Chest X-ray Image Classification Using Transfer Learning and GANs

Ziad Ali
Electrical and Computer Engineering
North Carolina State University
Raleigh, NC, USA
zaali@ncsu.edu

FNU Vivek
Computer Science
North Carolina State University
Raleigh, NC, USA
vvivek@ncsu.edu

Ravindersingh Rajpal
Computer Science
North Carolina State University
Raleigh, NC, USA
rkrajpal@ncsu.edu

***Abstract*—Chest X-rays are among the most commonly used tools for diagnosing lung diseases by clinicians; however, X-ray images can be difficult to interpret, and radiologists can struggle to distinguish which specific disease pathologies are present within a scan. In this work, convolutional neural networks are trained on the NIH chest X-ray dataset, consisting of fourteen different thoracic pathologies, and used to classify diseases present within scans. A basline model is built from scratch and its results are compared to those of two models built using transfer learning. The models are then re-trained using images generated through standard augmentation approaches (e.g. flipping, rotation) as well as generative adversarial networks (GANs). The transfer learning models perform better than the baseline model in almost all cases, except when a certain disease does not have many positive images. Standard augmentation has a limited effect on improving performance, while GAN-generated images can significantly boost performance if trained using a disease classification with many images in the dataset.**

I. Introduction

X-rays are used frequently in medical imaging because they are cheap, fast, simple, and provide relatively high-resolution images [1]. In the US, approximately 200 million X-ray exams are conducted each year [2], compared with 80 million CAT scans [3] and 35 million MRIs [4]. While they are used ubiquitously, there are several disadvantages to X-rays that limit their efficacy; primarily, X-rays do not generate scans as detailed as other imaging techniques and do not obtain 3D information, meaning 2D images can be obscured and hard to interpret [5].

X-rays are commonly used to diagnose thoracic (lung) diseases in particular. There are many different types of lung diseases, and over 35 million Americans suffer from a chronic lung disease [6]. Lung diseases are the third leading cause of death in the United States, and lung cancer is the leading cause of cancer deaths among both men and women [7]. However, clinicians are not always able to properly diagnose thoracic pathologies using conventional methods - in [8], Rajpurkar et al. demonstrated that radiologists obtained an F1 score of only 0.387 when tasked with diagnosing X-ray scans of lungs with pneumonia. However, in that same study, a machine learning model (CheXNet), was able to achieve an F1 score of 0.435. This demonstrates that neural networks and deep learning can be used to aid clinicians in performing diagnoses even when using limited data of the type provided by an X-ray.

In this work, the feasibility of applying deep learning to the task of diagnosing thoracic diseases was examined. In particular, models were trained and evaluated on scans obtained from the NIH Chest X-ray Dataset of 14 Common Thorax Disease Categories [9]. This dataset contains images of X-rays obtained from patients with one or more of atelectasis (11559), cardiomegaly (2776), consolidation (4667), edema (2303), effusion (13317), emphysema (2516), fibrosis (1686), hernia (227), infiltration (19894), mass (5782), nodule (6331), pleural thickening (3385), pneumonia (1431), and pneumothorax (5302). The number in parentheses indicates the number of images associated with the pathology.

Convolutional neural networks (CNNs) were developed for this task both from scratch (our baseline model) and using pre-trained weights and architectures (transfer learning). In addition, baseline models were re-trained using data obtained from a generative adversarial network (GAN). A GAN consists of a generator and a discriminator - the generator maps features of training images to a latent space and samples that latent space to generate images. The discriminator distinguishes which images look ”realistic”, forcing the GAN to learn to produce better images over time [10]. The amount of data required for training GANs plays a crucial part in training time and quality of artificial images. We discuss the ramifications of training data on GANs in Section III.

The F1 scores and AUC (area under curve) scores for the different models are compared. In addition, limitations of the dataset and future directions for this research are discussed.

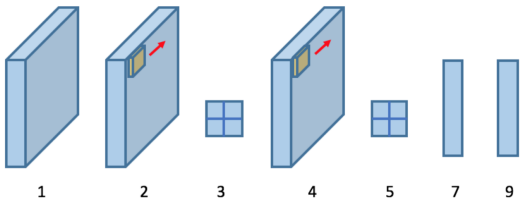
II. Methodology

A. Baseline Model

To begin with the task of classification, we decided to create a baseline CNN model and observe its performance on the given dataset. Since our dataset is a multi-label dataset, i.e. each X-ray image can be related to one or more disease category (a total of 14 categories), for the baseline model, we created 14 different binary classifiers with approximately the same architecture (but some different hyper-parameters). Binary classifiers were used instead of a general, multi-class and multi-label classifier because there were not enough training images for a single classifier with a simple architecture to perform multi-label prediction. Each classifier gave the

probability of that disease both being associated with and not associated with the given X-ray image (a positive classification was given if the disease probability was higher than the no-disease probability). The general structure for each binary classifier is shown in Fig. 1.

The input images were 128x128 pixels with 3 channels (RGB). The first convolutional layer applied 32 3x3 filters followed by 2x2 max pooling, while the second convolutional layer applied 64 4x4 filters followed by 2x2 max pooling. The outputs of that layer were fed into a dropout layer, followed by a fully connected layer with 1024 neurons, followed by another dropout layer, and finally into 2 output neurons corresponding to probabilities that the image does not depict a certain disease and that it does.



Layer	Dimensions
1) Input Image	128x128x3
2) Convolutional Layer – 32 3x3 filters, stride = 1, ReLU activation	128x128x32
3) Max Pooling Layer – 2x2	64x64x32
4) Convolutional Layer – 64 4x4 filters, stride = 1, ReLU activation	64x64x64
5) Max Pooling Layer – 2x2	32x32x64
6) Dropout Layer	32x32x64
7) FC Layer	1x1024
8) Dropout Layer	1x1024
9) FC Layer	1x2

Fig. 1. Architecture used for baseline convolutional neural network. The CNN is a binary classifier that predicts whether or not a single disease pathology is present in a chest X-ray.

B. Transfer Learning

To develop a single model that could perform multi-class and multi-label classification, a pre-initialized CNN architecture with more layers was needed. Ideally, this architecture would already have robust feature-detection built in due to being trained on previous large image datasets (not X-rays).

It has been noted that using a model pre-trained on a certain image set for a different application (transfer learning) has many advantages [11]. The two primary motivations behind transfer learning are to reduce training time and improve performance. For this study, two different pre-trained models, obtained from Francois Chollet [12] and Keras [13], were used and compared. Since our project deals with 14 different classes that are each Bernoulli-distributed variables, we chose to use a binary cross-entropy loss function and to apply sigmoid activation to our final output layer. The specific pre-trained networks used are now discussed.

1) Inception V3: Inception V3 is a 42-layer deep neural network created by Google [14]. This model was introduced in 2016 in ILSVRC (ImageNet Large Scale Visual Recognition Competition). While it is common, when using Inception V3, to freeze the pre-trained layers and only train added layers (which format the output to the number of classes for a specific application), we decided to train the whole model at once since we had approximately 51k images to use. We initialized the model with the pre-trained weights from the 'Imagenet' dataset, and added one global pooling layer along with one dropout and a fully connected layer as the output layer.

2) VGG16: The VGG network architecture was introduced by Karen Simonyan and Andrew Zisserman in 2015 [15]. VGG16 is a 16-layer deep neural network model created for the task of image classification. While Inception V3 was trained without freezing the layers, VGG16 was trained by freezing all the VGG16 model layers except for the last 4. The accuracy, recall, precision, F1 score, and AUC for both the VGG16 and Inception V3 augmented networks were measured and compared.

C. Generative Adversarial Networks

Our motivation to use Generative Adversarial Networks (GANs) is the limitation of the amount of data for each disease class. As will be discussed in Section IV, attempts to train models on augmented data obtained through simple image transformations did not demonstrate significant performance improvements. The benefit of using GANs is that they generate augmented images different from the original, yet nearly as good in terms of quality.

GANs are a type of unsupervised deep neural network. As proposed by Goodfellow et al [16], GANs consist of two key components, a generator and a discriminator. The optimization function of the GANs is based on the generator learning a latent space distribution of the input images that closely matches their real properties. The generator then samples from this latent space to generate images which it uses to try to "fool" the discriminator, which is trained on the real images.

The biggest problem with the original GAN concept is that it is difficult to train, and they have sometimes been associated with producing nonsensical output. Deep Convolution Generative Adversarial Networks (DCGANs) [17] were introduced by Radford et al. in an effort to train more stably than the original GANs. However, one prominent problem with DCGANs is that the representation remains entangled; meaning the latent code is not separable from the noise. InfoGAN [10] tries to solve this problem and provide a disentangled representation. InfoGAN tries to learn an input code that has a consistent effect on output. InfoGAN splits the generator input into two parts: a noise vector and a latent code. The latent code is then made meaningful by maximizing mutual information between the code and generator output. In essence, InfoGAN adds a regularization term to the original GAN's objective function. However, this is implemented as a lower bound approximation of mutual information. We use a Keras implementation of InfoGAN provided by Github user tdeboissiere [18].

III. Training and Hyperparameters

The NIH dataset used for this task had a very skewed data distribution. Apart from the data being multi-label and multi-class, out of approximately 110K images, 60K images were

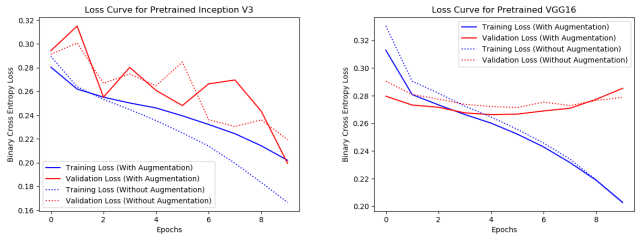


Fig. 2. Loss Curve - InceptionV3 and VGG16

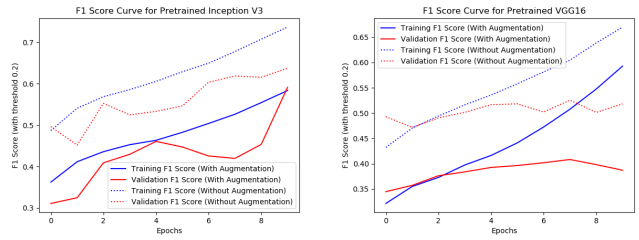


Fig. 3. F1 Curve - InceptionV3 and VGG16

labelled as 'No Finding' which we interpreted as not having a proper label or being a healthy person's X-ray. We removed those images and performed training on solely the labeled images.

A. Training with Original Dataset

Initially, training on the baseline and pre-trained models was performed only with the original images from the NIH dataset. Since the data wasn't equally distributed for each disease class, bias in the results was expected. The performance of the baseline model, pre-trained Inception-V3 model, and pre-trained VGG16 model on the original dataset can be seen in Table II.

B. Data Augmentation

After training with the original dataset, certain classes (e.g. pneumonia, hernia, fibrosis) demonstrated poor results. Given that these classes were among those with the fewest images, we hypothesized that our model failed to classify them well due to a lack of training data. To address this issue, simple image augmentation was attempted using the Keras Image generator [19]. We introduced left/right shifts and zooms-in/out of the images for every class in a weighted fashion.

From Fig. 2 and Fig. 3, we can see that data augmentation did not improve the F1 Scores in the case of both pre-trained models. Possibly, rotating and shifting the images is not enough to extract important features out of x-rays. Rather, the additional images may be confusing the model, leading to a higher loss and lower F1 score than normal (without data augmentation).

C. Augmented Images - InfoGAN

The discriminator D and the recognition network Q share most of the network. For this task, we use 4 ten-dimensional categorical codes, 4 continuous latent codes, and 32 noise variables, resulting in a concatenated dimension of 40. The network architecture is as shown in Table I. Other hyper-parameters associated with InfoGAN models were chosen as suggested by [10]. Batch size was selected to be 64. We trained one InfoGAN model per pathology. It was observed that InfoGAN produced visually recognizable results quicker for larger datasets than for pathologies with relatively lower numbers of X-rays. We trained InfoGAN for the infiltration, effusion, atelectasis, nodules, and pneumonia classes. As pneumonia had only 1431 images, InfoGAN could not generate an image recognizable to the human eye even after 20000 epochs of training, as shown in Fig. 4. On average, the other disease classes required 250-500 epochs to train. The generator model was later used to generate 10k images per disease and the discriminator identified which images would be used to re-train the baseline model.

TABLE I
NETWORK ARCHITECTURE

Discriminator D / Recognition Network Q	**Generator G**
Input 128x128 Color Image	Input $\in \mathbb{R}^{40}$
32x32 64 conv, lRelu, Stride 2	FC, 1 x 1 x 448 RELU, batch-norm
32x32 conv, 128 lRELU, Stride 2, batch-norm	16x16 upconv, 128 RELU, Stride 2, batch-norm
32x32 conv, 128 lRELU, Stride 2, batch-norm	32x32 upconv, 64 RELU, Stride 2, batch-norm
FC Output Layer for D FC, 128-batch-norm-lRELU-FC Output for Q	64x64 upconv, 1 Tanh, Stride 2,

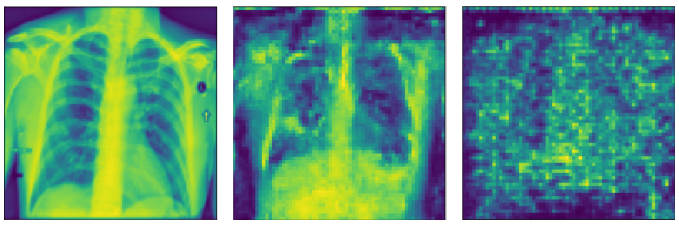


Fig. 4. Original image of an X-ray (left). GAN-generated image of an X-ray with characteristics of effusion (center). GAN-generated image of an X-ray with pneumonia with no identifiable features (right).

D. Hyper-Parameter Tuning

Hyper-parameter tuning was performed primarily on the baseline model on parameters such as learning rate, loss function, dropout, number of epochs, batch size, regularization parameter (beta), and number of layers. For transfer-learning, the specific set of pre-trained weights (either Inception V3 or VGG16) was our primary hyper-parameter; other parameters were largely set based on the characteristics of the dataset (e.g. adding layers with neurons to match the number of output classes).

For the baseline model, the learning rate was adjusted between 1e-6 and 1e-5; generally, disease classes with lots of images were assigned lower learning rates for their binary classifiers, while disease classes with few images were assigned higher learning rates. This is because training for diseases with many images proceeded steadily with every epoch, so

a small learning rate enabled a relatively consistent convergence. On the other hand, when learning rates were small for diseases with few images, the model would occasionally fail to converge if it initialized weights poorly. Thus, a larger learning rate was needed to "jump" out of regions of poor performance in the feature space. The dropout value after the fully connected layer was always set to 0.5, but the dropout after the max pooling layer was adjusted between 0.1 and 0.5. The batch size was adjusted between 128 and 512 - 512 was traditionally used for diseases with fewer images, so that at least one positive image would appear in every batch. Beta was adjusted between 0.001 and 0.1; this parameter was changed for every disease (there was no overall trend). For each binary classifier, either a traditional cross entropy loss function was used, or a weighted cross entropy function (which took the ratio of images positive for a specific disease in the dataset into account) was used. Generally, the weighted function was used for diseases with larger numbers of images, as low weight ratios tended to push the model to rarely output positive predictions for sparser disease classes.

For fine-tuning the pre-trained models, we tried experimenting on whether to freeze the layers of the models with the weights or not. While trying different settings for transfer learning, in the case of Inception-V3, we tried training the model from scratch with weights initialization and got acceptable performance and after freezing the layers, experienced poor performance on the validation set.

On the other hand, we experienced no change or little improvement in validation set performance for VGG16. Because of that, we froze all the layers of VGG16 network except the last 4 and kept a slow learning-rate (1e-4) to avoid over-fitting. Eventually, we got acceptable results for VGG16 as shown in Table II.

IV. Evaluation

A. Results

We used accuracy, recall, precision, F1 scores and AUC (area under curve) scores to evaluate our models. From the results, we can see that accuracy is not a good metric to evaluate performance alone; for a disease class with only a 1% incidence rate in the dataset, a model that never classified an image as belonging to that disease class would still have a 99% accuracy rate. We relied on AUC scores and F1 scores because AUC scores were widely used in the literature (e.g. [9]) and because these metrics penalized both false negative and false positive predictions, which is important in medical diagnostics.

We performed a comparative analysis between our baseline architecture and pre-trained models as shown in Table II. The first notable observation is that the pre-trained models outperformed the baseline models for 11 out of 14 disease classes (based on F1 score). The three classes which the baseline model classified better than the pre-trained models were hernia, pneumonia, and fibrosis, also the diseases with the fewest images in the dataset. The pre-trained model was not able to adequately classify these diseases (each had an F1 score at or close to 0) because it was skewed to better classify more common diseases. However, since the baseline model consisted of individual binary classifiers for each disease, it was able to have slightly better performance. The pre-trained models classified effusion and infiltration best, both of which had the highest number of images in the dataset.

We observe that there is a very minor difference between the performance of VGG-16 and InceptionV3 for our use case. This can be attributed to the fact that we use pre-trained weights and only train a portion of the model. The pre-trained weights are trained on 'imagenet'; hence, they are good at capturing general features of an image. Generally, these models both perform poorly for diseases with little data and perform well for diseases with lots of images. However, this trend does not hold for every individual disease class. Moreover, it is worth noting that VGG-16 performs better for 7 diseases while Inception V3 performs better for 4 (out of the 11 that performed better than the baseline model). We attribute this fact to the batch normalization layer that is present in InceptionV3. The Keras implementation of the BN layer differs from other frameworks; to take full advantage of the BN layer, we should retrain all BN layers to find the actual mean and variance corresponding to the input images which in our case is a general image of a chest X-ray which is very different from the images from imagenet.

Previous studies have used AUC to evaluate performance, so the AUC scores for each of our models are compared against those obtained by Wang et al. [9]. We found our pre-trained Inception-V3 and VGG16 models out-performed Wang et al.'s architecture for 8 out of 14 disease (in terms of AUC scores).

For the majority of the disease classes, we have fewer than 5k images for training. To extract better performance from the models we tried data augmentation. As explained earlier, simple image augmentation techniques such as rotating the image slightly did not improve performance. Thus, we used InfoGAN to generate artificial images to improve performance. As seen in Table III we find that for infiltration, effusion and atelectasis, we were able to increase the F1 score and recall of our baseline model consistently when using original images and GAN-generated images together. However, in the case of nodule we do not get a performance improvement. This likely happened because while we have more than 10k images for the first 3 pathologies, we have only half that number for the nodule class. As is shown in Fig. 4, GAN-generated images for diseases with few positive images in the dataset (such as pneumonia, with only 1431 images) are unrecognizable as even X-rays to the naked eye. This suggests that as data for a disease becomes more limited, so does the ability of the GAN to improve the model's performance. Based on our data, the InfoGAN architecture requires at least 10k images and 250 epochs to produce images that are visually recognizable as chest X-rays. The GAN-generated images could not be used to train the transfer learning models because the number of channels in the GAN images (1) did not match the channels required for the pre-trained models (3), and training a GAN to produce 3 channel output did not produce usable results.

TABLE II
EVALUATION RESULTS

Model	Metric	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis	Hernia	Infiltration	Mass	Nodule	Pleural Thickening	Pneumonia	Pneumothorax
BaselineModel	Accuracy	0.71113	0.9207	0.67334	0.9056	0.70967	0.9397	0.74443	0.99512	0.61221	0.86045	0.78594	0.84863	0.90322	0.88135
	Recall	0.2762	0.1882	0.4495	0.2289	0.3568	0.06694	0.4183	0.04651	0.2888	0.1104	0.1868	0.1612	0.0808	0.0807
	Precision	0.3356	0.2263	0.1273	0.148	0.4279	0.1728	0.057	0.1818	0.5031	0.2433	0.1586	0.09982	0.03235	0.2005
	F1	0.303	0.2055	0.1984	0.1798	0.3892	0.0965	0.1004	0.07407	0.367	0.1519	0.1716	0.1233	0.0462	0.1151
	AUC	0.5886	0.612	0.5346	0.723	0.6238	0.5346	0.573	0.6362	0.5839	0.5889	0.5445	0.5212	0.5053	0.5733
PretrainedInceptionV3	Accuracy	0.72865	0.92474	0.74661	0.94484	0.76062	0.94368	0.96251	0.99586	0.43653	0.72082	0.86727	0.74246	0.97102	0.89905
	Recall	0.43646	0.44247	0.51668	0.14377	0.71934	0.27634	0.05142	0	0.97292	0.68744	0.17952	0.477306	0	0.26518
	Precision	0.40916	0.35014	0.18085	0.28033	0.52804	0.38826	0.24324	0	0.40628	0.24181	0.37777	0.12371	0	0.45094
	F1	0.42237	0.39093	0.26793	0.19007	0.60902	0.32288	0.084905	0	0.5732	0.35777	0.24339	0.196503	0	0.33397
	AUC	0.69543	0.83152	0.70883	0.81682	0.81503	0.79543	0.73417	0.70723	0.65258	0.76837	0.67292	0.67055	0.63457	0.76401
PretrainedVGG16	Accuracy	0.679	0.92736	0.88717	0.9374	0.71194	0.9404	0.96513	0.99585	0.43267	0.87133	0.84177	0.92832	0.97102	0.8808
	Recall	0.62303	0.41947	0.08934	0.19957	0.75177	0.14712	0.00571	0	0.96001	0.27071	0.23964	0.041	0	0.36134
	Precision	0.3757	0.35855	0.20494	0.25272	0.4655	0.28244	0.13333	0	0.40357	0.39874	0.29589	0.24348	0	0.37188
	F1	0.46875	0.38662	0.12444	0.22302	0.57497	0.19346	0.01096	0	0.56826	0.32248	0.26481	0.07018	0	0.36653
	AUC	0.71502	0.84752	0.69333	0.82036	0.80145	0.77325	0.74444	0.79268	0.66223	0.73951	0.69555	0.67856	0.60539	0.78439
Basline	AUC	0.5886	0.612	0.5346	0.723	0.6238	0.5346	0.573	0.6362	0.5839	0.5889	0.5445	0.5212	0.5053	0.5733
Inception V3	AUC	0.69543	0.83152	0.70883	0.81682	0.81503	0.79543	0.73417	0.70723	0.65258	0.76837	0.67292	0.67055	0.63457	0.76401
VGG16	AUC	0.71502	0.84752	0.69333	0.82036	0.80145	0.77325	0.74444	0.79268	0.66223	0.73951	0.69555	0.67856	0.60539	0.78439
Wang et al	AUC	0.7003	0.81	0.7032	0.8052	0.7585	0.833	0.7859	0.8717	0.6614	0.6933	0.6687	0.6835	0.658	0.7993

TABLE III
COMPARATIVE ANALYSIS WITH GAN

Pathology→	Infiltration		Atelectasis		Effusion		Nodule	
Metric↓	w/ GAN	w/o GAN	w/GAN	w/o GAN	w/GAN	w/o GAN	w/GAN	w/o GAN
Accuracy	0.55166	0.61221	0.64658	0.71113	0.66055	0.70967	0.83643	0.78594
Recall	0.4758	0.2888	0.3731	0.2762	0.5409	0.3568	0.08799	0.1868
Precision	0.4309	0.5031	0.287	0.3356	0.3886	0.4279	0.159	0.1586
F1	0.4522	0.367	0.3244	0.303	0.4523	0.3892	0.1133	0.1716
AUC	0.5435	0.5839	0.5621	0.5886	0.6493	0.6238	0.5434	0.5445

B. Conclusions

While X-rays are among the most commonly used medical diagnostic tools, they suffer from serious limitations that prevent clinicians from using them to make accurate medical evaluations. To improve the utility of X-rays, and lung X-rays in particular, we propose to use deep convolutional neural networks, initialized with pre-trained architectures and weights, and augmented with GAN-generated images. Our results indicate that combining all of these approaches results in the best performance for the prediction of most diseases. However, for diseases with limited amounts of X-ray scans available, more data needs to be collected before these models can predict those diseases with any reasonable degree of accuracy. Until that point is reached, binary CNN classifiers are more likely to produce accurate results than larger architectures.

REFERENCES

[1] H. Chen, M. M. Rogalski, and J. N. Anker, "Advances in functional x-ray imaging techniques and contrast agents," Oct 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3569739/

[2] "Imv reports general x-ray procedures growing at 5.5% per year, as numb," Feb 2011. [Online]. Available: https://www.prweb.com/releases/2011/2/prweb8127064.htm

[3] "Dangers of ct scans and x-rays - consumer reports." [Online]. Available: https://www.consumerreports.org/cro/magazine/2015/01/the-surprising-dangers-of-ct-sans-and-x-rays/index.htm

[4] "Health care use - magnetic resonance imaging (mri) exams - oecd data." [Online]. Available: https://data.oecd.org/healthcare/magnetic-resonance-imaging-mri-exams.htm

[5] J. Mattoon and C. Smith, "Breakthroughs in radiography: Computed radiography," *Compendium on Continuing Education for the Practicing Veterinarian*, vol. 26, 12 2008.

[6] S. Pal, "Epidemiology of chronic lung diseases," Jul 2017. [Online]. Available: https://www.uspharmacist.com/article/epidemiology-of-chronic-lung-diseases

[7] "Thoracic diseases." [Online]. Available: https://hartfordhealthcare.org/services/cancer-care/departments-centers/thoracic-diseases

[8] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR*, vol. abs/1711.05225, 2017. [Online]. Available: http://arxiv.org/abs/1711.05225

[9] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. 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.

[10] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *CoRR*, vol. abs/1606.03657, 2016. [Online]. Available: http://arxiv.org/abs/1606.03657

[11] N. Donges and N. Donges, "Transfer learning," Apr 2018. [Online]. Available: https://towardsdatascience.com/transfer-learning-946518f95666

[12] Fchollet, "fchollet/deep-learning-models." [Online]. Available: https://github.com/fchollet/deep-learning-models/releases

[13] "Keras: The python deep learning library." [Online]. Available: https://keras.io/

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[18] T. de Boissiere, "Deeplearningimplementations/infogan," Oct 2018. [Online]. Available: https://github.com/tdeboissiere/DeepLearningImplementations

[19] "Keras: Preprocessing image generator class." [Online]. Available: https://keras.io/preprocessing/image/