Exploring Chest X-Ray Classification

Michal Shlapentokh-Rothman, Nouran Soliman

Abstract

Deep learning techniques are increasingly used for medical image interpretation due to the need for fast and accurate diagnoses. Work on chest x-rays is less developed compared to other diseases because there are only a few reliable datasets. Our work examines the recently released large (>200,000) chest x-ray dataset, CheXpert, using models that previously produced state-of-the-art results on earlier chest x-ray datasets. We experimented with different versions of an auto-encoder based CNN and Densenet-121 on the CheXpert dataset. Our results were promising: we achieved an average AUC (area under the curve) of 0.829 with the auto-encoder based CNN.

1. Introduction

Many heart and lung diseases require Chest X-Rays to be diagnosed.However, if the x-rays are not accurately and quickly examined, the consequences can be fatal.[18] Therefore, there is a crucial need for a high performing computer aided diagnosis system for chest x-rays. In this work, we investigate multi-label classification for chest x-rays using deep learning.

Compared to other disease classification problems, there is little work done on identifying heart and lung diseases from chest x-rays [19]. Radiologists tend to disagree about the appearance of diseases in chest x-rays [3] which has resulted in there being relatively few accurately labeled datasets. Recently, CheXpert, a large dataset consisting of more than 200,000 chest x-rays and 65,000 patients was released. [11] The goal of this work is to explore existing methods that were trained on other chest x-ray datasets and apply them to this new dataset. Ideally, these existing methods have high performance on the new dataset.

The rest of the paper is organized as follows: section 2 is Literature Review, section 3 contains dataset discussion, section 4 describes our methodology, section 5 presents our experiments and results, section 6 discusses the results and potential limitations and improvements, and we conclude in section 7.

2. Literature Review

2.1. Deep learning in medical imaging

Deep learning approaches in medical imaging are outperforming experts in interpretation tasks [22, 21, 6, 26] due to high-quality datasets [33, 11, 12] and high-performing network architectures [7, 9, 30, 35]. CNNs [16, 14] are dominating medical imaging detection and classification problems such as pulmonary tuberculosis detection [15], lung cancer detection [10], skin cancer classification [5] and others [20, 17].

2.2. Multi-label classification of chest x-rays (CXRs)

In multi-label classification, each sample is labeled with one or several classes [34, 32]. A lot of work has been done on multi-label classification of CXRs. CheXNet [22], a DenseNet-121 trained on ChestX-ray14 dataset [33], achieved state-of-the-art performance and exceeded radiologist performance on pneumonia using the F1 metric. Another approach trains a 121-layer DenseNet on CheXpert [11] dataset (224,316 scans) and handles uncertainty labels. This model achieved an AUC of 0.907 when tested on 5-class 500 test images. In another work, DualNet [24], dual convolutional networks jointly trained on both the frontal and lateral CXRs of MIMIC-CXR [12] (more than 350,000 scans), achieved better performance in classification when compared to baseline (i.e. frontal and lateral) classifiers.

2.3. Auto-encoders in medical imaging classification

Autoencoder-based representation learning is used in medical imaging applications [28, 1, 29] to learn representations of images for higher classification performance by stacking auto-encoders with classifiers. One method reduces resolution of images using an auto-encoder and then feeds the output to a CNN classifier [23]. This method was trained on chestx-ray14 dataset [33] and achieved state-of-the-art performance on 14-class classification.

2.4. Contribution

Our method compares the performance of a baseline DenseNet and an autoencoder-based CNN architecture similar to [23] applied to the new Chexpert dataset. As part of our experiments, we tested changing the classifier on the

auto-encoder based CNN as well as various pre-processing techniques. We conducted hyperparameter tuning on our best model. This method aims to test if we can learn the representation of chest x-rays through feature extraction using an auto-encoder. The ultimate goal is to try different techniques to achieve high classification performance on the Chexpert dataset.

3. Dataset

This work uses the CheXpert dataset [11], one of the largest publicly available chest x-ray datasets. It contains 224,316 scans of 65,240 unique patients, and each scan is labeled for the presence of 14 common chest radiographic observations. There are three types of labels possible. Positive or a 1, indicates the disease is present. Negative or 0 indicates the disease is not present or uncertain (u), means the expert radiologists could not agree on if the image contained the disease or not. The standard dataset is divided approximately into 223,413 training images (shown in Table 2), 232 validation images and 500 test images. However, the test images are not publicly available, so for a model to be evaluated on the test set it has to be sent to CheXpert competition. This evaluation process takes 2 weeks, which makes it impossible to test all the experiments on the test set during the project time frame. Therefore, we split the validation dataset into 2 halves, using the first half for validation and the second half for testing. The validation set is annotated by 3 board-certified radiologists taking the majority vote as the true label. CheXpert authors suggested an evaluation protocol over 5 classes: Cardiomegaly, Atelectasis, Pleural Effusion, Consolidation, and Edema, which are picked based on their prevalence from the validation set and clinical importance. The effectiveness of trained models is measured by AUC metric. Our model takes chest x-ray images as an input and outputs the probability of each of the 5 observations.

Pathology	Positive(%)	Negative(%)
Cardiomegaly	35087 (15.7)	188326 (84.3)
Atelectasis	67115 (30.0)	156298 (70.0)
Pleural Effusion	97815 (43.8)	125598 (56.2)
Consolidation	42525 (19.0)	180888 (81)
Edema	65230 (29.2)	158183 (70.8)

Table 1. This table reports the number of studies which contain these observations in the actual training set after applying U-Ones.

Pathology	Positive	Uncertain	Negative
No Finding	16627 (8.86)	0 (0.0)	171014 (91.14)
Enlarged Cardiom.	9020 (4.81)	10148 (5.41)	168473 (89.78)
Cardiomegaly	23002 (12.26)	6597 (3.52)	158042 (84.23)
Lung Lesion	6856 (3.65)	1071 (0.57)	179714 (95.78)
Lung Opacity	92669 (49.39)	4341 (2.31)	90631 (48.3)
Edema	48905 (26.06)	11571 (6.17)	127165 (67.77)
Consolidation	12730 (6.78)	23976 (12.78)	150935 (80.44)
Pneumonia	4576 (2.44)	15658 (8.34)	167407 (89.22)
Atelectasis	29333 (15.63)	29377 (15.66)	128931 (68.71)
Pneumothorax	17313 (9.23)	2663 (1.42)	167665 (89.35)
Pleural Effusion	75696 (40.34)	9419 (5.02)	102526 (54.64)
Pleural Other	2441 (1.3)	1771 (0.94)	183429 (97.76)
Fracture	7270 (3.87)	484 (0.26)	179887 (95.87)
Support Devices	105831 (56.4)	898 (0.48)	80912 (43.12)

Table 2. The CheXpert dataset consists of 14 labeled observations. This table reports the number of studies and % which contain these observations in the training set. (This data is taken from [11])

4. Methodology

4.1. Data Pre-processing

4.1.1 Image Processing

Before we trained the models, we made some modifications to the images. Our models used a Densenet [31] which requires the height and width of images to be at least 224 and be loaded in a certain range. All images were always normalized with the following values: mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]. As part of our experiments, we added some additional pre-processing in some of the models. We used common pre-processing guidelines that were discussed in class 4.1. In particular, we added Gaussian noise and randomly rotated the images before they were loaded into the model for training.

4.2. Handling of the Unknowns

We tried three methods for handling the unknown labels in the training set: ignoring them (U-Ignore), setting the unknown to always be one or positive (U-Ones) or setting the unknowns to be zero or negative (U-Zeros).

4.2.1 Cost-Sensitive Learning

To handle class imbalance, cost-sensitive learning is used where samples are assigned weights to match a certain data distribution. In our case, weighting by inverse class frequency is used. However, the results of this method were not good on our dataset using baseline classifier (average AUC = 0.537). One reason behind this could be the difference in class distribution between training and validation sets (Table 3 shows distribution of validation set).

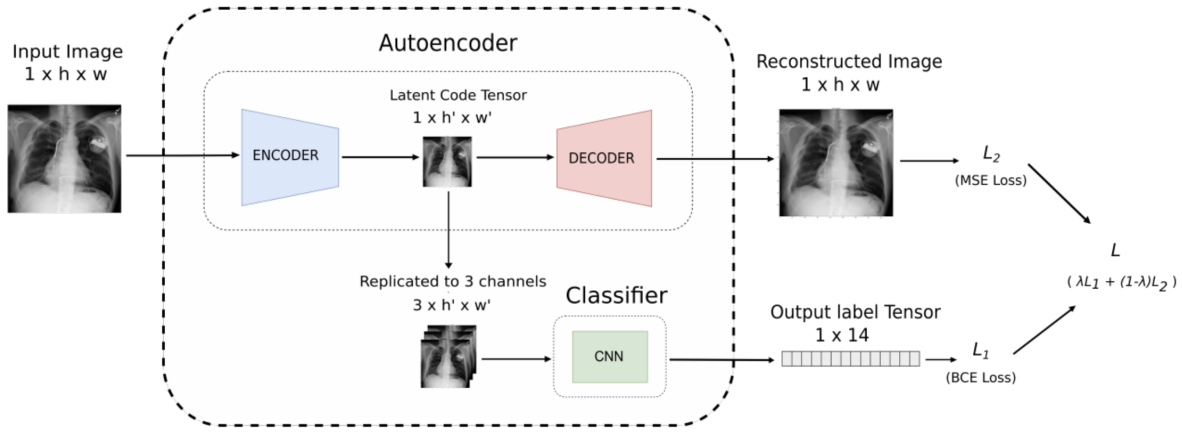

Figure 1. Auto-encoder-based CNN architecture. This diagram is taken from [23]

Pathology	Positive(%)	Negative(%)
Cardiomegaly	27 (23.3)	89 (76.7)
Atelectasis	28 (24.1)	88 (75.9)
Pleural Effusion	18 (15.5)	98 (84.5)
Consolidation	9 (7.8)	107 (92.2)
Edema	19 (16.4)	97 (83.6)

Table 3. This table reports the number of studies which contain these observations in the validation set after applying U-Ones.

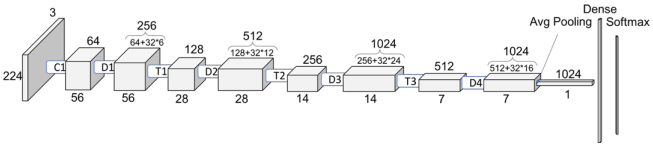

Figure 2. Overview of Densenet-121

4.3. Models

4.3.1 Baseline Model

Our baseline model consisted of a Densenet-121 [8]. We decided to use a Densenet-121 because the original paper [11] had the best results with it. The baseline model takes in an image of size 1×224×224 and outputs the probability of each of the 5 classes tested. We then used softmax to determine the most likely predicted classification of the image from the densenet. An overview of densenet architecture [25] can be seen in Figure 2. For the baseline model we used cross-entropy loss:

$$L1(Y, \hat{Y}) = \sum_{c=1}^{5} [y_c \log(1 - \hat{y}_c) + (1 - y_c) \log \hat{y}_c] \quad (1)$$

Where c is the disease index, y_c is the actual label of c^{th} disease, $\hat{y}_c$ is the predicted label of c^{th} disease, and Y and $\hat{Y}$ are actual label vector and predicted label vector respectively.

4.3.2 AutoEncoder-based CNN

Our work uses an AE-CNN model for disease classification as [23]. The architecture used consists of 3 main modules: encoder, decoder and classifier (Figure 1). The downsampled output of the encoder is fed as an input to the classifier for classification and to the decoder for reconstruction. The auto-encoder and classifier are trained jointly.

Encoder: The encoder takes in an image of size 1×224×224 and produces a latent code tensor of size it to 1×56×56. The encoder consists of 2 convolution layers: first layer of 32 kernels of size 5 × 5 each with a stride of 4 followed by exponential linear units activation function [2] (with zero padding to preserve image size), and a second layer of kernels of size 1 × 1 with stride of 1 with ClippedReLU activation. The definition of ClippedReLU is:

$$ClippedReLU(x) = min(max(0, x), 1) \quad (2)$$

Decoder: The decoder consists of a single convolution layer with kernel size 3 × 3 with a stride of 1 followed by sub-pixel shuffling [27] to reconstruct high-resolution image back. ClippedReLU activation is then used.

Classifier: Our baseline Densenet-121 classifier is integrated with this architecture. We used non-pretrained and CheXpert-pretrained versions of our baseline classifier.

Loss Function: The loss function used in the AE-CNN model is different from what is used in the baseline model. The loss is the weighted sum of classification loss and reconstruction loss.

$$L = \lambda L1 + (1 - \lambda)L2 \quad (3)$$

Where L1 is classification loss, L2 is reconstruction loss and λ is a hyperparameter from 0 to 1 indicating importance of

classification loss. We set λ = 0.9 as in [23]. Binary cross entropy (BCE) loss function is used for classification error during training. We used the same binary cross entropy (BCE) loss as used in the baseline training and is denoted by $L1$. Mean-squared error (MSE) is used for autoencoder error (L2).

4.4. Training and Testing

As discussed earlier, the actual test set for CheXpert is not publicly available and requires two weeks for evaluation. Our test set consisted of the second half of the validation set. To match the images with their appropriate classifications, we had a CSV file where each row contained the image file location, and a 0 or 1 for each class.

During training, we ran the model for 10 epochs and saved the weights that performed best on the validation data. We then used those weights for the test set.

We trained and tested the following the models: a baseline with U-ones, baseline with U-zeros, baseline with U-ignore, baseline with noise, AE-CNN with a non-trained baseline, AE-CNN with a trained baseline. We then tuned the model that had the best performance.

5. Experiments and Results

We used PyTorch in all of our experiments.

5.1. Area Under the Curve (AUC)

To measure how well we were performing, we used a metric called Area Under the Curve (AUC), which was the metric used in the original paper that contained the dataset [11]. The area under the curve refers to the area under the ROC or Receiving Operator Characteristic (ROC). The ROC curve is a measure that relates sensitivity and specifity. Sensitivity is the number of true positives divided by the number of samples that are actually positive(sum of true positives and false negatives). The ROC is a graph of 1-specifity. [4]. AUC is a more important measure than other measures like accuracy because we want to differentiate from models that may have a high accuracy but have a large rate of false negatives or positives. Since this dataset is part of the healthcare domain, a correctly identified false positive is just as bad as an incorrectly identifying a disease.

Model Name	Average AUC
Densenet w/U-Ones	.853
Densenet w/U-Zeros	.852
Densenet w/U-Ignore	.57
Densenet and Pre-Processing	.845
AE-CNN w/no pre-training	.829
AE-CNN w/pre-trained classifier	.756
AE-CNN w/pre-trained classifier and pre-processing	.72

Table 4. Experimental results

5.2. Baseline Results

All results can be seen Table 4. We ran three baseline experiments and the three experiments differed in how the unknowns in the training data were treated: ignoring (U-Ignore) the unknowns, mapping to positive (U-Ones), or mapping to negative (U-Zeros). U-Ones yielded the best results. The baseline experiment used a batch size of 16, learning rate of 10^{-4}, a dropout of .9 and the Adam optimizer [13].

5.2.1 Baseline with Pre-Processing

We ran the U-ones baseline model with the pre-training as described in Section 4.1. Everything else was kept the same.

5.3. AE-CNN

5.3.1 Not pre-trained

In this experiment, the untrained baseline classifier was integrated with the autoencoder. The model used the same hyperparameters as the baseline experiments described in Section 5.2.

5.3.2 Pre-trained

The exact same experiment as mentioned in Section 5.3.1 was repeated except that we used a trained version of our baseline model. 5.2

5.3.3 Preprocessing

Same experimental set up is used as in Section 5.3.1 except we added the pre-processing as described in Section 4.1.

Average AUC	Learning Rate	Dropout	Batch Size
0.829	0.0001	0.9	**16**
0.83	0.0002	0.9	16
.831	**0.001**	0.9	16
.829	.0001	.9	**32**
.825	.0001	.9	64
.832	.0001	**.5**	16
.831	.0001	.1	16

Table 5. Hyperparameter Tuning for AE-CNN

5.4. Best model: hyperparameters tuning

Comparing the results of the previous 3 AE-CNN experiments, we can see that the best model was the AE-CNN with no pre-training. The model had an average AUC = 0.829. Therefore, this model setting was picked to do hyperparameter tuning. All values were kept the same as the previous experiment except the specific hyperparameter being tuned. Table 5 shows the results of the experiments for each hyperparameter (learning rate, batch size and dropout) value. The top performing model setting is bolded.

6. Discussion

6.1. Effect of Pre-processing

We expected that by pre-processing the images as described in Section 4.1, we would have improved the performance model (compared to the original Densenet). This indicates that instead of learning the actual representation of the images, the model learned noise. The image that was the output of the pre-processing did not preserve the features of the original image that helped identify the various diseases. In the future, more variants of the parameters in the image processing should be looked at. Also, the images should be cropped so that the model does not learn noise that exists around the actual x-ray.

6.2. AE-CNN without training

As seen from the results, this model had a slightly lower AUC score than the baseline Densenet. This could be due to the fact that we used low resolution images for training, which were of size 390×320. We used small images due to our space capacity limitations. The encoder downsamples the images to 56×56 before feeding them to the classifier, which could be too low of a resolution to learn any useful feature representation. As a future improvement, the higher resolution version of this dataset can be used to test out this model. Since the current model has shown good results on the current low resolution images, we expect it to have better performance on the high resolution images.

6.3. AE-CNN with training

The AE-CNN that had a classifier from a pre-trained Densenet had worse performance than the AE-CNN that contained a Densenet that was not pre-trained. We think this is likely due the Densenet overfitting to the data. The classifier that was used was trained once by itself and then retrained again with the auto-encoder. To fix this in the future, the weights of the Densenet could be frozen during the training process of the auto-encoder.

6.4. Limitations and Future Work

There are also some other limitations with the current dataset. One limitation is the class imbalance of the dataset which we believe significantly affects the performance. Also, expanding the problem to include the 14 classes of all thoracic diseases will impose more bias in the data as well. For example, the 'No findings' class has very few samples in the dataset which means that most of the data is collected from diseased people. The samples are also all from a certain geographical area and therefore are not a good representation of healthy people. This is not reliable in developing a real world model that can ultimately outperform radiologists in such critical classification medical problems. One way to handle this would be expanding the dataset to include lateral views (not just frontal) with more balanced classes including healthy conditions. Also, handling uncertain labels is another limitation that can be studied further to improve the results.

Another limitation is that we considered the 5 classes to be completely independent of each other and this might not actually be the case. It could be the case that the diseases (or a subset of them) are correlated with each other and that having one means it is likely that you will have another. Future work should try to address this problem.

Lastly, there are several alternative methods that could have been used other than the Densenet and AE-CNN. A semi-supervised approach would likely be promising because it would be able to handle unlabeled datasets since there are not many labeled Chest X-Ray datasets.

7. Conclusion

We applied different versions of an existing AE-CNN model that had state-of-the-art performance on the chest-xray-14 [33] dataset to a brand new CheXpert dataset to explore its performance. The results of the AE-CNN (average AUC = 0.829) did not outperform the baseline Densenet model (average AUC = 0.853), but we believe that, given the limitations of the dataset and computational resources, the results of the AE-CNN are promising. If combined with the proposed suggestions, we think the results might outperform the baseline Densenet.

8. Work Distribution

Nouran worked on implementing the AE-CNN. Michal worked on implementing the baseline and pre-processing techniques. Together they worked on the experiments, hyperparameter tuning and writing the report.

References

[1] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani. Deep features learning for medical image analysis with convolutional autoencoder neural network. *IEEE Transactions on Big Data*, 2017.

[2] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[3] L. Delrue, R. Gosselin, B. Ilsen, A. Van Landeghem, J. de Mey, and P. Duyck. *Difficulties in the Interpretation of Chest Radiography*, pages 27–49. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[4] J. D'Souza. Let's learn about auc roc curve!, 2018.

[5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[6] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang. Diagnose like a radiologist: Attention guided con-

volutional neural network for thorax disease classification. *arXiv preprint arXiv:1801.09927*, 2018.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks, 2016.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[10] P. Huang, S. Park, R. Yan, J. Lee, L. C. Chu, C. T. Lin, A. Hussien, J. Rathmell, B. Thomas, C. Chen, et al. Added value of computer-aided ct image features for early lung cancer diagnosis with small pulmonary nodules: a matched case-control study. *Radiology*, 286(1):286–295, 2017.

[11] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *arXiv preprint arXiv:1901.07031*, 2019.

[12] A. E. Johnson, T. J. Pollard, S. Berkowitz, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, R. G. Mark, and S. Horng. Mimic-cxr: A large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] P. Lakhani and B. Sundaram. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2):574–582, 2017.

[16] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[18] H. H. Pham, T. T. Le, D. Q. Tran, D. T. Ngo, and H. Q. Nguyen. Interpreting chest x-rays via cnns that exploit disease dependencies and uncertainty labels, 2019.

[19] C. Qin, D. Yao, Y. Shi, and Z. Song. Computer-aided detection in chest radiography based on artificial intelligence: a survey. *BioMedical Engineering OnLine*, 1, 2018.

[20] C. Qin, D. Yao, Y. Shi, and Z. Song. Computer-aided detection in chest radiography based on artificial intelligence: a survey. *Biomedical engineering online*, 17(1):113, 2018.

[21] P. Rajpurkar, J. Irvin, R. L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. P. Langlotz, et al. Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists. *PLoS medicine*, 15(11):e1002686, 2018.

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

[23] E. Ranjan, S. Paul, S. Kapoor, A. Kar, R. Sethuraman, and D. Sheet. Jointly learning convolutional representations to compress radiological images and classify thoracic diseases in the compressed domain. In *11th Indian Conference on Computer Vision, Graphics and Image Processings*, 2018.

[24] J. Rubin, D. Sanghavi, C. Zhao, K. Lee, A. Qadir, and M. Xu-Wilson. Large scale automated reading of frontal and lateral chest x-rays using dual convolutional neural networks. *arXiv preprint arXiv:1804.07839*, 2018.

[25] P. Ruiz. Understanding and visualizing densenets, 2018.

[26] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh. Deep learning to improve breast cancer detection on screening mammography. *Scientific reports*, 9(1):1–12, 2019.

[27] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[28] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1930–1943, 2012.

[29] Q. Song, L. Zhao, X. Luo, and X. Dou. Using deep learning for classification of lung nodules on computed tomography images. *Journal of healthcare engineering*, 2017, 2017.

[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[31] P. Team. Densenet.

[32] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[33] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[34] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

[35] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.