TorchXRayVision

Cindy Lee, Ferhat Yilmaz, Wellington Cunha

Presentation Video

Demo Notebook

Technology has been a game-changing factor in the healthcare space for quite a time. Most recently, with the increase in computational resources and the huge amount of data available, the possibilities around the use of technology have reached a never seen state, where the potential applications surpasses, by far, the number of human resources (researchers, developers, among others) available.

One of the core areas that has been widely explored is in the use of technology in helping practitioners (doctors, radiologists, physical therapists, etc.) in assessing and getting information from patient data. By patient data, anything that is already available (such as biometrics information) to data produced for specific purposes (like blood and other types of tests) can be used to aid practitioners in providing the best outcome (treatment) for a patient.

In general, they aim to augment the capacity of practitioners by searching through that data, providing them with hints and tips on where to look first, potential causes and diagnosis, etc. The idea is not to replace humans, but there are already several examples of applications being more accurate than experienced doctors when diagnosing certain types of diseases based on the same available data (Richens et al., 2020).

One of the projects with that goal is TorchXRayVision, an open source software library that can be implemented in applications and help doctors and radiologists in the analysis of chest X-rays. But beyond that, it can also be used by other researchers, in understanding the best way of dealing with this DICOM (Bidgood et al., 1997) image type.

Among the goals of the project, we can highlight two. First one is to unify different deep and machine learning projects that have the goal of diagnosing lung related diseases using chest x-ray in one unique place, creating an abstraction layer that allows the use of their datasets with the same interface (Cohen et al., 2021).

But the main goal is to use all of those projects and all the knowledge created by them along with the data they deal with to create a model trained with all of the data available in each one of those projects.

Following a description of the datasets/projects currently implemented:

- **NIH ChestX-ray8**: contains over 100,000 frontal anonymized chest x-ray images from more than 30,000 patients. The images are all annotated and labeled, covering eight lung-related diseases (Wang et al., 2017).
- **PadChest**: almost 160,000 high resolution images from over 67,000 patients. Nineteen different diagnoses are annotated on images (Bustos et al., 2020).
- **RSNA Pneumonia Detection Challenge**: over 25,000 frontal anonymized chest x-ray. The project focuses on pneumonia, so, this is the only label available on the annotated images (Shih et al., 2019). The number of patients is not informed, but we are assuming that it contains one image per patient.
- **NIH Google**: this is a subset of NIH ChestX-ray14 dataset containing over 30,000 images (one per patient) re-labelled and re-annotated by experienced radiologists for four diseases (Majkowska et al., 2019). Annotation itself was part of the project.
- **CheXpert**: 224,316 images (not annotated) from 65,240 patients covering fourteen different diagnosis on text labels only (Irvin et al., 2019)
- **MIMIC-CXR**: contains 337,110 non-annotated images from 227,827 studies (we are assuming that each study corresponds to a patient) and labels for fourteen different diagnostics that were extracted from respective medical reports using Natural Language Process (Alistair et al., 2019)
- **OpenI**: dataset with 7,470 chest x-rays along with 3,995 reports (we are also assuming each report corresponds to a patient). The images are not annotated and eighteen labels were manually extracted from the respective reports (Demner-Fushman et al., 2012)
- **SIIM-ACR Pneumothorax Segmentation**: it was a Kaggle challenge posted by the Society for Imaging Informatics in Medicine (SIIM). It contains 12,954 non-annotated images for one label (pneumothorax). Although non-annotated, the label contains details that can be used to annotate the images with the exact location of masks on the images (Society for Imaging Informatics in Medicine (SIIM), 2019)
- **VinBrain**: the project contains 18,000 images (one per patient) that were annotated by radiologists (as part of the project). There are 22 localized labels - for local lesions, fibrosis, edemas, etc. - and 5 global labels - for diagnostics like tumor, pneumonia, tuberculosis, etc. (Nguyen et al., 2022) that were “translated” to fourteen (out of eighteen) labels used on TorchXRayVision

Below is a summary of the datasets:

Dataset	Images	Patients	Original Labels	Annotated images
NIH	112,120	32,717	8	Yes
PadChest	158,626	> 67,000	19	Yes
RSNA Pneumonia Challenge	26,684	Not informed	1	Yes
Google	30,805	30,805	4	Yes
CheXpert	224,316	65,240	14	No
MIMIC-CXR	337,110	227,827	14	No
OpenI	7,470	3,995	18	No
SIIM-ACR	12,954	Not informed	1	No
VinBrain	18,000	18,000	22 + 5	Yes

The Current version of TorchXRayVision uses almost a million images to train its disease prediction models. Around 350 thousands of those images are annotated and were used to train the models responsible for extracting the features (autoencoder).

According to Cohen, Hashir, Brooks & Bertrand (2020), the main challenge when implementing models using x-rays from different datasets for prediction happens because of the labels, and not the images. As each practitioner has his/her own background, and procedures in different locations and even hospitals vary a lot, uniforming those labels (diagnosis) is what makes it hard to use those datasets altogether.

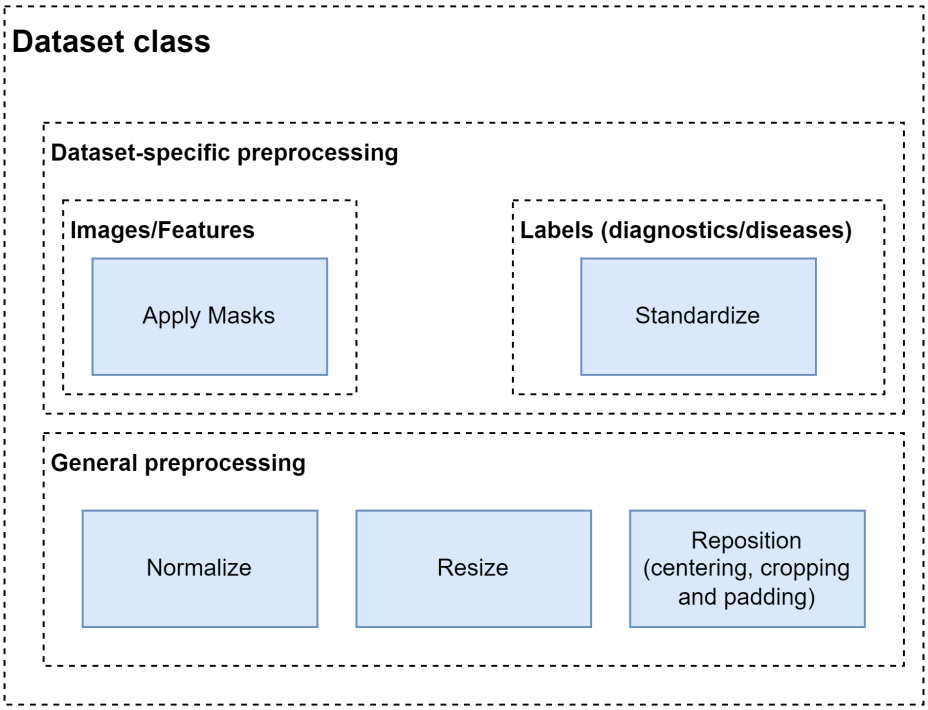
There are other 5 datasets available in the library, totalling 14, but their implementations are still in progress (one of them was not even in the paper). VinBrain dataset was recently added to the datasets used to train the autoencoder model, although it is not yet described in any paper.

Besides the overall model trained with all images, there are models trained using each dataset and their labels (translated to TorchXRayVision standards) using the same architecture.

Architecture

As previously mentioned, the two main goals of the TorchXRayVision project are (1) creating an unique interface for chest x-ray images from other projects and (2) attempting to create a predictive model using all of the available dataset. To achieve that, the project created a clear segregation between datasets (and everything needed to deal with them) and models (predictive and autoencoder). Below we provide information about each one of those three modules.

Datasets module/class



The datasets module, that implements a superclass on the library, contains all the functions required to deal with the dataset in preparation for both training and predicting. One important point is that it does not acquire any dataset used for training: datasets should already be available on the local machine that will be training the models.

We can break the dataset functions in two main categories. The general pre-processing functions, that are applicable to any dataset being used for training and any image used for prediction, and the dataset-specific preprocessing functions, that are available for each one of the dataset covered by the project.

General pre-processing

In the general pre-processing tasks, we have the **normalization** function, that converts the image to be between [-1024, 1024] pixels and converts the color channel from the grayscale channel to RGB. Here it is important to point out that the function only scales the original dataset pixels to the arbitrarily defined pixel range and does not use min/max to rescale that. If the pixel values of a dataset are, for instance, in between 0 and 65,536 (16-bit image), those values will be rescaled between -1024 and 1024.

The next function is to **resize** the image. Models were trained using a size of 224 x 224, but there is one model using 512 x 512 (ResNet) and the option to resize them to 1024 x 1024, but no reference on training using this size. We are assuming the different sizes are still under development or were attempts that the authors decided not to follow. Images are resized using OpenCV pip package applying bilinear interpolation option.

The third most important preprocessing task can be called by **reposition**. This function centers the image, cropping and padding it, to ensure they will start with exactly the same size (224 x 224).

In predicting mode, the three functions above are automatically applied to the image (they are called by the model class). Also, those functions above contain parameters to be used to augment the data when in training mode.

Dataset-specific preprocessing

In this category of functions, we have one (big) class per dataset covered and those classes deal with preprocessing tasks that pertains to that dataset. As explained above, one of the biggest challenges when dealing with medical datasets is the inconsistency among the labels (diagnostics), even inside the datasets.

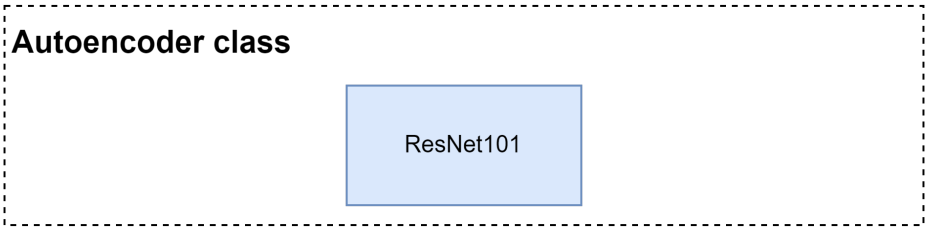
In order to deal with this, each dataset class contains a **standardize** function to “translate” original labels to the eighteen implemented in the project. If one of the pathologies doesn't exist on the dataset, a NaN value will be assigned to it. So, mainly if it was not possible to translate the original diagnostics into one of the eighteen covered by TorchXRayVision, it will be ignored. And the same happens if the original label brings a disease not covered in the project: it will be ignored.

The other important dataset-specific preprocessing function is to **apply masks** to the images for the five annotated datasets. Those masks would be used later to train the autoencoder model that will be responsible for extracting the features for all images.

The dataset-specific preprocessing tasks only happen during training.

Autoencoder module/class

The Autoencoder module has also its own class to be used during prediction. The function of this module is to use the images from the five datasets containing annotations (pathology masks), with the preprocessing tasks applied, to train a ResNet101 (He, Zhang, Ren & Sun, 2016) to extract the features from non-annotated images. It applies the same data augmentation techniques that are applied during training the model (described below).



The trained autoencoder model is then used to extract features from all the images both during training and prediction. For instance, when predicting, Models class will apply it automatically to the image being assessed, as it does during training with the preprocessing tasks.

Other architectures were attempted to create the autoencoder, but the one that provided the best results, according to the authors of TorchXRayVision, was ResNet101 without any change, besides the ones required to deal with the variable number of input and output channels.

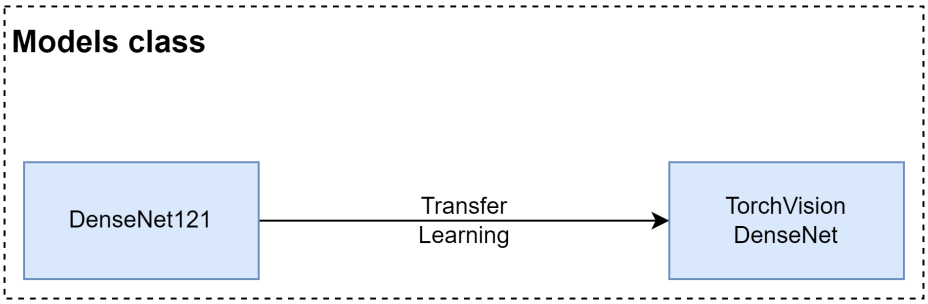
It is important to bring up the fact that lungs are composed of several overlapping layers of tissue that, if extended, would cover a tennis court (for a regular adult). As the images being analyzed are 2D frontal images, a feature may be easily “visible” because they are present on the highest layer, while another similar feature (like a nodule with the exact same size) that is present in deep layers will be hidden.

Creating and training a model to extract the pieces of information that could be hidden underneath several layers of a 2D image and exaggerating the hidden features in order to give them similar weights (Cohen et al., 2021) as a feature present on the shallow layers was the solution found by TorchXRayVision to deal with this challenge.

Models module/class

Models module trains the nets and encapsulates the models for prediction on the Models class of TorchXRayVision library. On training, it transfers the learning of

DenseNet121 (Huang, Liu, van der Maaten & Weinberger, 2017) to create the TorchVision DenseNet (as named by authors).



Besides some adjustments to allow variable number of input channels, it adds a normalization layer, a linear layer and finally a sigmoid function, outputting a vector with the probabilities for the eighteen diseases covered. Below is an example of the output:

```
{'Atelectasis': 0.32797316,
 'Consolidation': 0.42933336,
 'Infiltration': 0.5316924,
 'Pneumothorax': 0.28849724,
 'Edema': 0.024142697,
 'Emphysema': 0.5011832,
 'Fibrosis': 0.51887786,
 'Effusion': 0.27805611,
 'Pneumonia': 0.18569896,
 'Pleural_Thickening': 0.24489835,
 'Cardiomegaly': 0.3645515,
 'Nodule': 0.68982,
 'Mass': 0.6392845,
 'Hernia': 0.00993878,
 'Lung Lesion': 0.011150705,
 'Fracture': 0.51916164,
 'Lung Opacity': 0.59073937,
 'Enlarged Cardiomediastinum': 0.27218717}
```

Before training, the data was augmented by randomly rotating the images up to 45 degrees, translated (shift right or left and up or down) by up to 15% and scaled larger or smaller up to 10% of the size.

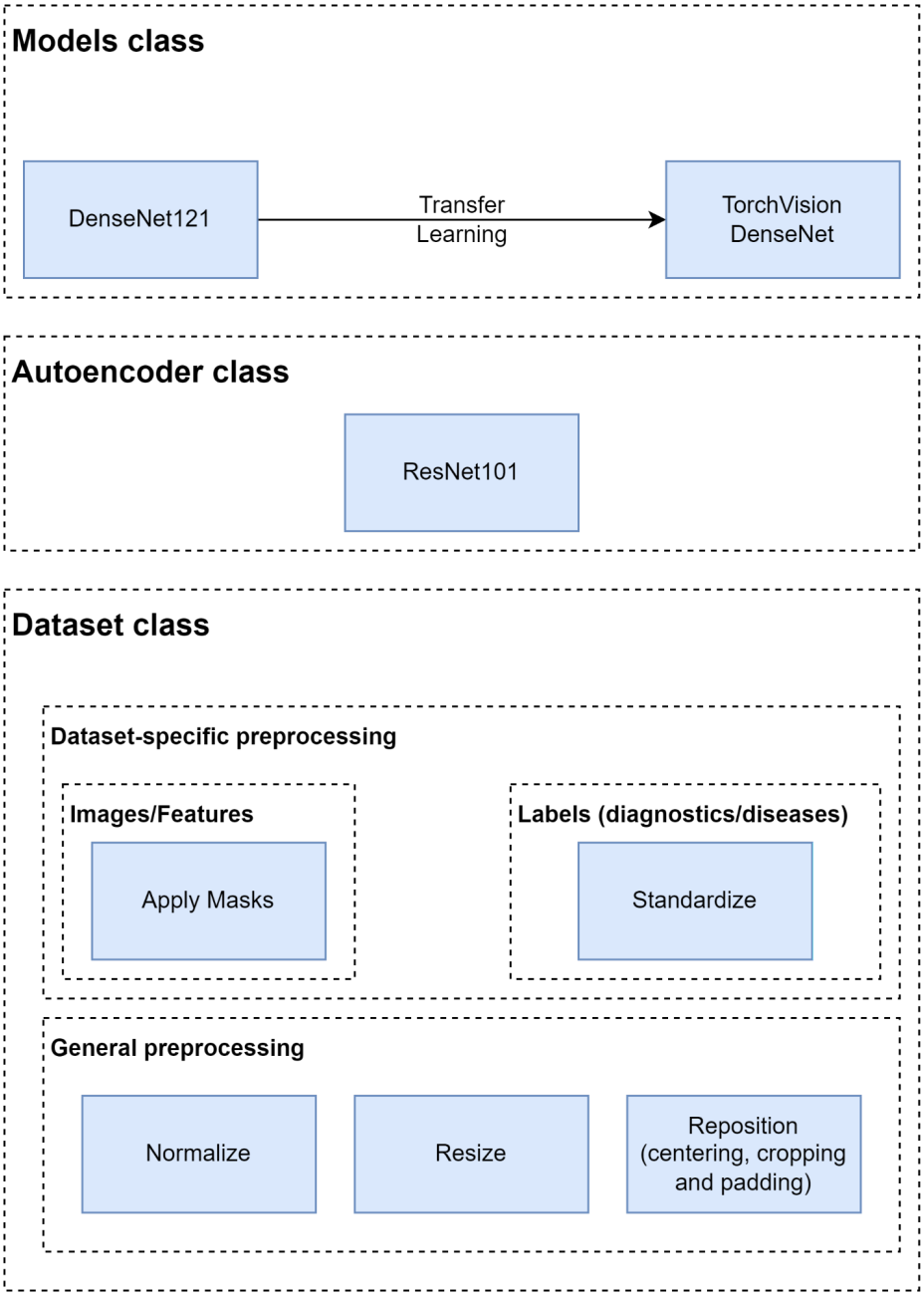
One important fact to consider here is that the datasets were unbalanced, with way more observations with no pathologies than with pathologies and unbalanced even among the eighteen pathologies being covered.

As explained before, besides training a model using all images, it also trains models for each one of the individual datasets, using the same network. Models are saved and they can be loaded by calling a function passing parameters for the net, the image size and the dataset. For instance, “densenet121-res224-all” is the parameter for the model

trained with all images of size 224 using densenet121 and “densenet121-res224-chex” is the parameter for the model trained with ChexPert images of size 224 using the same densenet121.

TorchXRayVision library

Below the overall architecture of the library:



Innovations

TorchXRay Vision is an excellent open-source python library that provides a unified and singular API that contains numerous datasets for chest X-rays and deep learning models. By unifying all the datasets, the authors have made it easier to facilitate evaluating models on data from multiple sources which will help prevent bias that may be present when only using a singular dataset.

Access to datasets is encapsulated in TorchXRayVision, so one does not need to have details on how to get access to them to start using them. They just need to be downloaded and placed on a folder structure defined inside the documentation. Models will automatically apply the preprocessing tasks and extract the features using autoencoder-trained models when predicting. Autoencoders can also be used alone to extract the features for use on other projects and the functions contained in the datasets class for preprocessing can also be used alone.

When working with chest X-ray datasets there is a lot that must be done, such as preparing data classes to adapt to heterogeneous structures, formats, labels, and source-specific fields. This is even more difficult when integrating multiple datasets in projects for applications like domain adaptation, continual learning, and transfer learning. TorchXRay Vision remedies this as the standardization of data handling, preprocessing and corresponding standard baselines and pre-trained models establishes a reusable framework that can be used for reproducible research and consistent baseline experiments.

The idea of merging different datasets and projects into one is innovative in itself. The pre-trained models contained in TorchXRay Vision have already been used in a host of different applications. The reuse of the knowledge acquired on projects, both technical and functional, is what we call transfer learning. This library has already been used and referenced in papers related to transfer learning to related chest x-ray tasks like patient severity scoring (Gomez et al., 2020) and patient clinical trajectory predictions (Maurya, 2020). The pre-trained models provided by TorchXRay Vision have also been used as feature extractors of images for multi-modal models (Delbrouck et al., 2021).

Being able to produce better x-ray prediction models in the future will be very helpful in the medical field as it is more economical and more available in comparison to a CT scan. Although models right now cannot replace clinical judgment, they can be very helpful if used complementary to a clinician’s expertise to improve triaging and quarantining decisions. Being able to predict severity is important in helping hospitals understand whether the resources that they currently have can support the patient’s status.

Future applications and enhancements

The pre-trained models for the eighteen diseases are ready to be implemented on solutions, such as an application where the x-rays of a patient are uploaded and it presents the probability of each one of the eighteen diseases for that patient to the practitioner. But enhancements to the solution and other applications can still be made.

Add new datasets

As per documentation, five new datasets are being currently implemented (but only three were found on the code base). They can be used to increase the knowledge of the subject, enhance the autoencoder (if annotated) and retrain the model that used all datasets.

However, adding them, besides all the work in adjusting the labels, will also require the re-assessment of architecture and some decisions already made.

Add new labels / diseases

Another enhancement is to increase the number of diseases (labels) covered. That also would require a complete review of existent models. All of the three datasets that are in the implementation phase are enhancing the library in this direction, as per the table below:

Dataset	Images	Patients	Original Labels	Annotated images
COVID-19	679	412	COVID, MERS, SARS and ARDS	No
NLMTB	800	800	Tuberculosis	No
Stonybrook COVID	2373	552	2 local + 1 global (COVID)	No (but metadata can be used to add masks in macro regions)

The idea presented on VinBrain (and now in Stonybrook) of creating two categories of labels (local and global) is also something that can potentially be explored and would, for sure, require a review on the whole library, including retraining all models (even the ones trained with specific datasets).

Another idea on dealing with labels would be outputting the images with the annotations (masks) using the autoencoder class in order to help practitioners by pointing out specific places for them to look at (like an attention mechanism for humans).

Use additional “features”

The idea here is to use more data in order to increase the performance of models. Some of the datasets already have demographics (gender, age, etc.) and biometrics (height, weight, race/ethnicity) information. Other types of information could also be explored, such as the sound of patients breathing, the results of blood samples, blood pressure measures.

The information collected by wearable devices (such as FitBit and Apple Watch) and their respective applications are other sources of valuable information. Devices like that capture sleep patterns, heart rate, calories and their respective applications can be used to track things like food and water intake, smoking habits, weight. All of that information could be used to augment the image data and provide a most comprehensive analysis of all factors that impact the development of diseases.

Stonybrook COVID dataset, which is currently being implemented, has, for most patients, more than one x-ray taken at different points in time. So, for those patients, the evolution of disease can be also understood and, maybe, that information could be used to train models to detect diseases at early stages.

Medical images framework

This enhancement has more to do with processes than with technology itself. The idea of TorchXRayVision of aggregating different projects into one library, that uses the data already available for each of them and, maybe more important, the lessons learned in each one of them could be extended to other types of DICOM, such as ultrasound, mammograms, CT scans, among others. And for that, a framework for the implementation, which would include base classes and scripts that could be reused, along with instructions on how to proceed, could be very valuable.

References

Alistair, J. E. W., Pollard, T. J., Greenbaum, N. R., Lungren, M. P., Deng, C., Peng, Y., Lu, Z., Mark, R. G., Berkowitz, S. J., & Horng, S. (2019, December 12). MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs. *Scientific data*, *6*(1), 317. https://doi.org/10.1038%2Fs41597-019-0322-0

Bidgood, W. D., Horii, Jr, S. C., Prior, F. W., & Van Syckle, D. E. (1997, May 1). Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. *Journal of the American Medical Informatics Association*, *4*(3), 199-212. 1997.0040199

Bustos, A., Pertusa, A., Salinas, J.-M., & de la Iglesia-Vayá, M. (2020, December). PadChest: A large chest x-ray image dataset with multi-label annotated reports,. *Medical Image Analysis*, *66*. https://doi.org/10.1016/j.media.2020.101797

Cohen, J. P., Brooks, R., En, S., Zucker, E., Pareek, A., Lungren, M. P., & Chaudhari, A. (2021, August). Gifsplanation via latent shift: a simple autoencoder approach to counterfactual generation for chest x-rays. *Medical Imaging with Deep Learning*, 74-104. https://openreview.net/forum?id=rnunjvgxAMt

Cohen, J. P., Hashir, M., Brooks, R., & Bertrand, H. (2020, September 21). On the limits of cross-domain generalization in automated X-ray prediction. *Medical Imaging with Deep Learning*, 136-155. https://arxiv.org/pdf/2002.02497.pdf

Cohen, J. P., Viviano, J. D., Bertin, P., Morrison, P., Torabian, P., Guarrera, M., Lungren, M. P., Chaudhari, A., Brooks, R., Hashir, M., & Bertrand, H. (2021, October 31). TorchXRayVision: A library of chest X-ray datasets and models. 1-14. https://arxiv.org/pdf/2002.02497.pdf

Demner-Fushman, D., Antani, S., Simpson, M., & Thoma, G. R. (2012, June). Design and development of a multimodal biomedical information retrieval system. *Journal of Computing Science and Engineering*, *6*(2), 168-177. http://lhncbc.nlm.nih.gov/system/files/pub2012019.pdf

Delbrouck, Jean-Benoit & Zhang, Cassie & Rubin, Daniel. (2021). QIAI at MEDIQA 2021: Multimodal Radiology Report Summarization. 285-290. 10.18653/v1/2021.bionlp-1.33.

Gomes, D. P., Horry, M. J., Ulhaq, A., Paul, M., Chakraborty, S., Saha, M., ... & Rahaman, D. M. (2020). MAVIDH Score: A COVID-19 Severity Scoring using Chest X-Ray Pathology Features. arXiv preprint arXiv:2011.14983.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778. https://arxiv.org/pdf/1512.03385.pdf

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700-4708. https://arxiv.org/pdf/1608.06993.pdf

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., Seekins, J., Mong, D. A., Halabi, S. S., Sandberg, J. K., Jones, R., Larson, D. B., Langlotz, C. P., Patel, B. N., Lungren, M. P., & Ng, A. Y. (2019, January 21). CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. *Proceedings of the AAAI conference on artificial intelligence*, *33*(01), 590-597. https://doi.org/10.48550/arXiv.1901.07031

Majkowska, A., Mittal, S., Steiner, D. F., Reicher, J. J., McKinney, S. M., Duggan, G. E., Eswaran, K., Chen, P.-H. C., Liu, Y., Kalidindi, S. R., Ding, A., Corrado, G. S., Tse, D., & Shetty, S. (2019, December 3). Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation. *Radiology*, *294*(2), 421-431. https://doi.org/10.1148/radiol.2019191293

Maurya, A. (2020). Predicting intubation support requirement of patients using Chest X-ray with Deep Representation Learning. arXiv preprint arXiv:2011.01787.

Nguyen, H. Q., Lam, K., Le, L. T., Pham, H. H., Tran, D. Q., Nguyen, D. B., Le, D. D., Pham, C. M., Tong, H. T.T., Dinh, D. H., Do, C. D., Doan, L. T., Nguyen, C. N., Nguyen, B. T., Nguyen, Q. V., Hoang, A. D., Phan, H. N., Nguyen, A. T., Ho, P. H., … Vu, V. (2022, July 20). VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations. *Scientific Data*, *9*(1), 1-7. https://www.nature.com/articles/s41597-022-01498-w

Richens, J. G., Lee, C. M., & Johri, S. (2020, August 11). Improving the accuracy of medical diagnosis with causal machine learning. *Nature Communications*, *11*(1). 10.1038/s41467-020-17419-7

Shih, G., Wu, C. C., Halabi, S. S., Kohli, M. D., Prevedello, L. M., Cook, T. S., Sharma, A., Amorosa, J. K., Arteaga, V., Galperin-Aizenberg, M., Gill, R. R., Godoy, M. C.B., Hobbs, S., Jeudy, J., Laroia, A., Shah, P. N., Vummidi, D., Yaddanapudi, K., & Stein, A. (2019, January 30). Augmenting the National Institutes of Health Chest Radiograph Dataset with Expert Annotations of Possible Pneumonia. *Radiology: Artificial Intelligence*, *1*(1). https://doi.org/10.1148/ryai.2019180041

Society for Imaging Informatics in Medicine (SIIM). (2019, August). *SIIM-ACR Pneumothorax Segmentation*. Kaggle. Retrieved November 25, 2022, from https://www.kaggle.com/competitions/siim-acr-pneumothorax-segmentation/overview

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2097-2106. https://openaccess.thecvf.com/content_cvpr_2017/papers/Wang_ChestX-ray8_Hospital-Scale_Chest_CVPR_2017_paper.pdf