ORIGINAL ARTICLE

Comparison of CNN Architectures for Diagnosing Pneumonia in Chest X-Ray Images

Balkaran Singh[1]

[1]Auckland University of Technology

Correspondence
Email:

Funding information

This paper presents, methodical comparisons between four CNN architectures and different learning approaches, for detecting pneumonia in X-Ray images. We evaluate 12 different models obtained by applying three different learning approaches on four different CNN architectures. The results show that transfer learning using fine-tuning performs quite well on all cnn architectures, showing little or no overfitting in most cases. For the overall top model, we find that ResNeXt-50 with fine tuning performs the best. Achieving a high sensitivity (recall) of 98.7%, 75.6% specificity and AUROC of 0.87.

1. Introduction

Pneumonia is the world's leading infectious cause of death for children under the age of 5. This is caused due to an immune response to infectious pathogens like bacteria, viruses and other microorganisms. This causes inflammation in the alveoli, a small hollow sac found in the lungs and limits the individual's oxygen intake (McLuckie, 2009). The diagnosis process for pneumonia usually requires examination of Chest-X rays from radiologists. Essentially, this is a classification task which requires precise detection of structural abnormalities in radiographs. Many countries around the world face severe workforce shortages in radiology. In the UK, 97% of radiology departments, were unable to meet their diagnostic reporting requirements in 2016 (RCR, 2016). In New Zealand, the Midcentral District Health Board has labelled, radiologist shortages as a nation-wide problem (MCDHB, 2019).

In deep learning, convolutional neural networks (CNNs) have shown to surpass human level performance for many image classification tasks (He et al., 2015). CNNs are also anticipated to help facilitate the workflow of professional radiologists to help achieve faster and more accurate diagnosis (Yasaka and Abe, 2018). Several studies on chest X-Ray classification can be found in literature thanks to the publicly available data sets like CheXpert and ChestX-ray14. Most notably researchers developed CheXNet, a 121-layer CNN which has shown accuracy which exceeds expert radiologists. CheXNet was trained on the ChestX-ray14 dataset and can detect between 14 classes of chest related diseases (Rajpurkar et al., 2017). Another study developed a three-branch CNN, which learns from disease specific

areas. This approach is said to avoid noise and compensate for the poorly aligned images in the data (Guan et al., 2018). A recent paper, experiments with different depths of ResNet architecture on the ChestX-ray14 dataset, comparing models trained from scratch and their fine-tuned counterparts. It concludes that ResNet38 gives state-of-the-art accuracy on classification of 5 different chest diseases (Baltruschat et al., 2019). More work includes, development of a simple CNN architecture optimized for speed while achieving high accuracy in tuberculosis classification. This work also presented unique visual maps, which highlight the area of disease(Pasa et al., 2019).

This paper compares the capabilities of four different CNN architectures and 3 different learning approaches for classification of x-ray images as normal or 'with pneumonia'. The candidate architectures considered for study are VGG16, DenseNet121, ResNeXt50 and InceptionV3. The motivation for using these sets of models, comes from high accuracy measures and success stories achieved in recent studies. The ChexNet architecture, which exceeds classification accuracy of even expert radiologists is a 121-layer DenseNet. ResNeXt50 is newer variant of the ResNet architecture, which was was shown to achieve state-of-the art results for classification of chest 5 diseases in X-Rays (Baltruschat et al., 2019). VGG16 was used by Islam et. el. to achieve sensitivity of 96% for abnormality detection in chest X-Rays (Mohammad Tariqul Islam, 2017). InceptionV3 is a popular CNN architecture, which was also the first runner up for ILSVRC image classification challenge on the large ImageNet database. InceptionV3 has also been applied in medical imaging by many studies, most notably (Gulshan et al., 2016) uses it for detection of diabetic retinopathy, achieving over 97% sensitivity and 93.4% specificity. We train these architectures using three different learning approaches. Specifically, we train the models from scratch with random weight initialization and use two different transfer learning methods on pre-trained weights from ImageNet. For transfer learning we use the feature extraction and the fine tuning approach. The 12 models are evaluated using a test set, which is independent from the training and the validation set. For comparisons, we use area under the curve from ROC and PRC, additionally we also use sensitivity and specificity measures to compare model performances in terms of real applications. Overall, this study aims to 1) Identify which learning approach performs the best across all architectures. 2) Identify which of the 12 models is the best overall in terms of real life application. As per our knowledge, this is the first study in pneumonia detection, which has employed the ResNeXt50 model and has used these learning approaches.

This paper is organised as follows, in section 2 we give descriptions about data preprocessing and how we dealt with class imbalance issues. Section 3 gives an overview of CNN architectures and the different learning approaches used in this paper. We also give brief descriptions about how the experiments were set up and the metrics used for evaluation. Section 4 contains discussion and the results obtained from the experiments. Finally, section 5 gives the conclusion and suggestions for future work.

2. Data

For this study, we have considered the publicly available Chest X-Ray images data set, containing labelled cases of pneumonia and no pneumonia (normal) (Daniel Kermany, 2018). The training data contains 5,232 X-ray images collected from children, depicting 3883 cases of pneumonia and 1349 normal cases. The pneumonia cases contained, instances of the disease contracted from bacterial and viral infection. The test set contains data from 624 patients, including 390 pneumonia and 148 normal. The images are high resolution with some ranging between 70 and 700KB. The was organised into hierarchy of folders, the main directory contained a test and training folder. The test and training directory contained two folders each for pneumonia and normal cases. The dataset is highly imbalanced with pneumonia cases being nearly 3 times greater than normal. In this state, It is very likely that the models will always predict pneumonia. To deal with this we use image augmentation and cost sensitive learning.

2.1. Augmentations

Class imbalance in CNN models can have detrimental effects on the performance, there may be a bias towards the class with higher proportion of images and the most effective method to address this is issue is oversampling (Buda et al., 2018). In this paper we use image augmentation to artificially generate new samples for the normal images training class . Many different image augmentation techniques are cited in literature however, one of the most successful is the traditional strategy of applying random transformations like flips, crops and rotations (Luis Perez, 2017). We iteratively performed random augmentations on the original images and the new count stood to 3682 images for the Normal class and 3875 images for the pneumonia class.

The chosen augmentation parameters (see Table 1) applied minimal transformations to the original images because X-Rays are usually taken in controlled environments. Apart from anatomy differences, the images are quite similar. The chest is usually always centered and the orientation is upright across all images. More extreme augmentations may introduce too much random noise to the data, making it challenging for the models to learn. Our main purpose was to produce more samples for the normal class for balancing, so more extreme transformations were not necessary.

Augmenter	Description
Flips	Flip images horizontally
Rotate	1.2 - 1.3 degress
Zoom	Scale in by 1.1 - 1.2
Brightness	Multiply by 1.1 - 1.2
Random Blur	Kernel Size of 1.2 - 1.4

TABLE 1 Augmentations were randomly applied from this set of augmenters.

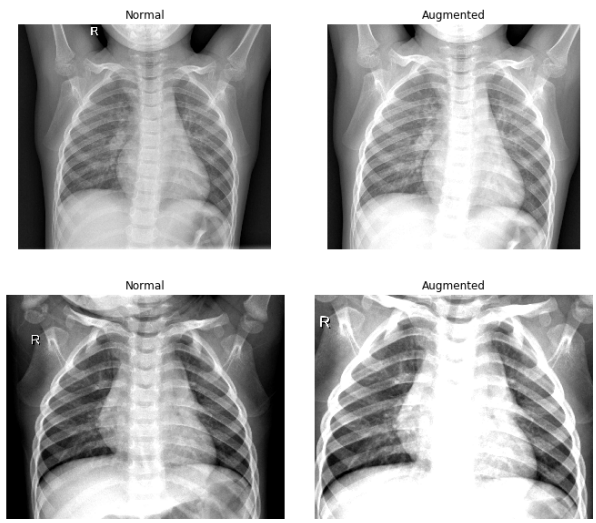

FIGURE 1 Example of Images from the Normal class (left) and its augmented counterpart (right)

| 2.2. Preprocessing

To prepare the data for training we resized all images into fixed size of 150x150 as most network architectures assume square sized images. Each image was converted into a 150x150x3 volume as some images had three channels (RGB pixels). We constructed a single numpy array with 7557 elements representing all training images (after augmentation). A single array containing all training labels for the respective images was also used. We perform one hot encoding on the training labels array, to represent categorical variables as binary vectors, which is an essential format for feeding our data into the model. For each image, we also scale the min/max pixel values between 0 and 1 as it allows for faster convergence during training. Finally, we shuffle the training data (images) and training labels in unison, this helps the model to generalize (distinguish between classes) better and also allows for faster convergence (Bengio, 2012).

3. Methods

| 3.1. Convolutional Neural Networks

Convolutional Neural Networks come from a family of deep learning neural networks which are prominently used in tasks such as object detection and image recognition. CNNs take images as inputs, which are then passed to series of hidden layers such as convolutional and pooling for feature extraction. Then fully connected layers serve as a classifier for the extracted features, We use an activation function to get an array of probabilities for each class. The class with the highest probability will be the predicted output. Contrary to regular neural networks, the neurons in CNNs are not connected to all neurons in the next layer, rather only a small portion is connected. This is done to reduce the number of weights, because in CNNs neurons are organised into 3 dimensions of width x height x depth (RGB color channels). For example, if we have a image of size 150x150, a single fully connected neuron will have 150x150x3 = 67500 weights. Such a huge number of weights at each neuron, in every layer would lead to the model being very slow and prone to overfitting. In CNNs connectivity to a small region of a prior layer lessens the number of wasteful parameters as images usually have similar features across different regions (edges etc). The convolutional layer in a CNN computes the dot product of neurons which are connected to regions of the input by convolving across the width and height of the input volume. As a result the network is able to learn certain features at spatial positions of a given input. The pooling layer performs downsampling, to continuously reduce dimensionality and the number of computations in the network. Altogether, the hierarchical structure of these layers allow CNNs to learn features at different levels of abstractions, the lower layers describe features such as edges while higher layers may describe bigger parts of a image. The components (convolutional, pooling) we have discussed above are the basic building blocks of CNNs. More complex architectures have been proposed in literature such as ResNeXt, DenseNet, InceptionV3 and VGG16 all of which are also used in this paper.

| 3.2. Transfer Learning

In practise, it is also common to adapt pretrained model weights to our working domain, this type learning is called transfer learning. As mentioned earlier, CNNs learn basic features such as edges and lines in the first few layers. The subsequent layers learn to detect more trivial shapes and objects using features learnt from previous layers. Lower level features such as edges and lines are common in many different types of images, we can use this to our advantage by using the network as a feature extractor or a starting point for the task of interest. This approach is most commonly used with small datasets to avoid overfitting and to compromise with limited computational power. To implement this, we simply remove the last fully connected layers (which act as classifiers) and **freeze** all other layers in the pretrained

network. The weights for the frozen layers are not changed during back propagation. To adapt this model to our domain, we then add a new classifier which conforms to our given task (classifying between two classes). This is the first approach we have used in this paper for transfer learning. One possible limitation of this approach is that it requires the pre-trained dataset to have some similarities to our target domain.

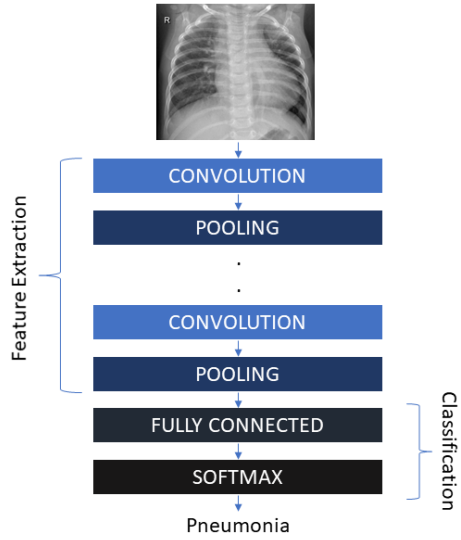

FIGURE 2 Overview of a simple CNN architecture.

The second approach used in this paper is called fine tuning. This is quite similar to the approach described above but now instead of freezing all layers, we only freeze the first few. The weights for unfrozen layers are initialized using the pretrained network. Previous studies have shown that this better than random initialization of weights, which is used when training models from scratch (Becherer et al., 2017). The weights for the unfrozen layers will be fine tuned according to our dataset by continuation of back propagation. In this approach we only freeze the first few layers as they will be used as feature extractors for only the most basic shapes. In this study, our models have been pretrained on the ImageNet dataset, which contains more than 14 million images and over 20,000 classes. Although, ImageNet does not have data relating to medical imaging or chest X-Rays, other studies in this domain have successfully used this dataset for transfer learning. For example, (Kermany et al., 2018) has used transfer learning using ImageNet on the same dataset as used in this study. Another paper, also used ImageNet for initializing the model weights for detecting pneumonia in chest x-rays(Benjamin Antin, 2017).

3.3. Network Architectures

3.3.1. VGG16

VGG16 is a 16 layer convolutional neural network, which achieves accuracy of 92.7% on the ImageNet dataset (Simonyan and Zisserman, 2015). This network uses 3x3 filter sizes (convolutions) with stride of 1 and 2x2 pooling layers with stride of 2 to progressively perform downsampling, for reducing the computational load. So, if we pass a 150x150x3 volume (raw RGB pixel data) from our dataset to such pooling layers, the resulting volume will be reduced

to of the size $H = \frac{(150-2)}{2+1} = 75$ and $W = \frac{(150-2)}{2+1} = 75$ (75x75x3). There are 5 pooling layers which trail some of the 15 convolutional layers, this stack is followed by 3 fully connected and a softmax layer for classification. VGG16 is shallowest of all the models considered in this paper, however this is also the heaviest with weights file size of 528MB (when trained on 224x224 images)

3.3.2. InceptionV3

The inception architecture also known as GoogleNet, takes a different approach than the traditional practice of sequentially stacking deeper convolutional layers. In a inception network, multiple convolutional and pooling layers operate on the same level, making the network more wider rather than deeper (Szegedy et al., 2016). Such organisation of layers are called inception modules and in InceptionV3, these are linearly stacked into 9 layers. This architecture maintains the complexity and accuracy of VGG16, while eliminating a large number of trainable parameters. The weights size for this architecture is only 92 MB, despite being deeper than VGG16.

3.3.3. ResNeXt50

It is generally agreed that deeper neural networks are capable of learning more complex representations of the input. However, researchers have observed that going too deep may have a negative effect on the models performance due to the degradation problem (He et al., 2016). The proposed solution to this problem is a residual neural network (ResNet) where convolutional layers are organised into residual blocks with skip connections. This allows the network to add feature maps (features convolved from previous layers) to next layers by skipping. The ResNext architecture further builds on this idea by using residual blocks containing convolutional layers operating on the same level (similar to inception modules) (Xie et al., 2017). ResNext50 weights file size is 96 MB and it achieves higher accuracy than its ResNet counterparts on the ImageNet dataset.

3.3.4. DenseNet121

The last CNN architecture we have considered for experimentation in this study is the DenseNet. These models present another approach for making networks deeper without the degradation problem. DenseNets are composed of entities called dense blocks, in these blocks each layer is densely connected to all its subsequent layers (Gao Huang, 2017). DenseNet121 has four such dense blocks each containing 6, 12, 24 and 16 dense layers respectively. Dense layers are made up of 2 layers, a convolutional layer with 1x1 filter and another with 3x3 filter. The layers between two adjacent dense blocks are known as transition layers and contain a convolutional and a pooling layer which are used for down sampling. All together, the network has 121 layers, the highest of all models considered in this report. DenseNets avoid degradation thanks to bypass connections (layers are connected to all subsequent layers), which allow for feature reuse. Despite being very deep, DenseNet121 weighs at only 33 MB while also achieving high accuracy measures.

3.4. Experimental Setup

In this study, we sought to compare four different CNN architectures using three different learning approaches for detecting pneumonia in X-ray images. We train each architecture from scratch on the Chest X-Ray images data set and also employ two different transfer learning approaches. The first approach is to adopt the pretrained model as

a feature extractor for our task, the second is to use it for initialising new weights (fine tuning). For all 3 training approaches, we first remove the original top layers used for classification with our own softmax classifier (number of classes set to 2). We also add a global average pooling layer before the softmax classifier to help reduce the computational load and minimize overfitting. For the first transfer learning approach, we freeze all layers and only train the newly added classification layers. For the second option, we only freeze the bottom few layers and train the rest including the newly added ones. For VGG16 we freeze the first 3 layers and use the rest, with InceptionV3 we only freeze first 2 inception modules, in ResNeXt we freeze first 2 convolutional blocks. Finally with DenseNet121, we freeze all layers up to the first Dense Block and transition layer. Notice that the layers we choose to freeze are the first few layers, which identify basic edges and lines.

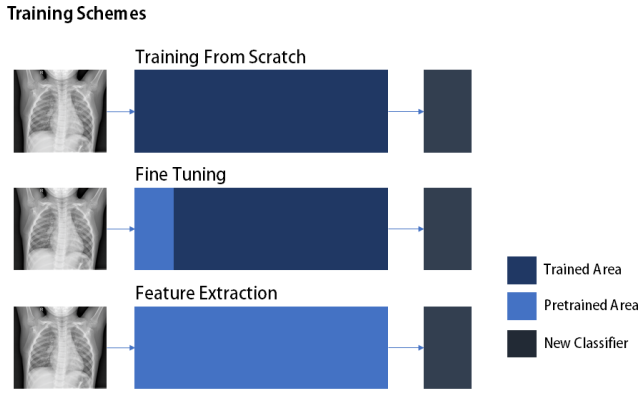

FIGURE 3 Visualisation of the training schemes discussed above

6801 samples are used for training and the validation set consists of 756 images(10% split). As optimizer, we use RMSprop with learning rates of 0.0001 for training from scratch and 0.001 for transfer learning methods. Smaller learning rate in transfer learning decreases the risk of distorting weights trained on the previous dataset (ImageNet), other papers in literature have also used this technique (Yang et al., 2018). We use the standard batch size of 32 for training over 6 epochs, after each iteration the data is shuffled. the data is shuffled to reduce chances of over fitting. A larger epoch size would have been preferable, but is avoided due to larger training times and computational load. We also employ cost sensitive learning to add more weight to the under represented normal class. After training we evaluate the model using a test set containing 624 samples (234 Normal and 390 from Pneumonia class). For comparison we use area under the receivers operating characteristic curve (ROC) and precision-recall curve (PRC). We have considered accuracy as an inadequate measure due to the imbalanced nature of this data. Higher AUROC and AUPRC values will indicate how well, the model is able to distinguish between classes. Ideally we want both of these to be close to 1, a high AUROC and very low AUPRC value will indicate bad performance (Davis and Goadrich, 2006). We also include the mean training accuracy and mean validation accuracy to check if the model is over fitting. After evaluation with these metrics we will choose four best models for further assessment using sensitivity and specificity. Sensitivity is the true positive rate and specificity is the true negative rate (in the next section we describe this in context). These will calculated from the confusion matrix using formulas, $Sensitivity = \frac{TP}{TP+FN}$ and $Specificity = \frac{TN}{TN+FP}$. We train and build our models using the Keras and Tensorflow backened libraries in python. For training we use the google collaboratory environment equipped with Tesla K80 GPU.

	AUROC	AUPRC	Train Acc	Valid Acc
VGG16 - Fine Tuned	0.77	0.89	98%	88%
VGG16 - Feature Extraction	0.81	0.91	94%	95%
VGG16 - From Scratch	0.80	0.90	90%	87%
InceptionV3 - Fine Tuned	0.86	0.93	98%	90%
InceptionV3 - Freeze Layers	0.61	0.82	91%	72%
InceptionV3 - Full Train	0.78	0.90	95%	88%
DenseNet121 - Fine Tuned	0.79	0.90	98%	98%
DenseNet121 - Freeze Layers	0.77	0.88	85%	77%
DenseNet121 - Full Train	0.64	0.82	94%	63%
ResNeXt50 - Fine Tuned	**0.87**	**0.93**	**98%**	**98%**
ResNeXt50 - Freeze Layers	0.62	0.84	90%	69%
ResNeXt50 - Full Train	0.54	0.73	92%	57%

TABLE 2 Results obtained from experiments (Top performance highlighted in bold)

4. Results and Discussion

Table 2 summarizes the outcomes from our experiments. In total we have 3 experimental setups for each architecture, giving us 12 models for comparison. We have 3 experiments for each architecture, giving us 12 models for comparison. The results indicate low performance from models trained from scratch with an exception of VGG16. We can observe that fully trained DenseNet121 and ResNext50 are severely overfitting due to the large variance between training and validation accuracy. This maybe because DenseNet121 and ResNext50 are very deep and require more training data. Transfer learning for feature extraction gives higher AUROC and AUPRC values from models trained from scratch. Overfitting is still clearly prevalent in all model architectures apart from VGG16, although here it seems to be less severe. For transfer learning using fine tuning, we can rule out overfitting for RexNext50 and DenseNet121 as we observe very little difference between the training and validation accuracy. Fine tuning gives the highest AUC values for all architectures apart from VGG16, this is quite interesting to observe as VGG16 performed the best on the other two learning approaches. For InceptionV3, all learning approaches seem to have at least some overfitting, although fine tuning is not severe. Overall, we see that the fine-tune approach works well with all architectures, some overfitting is happening with InceptionV3 and VGG16 but it is not as severe compared to previous examples. (maybe talk about why it's overfitting model complexity etc)

To proceed with our analysis, we choose the the fine tuned models for ResNeXt50, DenseNet121 and InceptionV3. For VGG16 we choose the feature extraction model. All other models have been disregarded because either they were overfitting or the AUC values were lower than current chosen models. The AUROC and AUPRC for all selected models are not significantly different, so we know that these metrics are not misleading due to class imbalance. Now we will also consider sensitivity and specificity to identify, what our results mean in terms of identifying pneumonia from X-ray Images. Sensitivity is the true positive rate or in our case the percentage of patients who were correctly diagnosed with pneumonia. Specificity is the percentage of patients who were correctly identified as belonging to the normal class by the model. Ideally we want to choose the overall best model by find the right balance between these metrics.

	Sensitivity	Specificity
ResNeXt50 - Fine Tune	0.987	0.756
DenseNet121 - Fine Tune	0.997	0.581
InceptionV3 - Fine Tune	0.943	0.777
VGG16 - Feature Extraction	0.984	0.64

TABLE 3 Sensitivity and Specificity for selected models

High sensitivity will mean that there are less number of people who actually have pneumonia and were diagnosed as normal. This is very important in this scenario because diagnosing a pneumonia victim as normal can be very fatal and even cause death. On the other hand, we also don't want normal people being diagnosed with pneumonia, this may harm clinical reputation and incur unnecessary costs. DenseNet121 and VGG16 all give sensitivity above 98%, however the specificity values for these models are quite low. Meaning lots of normal people are being diagnosed with pneumonia. Top specificity value is given by InceptionV3 but it also gives the lowest sensitivity. ResNeXt50 gives a high sensitivity value at 98.7% and fair specificity value of 75.6%. This value isn't ideal but we deem it as the best out of all the selected models. This model also showed less signs of overfitting as training and validation accuracy was quite close. Overall, we conclude that ResNeXt50 has provided the best performance out of all the 12 models tested.

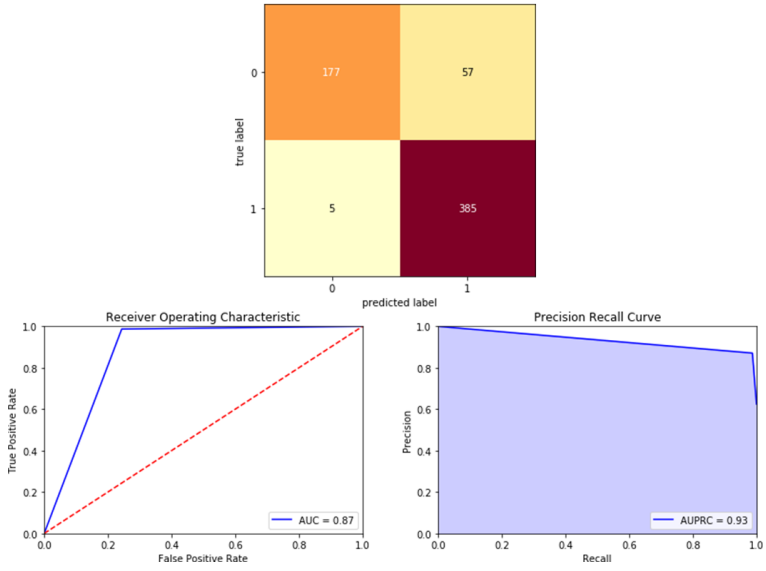

FIGURE 4 Confusion matrix, ROC and PRC for ResNeXt50 - Fine Tuned. (1 is pneumonia, 0 is normal)

Confusion matrix in Fig. 5, shows 57 normal people have been incorrectly diagnosed with pneumonia and only 5 victims were incorrectly identified as normal by the model. We observed that almost all models had some bias towards false positive (normal identified as pneumonia). This maybe due to the model not being able to generalize well

between normal and pneumonia, due to imbalanced data. In comparison to other studies, this model has achieved a high sensitivity value. The paper by (Kermany et al., 2018) was trained and tested on the same dataset for 100 epochs, they achieved sensitivity of 93.2% and specificity of 90.1%. Another paper trained on this dataset along with the ChestXray-14, achieved 96.1% sensitivity and 91.03 specificity (Mrinal Haloi, 2018). Although ResNeXt50, overall performed the best, it is still inadequate to be used in real life medical applications, better performance is needed due to the seriousness of this context. The specificity value is especially alarming, which may be caused by a number of limitations and could be improved further. We have only trained our models for 6 epochs to reduce training times, ideally the model should be trained on larger number of epochs until training and validation accuracies start diverging. Oversampling (augmentations) was used to generate new samples for the minority class, this may be introducing some similarity between the the classes (augmented samples might start looking like pneumonia). It would be interesting to experiment with different settings for image augmentation and check how they are effecting the accuracy.

5. Conclusion

We have presented methodical comparisons between CNN architectures and different learning approaches for detecting pneumonia in X-Ray images. We evaluated four different CNN architectures using three different learning approaches on the publically available Chest X-Ray images . The results show that transfer learning using fine-tuning performs quite well on all cnn architectures, showing little or no overfitting in most cases. For the overall top model, we find that ResNeXt-50 with fine tuning performs the best. Achieving a high sensitivity (recall) of 98.7%, 75.6% specificity and AUROC of 0.87. 98.7% sensitivity is exceptional in this case as we have very small false negatives, however specificity for this model definitely needs improvement. We propose that with further tweaking this model can be used for real-life applications. For further work, we suggest that hyper parameters for the CNN models should be chosen using randomized grid search. In this paper we chose hyper parameters based general preferences from other studies in this domain due to computational restrictions. However, since the models used here are different from other studies, an accuracy boost maybe observed by using this approach. Next, we suggest the inclusion of patients history and other clinical variables with X-ray images as in practise radiologists review this information in the diagnosis process. We also suggest that tweaking/optimizing the model architecture for ResNext-50 along with transfer learning may further improve the results.

References

Ivo M. Baltruschat, Hannes Nickisch, Michael Grass, Tobias Knopp, and Axel Saalbach. Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification. *Scientific Reports*, 9(1), apr 2019. doi: 10.1038/s41598-019-42294-8. URL https://doi.org/10.1038%2Fs41598-019-42294-8.

Nicholas Becherer, John Pecarina, Scott Nykl, and Kenneth Hopkinson. Improving optimization of convolutional neural networks through parameter fine-tuning. *Neural Computing and Applications*, nov 2017. doi: 10.1007/s00521-017-3285-0. URL https://doi.org/10.1007%2Fs00521-017-3285-0.

Yoshua Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures. In *Lecture Notes in Computer Science*, pages 437–478. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-35289-8_26. URL https://doi.org/10.1007%2F978-3-642-35289-8_26.

Emil Martayan Benjamin Antin, Joshua Kravitz. Detecting Pneumonia in Chest X-Rays with Supervised Learning. 2017.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, oct 2018. doi: 10.1016/j.neunet.2018.07.011. URL https://doi.org/10.1016%2Fj.neunet.2018.07.011.

Michael Goldbaum Daniel Kermany, Kang Zhang. Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification. Technical report, 2018. URL http://dx.doi.org/10.17632/rscbjbr9sj.2.

Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*. ACM Press, 2006. doi: 10.1145/1143844.1143874. URL https://doi.org/10.1145%2F1143844.1143874.

Laurens van der Maaten Kilian Q. Weinberger Gao Huang, Zhuang Liu. Densely Connected Convolutional Networks. *CVPR*, 2017.

Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng, and Yi Yang. Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification. *CoRR*, 2018.

Varun Gulshan, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip C. Nelson, Jessica L. Mega, and Dale R. Webster. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, 316(22):2402, dec 2016. doi: 10.1001/jama.2016.17216. URL https://doi.org/10.1001%2Fjama.2016.17216.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015. doi: 10.1109/iccv.2015.123. URL https://doi.org/10.1109%2Ficcv.2015.123.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. doi: 10.1109/cvpr.2016.90. URL https://doi.org/10.1109%2Fcvpr.2016.90.

Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5):1122–1131.e9, feb 2018. doi: 10.1016/j.cell.2018.02.010. URL https://doi.org/10.1016%2Fj.cell.2018.02.010.

Jason Wang Luis Perez. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *Computer Vision and Pattern Recognition*, 2017.

MCDHB. Health & Disability Advisory Committee Meeting. Technical report, 2019.

A. McLuckie. *Respiratory Disease and its Management*. Springer-Verlag London, 2009.

Ahmed Tahseen Minhaz Khalid Ashraf Mohammad Tariqul Islam, Md Abdul Aowal. Abnormality Detection and Localization in Chest X-Rays using Deep Convolutional Neural Networks. *Computer Vision*, 2017.

Pradeep Walia Mrinal Haloi, Raja Rajalakshmi K. Towards Radiologist-Level Accurate Deep Learning System for Pulmonary Screening. *Computer Vision and Pattern Recognition*, 2018.

F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer. Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization. *Scientific Reports*, 9(1), apr 2019. doi: 10.1038/s41598-019-42557-4. URL https://doi.org/10.1038%2Fs41598-019-42557-4.

Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis P. Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *CoRR*, 2017.

RCR. Clinical radiology UK workforce census 2016 report. Technical report, 2016. URL https://www.rcr.ac.uk/system/files/publication/field_publication_files/cr_workforce_census_2016_report_0.pdf.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR*, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. doi: 10.1109/cvpr.2016.308. URL https://doi.org/10.1109%2Fcvpr.2016.308.

Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017. doi: 10.1109/cvpr.2017.634. URL https://doi.org/10.1109%2Fcvpr.2017.634.

Yang Yang, Lin-Feng Yan, Xin Zhang, Yu Han, Hai-Yan Nan, Yu-Chuan Hu, Bo Hu, Song-Lin Yan, Jin Zhang, Dong-Liang Cheng, Xiang-Wei Ge, Guang-Bin Cui, Di Zhao, and Wen Wang. Glioma Grading on Conventional MR Images: A Deep Learning Study With Transfer Learning. *Frontiers in Neuroscience*, 12, nov 2018. doi: 10.3389/fnins.2018.00804. URL https://doi.org/10.3389%2Ffnins.2018.00804.

Koichiro Yasaka and Osamu Abe. Deep learning and artificial intelligence in radiology: Current applications and future directions. *PLOS Medicine*, 15(11):e1002707, nov 2018. doi: 10.1371/journal.pmed.1002707. URL https://doi.org/10.1371%2Fjournal.pmed.1002707.