# Jointly Learning Convolutional Representations to Compress Radiological Images and Classify Thoracic Diseases in the Compressed Domain (Reimplementation by Viraj Bagal)

Ekagra Ranjan[1], Soumava Paul[2], Siddharth Kapoor[3], Aupendu Kar[2], Ramanathan Sethuraman[4], Debdoot Sheet[2]

[1]Indian Institute of Technology, Guwahati, India
[2]Indian Institute of Technology, Kharagpur, India
[3]National Institute of Technology, Karnataka, India
[4]Intel Technology India Pvt. Ltd., Bangalore, India

## Abstract

*Convolutional Neural Networks are widely used for image classification in all walks of life including healthcare. But in the case of medical data, large image size is one of the major problems. It exhibits large cost on computation. The common way of dealing with large image size is 'resizing' using interpolation techniques. However, this might lead to loss of information. To mitigate this issue, the paper trains Autoencoder on the dataset, and then uses the encoder to downscale the images. Furthermore, Densenet121 is used to classify the latent vector obtained from the encoder. It is observed that the encoder approach outperforms the plain resizing approach. ChestX-Ray14 dataset is used in the implementation. This is the report of an attempt to reproduce the results of the mentioned paper [1]. Pytorch is used for the implementation. Re-implementation code can be obtained at:* `https://github.com/VirajBagal/ChestXRay14-Reimplementation/tree/master`

## 1. Introduction

The advent of Convolutional Neural Networks has accelerated the field of computer vision. CNNs are revolutionizing all the sectors dealing with image-like data and medical field is no exception to it. CNNs have been widely used in the classification of diabetic retinopathy, prostate cancer; segmentation of brain MRI scans. However, large image size is pretty common in the medical data. This hampers the speed of training and evaluation process and leads to exorbitant cost on computation. One of the common techniques observed is resizing image using interpolation techniques. These techniques might lead to some loss of information. This motivates to circumvent the problem without loss of much information. This paper finds Autoencoders as a plausible solution to the loss of information problem.

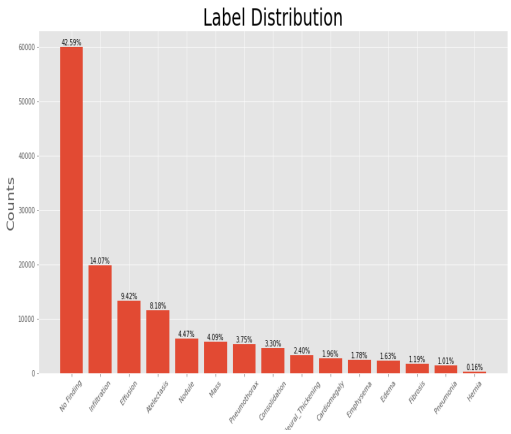


Figure 1. Label Distribution Histogram. Data is highly imbalanced. 'No Finding' is the most prevalent while Hernia is the least prevalent.

In this reimplementation, I use DenseNet121 BL5 for baseline results. BL5 refers to *BaseLine 5* since five preprocessing steps are used during training. To improve on that, Autoencoder, rather than interpolation techniques, is used for forming compressed representations of large images, followed by DenseNet121 for classification. ChestX-Ray14 dataset is the publicly available largest dataset consisting of chest xrays. Chestx-ray14 dataset is used for evaluating this approach.

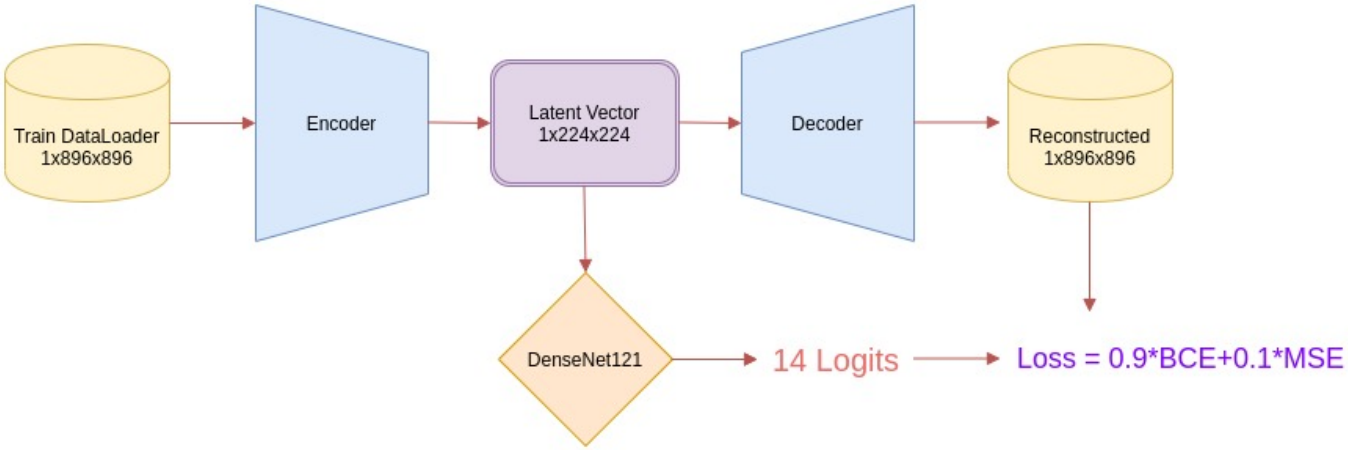


Figure 2. Stage 3 Model

## 2. Methodology

### 2.1. Dataset

ChestX-ray14 dataset was used. It is 42 GB in size and available on Kaggle to directly use. It has 1,12,120 chestx-ray images in total with 14 disease labels. It is multilabel problem, that is, each image can have more than 1 disease label. Images are 1024x1024 in size. The dataset has 51,708 images with 1 or more diseases and 60,412 images with no disease. The dataset is highly imbalanced as observed in Figure 1. Official train-test split was obtained from the official website. Training dataset has 86,524 images while the testing dataset has 25,596 images. The training dataset is further split into train-val patientwise randomly. Validation dataset has 10,000 images while train dataset has 76,524 images.

### 2.2. Architecture

DenseNet121 was used as the classifier. The Autoencoder was designed such that the encoder downsamples the input by the factor of 4 in the height and width dimensions and the encoder then resconstructs it. The encoder has 2 convolutional layers. The first layer has 32 5x5 filters with stride 4 followed by the exponential linear unit as the activation function and zero padding is used to preserve the spatial dimensions during convolution. The second layer has 1 1x1 filter with stride 1. The encoder has 16 3x3 filters with stride 1 and zero padding is used. It is followed by the PixelShuffle layer with r factor 4. PixelShuffle(r) converts the input (BS,$Cxr^2$,H,W) to the output (BS,C,Hxr,Wxr). This is followed by the ClippedReLU layer. The formula for the ClippedReLU layer is

$$\text{ClippedReLU} = \min(\max(0,x),1)$$

### 2.3. Loss Function

In contrast to the BCELoss used in the paper, I used BCEWithLogitsLoss (L1) for the classifier. The Pytorch documentation mentions that BCEWithLogitsLoss is numerically more stable than using plain sigmoid followed by the BCELoss. MSELoss (L2) is used for the autoencodger. The total loss function for the Autoencoder-CNN (AECNN) end-to-end training is

$$L = \lambda*L1 + (1-\lambda)*L2, \text{ with } \lambda=0.9$$

### 2.4. Training

3-stage training is implemented in this paper.

**Stage 1**: In this stage, DenseNet121 (D121) is trained on the dataset. The model parameters are initialized with imagenet pretrained weights. 1024x1024 original images are first resized to 256x256, followed by RandomCrop of 224x224. RandomHorizontalFlip, RandomRotation(degrees=5) and ColorJitter(contrast=0.25) are applied after the RandomCrop. Imagenet normalization is applied on the images before forward pass. Torchvision transforms are used for this augmentation. The model is trained for 20 epochs with the Adam optimizer with the initial learning rate 1e-4 and other default hyperparameters. The learning rate is reduced by 0.1 factor if the validation loss doesn't decrease for straight 5 epochs. ReduceLROnPlateau is used to bring that into effect. Just BCEWithLogitsLoss is used in this stage. Training was done on Colab. It took around 3.5 hours for stage 1 training.

**Stage 2**: In this stage, the Autoencoder is trained on random 128x128 patches of the original dataset. This is done just to get an headstart in the training of stage 3. RandomHorizontalFlip and RandomVerticalFlip are used. No normalization is applied on the images. Just MSELoss is used in this stage. Same number of epochs, optimizer and

| Labels | D121 (O) | D121 (R) | AECNN (O) | AECNN (R) | FMix |
|---|---|---|---|---|---|
| Atelectasis | 0.7829 | 0.786 | 0.7848 | 0.7844 | 0.7810 |
| Cardiomegaly | 0.8941 | 0.8811 | 0.8951 | 0.9006 | 0.8593 |
| Consolidation | 0.7565 | 0.7545 | 0.7584 | 0.7514 | 0.7481 |
| Edema | 0.8518 | 0.8583 | 0.8506 | 0.8586 | 0.8533 |
| Effusion | 0.8340 | 0.8387 | 0.8367 | 0.8354 | 0.8326 |
| Emphysema | 0.9271 | 0.9242 | 0.9205 | 0.9206 | 0.9229 |
| Fibrosis | 0.8340 | 0.8314 | 0.8389 | 0.8458 | 0.8283 |
| Hernia | 0.9117 | 0.9055 | 0.9178 | 0.9105 | 0.9220 |
| Infiltration | 0.7044 | 0.7064 | 0.7050 | 0.7027 | 0.7000 |
| Mass | 0.8366 | 0.8358 | 0.8390 | 0.8305 | 0.8164 |
| Nodule | 0.7771 | 0.7772 | 0.7789 | 0.7787 | 0.7671 |
| Pleural Thickening | 0.7855 | 0.7895 | 0.7847 | 0.7910 | 0.7806 |
| Pneumonia | 0.7434 | 0.7381 | 0.7459 | 0.7343 | 0.7147 |
| Pneumothorax | 0.8690 | 0.8750 | 0.8698 | 0.8647 | 0.8570 |
| **Mean** | **0.8220** | **0.8216** | **0.8226** | **0.8221** | **0.8131** |

Table 1. Labelwise ROC-AUC scores. Original (O) vs Reproduced (R)

learning rate scheduler is used in this stage as in the previous stage. Training was done on Kaggle. It took about 6 hours for stage 2 training.

**Stage 3**: In this stage, the Autoencoder-CNN are fine-tuned on the dataset in the end-to-end fashion. Autoencoder and CNN are loaded with the pretrained weights of stage 2 and stage 1 respectively. 1024x1024 images are first randomly cropped to size 896x896. RandomRotation of 5 degrees and RandomHorizontalFlip are used before forward pass through the decoder. The latent vector obtained from the decoder is then normalized with imagenet norms and forward passed through the classifier. Loss (L) as mentioned in the subsection 2.3 was used in this training stage. Same number of epochs, optimizer and learning rate scheduler is used in this stage as in the previous stages.

### 2.5. Validation

Stage 1 and 2 validation is done on 256x256 and 128x128 resized images respectively, while stage 3 validation is done on 896x896 resized images. No augmentation is used during validation.

## 3. Evaluation

D121 BL5 and AECNN are evaluated on the official test set of the ChestX-ray14 dataset. The original and reproduced ROCAUC values are reported in the table 1. D121 corresponds to the Stage 1 training evaluation only while AECNN refers to evaluation of stage 3 AECNN model. Clearly, the AECNN model is performing better than the baseline D121 model and the same is observed during the re-implementation. However, slight deviation is observed in the original and reproduced results. Generally, training is carried out multiple times and then the average stats with the deviations are reported. The slight deviation can be accounted by the inherent randomness in the training.

## 4. Ablation Study

In this study, the augmentations in stage 1 are supplanted by FMix [2] (Mixed Sample Data Augmentation technique) while the model of stage 1 is retained. It is observed that D121 FMix performed worse than D121 BL5 and AECNN. FMix mixes two samples from the same batch using masks sampled from the Fourier Space. Due to masks, it might happen that the deciding features of the image get occluded and so, the model learns slowly compared to the previous processes. Training for more number of epochs might improve its score.

## 5. Conclusion

I have re-implemented the mentioned paper and reported the effects of applying FMix instead of the normal augmentations reported in the original paper. The reproduced values slightly deviate from the original reported values. This can be accounted by the randomness in the training process. On the other hand, even if FMix performed worse than the D121 BL5 and AECNN, training for more epochs might ameliorate its results.

## References


[1] S. K. A. K. R. S.-D. S. Ekagra Ranjan, Soumava Paul. Jointly learning convolutional representations to compress radiological images and classify thoracic diseases in the compressed domain. *ICVGIP*, 2018.

[2] M. P. M. N. A. P.-B. J. H. Ethan Harris, Antonia Marcu. Understanding and enhancing mixed sample data augmentation. *arXiv:2002.12047*, 2020.