Draft Project

Team: Martina A. Nacheva, OMCS, Georgia Institute of Technology

Code: https://github.gatech.edu/mnacheva3/CSE6250
Presentation: https://github.gatech.edu/mnacheva3/CSE6250
(Note: To play presentation recording, go to 'Slide Show' > ''From Beginning')

Abstract
Chest diseases are a serious health issue, affecting the lives of many people. Their correct and early diagnosis is essential for prognosis. X-ray imaging are among the most frequently used methods for diagnosis, but their interpretability can be challenging. Need exists for an accurate and cost-effective method to support diagnosis based on x-rays. This study implements a range of CNN models to classify chest disease using the NIH Chest X-ray Dataset, containing 112,120 sample of 14 pathologies. As part of data exploration, class imbalances become an obvious defining feature of the dataset, for which we take several measures to correct. Several architectures are implemented, including ResNet50, VGG19 and a simplified 8-layer model termed LightNet. Results are evaluated based on AUC. Findings show that VGG and ResNet produce almost the same results, with VGG averaging 63% AUC and ResNet model averaging 62% AUC on the test set. LightNet underperforms, averaging 54% AUC.

Introduction
Chest diseases represent a serious issue in the field of healthcare and their timely diagnosis is very important. The Chest X-ray is among the most economical and frequently used medical imaging examinations as a method for diagnosis. However, the correct interpretation of X-rays and subsequent diagnosis can be challenging to achieve. Not only is diagnosis subject to the judgement-error of radiologists, but also in underdeveloped countries, where qualified radiologists are in short-supply, the X-rays may remain unevaluated. Therefore, these limitations give rise to the need for accurate and accessible methods of x-ray diagnosis.

In this context, an automated algorithm designed for chest disease diagnosis presents one possibility for a cost-efficient and effective method to make widespread chest-disease screening viable. This project proposes to demonstrate the feasibility of diagnosis through classification of pathologies captured in X-rays using deep-learning approaches, namely convolutional neural networks (CNNs). During the last number of years, CNNs have demonstrated a significantly superior performance across numerous image analysis tasks in comparison to more classical machine learning algorithms, such as, for example, support vector machines [1]. CNNs have been applied to a myriad of medical image analysis and have demonstrated state-of-the art performance on, for example, breast cancer classification [2] to organ [3] and tumor segmentation [4] [5]. Unsurprisingly, the use of deep learning on the diagnosis of chest X-Rays has also caught interest [6] [7], on account of its cost-effectiveness and increasing access to quality data.

Literature Survey
Over the past decades, computer-aided diagnosis (CAD) systems have been developed to extract useful information from X-rays to help doctors in having a quantitative insight about an X-ray. However, these CAD systems could not have achieved a significance level to make decisions on the type of conditions of diseases in an X-ray [8]. Deep learning, and more concretely, convolutional neural networks (CNNs) have become the state-of-the-art method to image classification. As Litjenset al [9] summarize, "Out of the 47 papers published on exam classification in 2015, 2016, and 2017, 36 are using CNNs, 5 are based on AEs and 6 on RBMs. The application areas of these methods are very diverse, ranging from brain MRI to retinal imaging and digital pathology to lung computed tomography (CT)."

For example, Wang et al [10] apply a CNN for the purposes of detecting lung nodules. Their analysis is conducted on a database acquired by the Japanese Society of Radiological Technology, which contains 154 cases of confirmed lung nodes. Wang et al. pre-process the data by conducting rib suppression methods (based on principle component analysis) in order to reduce the visibility of the ribs and improve the detectability of lung nodules. They also implore lung segmentation based on an active shape model. As a feature extractor, the authors use AlexNet. Their results are

found to be promising in terms of sensitivity and specificity (69.3% and 96.2%), outperforming hand-crafted features from structured data.

Similarly, Shin et al [11] apply an approach consisting of both CNNs and RNNs on a dataset of almost 3, 955 patients to classify 17 unique disease annotations. They employ a pre-trained RNN to generate the context of the annotations and recurrently use the output as an attribute in the CNN. Thus, the study uses the dataset's image annotations to mine disease names to train CNNs, and then RNN are trained to describe the contexts of a detected disease. The results are evaluated based on BLEU scores and show significant improvement of image annotation. Schlegl et al [12] also demonstrate that incorporation of semantic information from patient medical records can improve the accuracy of medical image classification. The authors generate a semantic representation of clinical records and show an increase of classification accuracy for intraretinal cystoid fluid, subretinal fluid and normal retinal tissue.

Some studies on medical image classification report results derived by CNN, which are comparable to human experts. Becker et al [13] claim that deep learning achieves human-level accuracy for breast cancer detection using mammogram data. The study trains a CNN on a dataset comprising of mammograms for 3228 patients. They report test results of area under the curve of 0.79, which is rivals that of radiologists, even in a screening cohort with low breast cancer prevalence. Similarly, Kermany et al [14] propose a transfer learning system to classify 108,309 optical images coherence tomography. The results report that weighted average error is equal to the average performance of 6 human experts. Of course, this indicates that scope remains for improving the CNN algorithms, so as to exceed human – level performance

Data

The proposed dataset on which to conduct the analysis is the NIH Chest X-ray Dataset. The dataset contains a total of 112,120 frontal-view X-ray images of a total of 30,805 unique patients. The images are labeled to provide information on a total of 14 pathologies, which include Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural thickening, Cardiomegaly, Nodule, Mass and Hernia. Hence, 15 classes in total are found as target labels, including 'No Finding' class.

The data exhibits a clear class imbalance between the 15 classes. As demonstrated in Table 2 in the appendix (A1), the data is strongly skewed in favour of the 'No Finding' class, which accounts for approximately 43% of samples. Among the pathologies, Infiltration is most frequently observed, found in 14% of samples, followed by Effusion at 9%. Importantly, the dataset contains very few sample images on some classes, such as Pneumonia, Fibrosis, and Hernia, which can be expected to challenge training, given the low number of samples, and to pose overfitting risks for these classes in particular. As detailed in the 'Approach' section further below, measures are taken to address the class imbalance, namely through imposing class weights in the loss function and applying data augmentation.

Further, the data is multi-labeled, as a sample image can exhibit several pathologies. As demonstrated in Figure 1 in the appendix (A2), of the sample images forming the dataset, 91,324 (81 %) images exhibit only one class (i.e. either 'No Finding' or only 1 pathology), whereas the remaining 19% exhibit two or more pathologies.

With regards to the non-image features, the data shows a slight imbalance between gender in favour of males, accounting for 56% of the sample, as shown in Figure 2 in the appendix (A3). With regards to age, as Figure 3 in the appendix (A4) shows, the distribution ranges from 0 years to 95 years, with the median reported at 53 years. The distribution appears only very slightly skewed to the left, but overall it resembles a distribution close to the normal. As Figure 4 in the appendix (A5) shows below, the view position of the images taken are somewhat skewed in favor of PA (Posterior – anterior) as compared to AP (Anterior – posterior).

The data is split 70-10-20 between the training, validation and test sets, respectively. The training and validation datasets are split from the testing set such that samples from a given patient do not cross over. The reasoning behind this is that different images of patients, who have had multiple screenings, would have a high degree of correlation, which could positively bias performance in the test set through memorization of the patient's idiosyncrasy.

Approach

This section overviews the methodological approach adopted. Whilst the data exploration phase is conducted in Pyspark, the model deployment is conducted in Python, primarily utilizing the Keras package. Models were trained on Google Colab in order to utilize GPU computational power. Whilst GPU access significantly aided training, resources were nevertheless limited, which placed limits to the optimization possible.

Preprocessing

As part of the pre-processing stage, the data undergoes several manipulation steps. Firstly, given the multi-label nature of the problem task, the target classes are one-hot encoded to produce target vectors, which are of shape (15,) for any given sample point. Further, the input feature data are normalized in order to ensure similar scales of various input features in order to mitigate the problem of vanishing gradients and slow learning. The data is read in size (224, 224).

The data is run on three separate CNN models. ResNet50, VGG19, and a simplified 8-layer model, which shall henceforth be referred to as LightNet.

ResNet-50 Architecture

The Residual neural networks (ResNet) was developed by He et al.16 and showed a 3.6% error rate on the ImageNet SVRC database. The authors built a 152 layer deep CNN architecture. It is deeper than earlier networks such as AlexNet, which has eight layers, and VggNet with up to 19 layers. The ResNet architecture introduced connections (residual connections) which allows to avoid information loss during training. Skip connection technique enables to train very deep networks and can boost the performance of the model. For the purpose of applying the model to our dataset, the model is implemented with transfer learning with weights trained on the ImageNet dataset. Hence, the top classification layer is removed and the model is extended by appending layers fully connected layers, which are trained on the NIH dataset. The final layer comprises a sigmoid activation function in order to reflect the multi-label nature of the problem task, such that more than one classes can be identified in a sample image. Two different structures are tested, one comprising of 3 dense layers for classification, and another deeper structure, comprising of 7 dense layers for classification learning, the latter of which was found to perform better and which comprises 127,034,251 parameters, of which 103,465,389 are trainable. For a model summary refer to A6.

VGG-19 Architecture

VGG-19 was developed by Simonyan and Zisserman[18] in 2015 and it holds a 7.3% error rate on the ImageNet SVRC database. is a convolutional neural network that is 19 layers deep. The feature extractor part of the model comprises of 16 convolution layers, followed by relu and max pooling layers, as well as dropout of 5% as a regularization technique. A further 3 fully connected layers are added, leading to classification. Similarly to ResNet, this model is applied to the Chest Disease problem task by implementing the pretrained weights from ImageNet, removing the top layers (three fully connected layers). Appended are 3 fully connected layers, two of which are followed by relu activation and dropout layers, and the last of which applies a sigmoid activation function for classification of the multi-label dataset, and which are trained on the NIH dataset. The total number of parameters are 21,281,209, of which trainable are 1,256,635. For a model summary refer to A7

LightNet Architecture

LighNet is a simplified CNN model created specifically in the context of classification of our chest disease images. For the purpose of this study, it serves as a comparable baseline of a simple model to which we can compare the performance of the deeper VGG and ResNet models. LightNet comprises of 6 convolutional layers, which sandwich relu activation, maxpooling for down-sampling, and dropout layers for regularization. An additional 2 fully connected are appended at the end, leasing to classification, with a final sigmoid activation function. The total number of parameters are 19,051,695. For a model summary refer to A8.

Loss Function and Optimizer

The loss function applied to all three models is the binary cross-entropy loss function. The crossentropy function can be defined as follows where t_i and s_i are the groundtruth and the CNN score for each class i in C:

$$CE = -\sum_{i=1}^{C} t_i \log (s_i)$$

Hence, in a binary classification problem, where $C = 2$, the loss function can be presented as follows:

$$L = -\sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1)\log(1 - s_1)$$

The preliminary results presented are based on the Adam optimizer. For example, Yaqub et al [19] employ the Adam optimizer in segmentation of brain images using CNNs. Similarly, Baltruschat et al [20] also deploy the Adam optimizer in training the ResNet on the NIH Database. Similarly, Grewal et al [21] apply it in the deep learning of brain hemorrhaging in CT scans, while Oakden [21] deploy it in the classification of pneumonia. Nonetheless, this remains subject to possible change as part of further optimization. The learning rate is currently initialized at 0.01 and is decreased by a factor of 100 subject to no improvement in the validation AUC for 3 consecutive epochs. This, of course, is a hyperparameter, which remains to be further tuned.

Class Weights, Undersampling and Data Augmentation

The data is read in batches through means of generators in order to manage memory requirements of loading the dataset. At this stage, data augmentation is also applied. Augmentation generates new images through rotating, zoom range and shear range. Special consideration has been given to the types of augmentations applied, as not all methods would be appropriate to medical data. For example, horizontal flipping would reverse the position of the heart (on the left-hand of the body), and this could hinder the learning process in the context of pathogens relating to the heart, such as Cardiomegaly.

The rationale for implementing data augmentation is to increase the size of the dataset and introduce variability, which is particularly relevant for underrepresented classes, for which the number of samples could be as few as 227 (as in the case of Hernia).

In order to address the imbalance between the 15 classes, two measures are taken. First, the majority class 'No Findings' is undersampled in the training dataset to reduce the number of samples to 10,500, which is proportionate to the other major classes such as Infiltration and Effusion. As an approach, undersampling the majority class was preferred to oversampling the remaining for several reasons, notably more favourable implications for memory requirements and training times.

Secondly, weights are applied for each class within the loss function. These weights serve as a penalty, increasing the loss, for incorrect classifications. Thus, the weights impede the model from turning 'greedy', whereby it focuses entirely on predicting the majority classes, which in this case is 'No Finding'. The weights are calculated as follows, where the weight for a given class if a function of the total number of samples N, the number of classes C and the number of occurrences per class n_c:

$$w_i = \frac{N}{C * n_c}$$

Hyperarameter Tuning

For the ResNet model, parameters which were optimized through a grid search include optimizer, learning rate, batches and architecture structure. Two optimizers were tested, including Adam and SGD. Learning rates tested include 0.001, 0.0001, and 0.00001. Batches tested include 16, 32, 64, 128, 256 and 512. All permutations between these parameters were tested. Additionally, apart the structure of the classification portion on the model was tested, with a simpler structure comprising of 3 layers, and a deeper structure comprising of 7 layers. In total, 45 models were tested. For loss and AUC (validation) results, please refer to A9 and A10.

For the VGG model, parameters tuned include the learning rate for the Adam optimizer, and the number of batches. Learning rates tested include 0.01, 0.001, 0.0001, 0.00001. Batches tested include 16, 32 and 64. All permutations were tested. In total, 12 different models were tested. For validation loss and AUC results please refer to A10 and A11.

For LightNet, parameters hypertunined include the learning rate for the Adam optimizer. Rates tested include 0.001, 0.0001 and 0.00001. Validation loss and AUC metrics are available in A12 and A13.

Evaluation Metrics

The results will be evaluated on AUC, as it relates to both sensitivity and specificity and is therefore not biased towards the majority class. This approach is in line with others, such as Baltruschat et al [20], with Shen et al [22] as part of their work on classification of skin cancer. Training utilizes this metric through several callbacks, which adjust the learning rate and number of epochs depending on changes in the validation loss with the overarching purpose of optimizing model performance on this metric.

Results

Presented here are final results for the VGG, ResNet and LightNet models following hyperparameter optimization. The model has been trained on the full sample of the training data for 50 epochs. The results are presented in Table 1 below. Considering 50% is equivalent of no knowledge, the preliminary results show that all three models have learnt to differentiate between the classes. VGG and ResNet models have outperformed LightNet, which is unsurprising considering its simplified structure. On average, VGG shows 63% AUC on the test set, a similar performance to ResNet which shows 62% AUC. The models' performance is heterogeneous across the classes, with certain pathologies such as Cardiomegaly, Effusion and Hernia showing AUC exceeding 70%, whilst other classes such as Pneumonia and Nodule ranking just under 60% AUC.

For comparison, Table 4 also provides results from studies, which also aim to classify chest pathologies using the NIH dataset. Differences in the models applies exist. Baltruschat et al [22] implement a ResNet38 model, trained on the full sample set. They further integrate the non-image features of the dataset, such as age and gender. Gunder et al[24], on the other hand, implement a DenseNet model, and also used an additional dataset – PLCO dataset25 – with 185,000 images. Both of these measures could be considered as future steps to continue to improve the results.

Training and Validation losses and AUC metrics for the three models are presented in the figures available in the appending from A15 to A20.

Table 1. AUC Test set results, VGG model

	VGG	*ResNet*	*LightNet*	*Gundel et al*	*Baltruschat et al*
Atelectasis	0.61	0.62	0.525	76%	72%
Cardiomegaly	0.74	0.71	0.581	88%	73%
Consolidation	0.60	0.63	0.533	75%	74%
Edema	0.73	0.72	0.601	83%	84%
Effusion	0.70	0.70	0.542	82%	79%
Emphysema	0.65	0.72	0.534	89%	78%
Fibrosis	0.65	0.66	0.515	82%	72%
Hernia	0.75	0.75	0.582	89%	79%
Infiltration	0.63	0.63	0.543	71%	66%
Mass	0.60	0.63	0.537	82%	67%
No Finding	0.62	0.64	0.541	80%	72%
Nodule	0.57	0.58	0.511	76%	65%
Pleural Thickening	0.61	0.60	0.52	76%	69%
Pneumonia	0.55	0.57	0.56	73%	66%
Pneumothorax	0.70	0.69	0.578	84%	77%
Total	0.63	0.62	0.54	80%	73%

Discussion

The results for the VGG, ResNet and LightNet models indicate that all three have been able to learn during training, as demonstrated by the AUC score exceeding the 50% threshold. VGG and ResNet clearly outperform LightNet, as to be expected, owing to their deeper structure and implied ability to detect more complex and refined patterns in the images. VGG averages an AUC score of 63%, ranging from 75%(Hernia) to 55% (Pneumonia). ResNet averages an AUC score of 62%, reaching ranging from 75% (Hernia) to 57% (Pneumonia).

Significant efforts were implemented to optimize model performance through tuning hyperparameters. For the ResNet model, 46 different model variants were run. For VGG, 12 model variants were run. Testing shows that tuning the learning rate led to significant improvements in all cases, while tuning the batch size showed more conservative impacts. Further, the Adam optimizer was found to perform best.

Performance has scope to be further improved as demonstrated by the alternative studies by Gundel et al [24] and Baltruschat et al [22]. Baltruschat et al also include non-image features in the classification, whilst Gundel et al utilize an additional dataset, which increases the number of training images for underrepresented classes. Undoubtedly, these approaches present potential for further gains in training and will be considered in the future as possible avenues for further work.

Conclusion

This study presents a set of three CNN models for the classification of chest pathogens. Models are based on ResNet50, VGG19 and a simplified tailor-made LightNet model. The former two models are run using transfer learning. Among the key characteristics of the dataset is a strong class imbalance across the 15 target classes. Careful measures were implemented to mitigate the bias implications of this, including undersampling, class weights in the loss function and data augmentation. All models were optimized for hyperparameters, with tested permutations as many as 46 for ResNet. Results report the ResNet and VGG models having learnt from training, with test AUC metrics averaging 63% and 62% respectively. The LightNet model by comparison performs less well considering its simplicity. Further avenues of work could consider introducing non-image features to further improve results, and incorporating images of the underrepresented classes through other datasets to aid training.

References

1. Yadav, S.S., Jadhav, S.M. Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data; 2019, 6, 113
2. Rouhi, R., Jafari, M., Kasaei, S. & Keshavarzian, P. Benign and malignant breast tumors classification based on region growing and CNN segmentation. Expert Systems with Applications;2015, 42, 990–1002
3. Roth, H. R., Lu, L., Farag, A., Shin, H. & Liu, J. DeepOrgan: Multi-level Deep Convolutional Networks for Automated Pancreas Segmentation.2015, 9349,1–12
4. Milletari, F., Navab, N. & Ahmadi, S.-A. V-net: Fully convolutional neural networks for volumetric medical
5. image segmentation, In 3D Vision (3DV). Fourth International Conference; 2016, 565–571
6. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention; 2015, 234–241
7. Shin, H. et al. Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016, pp. 2497–2506.
8. Li, Z. et al. Thoracic Disease Identification and Localization with Limited Supervision. IEEE Conference Computer Vision Pattern Recognition; 2018, 8290–8299
    A. A. El-Solh, C.-B. Hsiao, S. Goodnough, J. Serghani, and B. J. B. Grant, "Predicting active pulmonary tuberculosis using an artificial neural network," Chest; 1999, 116, 968–973
9. G. Litjens, T. Kooi, E. B. Bejnordi et al., "A survey on deep learning in medical image analysis," Medical Image Analysis; 2017, 42, 60–88.
10. Wang C., A. Elazab, Wu J. and Hu Q. Comput. Med. Imaging Graph; 2016; 57, 10-18
11. Shin, H.-C., Roberts, K., Lu, L., Demner-Fushman, D., Yao, J., Summers, R. M.. Learning to read chest x-rays: recurrent neural cascade model for automated image annotation. IEEE Conference on Computer Vision and Pattern Recognition; 2016.
12. T. Schlegl, S.M. Waldstein, W.-D. Vogl, U. Schmidt-Erfurthand Langs G. Proceedings of the Information Processing in Medical Imaging, Lecture Notes in Computer Science; 2015, 9123, pp. 437-448,
13. Becker AS, Marcon M, Ghafoor S, Wurnig MC, Frauenfelder T, Boss A. Deep Learning in Mammography: Diagnostic Accuracy of a Multipurpose Image Analysis Software in the Detection of Breast Cancer. Invest Radiol; 2017, 52(7), 434-440
14. Kermany DS, Goldbaum M, Cai W, Valentim CC, Liang H, Baxter SL, McKeown A, Yang G, Wu X, Yan
15. F, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell. 2018; 172(5):1122–31.
16. Summers, Ronald (NIH/CC/DRD) (2017) NIH Chest X-Ray Dataset. Location: National Institutes of Health - Clinical Center
17. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition.
18. 2015.
19. Huang, G., Liu, Z., Laurens, V.D.M., Weinberger, K.. Densely Connected Convolution Networks. 2016
20. Simonyan,K., and Zisserman, A.. Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, 1409. 2015
21. Yaqub M, Jinchao F, Zia MS, et al. State-of-the-Art CNN Optimizer for Brain Tumor Segmentation in Magnetic Resonance Images. Brain Sci. 2020;10(7):427. Published 2020 Jul 3. doi:10.3390/brainsci10070427
22. Baltruschat, I., Nickisch, H., Grass, M. Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification. JOUR Scientific Reports. doi : 10.1038/s41598-019-42294-8
23. Grewal M., Srivastava M.M., Kumar P., Varadarajan S. Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans; Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging; Washington, DC, USA. 4–7 April 2018; pp. 281–284.
24. Oakden-Rayner, L. Exploring the ChestXray14 dataset: Problems. Available at, https://lukeoakdenrayner.wordpress.com/2017/12/18/ (2017).
25. Shen D., Wu G., Suk H.I. Deep learning in medical image analysis. Annu. Rev. Biomed. Eng. 2017;19:221–248. doi: 10.1146/annurev-bioeng-071516-044442.
26. Guendel, S. et al. Learning to recognize abnormalities in chest x-rays with location-aware dense networks. arXiv preprint arXiv:1803.04565 (2018).

Appendix

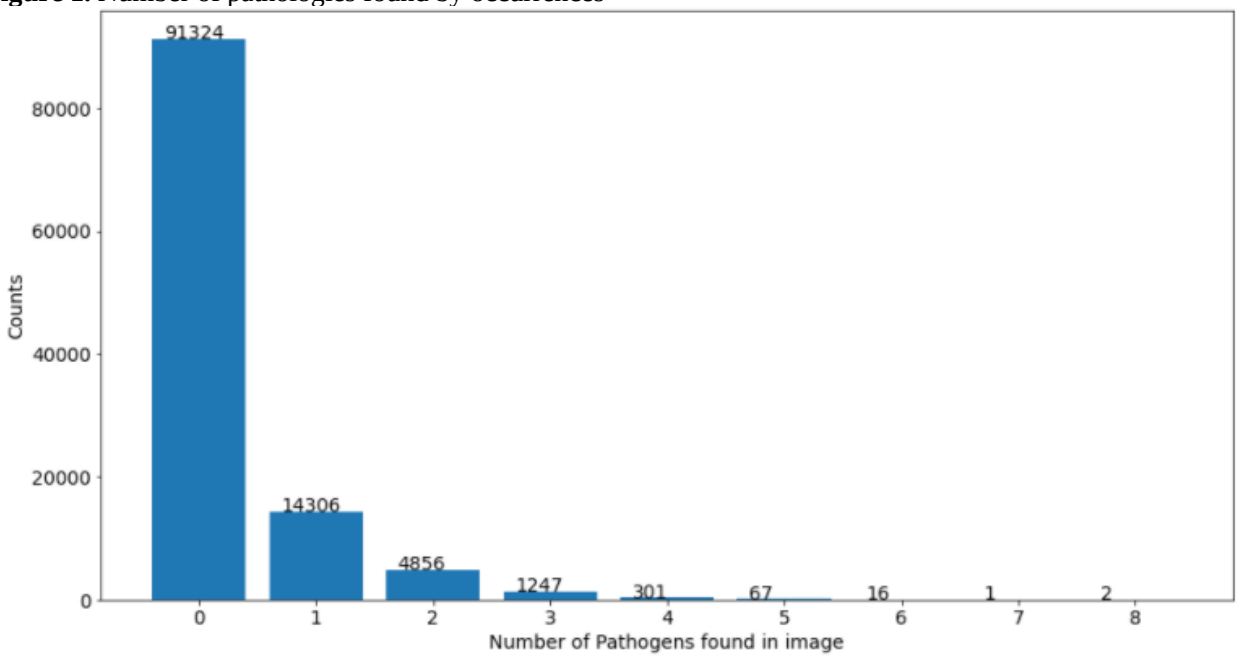
A1

Table 2. Distribution of chest pathologies across sample images

Pathology	*Counts*	*%*
Atelectasis	11,559	8%
Cardiomegaly	2,776	2%
Consolidation	4,667	3%
Edema	2,303	2%
Effusion	13,317	9%
Emphysema	2,516	2%
Fibrosis	1,686	1%
Hernia	227	0%
Infiltration	19,894	14%
Mass	5,782	4%
No Finding	60,361	43%
Nodule	6,331	4%
Pleural Thickening	3,385	2%
Pneumonia	1,431	1%
Pneumothorax	5,302	4%
	141,537	100%

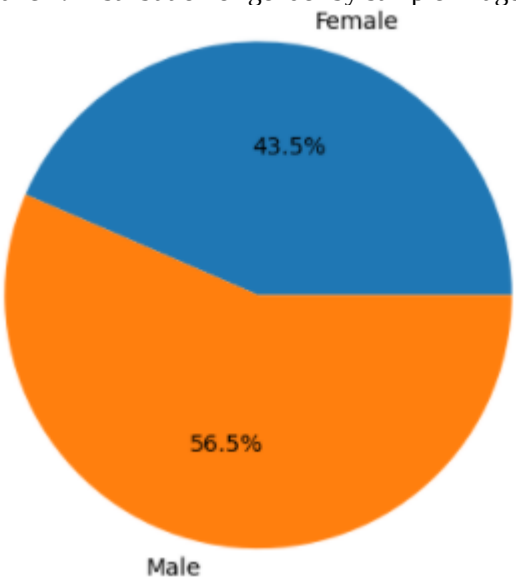
A2

Figure 1. Number of pathologies found by occurrences



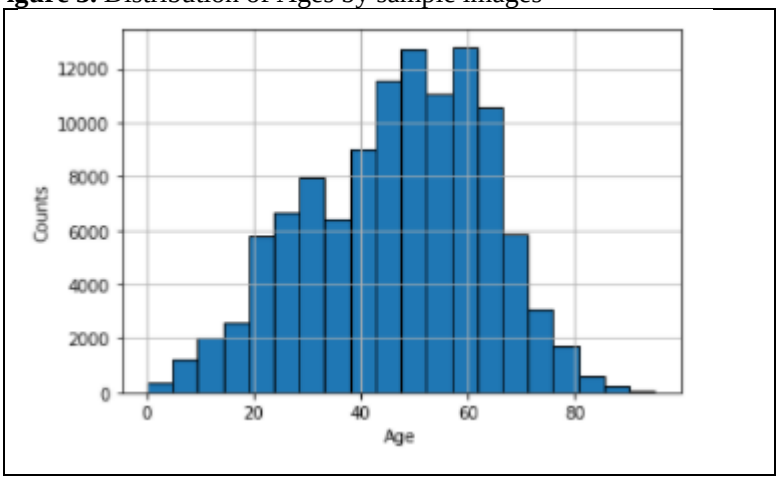
A3

Figure 2. Distribution of gender by sample images



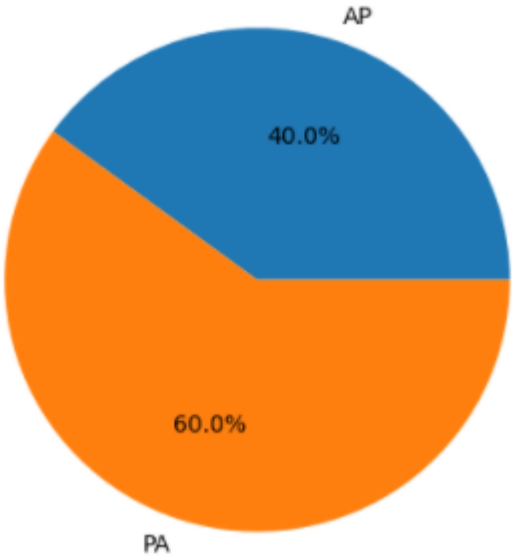
A4

Figure 3. Distribution of Ages by sample images



A5

Figure 4. Distribution of image view angle by sample images



A6

Table 2. Model Summary ResNet model with 'Deep Classification'

Layer (type)	Output Shape	Param #
resnet50v2 (Functional)	(None, 7, 7, 2048)	23564800
flatten_1 (Flatten)	(None, 100352)	0
dense_3 (Dense)	(None, 1024)	102761472
batch_normalization_3 (Batch	(None, 1024)	4096
activation_3 (Activation)	(None, 1024)	0
dropout_2 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 512)	524800
batch_normalization_4 (Batch	(None, 512)	2048
activation_4 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 256)	131328
batch_normalization_5 (Batch	(None, 256)	1024
activation_5 (Activation)	(None, 256)	0
dropout_4 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 128)	32896
batch_normalization_6 (Batch	(None, 128)	512

activation_6 (Activation)	(None, 128)	0
dropout_5 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 64)	8256
batch_normalization_7 (Batch	(None, 64)	256
activation_7 (Activation)	(None, 64)	0
dropout_6 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 32)	2080
batch_normalization_8 (Batch	(None, 32)	128
activation_8 (Activation)	(None, 32)	0
dropout_7 (Dropout)	(None, 32)	0
dense_9 (Dense)	(None, 15)	495
batch_normalization_9 (Batch	(None, 15)	60
activation_9 (Activation)	(None, 15)	0

Total params: 127,034,251
Trainable params: 103,465,389
Non-trainable params: 23,568,862

A7

Table 3. Model summary VGG model

Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 7, 7, 512)	20024384
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 50)	1254450
batch_normalization (BatchNo	(None, 50)	200
activation (Activation)	(None, 50)	0
dropout (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 30)	1530
batch_normalization_1 (Batch	(None, 30)	120
activation_1 (Activation)	(None, 30)	0
dropout_1 (Dropout)	(None, 30)	0
dense_2 (Dense)	(None, 15)	465

```
batch_normalization_2 (Batch (None, 15)                60
_________________________________________________________________
activation_2 (Activation)    (None, 15)                0
=================================================================
Total params: 21,281,209
Trainable params: 1,256,635
Non-trainable params: 20,024,574
_________________________________________________________________
```

A8

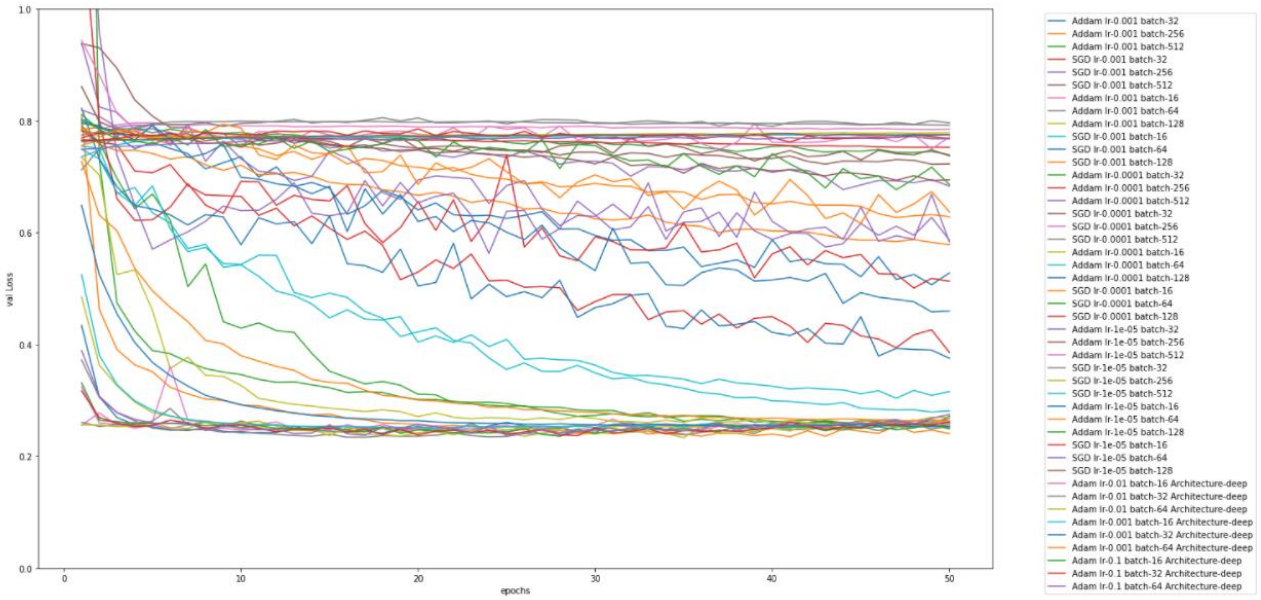
Table 4. LightNet Model Summary

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 32)      896
_________________________________________________________________
batch_normalization (BatchNo (None, 224, 224, 32)      128
_________________________________________________________________
activation (Activation)      (None, 224, 224, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 222, 222, 32)      9248
_________________________________________________________________
batch_normalization_1 (Batch (None, 222, 222, 32)      128
_________________________________________________________________
activation_1 (Activation)    (None, 222, 222, 32)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 32)      0
_________________________________________________________________
dropout (Dropout)            (None, 111, 111, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 111, 111, 64)      18496
_________________________________________________________________
batch_normalization_2 (Batch (None, 111, 111, 64)      256
_________________________________________________________________
activation_2 (Activation)    (None, 111, 111, 64)      0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 109, 109, 64)      36928
_________________________________________________________________
batch_normalization_3 (Batch (None, 109, 109, 64)      256
_________________________________________________________________
activation_3 (Activation)    (None, 109, 109, 64)      0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 52, 52, 64)        36928
_________________________________________________________________
batch_normalization_4 (Batch (None, 52, 52, 64)        256
_________________________________________________________________
activation_4 (Activation)    (None, 52, 52, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 64)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 24, 24, 64)        36928
_________________________________________________________________
batch_normalization_5 (Batch (None, 24, 24, 64)        256
_________________________________________________________________
flatten (Flatten)            (None, 36864)             0
_________________________________________________________________
```

```
dense (Dense)                (None, 512)               18874880
_________________________________________________________________
batch_normalization_6 (Batch (None, 512)               2048
_________________________________________________________________
activation_5 (Activation)    (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                32832
_________________________________________________________________
batch_normalization_7 (Batch (None, 64)                256
_________________________________________________________________
activation_6 (Activation)    (None, 15)                975
_________________________________________________________________
Total params: 19,051,695
Trainable params: 19,049,903
Non-trainable params: 1,792
_________________________________________________________________
```

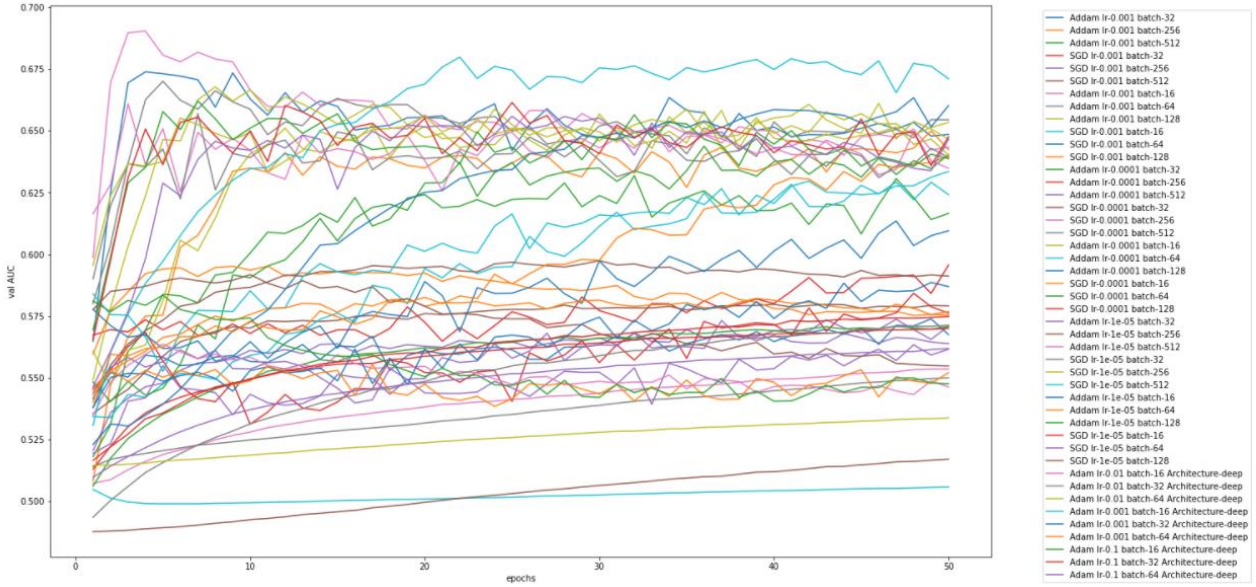
A9

Figure 5. ResNet validation loss results, hypertuning



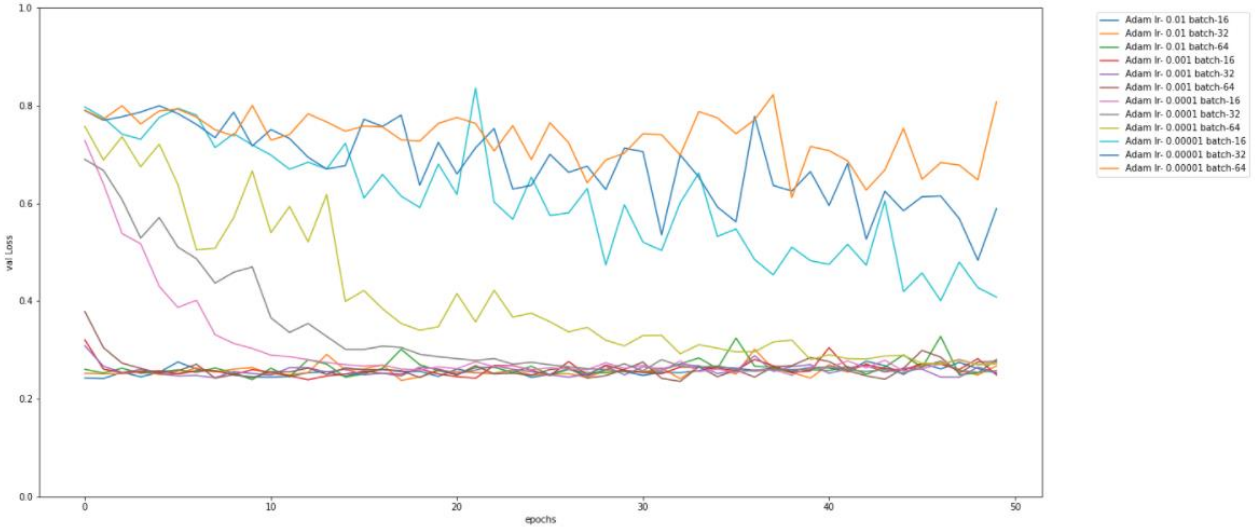
A10

Figure 6. ResNet validation AUC results, hypertuning



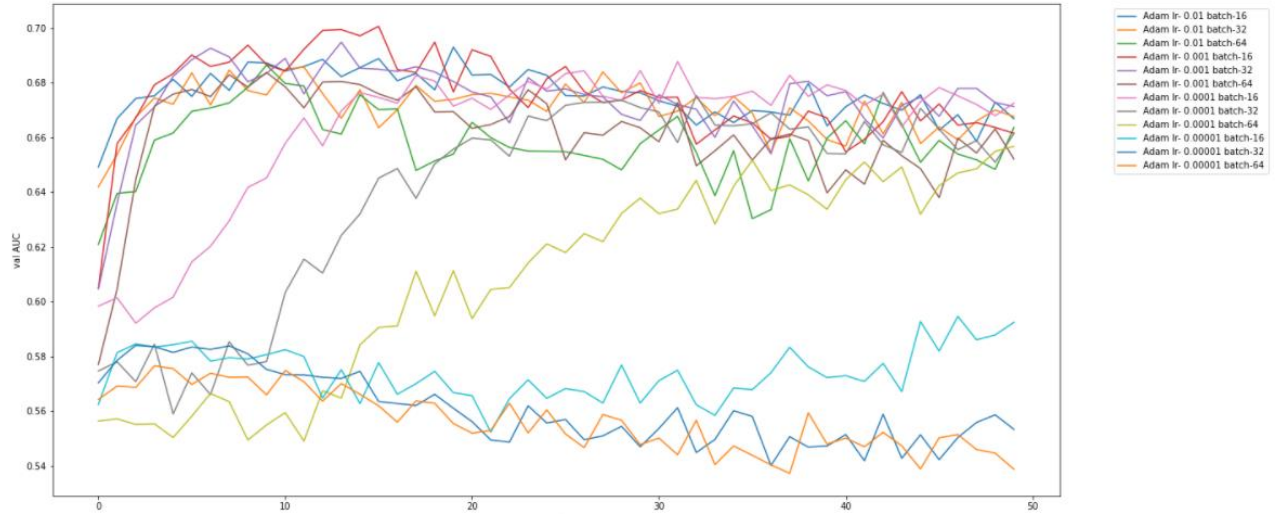
A11

Figure 7. VGG validation loss results, hypertuning



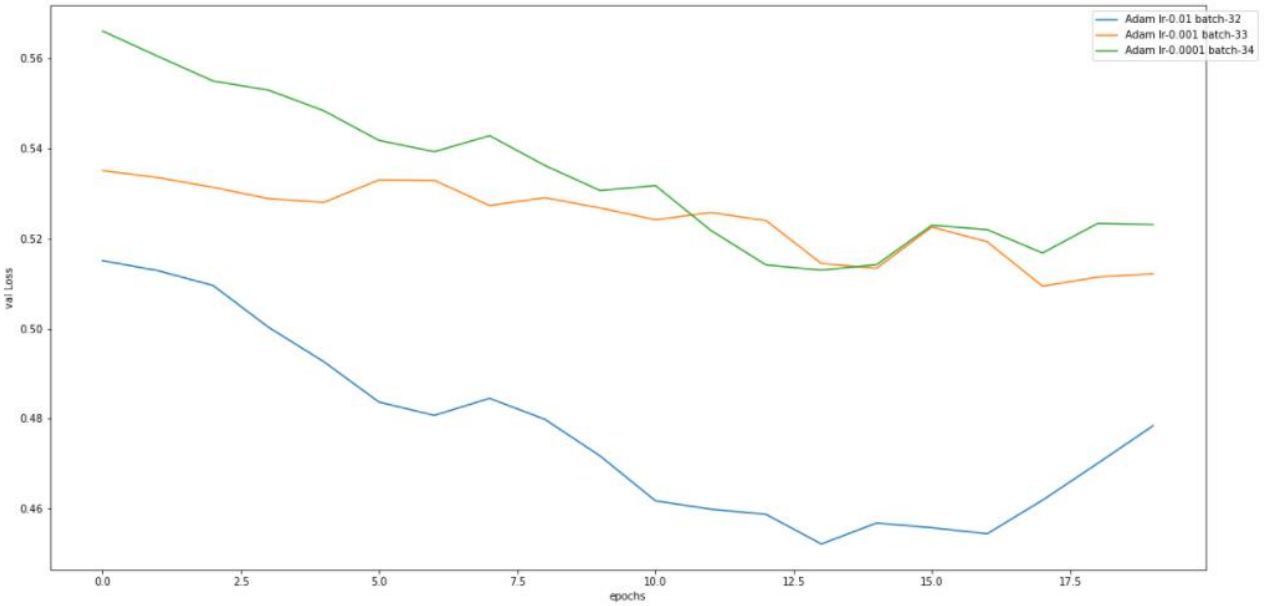
A12

Figure 8. VGG validation AUC results, hypertuning



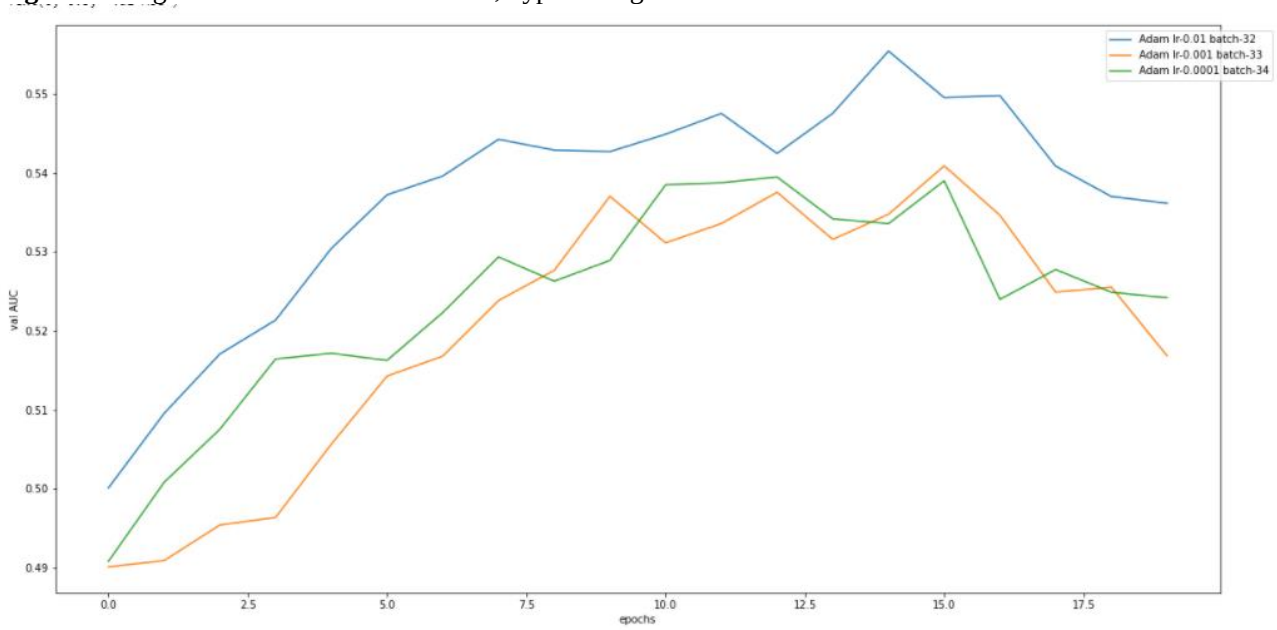
A13

Figure 9. LightNet validation loss results, hypertuning



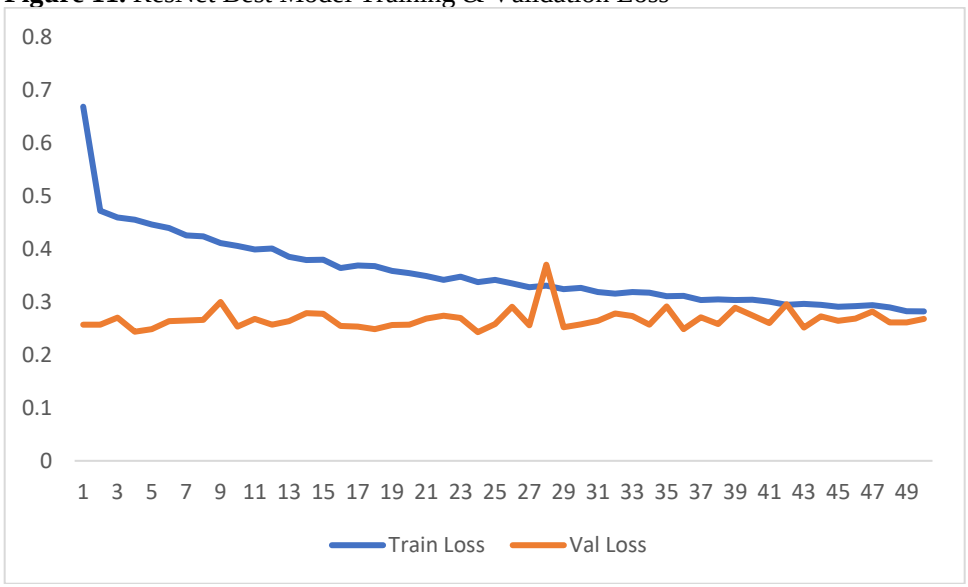
A14

Figure 10. LightNet validation AUC results, hypertuning



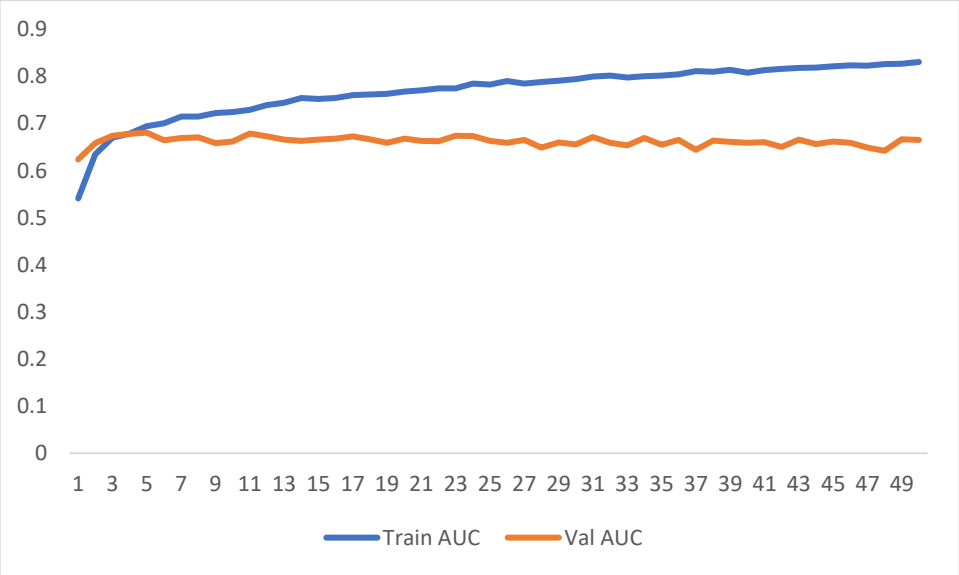
A15

Figure 11. ResNet Best Model Training & Validation Loss



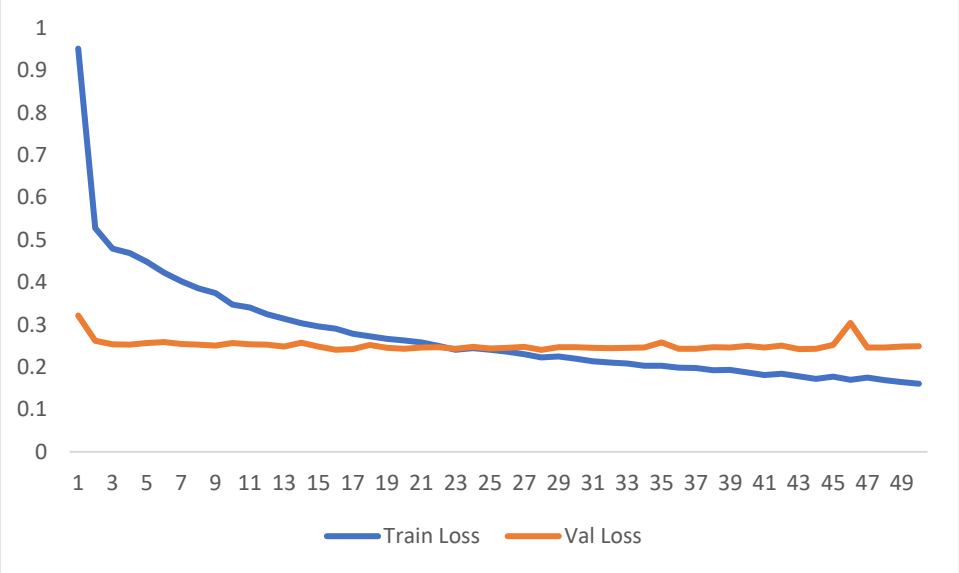
A15

Figure 11. ResNet Best Model Training & Validation AUC



A16

Figure 12. VGG Best Model Training & Validation Loss



A17

Figure 13. VGG Best Model Training & Validation AUC