COVID-19 classification of chest CT scans using Bidirectional LSTM and Attention trained on large, novel dataset

Sifan Ye
Department of Computer Science
Stanford University
sifanye@stanford.edu

Abstract

In this project, I used a large, novel COVID-19 chest CT dataset to train and compare several image sequence classification models to classify if a given CT scan is COVID-19 positive or not. The trained models achieves near perfect classification on in and out of distribution examples, showing robustness to unseen samples.

1 Introduction

Chest CT scans provide a fast complement to RT-PCR for COVID-19 diagnosis [1], [2]. However, in a pandemic like COVID-19, where fast diagnosis of a large population is needed, the scale of chest CT scan data puts significant pressure on radiologists. In this situation, computer aided diagnosis can be used to help radiologists to process CT scans faster, and one such application could be AI models trained for chest CT classification.

In contrast to chest X-ray scans, which usually contains a single image for each scan [3] [4] [5], chest CT scans usually contains a sequence of images, and hence classification of CT scans would involve training an image classification model on a large collection of CT scans, by the number of sequences not single slices. Many existing public datasets of COVID-19 chest CT scan suffers from limited size, either in the form of limited number of scans, (20 patients in [6], 50 patients in [7]) or limited sequence length (349 images from 216 patients in [8]). In this project, I will be using a large, non-public COVID-19 chest CT dataset which consists of chest CT scans of 450 patients, to train an image classification model. Since important features may be present across the slices of CT scans, the aim is to train a model that utilizes the sequential dimension of CT scans. Since the image output may vary based on different CT machines and settings, such as different contrast windows, spatial resolutions, patient position and orientation, etc., another challenge would be to train the model to be robust when applied to samples from different distributions.

2 Related work

There are many deep learning models for COVID-19 classification of chest X-rays, [3] [4] [5] and all of them employs deep convolutional neural network for image feature extraction and classification. Many existing models treat CT classification similarly as a single image classification task and would either require first extract Region of Interest images then feed single images to a classification model [9] or use a separate pipeline to extract 3D features using off-the-shelf softwares while keeping single image classification for the 2D slices [10]. Another approach used by existing models is to run each slice of the scan through the same CNN model to extract features, apply max pooling to combine the

2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

features into one then fed to a fully connected layer for classification [11]. Although this approach considers the entire CT scan as a whole, it is not order sensitive.

Treating CT classification as a sequence to sequence task, where the input sequence consists of the CT slices and the output consists of a single classification probability, sequential models such as LSTM [12] can be applied. Such application of LSTM can be seen in other image sequence recognition and description tasks [13]. Another common mechanism seen in sequence to sequence task is attention mechanism, and is widely applied in NLP tasks such as neural machine translation [14].

3 Dataset

The non-public COVID-19 chest CT scan dataset is provided by Shayan Alipour at Pi School. The dataset consists of anonymized CT chest scans of 450 patients, with each patient having 2 to 5 axial scans and most patients having 1 coronal scan. Each scan contains 40 - 300 slices depending on the slice thickness. The majority of the dataset is presented in the lung contrast window, but some are presented in the non-contrast window. The data is provided as 16-bit PNG images of pixel arrays extracted from DICOM files scaled by rescale slope and intercept and ordered by instance number, and metadata in the DICOM files such as slice thickness is not available. For this project, I have extracted a total of 757 sequences from 350 patients, and are split into 680 for training, 38 for validation and 39 for testing.

The negative samples consists of public datasets of non-COVID chest CT datasets. For this project I have extracted 246 sequences from approximately 60 patients from the RIDER lung CT dataset for lung cancer [15], which is split to 210 for training, 18 for validation and 18 for testing, and 200 sequences from 50 patients from ELCAP Public Lung Image Database [16], which is split to 160 for training, 20 for validation and 20 for testing.

Since the majority of the COVID CT dataset consists of sequences of approximately 40 to 80 slices, but some sequences of the COVID CT dataset and all of the non-COVID CT dataset have approximately 200 to 300 slices due to different slice thickness settings, the longer sequences are broken down into 4 sequences each by selecting every 4th image to increase the number of training samples.

To test the robustness of the model on out of distribution COVID-19 classification, I am using the 20 chest CT scans from COVID-19 image data collection [6] which consists of 10 non-contrast window sequences from coronacases, each consisting of 200 - 300 slices, and 10 lung contrast window sequences from radiopedia, with a more varied sequence lengths of 40 - 400. All those sequences are kept in their original length to test for model robustness for different scan thicknesses.

Dataset	Training	Validation	Test	Special Test
COVID-19	680	38	39	0
RIDER	210	18	18	0
ELCAP	160	20	20	0
Coronacases	0	0	0	10
Radiopedia	0	0	0	10

Table 1: Dataset Breakdown

4 Methods

4.1 Transfer Learning with ResNet and GoogLeNet

For this project, I applied the transfer learning paradigm, starting with using ResNet50 [17] and GoogLeNet [18] pretrained on ImageNet [19] to extract a sequence of features represented as vectors from the sequence of images of the CT scan by obtaining the input of the final fully connected layers of those models. The sequence of image feature vectors are then processed differently to obtain a classification result, and the performance of those different methods are compared in this project.

4.2 Max Pooling

Similar to the approach presented by Li [11], the baseline model employs max pooling to combine the results for each slice. Each slice of the CT scan is fed to the pretrained model with frozen weights to extract feature vectors, which are then fed to a shared weight trainable linear layer to obtain logits for each slice. The logits are then combined using max pooling and then the final probability of the sequence is produced using the sigmoid activation function. This method is not sequence sensitive and will act as a baseline for comparison with other sequence sensitive methods. (Figure 2 in Appendix)

4.3 LSTM

Since we can consider CT classification as a sequence classification problem, I applied the LSTM sequential model [12] to aggregate the slice features in an order sensitive way. As the CT dataset contains axial chest scans in both directions, the extracted feature vectors sequence is fed to a bidirectional LSTM for robustness in input order. The final hidden state of the LSTM is then fed to a linear layer which produces a single logit that is then fed to the sigmoid activation function for the final classification probability for the entire sequence. Since the ImageNet [19] pretraining task is vastly different from CT classification, especially the higher level features that need to be extracted, the weights of the last 2 modules of ResNet and GoogLeNet are unfrozen during training to finetune the models to recognize relevant high level features for this specific task. (Figure 3 in Appendix)

4.4 LSTM with attention

Since each CT scan is significant in length, a single hidden state from LSTM may not provide enough information for an accurate classification. The attention mechanism [14] provides a method for the model to search for slices from the input sequence that are relevant for the final classification, and also makes the model more explainable and helpful by providing an insight to which slices were important in the model's prediction.

Viewing the classification problem as a sequence to sequence task, where the output sequence always have length of 1, I implemented a simple attention mechanism as follows. For feature vectors of input sequence $\mathbf{X} = [\mathbf{x_0}, \mathbf{x_1}, ..., \mathbf{x_t}]$ of length t, we obtain a sequence of LSTM hidden states $\mathbf{H} = [h_0, h_1, ...h_t]$, where each h_t is a column vector representing the hidden state for input x_t. The attention weights are computed as $\mathbf{W} = \text{softmax}(\mathbf{H}^T \cdot h_t)$ which is then applied to the sequence of hidden outputs $\mathbf{H}$ to obtain a new last hidden output by $h_t^{'} = \mathbf{H} \cdot \mathbf{W}$. The final logit is then obtained by passing $h_t^{'}$ to a linear layer which is then converted to the final classification probability using the sigmoid activation function. (Figure 4 in Appendix)

5 Experiments and Results

5.1 Training Hyper-parameters

For this project, I am using the Adam optimizer [20] with learning rate 0.001 found during tuning. Due to limited computing resources available, I chose to train with batch size of 8, and bidirectional one-layer LSTM with hidden size of $\frac{1}{2}$ of the size of the extracted image features. Since the task is a binary classification task, binary cross entropy is chosen to be the loss function. Those hyper-parameters are held constant for all the models compared for the experiment.

5.2 Data augmentation and weighted sampling

Due to limited data available and the goal to train a model robust to varying inputs, the images are augmented to produce more samples for training. In contrast to other image classification tasks, for chest CT scans, flipping doesn't produce valid inputs, and so data augmentation is performed by random cropping and random rotation. As the input dimensions for both ResNet and GoogLeNet are both 224 by 224 pixels, the input images are first resized to 256 by 256 pixels, rotated randomly between -10 and 10 degrees as patient positions aren't expected to vary too much, then cropped to 224 by 224 pixels.

Since there is a class imbalance in the data, the input sequences are randomly sampled with the sample weights inversely proportional to the total number of samples in the class.

5.3 Results

Each model is trained for 12 epochs and the losses during training are shown in figure 1, with model performance similar on the training set and the validation set. We can observe that the choice of the pretrained feature extraction model has negligible impact on convergence speed. LSTM with attention shows the fastest convergence speed while LSTM without attention shows no signs of loss decrease. Max pooling shows loss decrease slower than LSTM with attention but is significantly faster than LSTM without attention.

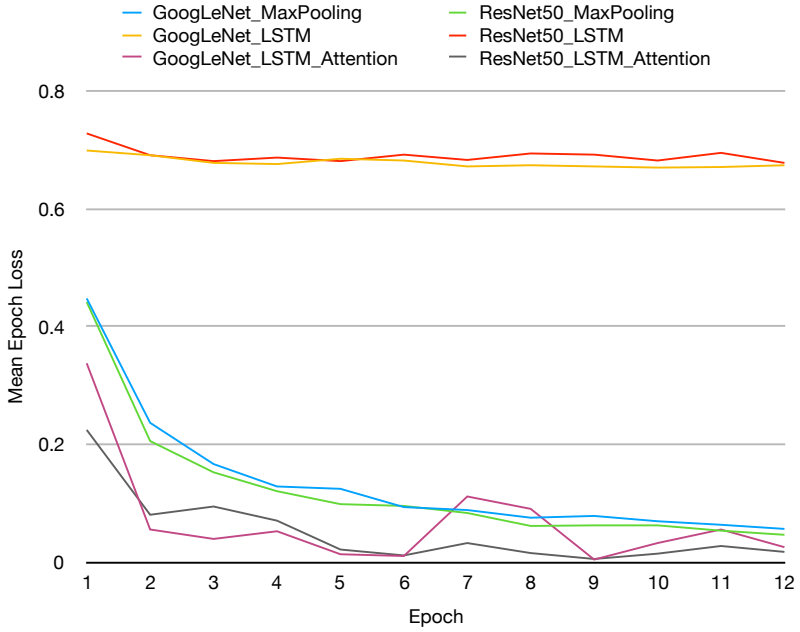

Figure 1: Mean Epoch Loss

The models are then compared by performances, measured by AP (average precision) AUC-PR (area under precision-recall curve) and AUC-ROC (area under receiver operating characteristic curve), on the test set with weights after the 6th epoch and the 12th epoch, as shown in table 2. We can see that the models have comparable, near perfect classification results on the held out datasets, with only ResNet50 with LSTM showing slightly worse performance at epoch 6.

Model	AP @ 6	AP @ 12	AUC-PR @ 6	AUC-PR @ 12	AUC-ROC @ 6	AUC-ROC @ 12
ResNet50 MaxPooling	1	1	1	1	1	1
GoogLeNet MaxPooling	1	1	1	1	1	1
ResNet50 LSTM	0.97	1	0.97	1	0.95	1
GoogLeNet LSTM	1	1	1	1	1	1
ResNet50 LSTM Attention	1	1	1	1	1	1
GoogLeNet LSTM Attention	1	1	1	1	1	1

Table 2: Performance of Different Models on the Test Set at Epoch 6 and 12

The models are also compared by performances on the out-of-distribution test set. The test set consists of only COVID positive cases, and the prediction probabilities are given in tables 3, 4 with probabilities < 0.5 marked in red. (See Appendix for LSTM with Attention sample outputs)

The results show that overall, the models with LSTM without attention performs the worst, with the max pooling models performing better and the models with LSTM and attention performing the best. This is consistent with the average loss during training, and can be attributed to the fact that without attention, the last hidden output of the LSTM model carries very limited information, compared to max pooling and LSTM with attention. The results also show that the LSTM with attention model

Model	1	2	3	4	5	6	7	8	9	10
ResNet50 MaxPooling @ 6	0.83	1.00	0.98	0.99	0.98	0.96	1.00	0.98	1.00	0.96
ResNet50 MaxPooling @ 12	0.83	1.00	0.99	0.99	0.99	0.98	1.00	0.99	1.00	0.99
GoogLeNet MaxPooling @ 6	0.64	0.97	0.97	0.96	0.73	0.95	0.64	0.97	0.97	0.89
GoogLeNet MaxPooling @ 12	0.76	0.99	0.99	0.98	0.83	0.97	0.76	0.99	0.99	0.96
ResNet50 LSTM @ 6	0.60	0.61	0.60	0.61	0.60	0.61	0.61	0.61	0.61	0.61
ResNet50 LSTM @ 12	0.57	0.57	0.57	0.54	0.57	0.57	0.57	0.53	0.57	0.57
GoogLeNet LSTM @ 6	0.61	0.61	0.60	0.61	0.61	0.61	0.60	0.61	0.61	0.61
GoogLeNet LSTM @ 12	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74
ResNet50 LSTM Attention @ 6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ResNet50 LSTM Attention @ 12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
GoogLeNet LSTM Attention @ 6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
GoogLeNet LSTM Attention @ 12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 3: Prediction Probabilities on 10 Positive Samples from coronacases

Model	1	2	3	4	5	6	7	8	9	10
ResNet50 MaxPooling @ 6	0.98	0.84	0.63	0.90	0.77	0.53	0.91	0.72	0.38	0.42
ResNet50 MaxPooling @ 12	0.99	0.92	0.65	0.94	0.82	0.58	0.93	0.78	0.43	0.56
GoogLeNet MaxPooling @ 6	0.38	0.32	0.48	0.15	0.19	0.24	0.26	0.89	0.19	0.28
GoogLeNet MaxPooling @ 12	0.46	0.21	0.40	0.06	0.09	0.17	0.28	0.96	0.10	0.25
ResNet50 LSTM @ 6	0.61	0.61	0.60	0.61	0.59	0.61	0.61	0.61	0.45	0.60
ResNet50 LSTM @ 12	0.57	0.57	0.57	0.57	0.55	0.56	0.56	0.57	0.23	0.54
GoogLeNet LSTM @ 6	0.61	0.58	0.61	0.59	0.61	0.58	0.60	0.57	0.56	0.60
GoogLeNet LSTM @ 12	0.74	0.74	0.73	0.76	0.73	0.74	0.74	0.69	0.65	0.73
ResNet50 LSTM Attention @ 6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ResNet50 LSTM Attention @ 12	1.00	0.83	1.00	0.99	1.00	0.98	1.00	0.01	0.97	1.00
GoogLeNet LSTM Attention @ 6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
GoogLeNet LSTM Attention @ 12	1.00	1.00	1.00	0.23	1.00	0.98	1.00	1.00	0.10	0.18

Table 4: Prediction Probabilities on 10 Positive Samples from radiopedia

almost classifies COVID-19 cases perfectly, showing robustness to inputs from different distributions, and also shows the model converges at approximately epoch 6 during training and longer training may end up hurting the model's performance as it start to fluctuate around the local optimum.

The models are also shown to perform slightly worse on the radiopedia cases which are in the lung contrast window, just like the majority of the training samples, and are also closer in sequence length to the training set. This is surprising since the coronacases samples are further from the training distribution.

The results show that the choice of different pretrained models almost shows no significant impact on the classification results. What is surprising is that the GoogLeNet with max pooling performs significantly worse on the radiopedia cases which cannot be explained without further experiments and investigation.

6 Conclusion and Further Work

In conclusion, in this project, I have successfully implemented and trained models that can almost perfectly classify COVID-19 cases from chest CT scans, by applying LSTM with attention to image features extracted using pretrained deep convolutional neural networks, showing the effectiveness of the attention mechanism in this application and robustness to samples out of the training distribution.

Due to the limited availability of public chest CT datasets of COVID-19 negative samples, the model can only classify between COVID-19 and lung cancer chest CT scans. Hence one possible further work is to train the model on large datasets of other lung diseases, such as pneumonia caused by other pathogens, pulmonary embolism, and datasets of healthy lungs, and investigate the effectiveness of the model in classifying COVID-19 cases from those conditions, especially those that may display similar CT image features. One other possible further work with a more diverse dataset would be to extend the task from binary classification to multi-class classification and train and evaluate the model on the more general-purpose task.

In the comparison experiments, max pooling has shown significantly better performance compared to LSTM without attention, which means that the order of the input sequence may not be as important compared to having a presentation of all the slides at the final layer, which can also be provided by the attention mechanism, with or without LSTM. To formally investigate the importance of

ordering, another further work would be to implement a model with attention only, and compare its effectiveness to a model with attention and LSTM.

7 Contributions

Non-COVID-19 data collection, literature review, model implementation, training and testing, and report writing is done by Sifan Ye.

The non-public COVID-19 dataset is provided by Shayan Alipour at Pi School, and is preprocessed for this project by Sifan Ye

References

[1] Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Tao Q, Sun Z, and Xia L. Correlation of chest ct and rt-pcr testing for coronavirus disease 2019 (covid-19) in china: A report of 1014 cases. *Radiology*, 2020.

[2] Dahai Zhao, Feifei Yao, Lijie Wang, Ling Zheng, Yongjun Gao, Jun Ye, Feng Guo, Hui Zhao, and Rongbao Gao. A comparative study on the clinical features of covid-19 pneumonia to other pneumonias. *Clinical infectious diseases: an official publication of the Infectious Diseases Society of America*, 2020.

[3] Mizuho Nishio, Shunjiro Noguchi, Hidetoshi Matsuo, and Takamichi Murakami. Automatic classification between covid-19 pneumonia, non-covid-19 pneumonia, and the healthy on chest x-ray image: combination of data augmentation methods. *Scientific Reports*, 2020.

[4] Asif Iqbal Khan, Junaid Latief Shah, and Mohammad Mudasir Bhat. Coronet: A deep neural network for detection and diagnosis of covid-19 from chest x-ray images. *Computer methods and programs in biomedicine*, 2020.

[5] Asmaa Abbas, Mohammed M. Abdelsamea, and Mohamed Medhat Gaber. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network, 2020.

[6] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv 2003.11597*, 2020.

[7] Paolo Zaffino, Aldo Marzullo, Sara Moccia, Francesco Calimeri, Elena De Momi, Bernardo Bertucci, Pier Paolo Arcuri, and Maria Francesca Spadea. An open-source covid-19 ct dataset with automatic lung tissue classification for radiomics. *Bioengineering*, 2021.

[8] Xuehai He, Xingyi Yang, Shanghang Zhang, Jinyu Zhao, Yichen Zhang, Eric Xing, and Pengtao Xie. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. *medrxiv*, 2020.

[9] Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, and Bo Xu. A deep learning algorithm using ct images to screen for corona virus disease (covid-19). *medRxiv*, 2020.

[10] Ophir Gozes, Maayan Frid-Adar, Hayit Greenspan, Patrick D. Browning, Huangqi Zhang, Wenbin Ji, Adam Bernheim, and Eliot Siegel. Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection patient monitoring using deep learning ct image analysis, 2020.

[11] Lin Li, Lixin Qin, Zeguo Xu, Youbing Yin, Xin Wang, Bin Kong, Junjie Bai, Yi Lu, Zhenghan Fang, Qi Song, Kunlin Cao, Daliang Liu, Guisheng Wang, Qizhong Xu, Xisheng Fang, Shiqin Zhang, Juan Xia, and Jun Xia. Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct. *Radiology*, 2020.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[13] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.

[14] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. January 2015. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

[15] Binsheng Zhao, Leonard P. James, Chaya S. Moskowitz, Pingzhen Guo, Michelle S. Ginsberg, Robert A. Lefkowitz, Yilin Qin, Gregory J. Riely, Mark G. Kris, and Lawrence H. Schwartz. Evaluating variability in tumor measurements from same-day repeat ct scans of patients with non–small cell lung cancer. *Radiology*, 252(1):263–272, 2009. PMID: 19561260.

[16] Anthony Reeves, Yiting Xie, and Shuang Liu. Large-scale image region documentation for fully automated image biomarker algorithm development and evaluation. *Journal of Medical Imaging*, 4:024505, 06 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Appendix

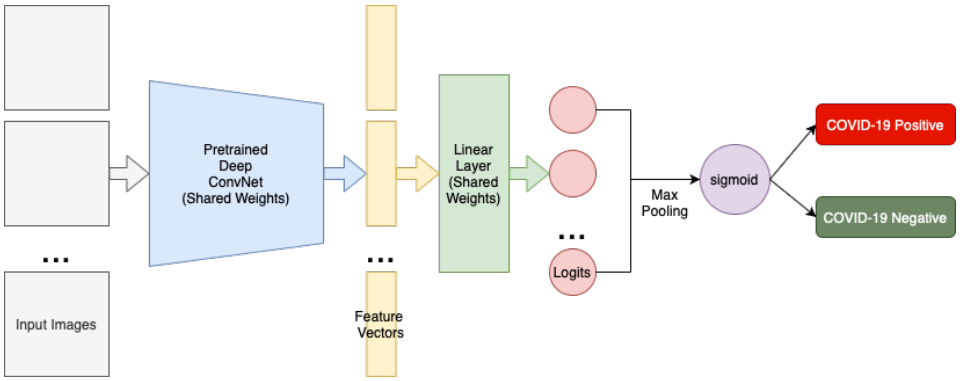

Figure 2: Max Pooling Model

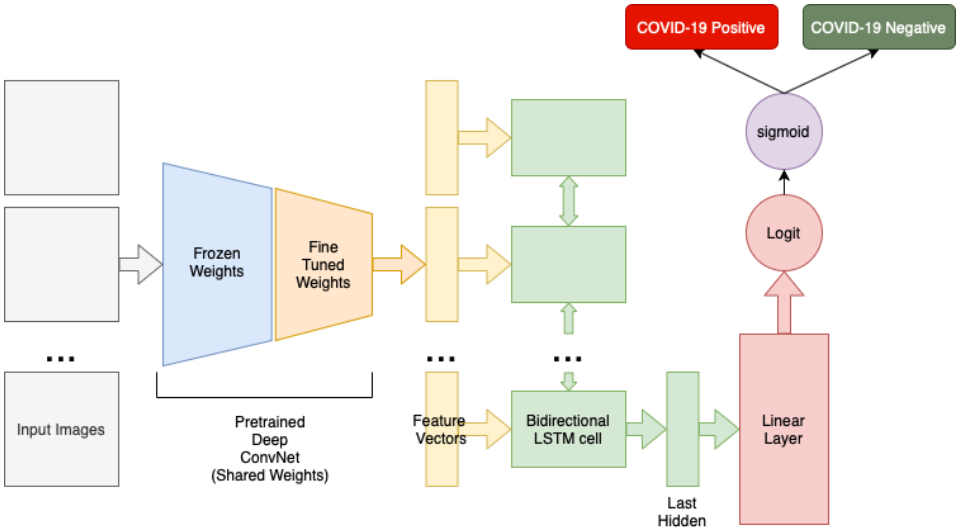

Figure 3: LSTM Model

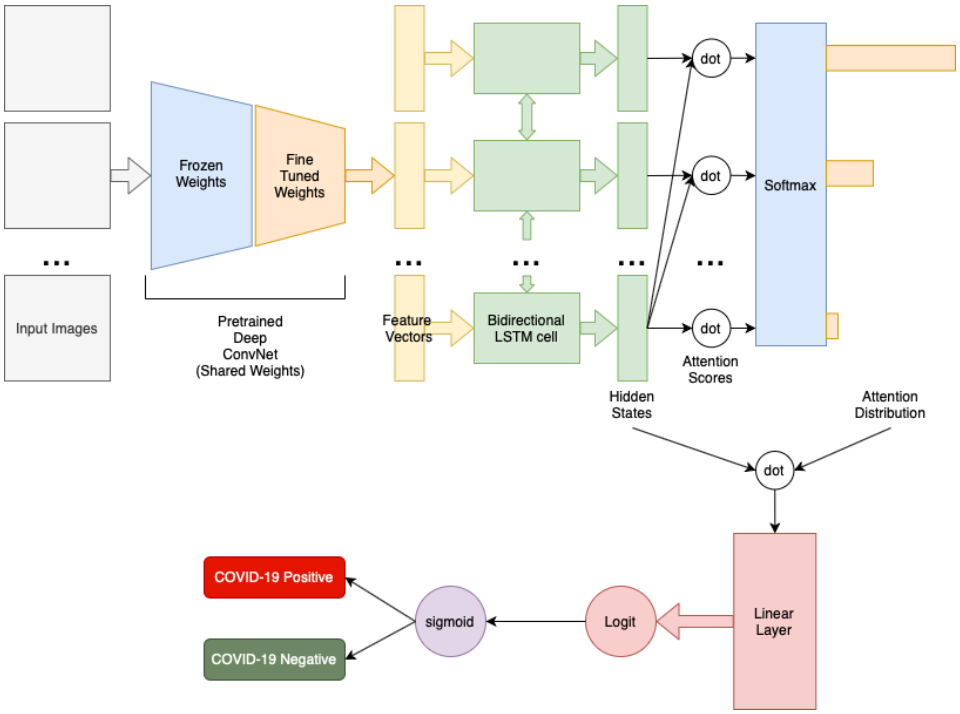

Figure 4: LSTM with Attention Model

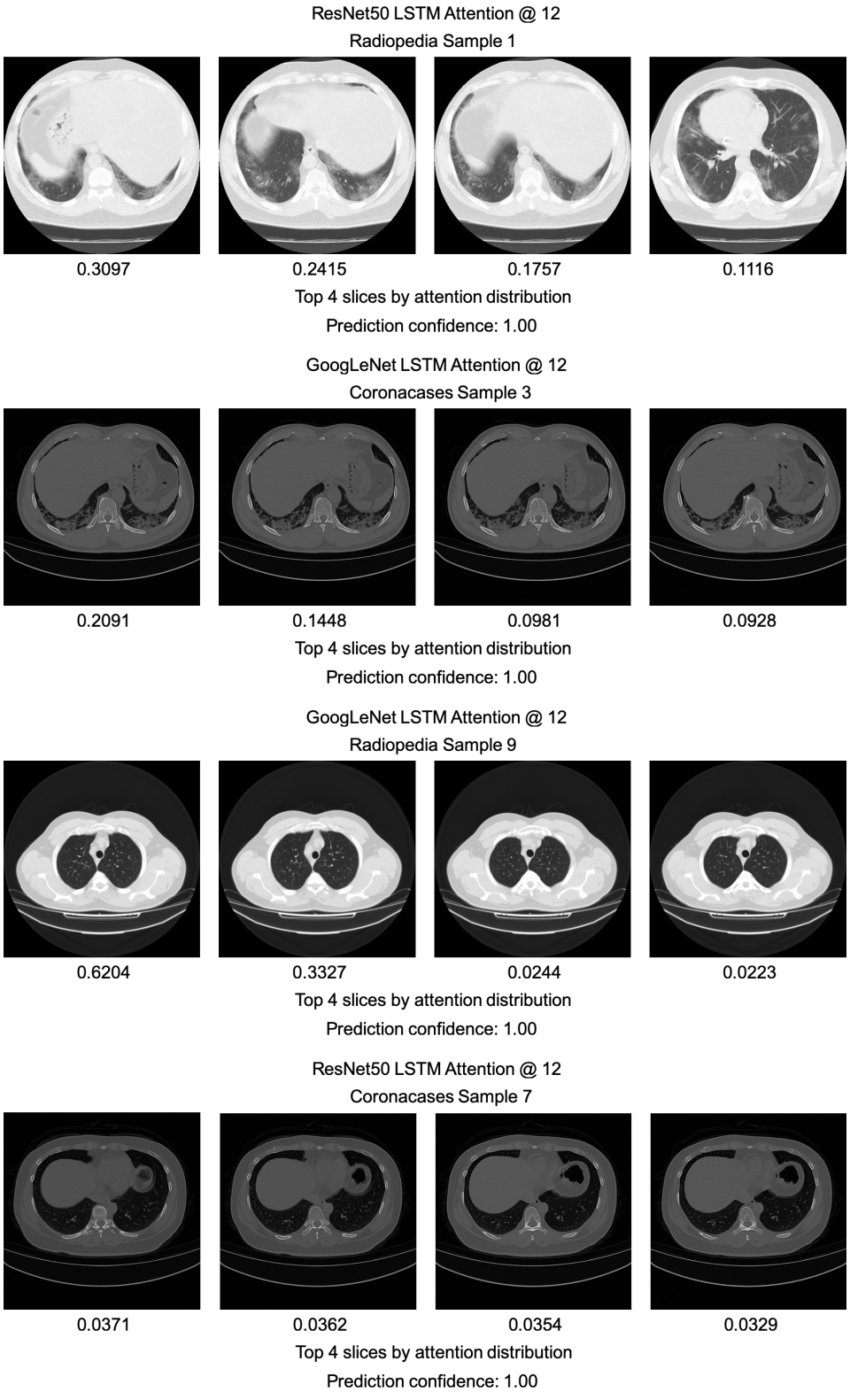

Figure 5: Sample Outputs