Predicting COVID-19 in Chest X-Ray Images

Healthcare/Computer Vision

Yash Maniyar
Computer Science
Stanford University
ymaniyar@stanford.edu

Madhu Karra
Computer Science
Stanford University
mkarra@stanford.edu

Arvind Subramanian
Computer Science
Stanford University
arvindvs@stanford.edu

1 Problem Description

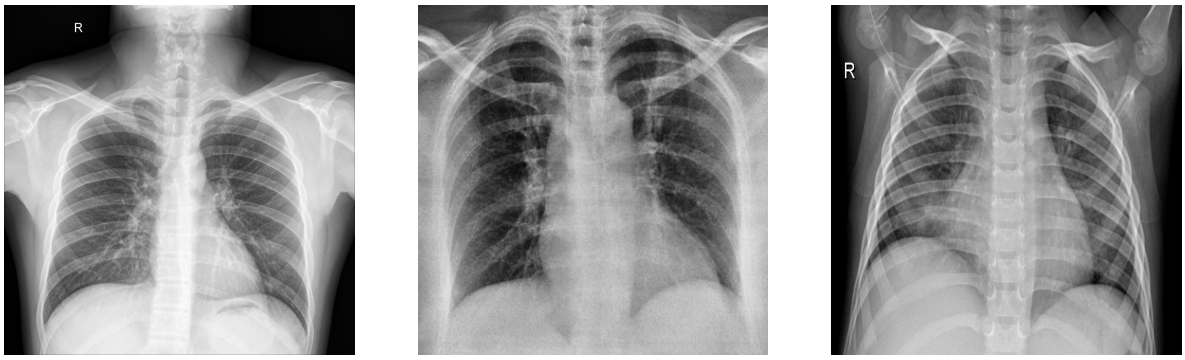
Most existing COVID-19 tests use nasal swabs and a polymerase chain reaction to detect the virus in a sample. We aim to develop an alternative, computer vision based method of identifying whether or not a patient is infected with COVID-19, viral pneumonia, or neither based on an X-ray image of their chest. We hope that such a model will expand access to quick, accurate diagnoses of COVID-19, and that the architecture we produce may be able to be re-purposed to detect other lung conditions.

2 Dataset and Preprocessing

2.1 Dataset

The dataset used in this project is composed of images from two separate chest x-ray datasets. Together, these images constitute 16 classes: 15 disease classes and 1 normal class.

The COVID-19 Radiography Database [2] is a collection of about 4000 chest X-ray images, each labeled as one of three classes: COVID-19, viral pneumonia, or “normal" (neither COVID-19 nor viral pneumonia). We used this dataset for the first part of our project.

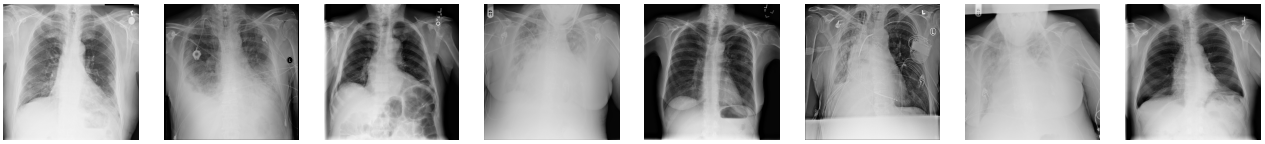


(a) Normal **(b)** COVID-19 **(c)** Viral Pneumonia

Figure 1: COVID Chest X-ray dataset split into 3 classes

CS230: Deep Learning, Winter 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

The NIH Chest X-Ray Dataset [6,7] is a collection of approximately 122,000 chest X-ray images, each labeled as one of 15 classes. We used this dataset in the second part of our project.



(a) Normal **(b)** Pneumonia **(c)** Atelectasis **(d)** Consolidation **(e)** Infiltration **(f)** Pneumothorax **(g)** Edema **(h)** Emphysema

(i) Fibrosis **(j)** Effusion **(k)** Pleural Thickening **(l)** Cardiomegaly **(m)** Nodule **(n)** Mass **(o)** Hernia

Figure 2: NIH Chest X-ray dataset split into 15 classes

We create a novel dataset for our project by sampling approximately 1000 images from each dataset. The total number of images from each class are shown in Table 1.

Table 1: Dataset classes split

Class	Label	# of Images
COVID	0	1000
Pneumonia	1	1000
Normal	2	1000
Atelectasis	3	1000
Consolidation	4	1000
Infiltration	5	1000
Pneumothorax	6	1000
Edema	7	628
Emphysema	8	892
Fibrosis	9	727
Effusion	10	1000
Pleural Thickening	11	1000
Cardiomegaly	12	1000
Nodule	13	1000
Mass	14	1000
Hernia	15	110

2.2 Preprocessing

Since the X-ray images are square and of varying resolution, we standardized the resolution to 256×256 pixels. Afterwards, each pixel value was normalized to be between 0 and 1. Additionally, we convert each image to grayscale. Finally, we subtract the mean value of each image's pixels then divide by the standard deviation.

We split the dataset into training, validation, and test sets. The training set used 98% of the data, the validation set used 1% of the data, and the test set used 1% of the data.

We created a dataloader to efficiently load the training and validation datasets, randomly shuffle the data, and batch the data. The dataloader performs the requisite transforms (resize to 256×256, scale pixel values between 0 and 1) when loading the particular batch to be used. We use a batch size of 32 for most of our models.

3 Experiments

3.1 Baseline: 4-Layer Conv Network

As a starting point, we wrote a 4 layer convolutional neural network with Dropout. Full details of this network are given in Table 2 in the Appendix. We trained this model separately on the COVID Radiography dataset and the merged dataset.

3.2 ResNet-Inspired Networks

In an attempt to capture the complexity of the merged, 16-class dataset, we drew inspiration from ResNet [8], a popular model used for multi-class image classification. We wrote two versions of a "Conv-Skip Block" (A and B), whose exact structure is defined in Tables 3 and 4 in the Appendix. The basic different between these blocks is that Conv-Skip Block A has three convolutional layers, and adds the identity of the input to the later activation, while Conv-Skip Block B has only two convolutional layers: one has a user-defined number (o) of 3x3 filters, while the other is a simple 1x1 convolution done on the identity of the input, to be added to the later activation. Both Blocks use Dropout and Batch Normalization.

3.2.1 Model A

Our first network (defined in Table 5 in the Appendix) stacks three Conv-Skip Block As, followed by fully connected layers. This model used no Dropout (p=0) and was trained on the merged, 16-class dataset.

3.2.2 Model B

In an attempt to counter over-fitting, Model B (Table 6 in the Appendix) has a very similar overall architecture as Model A (with one extra convolutional layer at the beginning), but with Dropout. This model was also trained on the merged, 16-class dataset.

3.2.3 Model C

To further reduce over-fitting, Model C (Table 7 in the Appendix) removed uses Conv-Skip Block B instead of A (making for a slightly smaller model). Dropout was also made more aggressive in this model; we increased drop probability. Model C was trained separately on both the merged 16-class dataset, and the smaller 3-class COVID Radiography dataset.

4 Results and Analysis

4.1 Trained on COVID-19 Radiography Dataset

Our first task was to train our models on the full set of images provided in the radiography 3-class dataset. The task was to classify X-ray images as either normal, pneumoniac or COVID lungs. For this dataset, we used a suite of CNNs with different variations and techniques, as described in the following sections. We split our data into train, test, and validation splits, each with a roughly even distribution of labels (we chose a 98-1-1 split).

4.1.1 Baseline: 4-Layer CNN

Our 4-layer baseline network converged quickly (Figure 6, Appendix); after training for 10 epochs, we observed a **test-set accuracy of 90%** and a **validation-set accuracy of 92%** Given that this relatively simple model performed so well, we try Model C, a more complex model, on the same dataset to see if we can boost performance.

4.1.2 Model C

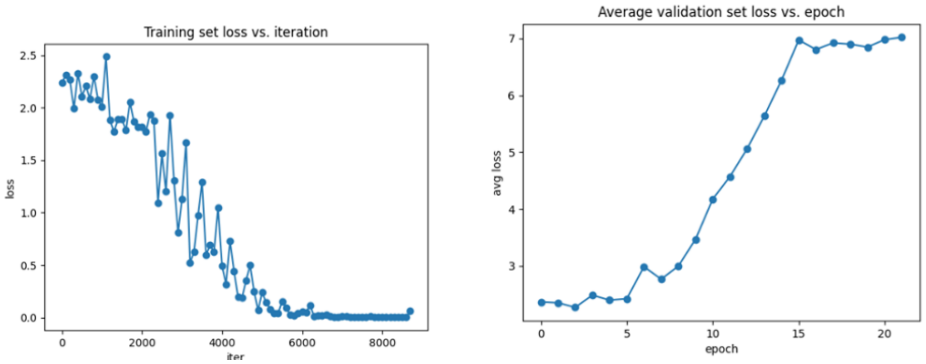
After running Model C for 15 epochs, we observe a **test-set accuracy of 92%** and a **validation-set accuracy of 91%**, slightly outperforming our 4-layer conv network. We further experimented with learning rate and hidden sizes, although none of these experiments yielded significant improvements in performance.

4.2 Trained on Novel Expanded Dataset: NIH Chest X-Ray

After exhausting our models' capabilities on the 3-class radiography dataset, we increase the complexity of our task by including 16 classes (see Dataset section for a description).

4.2.1 Model A

We first train model A, heavily inspired by the ResNet architecture, on the 16-class dataset. We observe dramatic over-fitting, with our train accuracy reaching a peak of **92%** after 20 epochs, but a validation accuracy that never exceeded **28%**. Figure 3 shows the learning curves; we see a steadily decreasing training loss alongside a monotonically increasing validation loss. For our next set of experiments, we decide to reduce the complexity of the model and introduce various regularization techniques in an attempt to combat the observed over-fitting problem.



(a) Train loss **(b)** Validation loss

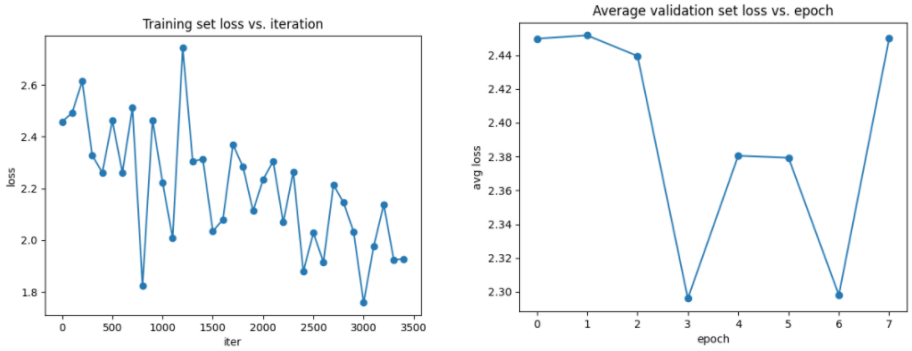
Figure 3: Learning Curves from Model A on 16-class dataset

4.2.2 Model B

We train Model B, hoping it might reduce over-fitting by adding a penalty to the loss function through dropout regularization. After 15 epochs, we observed a training-set accuracy of **73%** with a validation accuracy of **19%**. Figure 7 in the Appendix shows the learning curves; we see over-fitting once again, with worsened performance on both data splits, when compared to Model A.

4.2.3 Model C

We test out a much smaller and less-complicated model, Model C, on the same dataset and observe results. After 8 epochs, we observed a train accuracy of **65%** and a test-set accuracy of **30%**. Although the discrepancy between train and validation accuracies was smaller than those observed with Model A and Model B, it is clear that Model C has a harder time picking up features of the dataset, likely due to the smaller number of learnable parameters. See Figure 4 for learning curves from Model C. We treat Model C as our best model for this task, since it achieved the highest accuracy on unseen data.



(a) Train loss **(b)** Validation loss

Figure 4: Learning Curves from Model C on 16-class dataset

4.3 Analysis

Our final metrics on the 16-class extended task were much worse than our metrics for the 3-class dataset, showing us that the extended dataset presents a much more challenging classification task. We saw dramatic over-fitting with more complex models, and a lack of generalizability and sometimes under-fitting with the less-complicated models that we tried. These results suggest that a model, given more and more learnable parameters, tends to memorize this data rather than pick up on generalizable patterns. Since the 3-class version of this task seemed to be have a much more learnable objective, we hypothesize that some of the classes introduced in the NIH dataset are harder to differentiate from one another than the classes present in the Covid-19 Chest X-ray dataset (pneumonia, covid, healthy). To test this hypothesis, we generate a confusion matrix for Model C on the test set (see Figure 5 below). We see that our model was did a good job of detecting COVID lungs, healthy lungs, and pneumonia lungs, which were the three classes present in the Covid-19 Chest X-ray dataset. The model had a much more difficult time discriminating between the other lung diseases introduced in the NIH dataset. As a reference, we ran a pre-trained ResNet-18 model, and we observed the same trends (over-fitting to training dataset with poor performance on unseen data).

To further understand the errors our best model made on the 16-class data, we calculated the percentage of test examples whose correct class was within the top 3 from our predictions. It was good to see that although Model C achieved a test accuracy of 30%, around 53% of test examples had the correct label in the top 3 classes in our output.

Upon looking through the NIH dataset, it appears that certain diseases have a lot of variation in the images corresponding to that label. It's possible that some of these diseases manifest differently in different patients, leading to higher variation in chest X-ray features (see Figure 8 for example images for cardiomegaly). Furthermore, the labels for this dataset were created through an NLP bootstrapping algorithm; although the expected accuracy is said to be higher than 90%, perhaps noise introduced by mislabeled examples makes it harder for models to learn an objective function.

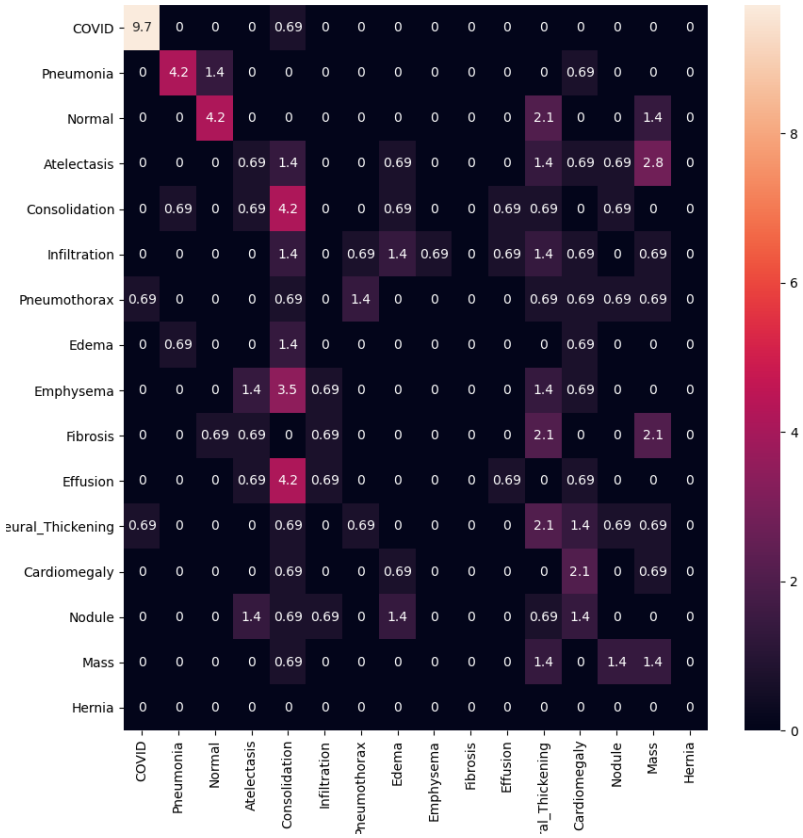


Figure 5: Confusion Matrix (y = true label, x = prediction)

References

[1] Pasa, F., et al. "Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization." Scientific reports 9.1 (2019): 1-9.

[2] Chowdhury, Muhammad EH, et al. "Can AI help in screening viral and COVID-19 pneumonia?." IEEE Access 8 (2020): 132665-132676.

[3] Hemdan, Ezz El-Din, Marwa A. Shouman, and Mohamed Esmail Karar. "Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images." arXiv preprint arXiv:2003.11055 (2020).

[4] Zhang, J., et al. "Viral pneumonia screening on chest x-ray images using confidence-aware anomaly detection (2020)." arXiv preprint arXiv:2003.12338.

[5] Abbas, Asmaa, Mohammed M. Abdelsamea, and Mohamed Medhat Gaber. "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network." Applied Intelligence (2020): 1-11.

[6] Wang, X., et al. "Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases." IEEE CVPR. 2017.

[7] https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

5 Appendix

Github repo: https://github.com/arvindvs/COVID-19-detection

Table 2: 4-layer CNN Architecture

Layer	num. output features	size of output	kernel size
Input	1	256 x 256	–
Conv2D C1	16	256 x 256	5 x 5
MaxPool2D	16	128 x 128	2 x 2
ReLU	16	128 x 128	–
Dropout(p=0.1)	16	128 x 128	–
Conv2D C2	32	128 x 128	5 x 5
MaxPool2D	32	64 x 64	2 x 2
ReLU	32	64 x 64	–
Dropout(p=0.2)	32	64 x 64	–
Conv2D C3	64	64 x 64	3 x 3
MaxPool2D	64	32 x 32	2 x 2
ReLU	64	32 x 32	–
Dropout(p=0.5)	64	32 x 32	–
Conv2D C4	128	32 x 32	3 x 53
MaxPool2D	128	16 x 16	2 x 2
ReLU	128	16 x 16	–
Flatten	1	32,768	–
Linear FC1	1	2048	–
ReLU	1	2,048	–
Linear FC2	1	512	–
ReLU	1	512	–
Linear FC3	1	16	–
Softmax	1	1	–

Table 3: Conv-Skip Block A (i, h, o, p)

Layer	num. output features	size of output	kernel size
Input	i	N x N	–
Conv2D C1	h	N x N	3 x 3
Batchnorm	h	N x N	–
ReLU	h	N x N	–
Conv2D C2	i	N x N	3 x 3
Add Input	i	N x N	–
ReLU	i	N x N	–
Conv2D C3	o	N x N	3 x 3
ReLU	o	N x N	–
Dropout(p)	o	N x N	–
MaxPool	o	N/2 x N/2	–

Table 4: Conv-Skip Block B (i, o, p)

Layer	num. output features	size of output	kernel size
Input	i	N x N	–
Conv2D C1	o	N x N	3 x 3
Batchnorm	o	N x N	–
ReLU	o	N x N	–
Conv2D C2(Input)	o	N x N	1 x 1
Add C2(Input)	o	N x N	–
ReLU	o	N x N	–
Dropout(p)	o	N x N	–
MaxPool	o	N/2 x N/2	–

Table 5: Model A

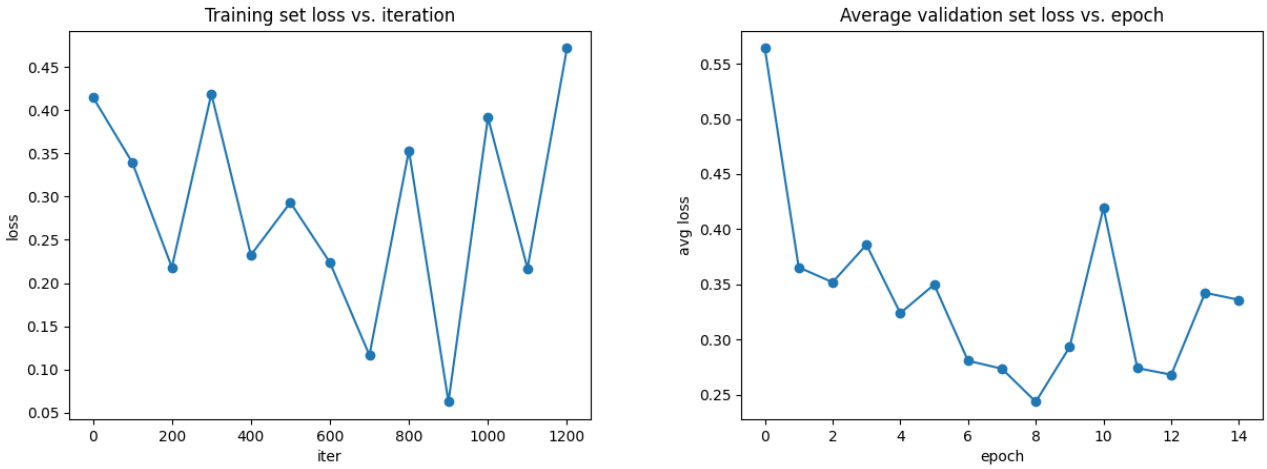
Layer	num. output features	size of output	kernel size
Input	1	256 x 256	–
Conv-SkipA(h=16, p=0.0) SB1	32	128 x 128	–
Conv-SkipA(h=16, p=0.0) SB2	64	64 x 64	–
Conv-SkipA(h=16, p=0.0) SB3	32	32 x 32	–
Flatten	1	32,768	–
Linear FC1	1	512	–
ReLU	1	512	–
Linear FC2	1	16	–
Softmax	1	1	–

Table 6: Model B

Layer	num. output features	size of output	kernel size
Input	1	256 x 256	–
Conv2D C1	16	256 x 256	5 x 5
ReLU	16	256 x 256	–
Conv-SkipA(h=16, p=0.1) SB1	64	128 x 128	–
Conv-SkipA(h=32, p=0.2) SB2	128	64 x 64	–
Conv-SkipA(h=64, p=0.3) SB3	128	32 x 32	–
Flatten	1	32,768	–
Linear FC1	1	512	–
ReLU	1	512	–
Linear FC2	1	16	–
Softmax	1	1	–

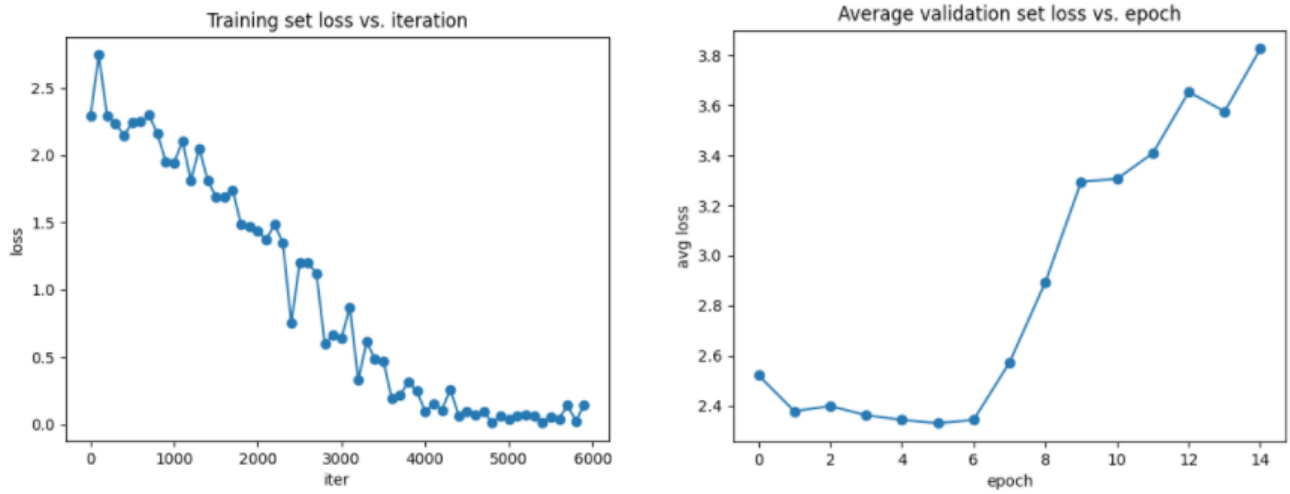
Table 7: Model C

Layer	num. output features	size of output	kernel size
Input	1	256 x 256	–
Conv2D C1	16	256 x 256	5 x 5
MaxPool2D	16	128 x 128	2 x 2
ReLU	16	128 x 128	–
Conv-SkipB(p=0.1) SB1	32	64 x 64	–
Conv-SkipB(p=0.3) SB2	64	32 x 32	–
Conv-SkipB(p=0.6) SB3	128	16 x 16	–
Flatten	1	32,768	–
Linear FC1	1	512	–
ReLU	1	512	–
Linear FC2	1	16	–
Softmax	1	1	–



(a) Train loss **(b)** Validation loss

Figure 6: Learning Curves on 4-layer Conv Network on 3-class dataset



(a) Train loss **(b)** Validation loss

Figure 7: Learning Curves from Model B on 16-class dataset

(a) Cardiomegaly 1 **(b)** Cardiomegaly 2 **(c)** Cardiomegaly 3

Figure 8: COVID Chest X-ray dataset split into 3 classes