Warm Up Assignments

Rishitosh Kumar Singh
Arizona State University
rksing18@asu.edu

Abstract

This report is part of the assignment that is due on 21st Jan 2024. This report consists of two practice problems and by completing these problems I got to know how datasets and dataloaders work in Pytorch, and how to train and test a neural network with Pytorch.

1. Introduction

This report documents the warm-up assignments and will be refined with each subsequent task completion. Additionally, it will incorporate details of new experiments undertaken in each assignment. The overarching goal is to achieve State-of-the-Art (SOTA) performance by employing a diverse array of techniques. A dedicated section labeled "To-Do" will delineate pending tasks for discussion and action.

2. Backbone models

Training a model from scratch consumes substantial resources, including data and time, which may not always be feasible. In scenarios where large datasets are lacking, transfer learning emerges as a practical solution. Transfer learning involves leveraging a pre-trained network with existing weights and fine-tuning it on a new dataset. Typically, backbone models are utilized in the initial layers as they excel at capturing fundamental features. Following these backbone networks, additional layers such as a linear layer or another classifier can be incorporated to adapt to the nuances of the new dataset.

2.1. ResNet

ResNet, an abbreviation for Residual Network, was pioneered by Kaiming He et al. [1]. Its hallmark innovation lies in the incorporation of residual connections, a breakthrough technique that remains integral in contemporary state-of-the-art networks, transcending beyond the realm of computer vision tasks. ResNet manifests in various versions, distinguished by the number of layers, including ResNet18, ResNet50, and beyond. Its widespread adoption stems from its capacity to facilitate the utilization of extremely deep networks, a feat made possible by the introduction of residual connections.

ResNet models are trained on ImageNet dataset, thus it have 1000 output nodes. In order to use it for our problems, we have to append few linear layers after this backbone layer.

3. Exercises

3.1. Read/Write Images

3.1.1 Problem

In this exercise, we were provided with two datasets. The objective was to proficiently load the data using PyTorch's dataloaders and dataset APIs, and subsequently, successfully generate matplotlib plots by writing the images.

3.1.2 Solution

To load the provided dataset, custom PyTorch dataset classes were crafted. These classes offer flexibility to handle datasets of various types and formats. The torchvision.io.read_image function is employed for reading PNG/JPG images, while the pydicom library is utilized for reading DICOM data.

Transformations

- When PNG/JPG images were loaded, a resize transformation was applied to change the images to a size of 256×256. Figure 1a displays a batch of 4 images from the PNG/JPG dataset.
- After loading DICOM data, a sequence of transformations was employed. At first, the image type is converted from uint16 to uint8, because PyTorch resize function does not support uint16 dtype. Subsequently, the tensor is converted to a PIL image, and the image is resized. Following the dataset loading, all the images are plotted in a matplotlib plot. Figure 1b displays a batch of 4 images from the DICOM dataset.

3.2. Find Chest XRay(s) Orientation

In this introductory warmup exercise, chest X-rays were provided, and the task involved training/fine-tuning a model to determine the orientation of these X-rays.

3.2.1 Dataset

The provided dataset consists of Grayscale X-ray images with 948 training samples depicting various orientations and an additional 40 samples for testing. All images are of size 128×128 and are oriented either as up (0), right (1), down (2), or left (4). Due to the absence of a dedicated validation set in the given dataset, I have randomly partitioned the training samples into training and validation sets in 9:1 ratio.

3.2.2 Model Used

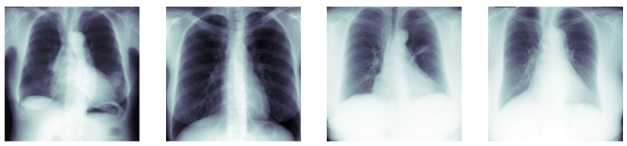
For this task, I utilized ResNet18 as the backbone model with pre-trained weights from ImageNet. Since ResNet18 was originally trained on ImageNet, its output tensor has a size of (n, 1000). However, in the given problem, we are specifically dealing with 4 classes (up, right, down, left). Therefore, I introduced few fully connected layer followed by a sigmoid layer for each class. Figure 6a shows the graphical representation of the used model architecture.

3.2.3 Training

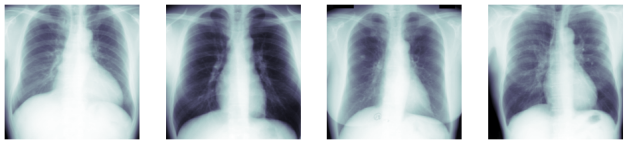
The model is trained using the Bbinary cross-entropy loss with adam as the optimizer. The model was trained for with learning rate ot 10^{-4} for 5 epochs. Figure 2 shows the learning curve and it can be infered from this that the model is neither overfitted, nor underfitted.

3.2.4 Results

After fine-tuning the model with the ResNet18 backbone, the model was transitioned to testing and evaluated on the provided test samples. The accuracy and AUC of the model is reported in Tab. 1. Figure 3 showcases some of the test samples along with model predection.



(a) PNG/JPG dataset



(b) DICOM dataset

Figure 1. XRay images batch

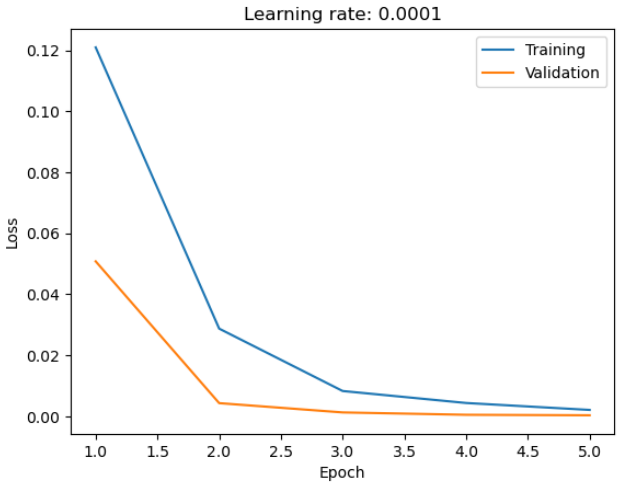


Figure 2. Learning curve of model trained for classifying the orientation of the chest xray.

3.3. Find patient gender from XRay(s)

In this preliminary practice activity, we were given chest X-ray images of patients using which we have to either train or fine-tune a model to determine the gender of patients.

3.3.1 Dataset

We were given three datasets for classifying gender of the patients; grayscale, RGB, and index. Out of these, I have used grayscale images for the experiment. The provided dataset have 154 training samples of patient X-Rays and an additional 93 samples for testing the trained model. The images in this dataset were color image with shape 256×256. Each training samples have either label "male", or "female". Due to the absence of a dedicated validation set in this dataset too, I have randomly partitioned the training samples into training and validation sets in 9:1 ratio.

3.3.2 Training

In this task, we are dealing with two labels; male and female. Therefore, I have constructed a model with resnet18 as feature extractor, and the model graph is described in Fig. 6b. A sigmoid layer is employed as the last activation function to get the output in range of $0 - 1$. The model is trained using the binary cross-entropy loss function with adam optimizer with a learning rate of 3.0×10^{-4}. The model was trained for a total of 20 epochs. Figure 4 illustrates the learning curve of the training and validation sets.

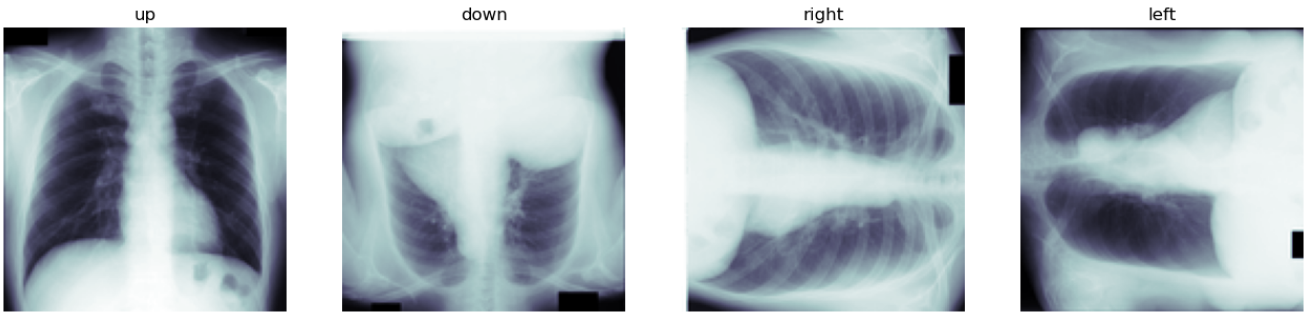


Figure 3. Predictions of few test samples to find orientation of chest xray

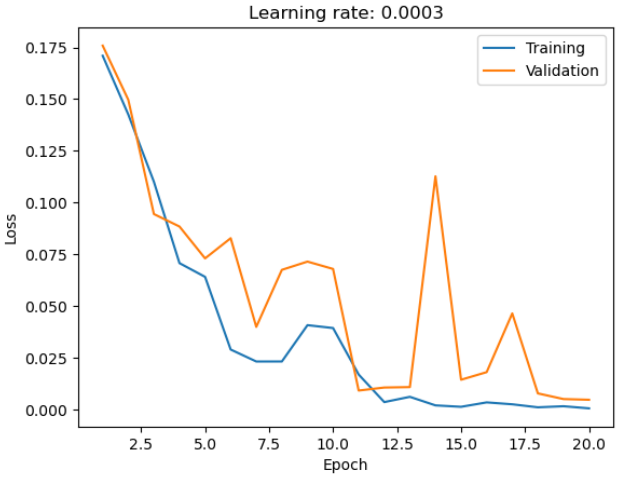


Figure 4. Learning curve of model trained on chest xray gender dataset.

	Orientation	Gender
ACC	100%	90.32%
AUC	1.0	0.947

Table 1. Classification model results

3.3.3 Results

After fine-tuning the model, it was evaluated on the provided test samples, and Tab. 1 reports the achieved accuracy and AUC. Figure 5 showcases some of the test samples along with their model predictions and target classes. The figure also includes GradCAM visualizations generated using the last layer of the ResNet18 backbone. The GradCAM visualization is helpful for understanding why the model classified certain images into specific classes.

3.4. Find patient age from XRay

We were given chest X-ray images of patients using which we have to fine-tune a model to determine the age of patients.

3.4.1 Dataset

The provided dataset have 80 training x-ray images and 165 testing images. As there was no validtion set, I have randomly splited the training dataset into validation and training set in ration 1:9 respictively. The images in this dataset were color image with shape 2048×2048, which is resized to image of size 224×224.

3.4.2 Training

This is a regression problem, thus a few changes is required to the model used in classifying gender of patients. The last Log Softmax layer is not required, thus eliminated from the model graph as shown in Fig. 6c. The model is trained using mean ssquare error function with Adam optimizer having a learning rate of 0.003. Additionally, in this experiment too, early stopping with a patience of 3 epochs is implemented to halt the model training. Figure 4 illustrates the learning curve of the training and validation sets.

3.4.3 Results

A total of 2 experiments were conducted with same hyperparameters and results are averaged. After fine-tuning the model, it was evaluated on the provided test samples on which it achieved a MAE of 5.57 Figure 9 showcases some of the test samples along with their model predictions.

3.5. Segment lungs in XRay(s)

Lung segmentation from X-rays are required to automate boundary detection, enhancing diagnostic accuracy and efficiency in medical imaging analysis. In this exercise, we were given chest X-ray images of patients, and the segmented masks of their lungs. We need to employ a deep learning model that can segment lungs from XRays or find boundaries of the lungs.

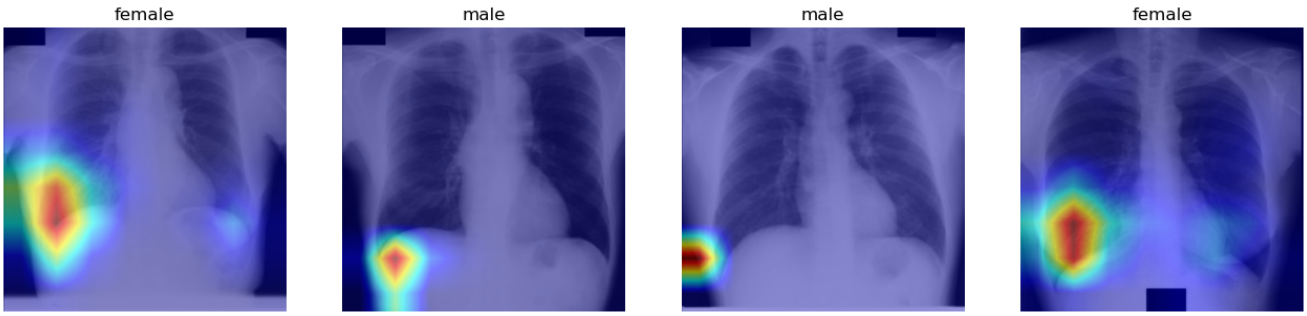
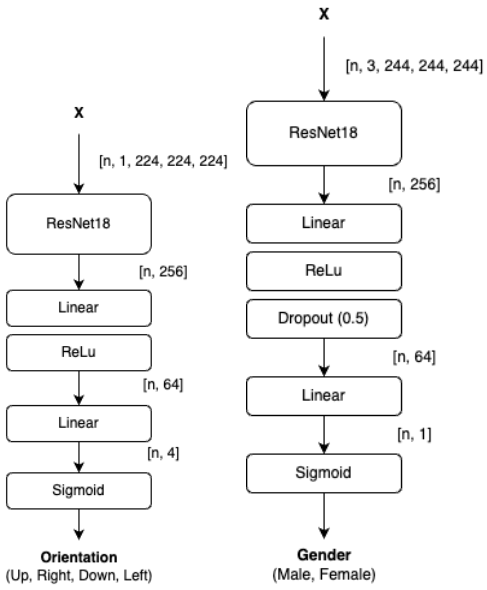
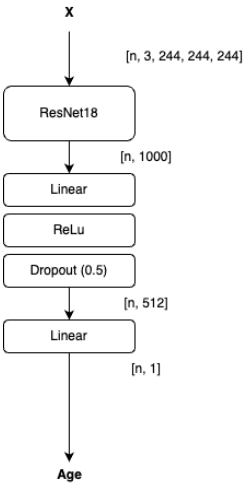


Figure 5. Gender predections on few test samples with GradCAM visualization.



(a) Orientation Model

(b) Gender Model



(c) Age Model

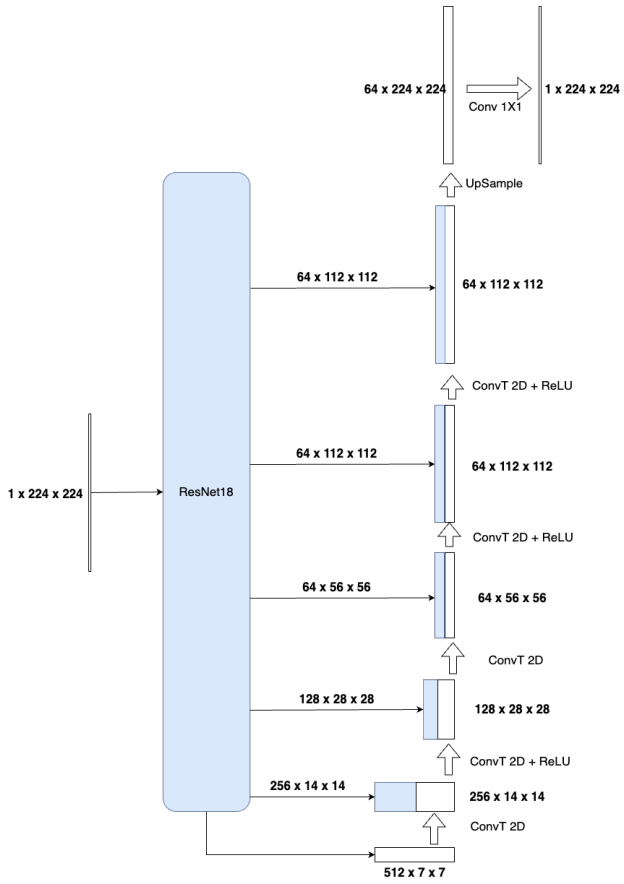


Figure 7. UNet for lung segmentation

3.5.1 Dataset

The provided dataset have 50 training x-ray images and 10 testing images, with their corresponding masks, thus we can observe that the provided dataset is very small. As there was no validtion set, I have randomly splited the training dataset into validation and training set in ration 1:9 respictively. The images in this dataset were converted to grayscale images with shape 256×256.

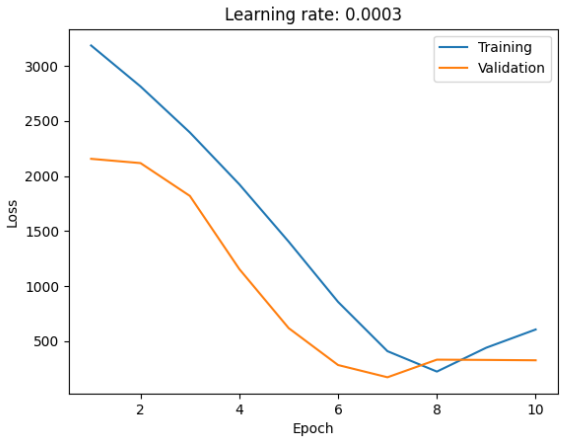


Figure 8. Age model learning curve on training and validation sets

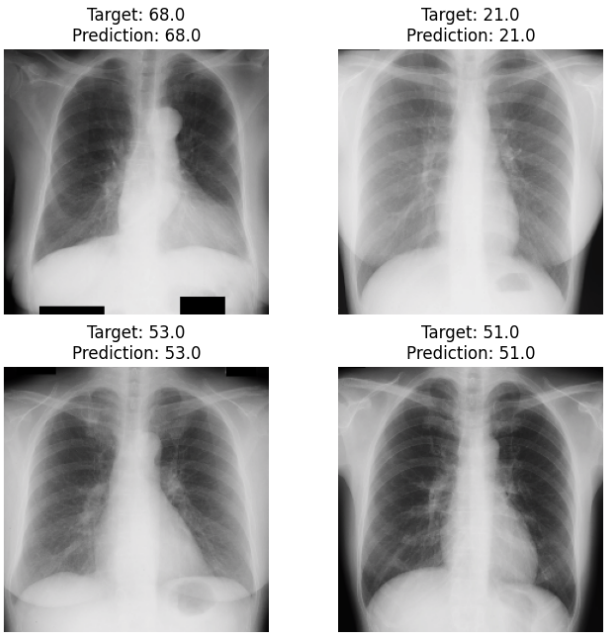


Figure 9. Age predictions on few samples of test data

3.5.2 Training

This being a segmentation problem, an encoder-decoder type of model needs to be employed. I have created a UNet as shown in Fig. 7. The model have a resnet18 encoder block which was pretrained on imagenet1k dataset and a decoder block. As the model might suffers from gradient vanishing, skip connections is used, thus a UNet is structure is achieved. Since, I am predicting the pixels of an image as 0 or 1, binary cross-entropy error was used for training with Adam optimizer with learning rate of 10^{-3}. Figure 10 illustrates the learning curve of the training and validation sets.

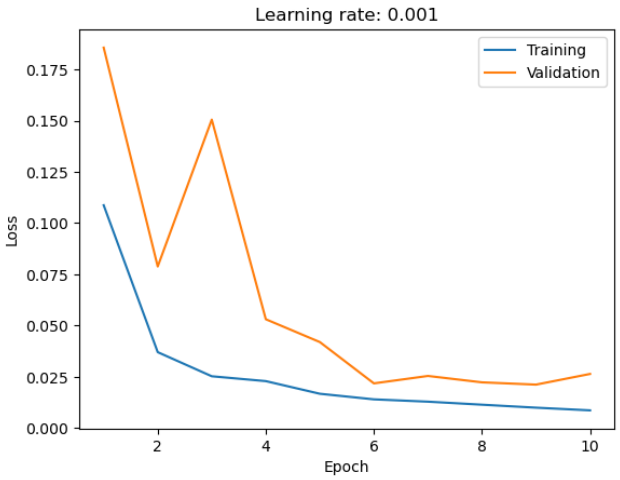


Figure 10. Lungs segmentation model learning curve on training and validation sets

3.5.3 Results

From the learning curve in Fig. 10 we can observe that the model have converged to lowest loss, and is not overfited on the dataset. The trained model is then tested on test set, and got mIoU score of 0.933 and mDICE score of 0.965. Figure 11 shows the predicted lung mask of a XRay and we can observe that model is able to generate masks which is similar to ground truth mask.

Figure 11. Predicted lung mask of an XRay by trained UNet

3.6. Segment chest XRay(s) into multiple classes

In Sec. 3.5, we predicted only one mask, that is of lungs. In this exercise, we have to segment an xray into 4 classes; lungs, heart, body and background. As the problem is similar to previous, we can utilize the same model as shown in Fig. 7 but with small modifications.

3.6.1 Dataset

The provided dataset have 200 training x-ray images and 47 testing images. As there was no validtion set, I have ran-

domly splitted the training dataset into validation and training set in ration 1:9 respictively. Validation set is important as it allows one to verify model performance on hold-out set. The images in the datset are of size 256×256. The mask images are grayscale image of same size (256×256) and have 4 unique pixel values (255:lungs, 170:body, 85:heart, 0:background).

The masks which are single channel image, have been converted to 4 channel image where each channel have information of a single mask.

3.6.2 Training

As we have to geenrate 4 masks, a small modification in model described in Fig. 7 has to be done before we can start the training. Instead of generating a single channel mask as output, we have to generate 4 channel masks, Thus, the output channel needs to be changed from 1 to 4. After this modification the model is trained for 10 epochs using Adam optimizer with binary cross-entropy loss and learning rate of 10^{-2}. Figure 12 shows the validation and training losses for each epochs.

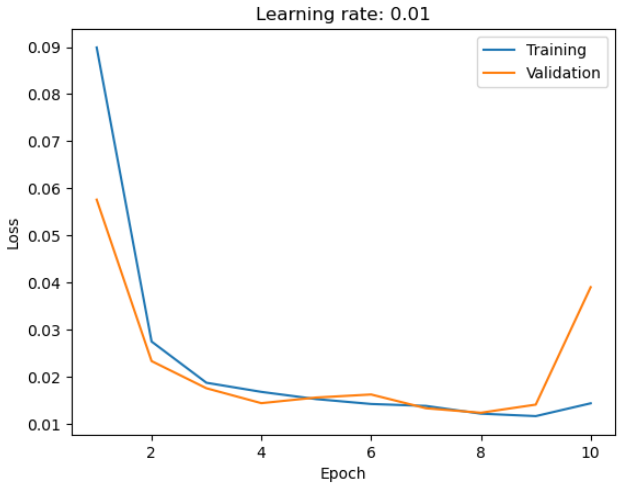


Figure 12. Learning curve of Modified UNet Fig. 7 for generating 4 masks.

3.6.3 Result

After training the model, the model was tested on the provided test set and IoU, DICE scores reported in Tab. 2. We can also observe the predicted masks in Fig. 13a and Fig. 13b and can say that model is able to generate near to accurate masks.

3.7. Locating lungs and heart in Chest XRays

Locating an organ is similar to finding boundary of an organ like we did in Sec. 3.6, infact it is slightly easier to to find location of an organ. Instead of generating masks for an image, we will pe predicting the bounding boxes of the organ, i.e. $(x_{min}, y_{min}, x_{max}, y_{max})$

	Background	Heart	Body	Lungs	Avg.
IoU	0.623	0.734	0.884	0.881	0.781
DICE	0.737	0.840	0.937	0.936	0.863

Table 2. Organs segmentation result

	Heart	Lungs	Avg.
IoU	0.803	0.907	0.855

Table 3. Heart and lungs localization result

3.7.1 Dataset

For this task also we will be utilizing the same dataset as we used in Sec. 3.6 which had 200 training x-ray images and 47 testing images. As this is a localization problem, I have used heart and lungs masks to find bounding boxes which will be used as target instead of organ masks. Thus, the shape of target will be $(2, 4)$, because we have to find 2 bounding boxes. The bounding boxes coordinates and also normalized to get these coordinates in range of $[0, 1]$

3.7.2 Training

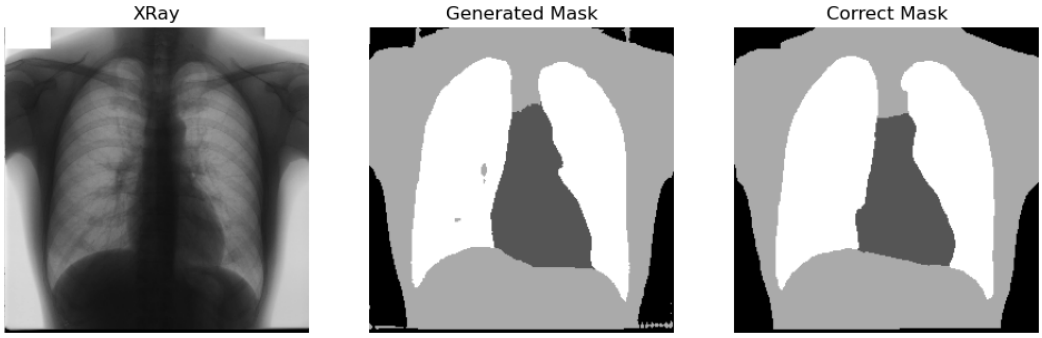
As this is simpler task than segmentation, a simpler model can be utilized, thus a resnet18 is used as backbone to extract features, with 8 as the output classes. Sigmoid activation function is also used as last layer to get output in range of $[0, 1]$. The first 4 outputs will be for first organ. The described model is trained for 10 epochs using Adam optimizer with binary cross-entropy loss and learning rate set to 10^{-2}. Figure 14 shows the validation and training losses for each epochs.

3.7.3 Result

After training the model, the model was tested on the provided test set and IoU reported in Tab. 3. We can observe from Tab. 3 and Tab. 3 that model is able to locate lungs with greater accuracy.

3.8. UnSupervised Learning - Anamoly Detections

There are scenarios in real-life where we have to find anamolies in images. An anamoly can also be find in chest xrays, and using those xrays might generate unsatisfactory results. In this task, we have to find anamolies in chest xrays (if present).



(a) Predicted masks, and target masks of a test XRay

(b) Individual predicted masks of a test XRay

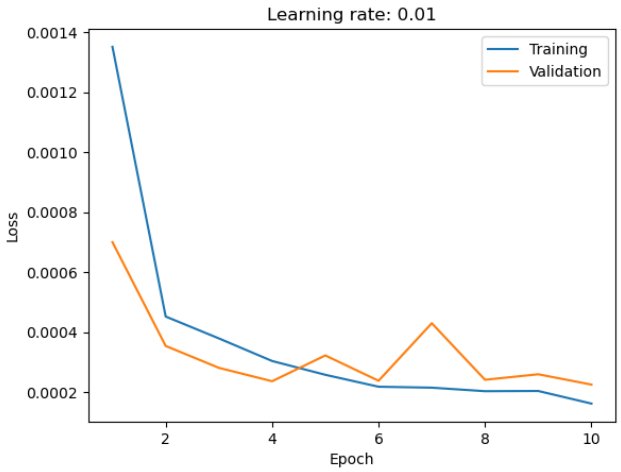


Figure 14. Learning curve of model with resnet backbone for predicting bounding boxes of heart and lungs

The basic methodology to tackle this problem is to first get a model to learn features from chest xrays, this can be done using unsupervised learning also if we don't have any labels of the xray. A model having encoder-decoder architecture can be trained to reconstruct the xray. Once trained, the encoder output can be utilized to find anamolies. An image having anamoly will have very different embeddings than a normal image.

3.8.1 Dataset

For this task, we were provided dataset having 200 normal xrays and 17 abnormal (flipped) xrays. For training, i.e. learning embeddings from a xray, all normal xrays are used, and for testing normal and abnormal xrays are used.

3.8.2 Training

An encoder-decoder model with resnet50 as encoder is employed to train to reconstruct the xray. The model is trained with Adam optimizer having learning rate of 10^{-3}. In initial experiments, MSE loss was utilized as reconstruction loss, but it dodn't gave satisfactory results as the generated xrays failed to generate the ribs and were quite blurry. To overcome this issue, SIMM loss was utilized which uses multiple quantities like luminance, to score the similarity between two images. The model was trained for 20 epochs, and Figure 16 shows the validation and training losses for each epochs.

3.8.3 Result

The trained model was able to generate xrays with an average SIMM score of 0.9707, and Fig. 17 shows two reconstructed xrays. The trained model is not able to generate the abnormal images very accurately as it is not able to capture flipped heart and ribs.

There are two methods in which we can use this model to find anamolies. First, we can use embeddings of encoder to cluster all the embeddings using a clustering algorithm,

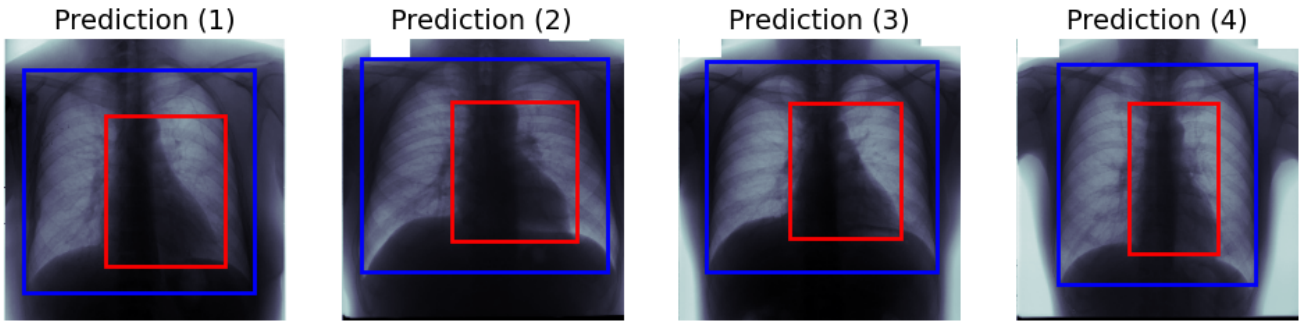


Figure 15. Predicted bounding boxes of heart and lungs

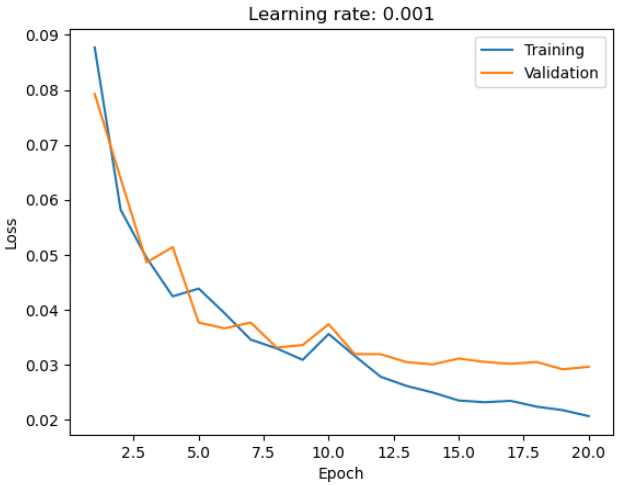


Figure 16. Learning curve of encoder-decoder model with resnet50 as encoder for reconstructing xray images

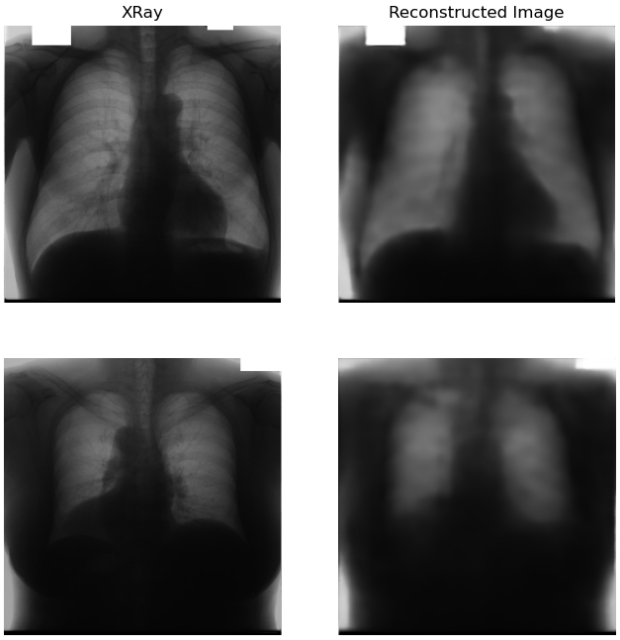


Figure 17. Reconstructed xrays of normal and abnormal xrays

and find clusters having maximum number of abnormal images. Second, this model can be utilized by anomalib library (github.com/openvinotoolkit/anomalib) to find anamolies. The first method didn't worked for this model, which might be due to small model.

This model will also be used in Sec. 3.9 for finding orientaion of xrays.

3.9. UnSupervised Learning - Finding orientation of xrays

This task is an extension of Sec. 3.8, where we had to find anamolies in xray images. In this task, we have to find orientation of xray images (Sec. 3.3) using unsupervised learning instead of supervised learning. The same idea will be used to find the orientation as we discussed in Sec. 3.8, i.e. apply clustering on learned embeddings of xrays.

3.9.1 Dataset

For this task, I have merged normal images xrays from Sec. 3.8.1 and xray images from Sec. 3.3.1. The orienation dataset as discussed in Sec. 3.3.1 is inversed so that they are constitent with images using which xray reconstruction model was trained in Sec. 3.8.

3.9.2 Result

The trained xray reconstruction model is used to find embeddings of all the xrays (normal, up, down, right, left). Figures 19a to 19d shows the reconstructed xray using original xray. With these images, we can observe that the model is not able to reconstruct xrays if they are not oriented properly, thus their embeddings can be used to cluster the xrays.

Figure 19e shows reconstructed xray of ”normal” image from dataset discussed in Sec. 3.8.1. The reconstructed image is better than other reconstructed images because the model was trained on this xray.

The embeddings of all the xrays are used to find 4 clusters using KMeans clustering algorithm. For testing this learned clusters, I have taken ”up” and ”normal” xrays as a single class because they are ooriented correctly. The predicted clusters and original clusters are used to calculate rand index which came out to be 0.98. This rand index means that the embeddings were enough to find cluster the xrays into 4 clusters. This result can also be visualized using t-SNE in Fig. 18, which shows the perfect 4 clusters.

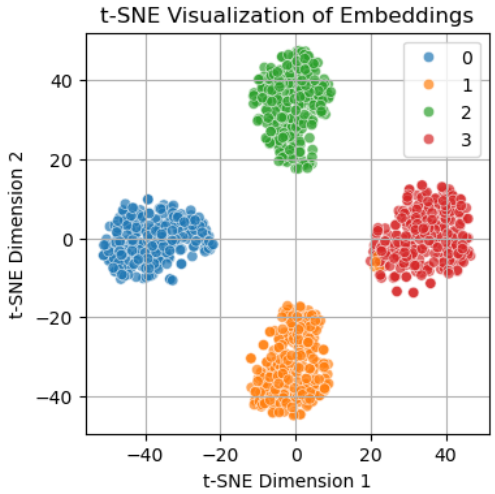


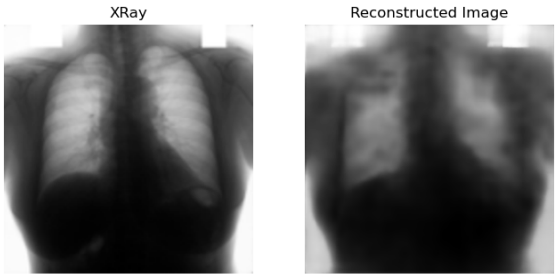
Figure 18

4. To-Do

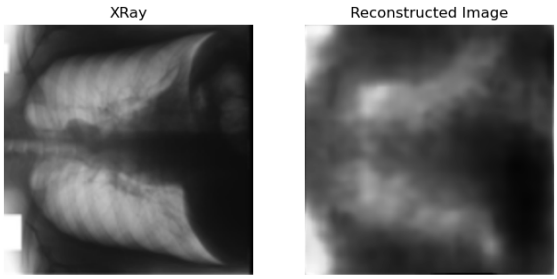
- Use anomalib with trained model to find anamolies in chest xrays.
- Improve report by paraphrasing.
- Include Conclusion

References

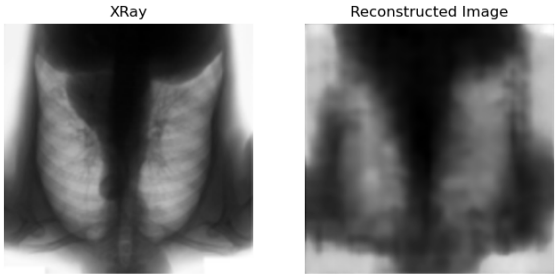
[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1



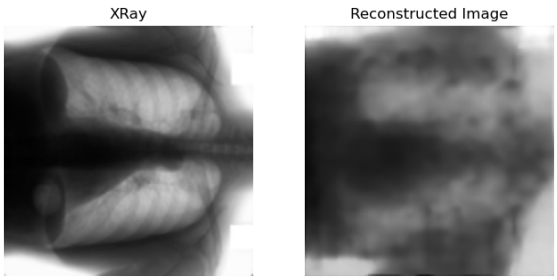
(a)



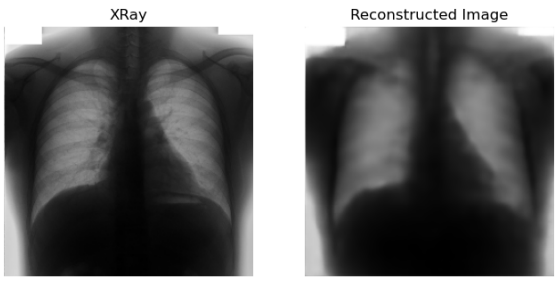
(b)



(c)



(d)



(e)