, and a transfer-learned ResNet18 performs a preliminary normal- versus-abnormal triage, collectively ensuring that computation- ally intensive classification is invoked only on radiographically valid inputs. Fourteen thoracic conditions are subsequently identified by DenseNet-121 trained on the NIH ChestX-ray14 benchmark, yielding an average area under the receiver operating characteristic curve (AUROC) of 0.8779. Prediction interpretability is provided through Grad-CAM++ attention heatmaps that spatially localise the image regions driving each classification decision. A Google Gemini large language model (LLM) integration then converts the structured numerical outputs into clinician-readable reports partitioned into summary, findings, and recommendations. The system additionally incorporates a Geoapify-powered specialist-finder that maps detected pathologies to nearby relevant clinicians. Fault-tolerant design across four operational tiers ensures graceful degradation rather than system failure under adverse conditions, establishing the pro- posed assistant as a practically deployable decision-support tool for resource-constrained healthcare environments.

***Index Terms*—Chest X-Ray Analysis, DenseNet-121, Convolutional Neural Networks, Binary Classification Pipeline, ResNet18, Transfer Learning, MobileNetV2, Grad-CAM++, Explainable AI, Large Language Model, FastAPI, Thoracic Pathology Detection, NIH ChestX-ray14, Medical Image Analysis**

I. INTRODUCTION

Chest radiography is the most frequently ordered imaging investigation worldwide, yet in a significant proportion of clinical settings the interval between image acquisition and qualified interpretation extends from several hours to multiple days [10]. This latency is not merely an operational inconvenience—for acute presentations such as pneumothorax, pulmonary oedema, or lobar consolidation, delayed diagnosis can directly worsen patient outcomes. The scarcity of trained radiologists is particularly acute in low- and middle-income healthcare systems, where a single specialist may be responsible for reviewing hundreds of studies daily.

Automated computer-aided detection (CAD) systems offer a viable complementary layer to address this imbalance. Prior work has demonstrated that deep convolutional networks can achieve radiologist-level discrimination on single-pathology tasks; however, most published frameworks treat the classification stage in isolation, offering no mechanism to reject invalid or irrelevant inputs, no visual attribution of model decisions, and no pathway from numerical output to clinically actionable guidance [10]. These omissions limit adoption in real-world deployments.

The system introduced in this paper addresses all three shortcomings through an integrated, layered architecture. In our approach, the system is divided

into four main components:

1) A *three-stage binary validation pipeline*: The system first goes through three validation steps. It checks whether the input is actually an X-ray, then verifies that it is a chest X-ray, and finally determines if the image appears normal or abnormal before passing it to the main disease detection model.
2) A *DenseNet-121 multi-label classifier* [5] trained on the NIH ChestX-ray14 dataset [10] to detect fourteen thoracic diseases simultaneously.
3) *Grad-CAM++* [1] is applied to create heatmaps that indicate the important areas in the image, based on features learned by the DenseNet-121 model.
4) Integrated *Google Gemini* to convert DenseNet-121 model output into simple clinical reports, making them easier to understand even for users without medical expertise.

The rest of the paper is organized as follows. Section II reviews the related literature. Section III explains the dataset and data preparation process. Section IV describes the system architecture and training approach. Section V presents the results and analysis. Finally, Section VI concludes the paper and discusses future work.

II. REVIEW OF RELATED LITERATURE

A. *Visual Explanation using Gradient-Weighted Class Activation Mapping*

Grad-CAM++ is an improved version of the original Grad- CAM technique, introduced by Chattopadhay et al. [1]. It pro- vides better visual explanations by highlighting the important regions in an image that influence the model's predictions. Un- like Grad-CAM [2], it uses more detailed gradient information, which helps produce clearer and more precise results. This improvement is especially useful in medical images, where multiple abnormalities can appear in the same X-ray. In such cases, Grad-CAM++ is able to highlight multiple relevant re- gions more accurately. In our system, Grad-CAM++ is imple- mented using a HeatmapGenerator module. This module connects to the final convolutional layer of the DenseNet-121 model and captures forward and backward information during prediction. The generated heatmap is then combined with the original image and encoded so that it can be easily sent as part of the API response.

B. *Efficient Image Classification with MobileNetV2*

MobileNetV2, presented by Sandler et al. [3], is a lightweight deep learning network that works well with fewer parameters and less memory than conventional convolutional networks. It uses techniques such as inverted residual blocks and linear bottlenecks to achieve this efficiency. MobileNetV2 is employed during the initial step in our validation pipeline because it is fast and has low computation costs. Here, the aim is to rapidly weed out non-X-ray images. A heavier model in this case would not improve the speed of the system much, but since MobileNetV2 can screen fast and efficiently, it is a good alternative.

C. *Deep Residual Networks for High-Capacity Feature Learning*

ResNet, proposed by He et al. [4], enabled training of extremely deep neural networks with the help of skip con- nections. These connections enable the flow of information between layers without being lost and therefore aid in pre- venting some typical training problems including vanishing gradients. Consequently, more effective training of deeper networks is possible, and better performance is attained. We apply ResNet18 in two steps of the validation process in this work. It is first used to determine whether the input image is specifically a chest X-ray. It is then applied to categorize the image as normal or abnormal. ResNet18 is efficient in these tasks as it can extract relevant features while remaining computationally efficient.

D. *Language Models for Clinical Report Synthesis*

Large language models have reached maturity, thus enabling a new generation of tools able to generate written clinical prose based on numerical model outputs [6], [7]. This transformation is practically important since raw per-class probability scores are non-actionable even to general users and non-radiologist clinical staff. By posting a structured prompt which encodes the binary pipeline verification outcomes, the per-pathology confidence scores, and the overall severity assessment to Google Gemini, the system generates summaries that present diagnostic results in understandable language without losing the specificity required for clinical decision support.

E. *Transfer Learning in Medical Imaging*

Medical imaging datasets can be small since the

process of data collection and labelling is time-consuming and requires expert knowledge. Consequently, it is hard to train deep learning models on this data alone. To overcome this issue, transfer learning is commonly used. In this method, the models are initially trained on large general datasets and then fine- tuned on medical images. This enables the model to share simple visual features and adjust to the task at hand [9]. We use a two-stage training approach with ResNet18 in our work. In the first stage, the early layers are fixed and only the last layers are trained. In the second stage, some of the deeper layers are gradually unfrozen. This assists the model to learn patterns particular to chest X-rays whilst preserving general features.

III. DATASET AND PREPARATION

A. NIH ChestX-ray14 (CXR8)

The NIH ChestX-ray14 dataset [10] is provided by the Na- tional Institutes of Health Clinical Center and contains 112,120 chest X-ray images collected from 30,805 patients. Each image is labeled for fourteen different thoracic diseases, including Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Oedema, Emphysema, Fibrosis, Pleural Thickening, and Hernia. An important feature of this dataset is that it supports multi- label classification. This means a single X-ray image can have more than one disease at the same time, which reflects real- world medical scenarios where multiple conditions can occur together.

The labels in this dataset are generated from radiology reports using NLP techniques. Because of this, there may be some noise in the labels, which is a known limitation of the dataset. This has been taken into account while analyzing the model performance and AUROC values. The dataset already provides separate train, validation, and test splits. In addition, we created an internal validation set by applying an 80/20 stratified split on the training data. A fixed random seed (42) was used to ensure that the results are consistent and reproducible.

B. Binary Validation Pipeline

Each of the three binary classifiers in the validation pipeline required an independently labelled image corpus. The data_organizer.py utility was developed to construct the six-directory folder structure expected by PyTorch's Image Folder loader: model1_garbage_vs_xray/{xray, garbage}, model2_chest_vs_other/{chest, other}, and model3_normal_vs_abnormal/{normal, abnormal}.

An integrated --validate option is used to verify the dataset before training. It counts the images in each class and highlights any missing or empty directories, helping to detect issues early and avoid unexpected errors during training.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system is organised as a sequential processing pipeline in which each stage has clearly delineated inputs, responsibilities, and outputs. The five stages—input validation, triage classification, multi-label pathology inference, visual explanation, and natural language synthesis—are described below.

A. Standardised Image Pre-Processing

A uniform pre-processing routine is applied to every image, regardless of the model being served: (i) spatial resizing to 224 × 224 pixels using bilinear interpolation; (ii) conversion to a three-channel floating-point tensor; and (iii) channel- wise normalisation with the ImageNet population statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$). Stan-dardising pixel statistics across the input distribution stabilises the gradient flow during stochastic optimisation and reduces the covariate shift introduced by differences in scanner models and exposure protocols across clinical sites.

For the normal-versus-abnormal ResNet18 (Model 3), training additionally incorporated stochastic augmentation: random horizontal reflection, rotations drawn from $U(-10^{\circ}, +10^{\circ})$, affine translations of up to ±10% of image dimensions, and brightness and contrast perturbations of up to ±20% via torchvision.transforms.ColorJitter. These transforms model the intra- and inter-site acquisition variabil- ity that a deployed system will inevitably encounter.

B. Three-Stage Binary Validation Cascade

The cascade acts as a computational gatekeeper, preventing invalid or irrelevant images from ever reaching the pathology classification stage.

Stage 1 – Content Screening (Model 1): A

MobileNetV2 backbone (≈14 MB on disk) performs binary discrimina- tion between radiographic and non-radiographic content. Pho- tographs, scanned documents, and other unrelated images are rejected at this stage with minimal latency.

Stage 2 – Anatomic Specificity (Model 2): A ResNet18 network (≈45 MB) confirms that the accepted radiograph depicts the thoracic region. Images of other body parts— extremities, the skull, the abdomen—are excluded here, pre- venting anatomy-specific mismatches from propagating into the classification stage.

Stage 3 – Normality Triage (Model 3): A transfer-learned ResNet18 (≈45 MB) assigns a preliminary binary clinical label (Normal or Abnormal) to each confirmed chest radiograph. Only images identified as abnormal are passed to the DenseNet-121 model. The result from this step is also included in the LLM input to provide better context for the final explanation. The fine-tuning procedure for this model is model3_normal_vs_abnormal/{normal, abnormal}.detailed in Section IV-D.

C. Pathology Classification: DenseNet-121

The main pathology classification model is DenseNet- 121 [5]. Its dense connectivity structure—where every layer receives the concatenated feature maps of all previous layers within a block—maximises gradient flow during training and feature reuse during inference, producing richer representa- tions per parameter than conventionally connected architec- tures. As depicted in Fig. 1, the network comprises an initial 7 × 7 convolution (stride 2) and max-pool stem, four dense blocks with 6, 12, 24, and 16 composite layers respectively, three transition blocks (each consisting of a 1 × 1 bottle- neck convolution and 2 × 2 average-pooling), global average pooling, and a 14-unit sigmoid output head for multi-label prediction.

1) Per-Disease Operating Thresholds: The model gener- ates independent probability scores for each of the four- teen pathologies. Instead of using one universal threshold, condition-specific decision thresholds were calibrated to the sensitivity–specificity operating points that best suit each dis- ease, informed by the clinical consequences of false negatives relative to false positives. Cumulative detections above their respective thresholds determine whether the overall scan is classified as NORMAL, ABNORMAL, or BORDERLINE; this aggregate assessment is passed directly to the Gemini prompt builder to ensure that the severity framing of the generated report is consistent with the quantitative findings.

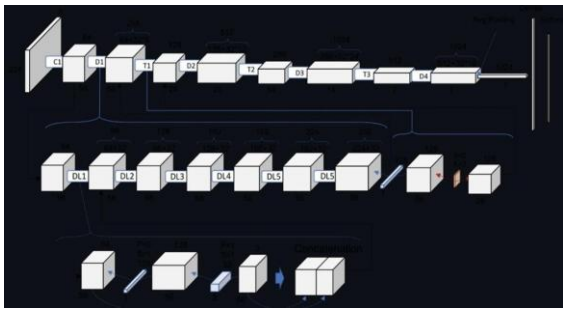

Fig. 1. DenseNet-121 architecture. The upper row shows the macro-level structure: initial convolution (C1), dense blocks D1–D4 interleaved with transition layers T1–T3, and the global average-pooling and classification head. The middle row expands the dense-layer (DL) internal connectivity, and the lower row depicts the 1×1–3×3 bottleneck unit and the feature- concatenation mechanism that characterises each composite layer.

D. Two-Stage Transfer Learning for ResNet18 (Model 3)

A deliberate two-stage fine-tuning protocol was adopted to prevent catastrophic forgetting of the broad visual representations acquired during ImageNet pretraining.

Stage 1 (Epochs 0–5): All convolutional backbone layers were frozen. In the first stage, only the newly added classification head was trained while keeping the rest of the network fixed. This head consists of three fully connected layers: 2048→512, 512→256, and 256→2, each with BatchNorm,

ReLU, and Dropout. A learning rate of 1×10^{-3} was used so that the head could stabilize before fine-tuning the rest of the model.

Stage 2 (Epochs 6–50): In the next stage, layers 3 and 4 of the ResNet18 backbone were gradually unfrozen. The learning rate was reduced to 1×10^{-4} so that the model could slowly adapt to the dataset, while the earlier layers continued to retain basic features like edges and textures.

This approach improved the validation accuracy for normal versus abnormal classification from around 76% using a basic CNN to over 90%. This shows that transfer learning is effec- tive, especially when working with limited medical imaging data.

E. Grad-CAM++ Interpretability Module

In medical applications, it is important to not only

get accu- rate predictions but also understand the regions in the image that influenced those predictions. The HeatmapGenerator module attaches forward and backward hooks to the final convolutional block of DenseNet-121. During inference, Grad- CAM++ [1] computes per-pixel importance weights via a second-order weighting of the back-propagated class activation gradients. The resulting activation map is bilinearly upsampled to the original image resolution and alpha-composited with the input radiograph at a configurable blending factor (default $\alpha = 0.5$). The composite heatmap is base64-encoded and in- cluded in the API JSON response, enabling the React frontend to render the original image and its explanation overlay side- by-side without requiring any additional network round-trips.

F. LLM-Based Clinical Report Generation

Raw numerical confidence scores are uninformative for most end users, including patients and non-radiologist clinical staff. The ExplainabilityAI module addresses this gap by constructing a structured prompt that encodes the binary pipeline outcomes, per-pathology confidence scores, detected conditions, and the aggregate severity level, then submit- ting the prompt to Google Gemini. The returned report is parsed and divided into three labelled sections: *Summary* (one-paragraph clinical overview), *Findings* (per-pathology narra- tive), and *Recommendations* (suggested specialist referrals and follow-up actions).

In the event of Gemini API unavailability—arising from ab- sent credentials, network partition, or service interruption—the module falls back to a pre-authored template report populated with the raw model outputs, ensuring that diagnostic workflow is never fully blocked by an external service failure. All API credentials are stored exclusively in environment configuration files and are never embedded in application source code.

G. Application Interface and Fault Tolerance

The system front-end is implemented as a modular Re- act 18 application comprising seven components: Header; UploadSection (drag-and-drop image ingestion with for- mat validation); BinaryPipelineStatus (real-time per- stage validation feedback); ImagePanel (side-by-side orig- inal and Grad-CAM++ overlay); PathologyResults (14-disease confidence grid); AIExplanationCard (Gemini report renderer); and NearbyDoctors (Geoapify specialist finder).

The NearbyDoctors component maps the dominant de- tected pathology to an appropriate specialist type—for ex- ample, Pneumonia findings map to Pulmonologist clinics, while Cardiomegaly maps to Cardiology services—and queries the Geoapify Places API with browser-supplied geolocation coordinates. When no specialist practice is found within the configured search radius, the component substitutes a Gen- eral Physician recommendation, guaranteeing that every user receives an actionable referral pathway.

The FastAPI backend is designed to handle different types of failures in a reliable way. For example, if the GPU is not available, the system switches to CPU-based inference. If API credentials are missing or invalid, it uses predefined responses. It also handles invalid image inputs by returning clear error messages, and rejects unsupported file formats before processing. This approach ensures that even if some parts of the system fail, it continues to work without crashing completely.

V. RESULTS AND EVALUATION

A. Integration Test Suite

In the integration test suite (test_integration.py), we confirmed five conditions during pre-deployment. This involved ensuring that all the modules were imported, that the trained model files were in the correct places, the FastAPI application loaded successfully, the environment configuration was correct, and that inference on the GPU was functioning correctly. Moreover, Model 3 was pre-trained with a separate validation script. It verified the important details like the expected number of parameters (11.7M), proper freezing and unfreezing of layers, and loading of all the data augmentation steps.

B. DenseNet-121 Training Dynamics

DenseNet-121 was trained on the CXR8 benchmark using train_model.py. The validation AUROC curve showed the characteristic two-phase pattern: a steep initial improve- ment as the network switched from ImageNet-based rep-resentations to chest radiograph-specific characteristics, and then slower, oscillatory convergence. The global optimum was achieved at Epoch 12, with a checkpoint validation AUROC of 0.8780.

C. Per-Pathology AUROC Performance

Table I shows the per-condition AUROC values at the optimal checkpoint, and Fig. 2 plots the ROC curves simulta- neously across all fourteen disease classes.

TABLE I
PER-PATHOLOGY AUROC AT THE BEST CHECKPOINT (EPOCH 12, OVERALL VALIDATION AUROC = 0.8780)

No.	Pathology	AUROC
1	Atelectasis	0.8333
2	Cardiomegaly	0.9434
3	Effusion	0.7848
4	Infiltration	0.9050
5	Mass	0.8628
6	Nodule	0.9614
7	Pneumonia	0.9138
8	Pneumothorax	0.9857
9	Consolidation	0.7665
10	Oedema	0.9005
11	Emphysema	0.8514
12	Fibrosis	0.8507
13	Pleural Thickening	0.7936
14	Hernia	0.9383
	Mean AUROC	0.8779

D. Analysis and Discussion

Pneumothorax had the best AUROC score (0.9857). This is not surprising, because its visual characteristics in a chest X-ray are often clear and easily discernible, like a distinct pleural line and absent lung markings along the edges. Others such as Nodule (0.9614), Cardiomegaly (0.9434), Pneumonia (0.9138), and Infiltration (0.9050) were also good performers because they have unique visual patterns.

Conversely, those that did not perform as well include Con- solidation (0.7665), Effusion (0.7848), and Pleural Thickening (0.7936). These conditions tend to appear similar to each other or even to normal cases and as a result, they are more difficult to distinguish. This is one of the known challenges of this

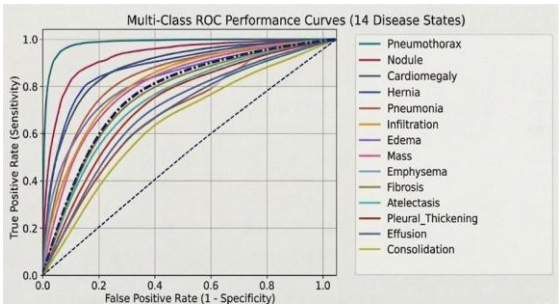

Fig. 2. Multi-class ROC curves for fourteen thoracic conditions (mean AUROC = 0.8779). Pneumothorax (teal, AUROC = 0.9857) and Nodule (red, AUROC = 0.9614) achieve the sharpest separation from the chance diagonal. Consolidation (yellow-green, AUROC = 0.7665) and Effusion (dark blue, AUROC = 0.7848) exhibit more modest separation, reflecting the known visual similarity of these conditions with adjacent findings and normal variants.

dataset [10]. Future research might concentrate on employing superior feature extraction algorithms or using multiple image views.

The model on the whole had a mean AUROC of 0.8779, which is comparable to other publications that employ the same method on the same dataset. Moreover, the binary classification phase with the help of ResNet18 increased the accuracy to approximately more than 90% compared to the initial 76%. This demonstrates that transfer learning can substantially enhance performance even without altering the model architecture or adding extra data.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, an end-to-end Chest X-Ray Diagnostic Assistant that integrates various elements into a complete opera- tional system is presented. It has a multi-step validation system, a DenseNet-121 disease detection model, a Grad-CAM++ visual explanation module, a Google Gemini report generation system, and a doctor recommendation system powered by Geoapify.

The validation stages ensure that only correct chest X- ray images are processed, helping to minimize redundant computation and keeping the system efficient even on com- mon hardware. Moreover, the system is capable of managing various failures. It does not halt but rather continues to operate with fallback options, making it more appropriate for real-life use.

There are a few potential directions for improvement that can be considered in future research:

- Multi-view input fusion: It is possible to extend the model to use both front and side-view X-ray images. This would enhance detection of conditions that are more easily detected at other angles.
- Federated model training: It is possible to train the model across multiple hospitals without disclosing raw patient data, which could achieve better and more generalised performance.
- Longitudinal change detection: Comparing the current X-rays of a patient to their previous scans may aid in tracking disease progression and generating more meaningful reports.
- Prospective clinical testing: Evaluating the system in an actual hospital setting should assist in measuring the effect of this system on diagnosis time, workload, and patient outcomes.
- Edge and mobile deployment: The model can be optimized to run on smartphones, enabling it to be useful in rural or resource-limited regions.
- Regional language support: Reports in local languages can be generated to make the system more accessible to patients and healthcare workers.

REFERENCES

[1] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Applications Com- put. Vis. (WACV)*, Lake Tahoe, NV, USA, pp. 839–847, Mar. 2018.

[2] 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 *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, pp. 618–626, Oct. 2017.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, pp. 4510–4520, Jun. 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, pp. 770–778, Jun. 2016.

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, pp. 4700–4708, Jul. 2017.

[6] Google DeepMind, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, Dec. 2023.

[7] T. B. Brown *et al.*, "Language models are few-shot learners," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[8] 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," *Quant. Imag. Med. Surg.*, vol. 4, no. 6, pp. 475–477, Dec. 2014.

[9] Y. Qin, S. Zheng, J. Huang, G.-S. Xie, and J. Yang, "ResNet-based architecture for thoracic disease detection in chest X-rays," *IEEE Access*, vol. 9, pp. 56–64, 2021.

[10] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, pp. 2097–2106, Jul. 2017.