Institute of Data

Data Science & Artificial Intelligence

Capstone Project

Pneumonia Detection in X-Rays Using Deep Learning

By
TAN YUE HANG
Date: 31 March 2024

Table of Contents

Abstract

Pneumonia, a severe lung condition, has impacted countless individuals globally. Some fatalities have resulted from misdiagnoses due to the complexities of image interpretation that require skilled radiologists. This issue is further complicated by the large volume of patients and X-ray images that need to be analysed daily. The objective of this project is to develop a deep learning model that employs transfer learning with a pre-trained CNN model to distinguish between healthy lungs and those infected with pneumonia. The findings indicate that a recall score of more than 98% can be attained by using DenseNet169 for feature extraction from the chest X-ray images, coupled with multiple dense layers with an optimized number of neurons for classification.

Problem statement

Pneumonia is an infection that inflames the air sacs in one or both lungs. It can be caused by bacteria, viruses (such as influenza or COVID-19), or fungi. The air sacs may fill with fluid or pus, causing discomfort and suffering.

The symptoms of pneumonia include a cough with phlegm, fever, chills, and difficulty in breathing. The severity of these symptoms could depend on factors such as age, overall health, and whether the infection is bacterial or viral. The mortality rate of hospitalized patients is about 5 – 10%, while it could also go as high as 30% for patients in the ICU.

Available treatments include antibiotics (for bacterial infections), cough suppressants, pain medications, etc. Depending on whether the treatment is invasive or comes with more serious implications, there may be more or less tolerance for having more false positives in the confusion matrix. This aspect needs to be discussed with certified medical practitioners. Some of the available diagnosis methods are Chest X-Ray, breathing test, sputum test, urine test, blood test, bronchoscopy, etc.

H. J. Koo et al. (2018) mentioned that various factors could affect how a pneumonic chest X-Ray image appears, such as whether it is a viral or bacterial infection, the family of the viruses, age, immune status, seasonal variation, and community outbreaks. [1]

The observation of pneumonia in lungs is generally complex. Depending on the type of infection, the observation could come in various sizes. [2] An example of how a radiologist interprets a chest X-Ray image is as follows:

"*The trachea is centrally located with no mediastinal shift. The costophrenic angles are sharp and unobscured. A non-uniform opacity is present in the right middle zone, characterized by visible air bronchograms and blurry borders. The right cardiac border is obscured, indicating right middle lobe pneumonia. The size of the heart appears to be normal.*" [2]

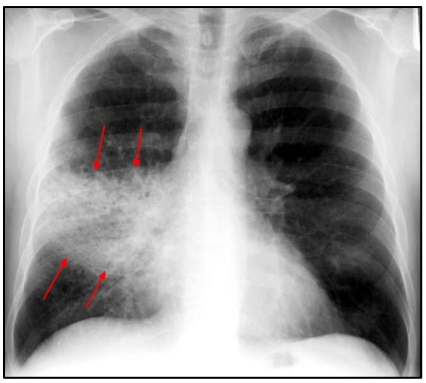


Figure 1: An X-Ray Image of a Pneumonic Lung. [2]

Domain

The area of study and the construction of deep learning models are focused on the medical field. Radiology is struggling with a surge in data due to an increase in exams and a shortage of staff. This results in a heavy workload for radiologists as the number of exams outpaces the number of specialists. An increased workload poses a risk of more interpretation errors. Reducing interpretation time by half increases the error rate by 16.6%. [4]

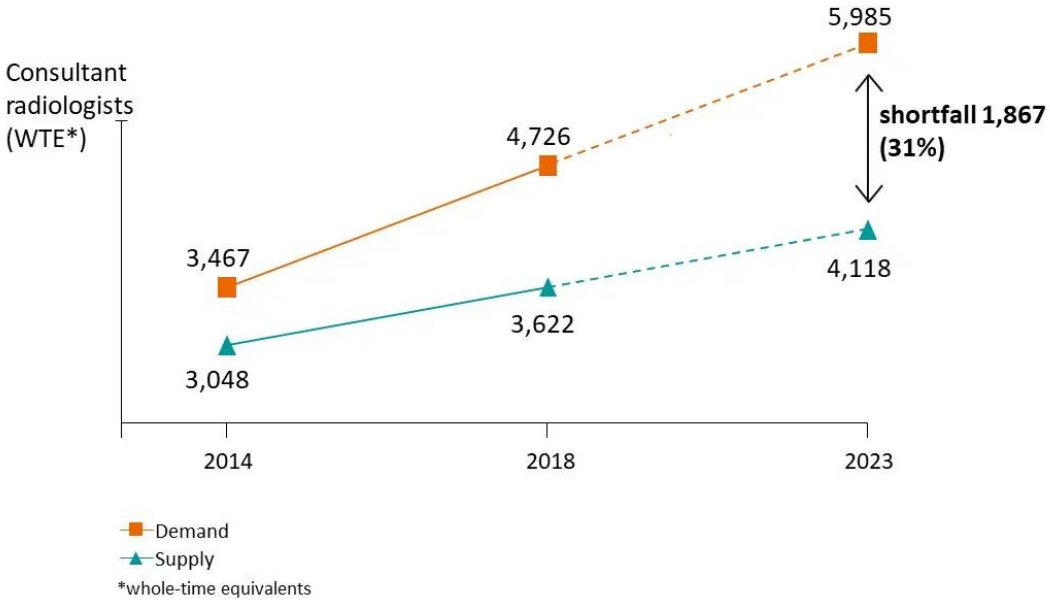


Figure 2: Gap Between Demand and Supply Over the Years

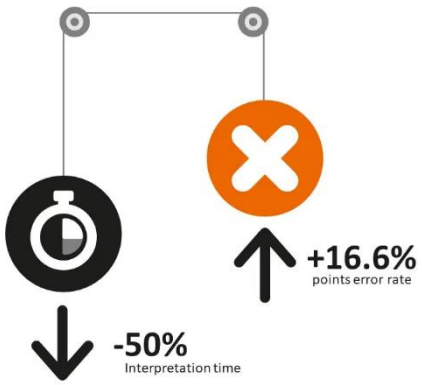


Figure 3: Increase in Interpretation Error with Decrease in Interpretation Time

Stakeholders

The key stakeholders for this project are medical professionals and healthcare workers, namely radiologists, physicians, and nurses. Their professional inputs are of great value in tuning the model to improve the performance (recall) of chest X-ray image classifications and the speed of diagnosis in real-life scenarios.

Business question

Early detection and treatment are key. Therefore, fast and effective diagnosis plays a crucial role in the early detection of pneumonia.

In the study by L. L. Plesner et al. (2023), AI was able to correctly identify approximately 99.1% of abnormal X-rays and about 99.8% of critical X-rays. On the other hand, human radiologists were able to correctly identify approximately 72.3% of abnormal X-rays and about 93.5% of critical X-rays. [5]

AI can identify features that are not easily detected by the human eye, thereby transforming radiology from a subjective perceptual skill to a more objective science. [6]

Table 1: Comparisons of AI Accuracy with Human Radiologists' Accuracy [5]

	Abnormal X-Rays Accuracy	Critical X-Rays Accuracy
AI	99.1%	99.8%
Human Radiologists	72.3%	93.5%

The enhanced efficiency offered by Artificial Intelligence empowers radiologists to undertake tasks of greater value, thereby increasing their visibility to patients. [6]

As a result, the two most pressing healthcare questions boil down to:

1. How accurately can the deep learning model classify images for pneumonia diagnosis?
2. How much time can it save for medical practitioners in the healthcare sector?

Data question

The data comes in the form of images, and the objective is to build an image classification deep learning model to classify chest X-ray images, determining whether the lung(s) are infected with pneumonia. Hence, the data questions are:

1. Is the sample size of the overall data sufficient to train a reliable deep learning model?
2. Is the image data RGB or grayscale?
3. Are the classes imbalanced?
4. Is the image data clean, or does it require augmentations to provide the model with more diversified features?
5. What is the suitable metric of choice for model optimization to achieve certain goals?
6. Which pre-trained CNN model provides the best possible performance?
7. What is the choice of hyperparameters for tuning?
8. What is the suitable methodology of hyperparameter tuning to efficiently optimize the model towards the intended goal?

Data

The dataset was acquired from Kaggle and originally sourced from Mendeley. The dataset is organized into three folders: train, val, and test. Each folder contains two sub-folders, namely Normal and Pneumonia, which represent the two classes of the X-ray images.

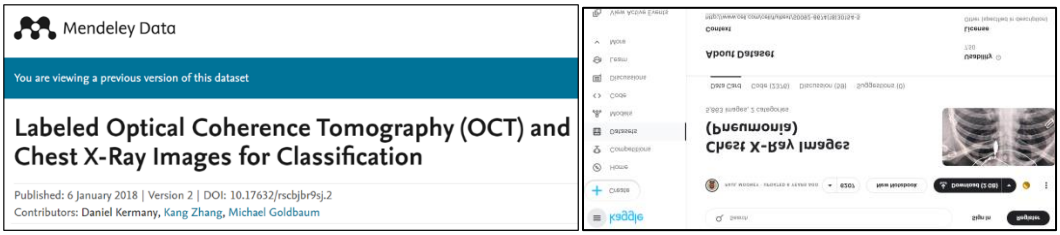
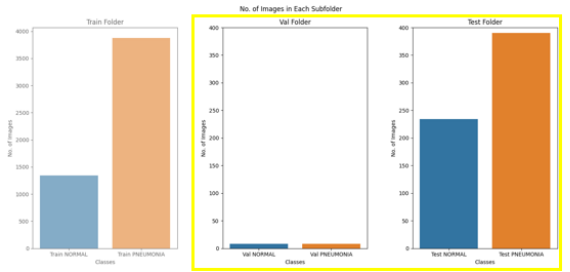


Figure 4: Source of Data

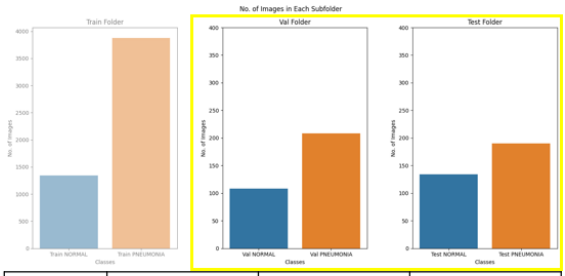
Data science process

Data cleaning and EDA

One of the issues with the dataset is that the training images are imbalanced, with the Pneumonia class being about 2.9 times larger than the Normal class. To address this, more class weights were applied to the Normal class than the Pneumonia class during model training (3:1). Another issue with the dataset is the insufficient number of validation images. Therefore, some image data from the test folder were moved to the validation folder to balance it out.



	Train		Val		Test	
Classes	Normal	Pneumonia	Normal	Pneumonia	Normal	Pneumonia
No. of Images	1341	3875	8	8	234	390



	Train		Val		Test	
Classes	Normal	Pneumonia	Normal	Pneumonia	Normal	Pneumonia
No. of Images	1341	3875	108	208	134	190

Figure 5: Before and After Moving Some of the Image Data from the Test Folder to the Validation Folder.

Some of the chest X-ray images, with and without image augmentations, are randomly loaded for preview. The settings for the augmentations are as follows:

```
train_datagen = ImageDataGenerator(
        rescale=1./255.,
        rotation_range=5,
        width_shift_range=0.1,
        height_shift_range=0.1,
        horizontal_flip=True
)
```

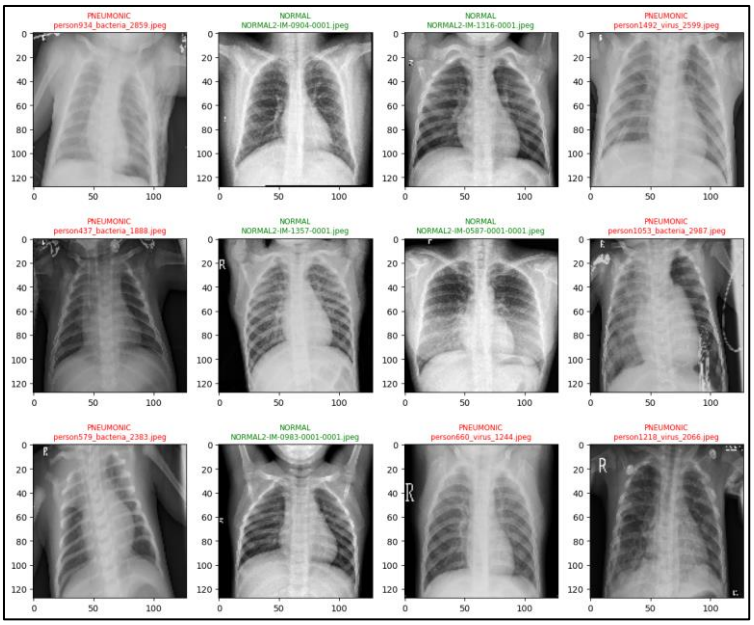


Figure 6: Chest X-Ray Images Without Augmentations

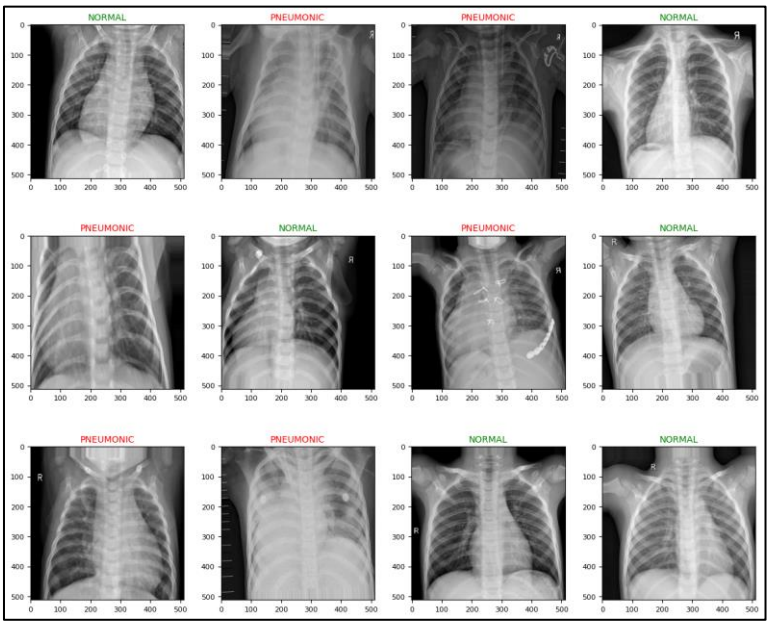


Figure 7: Chest X-Ray Images With Augmentations

Literature Reviews

Campos-Lopez et al. (2023) demonstrated that DenseNet201 and ResNet50 perform well on X-ray images of pneumonia [7]. This aligns with the findings from Sabottke and Spieler (2020) [8]. VGG16 and VGG19 were also used in the study by Campos-Lopez et al. (2023), but they did not offer comparable performance [7].

Sabottke and Spieler (2020) discovered that a lower resolution (64 x 64) works better for observations that are generally larger in size, whereas a higher resolution (256 x 256, 448 x 448) could work better for subtler observations and binary classification [8].

In the study by Q. Zheng et al. (2020), an improvement in accuracy of more than 8% was observed with a full-stage data augmentation framework on both CIFAR-10 (coarse-grained) and CIFAR-100 (fine-grained) datasets. Among these, translation, horizontal flip, and scale transformation seem to contribute more to the improvement [9]. Similarly, M. Nagaraju et al. (2022) also discovered a similar outcome: with full-stage data augmentation, the classification performance shows better performance than the existing method by up to 28.31% [10].

Batch normalization is a regularization method that standardizes the activations within a layer. This standardization is achieved by subtracting the mean of the batch from each activation and then dividing the result by the batch's standard deviation. [11]

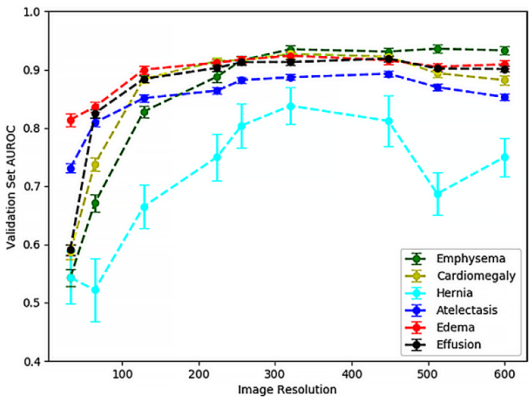


Figure 8: AUC with Different Illnesses at Various Image Resolutions. [8]

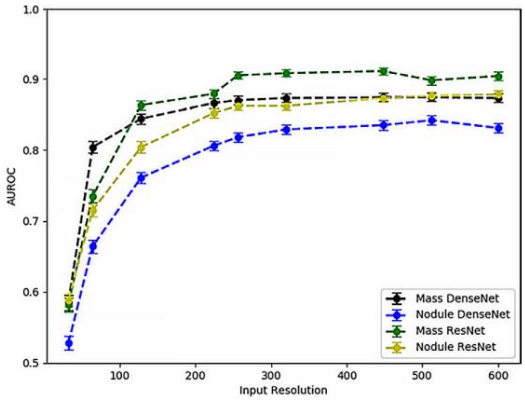


Figure 9: AUC with Different CNN Models at Various Image Resolutions. [8]

Modelling

A deep learning model for chest X-ray image classification will be built following the strategy illustrated below. The input images will be adjusted to different sizes and applied with various augmentations. A pre-trained CNN model will be chosen based on the performance from testing various pre-trained CNN models, and custom dense layers will be added on top of the pre-trained CNN model to perform classifications and output the results.

A few hyperparameters were selected for model tuning based on the literature, such as image size, classification threshold, and number of neurons. The Box-Behnken Design, a popular response surface methodology, was employed to narrow down the tuning range of these hyperparameters, and Keras Tuner libraries were used to choose the exact combination that gives the highest possible recall score.

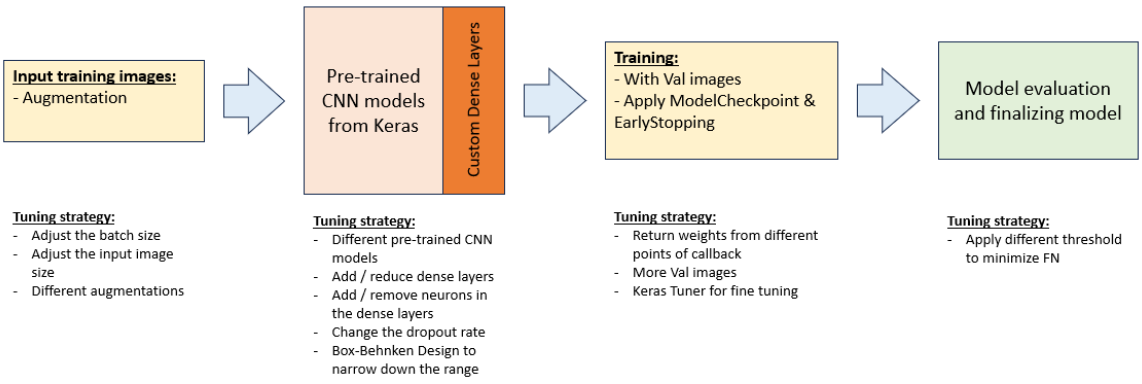


Figure 10: Modelling and Tuning Strategy

Pre-trained CNN Model Selection

Recall is the target metric to be maximized, as the goal is to reduce the number of False Negatives in the diagnosis as much as possible. Evidently, DenseNet169 appears to perform the best among the seven selected models. The performance results of the models generally align with other studies, where DenseNet and ResNet50 architectures perform better than VGG16 and VGG19 on X-ray images. [7][8]

Table 1: Performance of Various Pre-Trained CNN Models from Keras

Model	Accuracy	Precision	Recall	F1	AUC	True Normal	False Pneumonia	False Normal	True Pneumonia	Wall Time
DenseNet169	0.910494	0.885167	**0.973684**	0.927318	0.969	110	24	5	185	21min 16s
DenseNet201	0.873457	0.846512	0.957895	0.898765	0.933	101	33	8	182	16min 43s
VGG16	0.864198	0.837963	0.952632	0.891626	0.924	99	35	9	181	19min 26s
VGG19	0.858025	0.849515	0.921053	0.883838	0.917	103	31	15	175	21min 21s
ResNet50V2	0.839506	0.8	0.968421	0.87619	0.940	88	46	6	184	18min 17s
ResNet152V2	0.882716	0.85514	0.963158	0.905941	0.958	103	31	7	183	20min 35s
MobileNetV3Large	0.756173	0.751131	0.873684	0.807786	0.772	79	55	24	166	14min 27s

Fixed parameters:

```
BATCH_SIZE = 6
EPOCHS = 5
NEURONS = 512
img_input_size = 512
img_size = (img_input_size, img_input_size)
img_shape = (img_input_size, img_input_size, 3)
Threshold = 0.5
```

Fixed augmentations (training data only):

```
rotation_range=5,
width_shift_range=0.1,
height_shift_range=0.1,
horizontal_flip=True
```

Figure 11: Parameters That are Fixed When Training Different Pre-trained CNN Models.

Hyperparameters Correlation: Box-Behnken Design

Deep learning models operate like “black boxes”, where it is difficult to understand how the hyperparameters work and interact with each other behind the scene. A potentially feasible approach is to treat the “black box” deep learning model as a system, where we can model the input hyperparameters and output responses with the Design of Experiments (DoE) approach using Minitab software. This approach can potentially help the researcher to understand the correlations between the factors and responses in order to obtain a ballpark range of selected hyperparameters to start

with. Box-Behnken Design is selected in this study not only that it is one of the more popular Response Surface Methodology (RSM), it is also more efficient in terms of the number of experiments required.

Table 2: Selected Hyperparameters to Tune and Their Respective Parameter Windows

	Image Size	Threshold	Neurons
LL	128	0.3	128
HH	640	0.7	1024

Table 3: DoE Legs Constructed by Box-Behnken Design

	Predictors		
Leg	**Image Size**	**Threshold**	**Neurons**
1	640	0.7	576
2	640	0.3	576
3	384	0.7	1024
4	128	0.5	1024
5	640	0.5	128
6	384	0.3	1024
7	128	0.5	128
8	384	0.5	576
9	128	0.7	576
10	128	0.3	576
11	384	0.5	576
12	384	0.5	576
13	384	0.3	128
14	640	0.5	1024
15	384	0.7	128

Box-Behnken Design: Results

The regression model fits the data quite well, with an R-squared value of 93%. From the contour plot, the optimal image size appears to fall somewhere between 400x400 and 500x500. This result aligns with the findings from other researchers [8]. The optimal number of neurons seems to be somewhere below 300. The number of neurons for all dense layers is made equal in this case. It's noteworthy that there are potentially two main reasons why Box-Behnken Design may not give very accurate estimations on the response despite the high R-square value (93.09%). This is because Box-Behnken Design assumes quadratic and linear relationships in most cases, whereas in Deep Learning (DL) neural networks, the correlation could get very complex and often non-linear. Another reason is that there is inevitable randomness when deep neural networks train and make predictions. Therefore, the regression results from Box-Behnken Design should only be taken as a reference, or a narrowed-down ballpark range for us to start tuning the deep learning model.

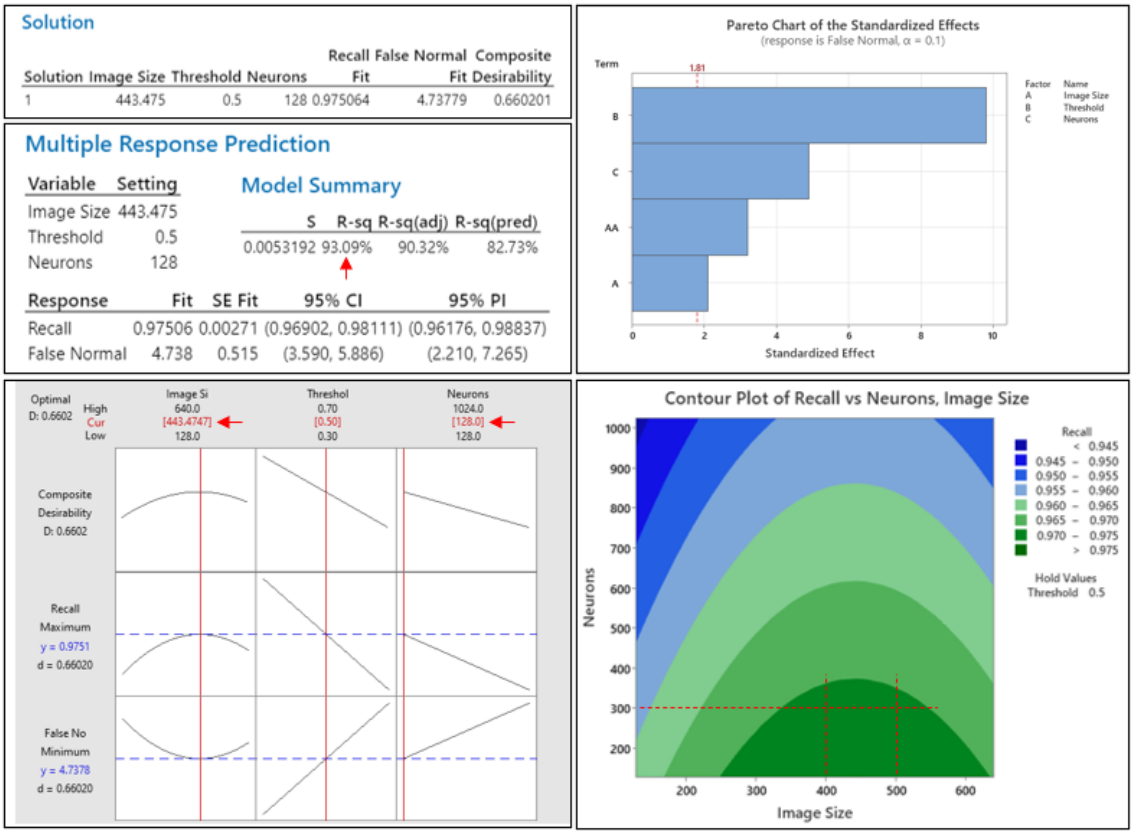


Figure 12: Optimization Results with Regression Modelling from Box-Behnken Design.

Table 4: Settings Identified by Regression That Are Likely to Give the Highest Possible Recall Score.

Image Size	Neurons
443	128

Model Optimization: Keras Tuner

Generally, there are a few types of dense layer neural structures, as shown in Figure 13 below. This is one of the parameters of interest in this study. To understand the impact of different image sizes, the number of neurons in the first dense layer, and the number of neurons in the second dense layer, the hyperparameters are selected as shown in Table 5.

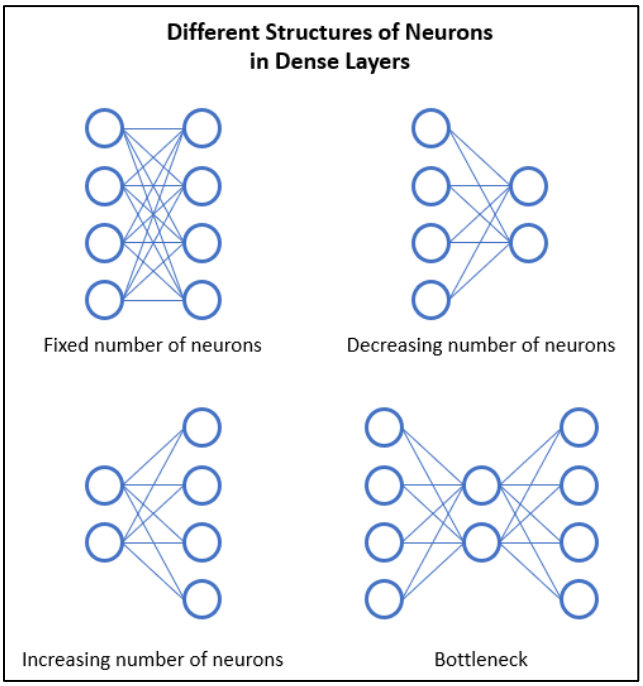


Figure 13: Different Types of Dense Layer Neural Structures.

Table 5: Selected Hyperparameters for Keras Tuner

	Image Size	Neurons 1	Neurons 2
LL	128 x 128	32	32
HH	640 x 640	512	512

With the RandomSearch function from Keras Tuner, it has been identified that the combination of hyperparameters shown in Table 6 appears to yield the highest recall score.

```
Trial 0
Hyperparameters {'image_size': 448, 'neuron_one': 32, 'neuron_two': 224}
Score 0.9855769276618958
----------------------
Trial 1
Hyperparameters {'image_size': 256, 'neuron_one': 480, 'neuron_two': 160}
Score 0.9855769276618958
----------------------
Trial 2
Hyperparameters {'image_size': 256, 'neuron_one': 480, 'neuron_two': 288}
Score 0.9839743773142496
----------------------
```

Figure 14: Optimal Hyperparameters Identified by Keras Tuner.

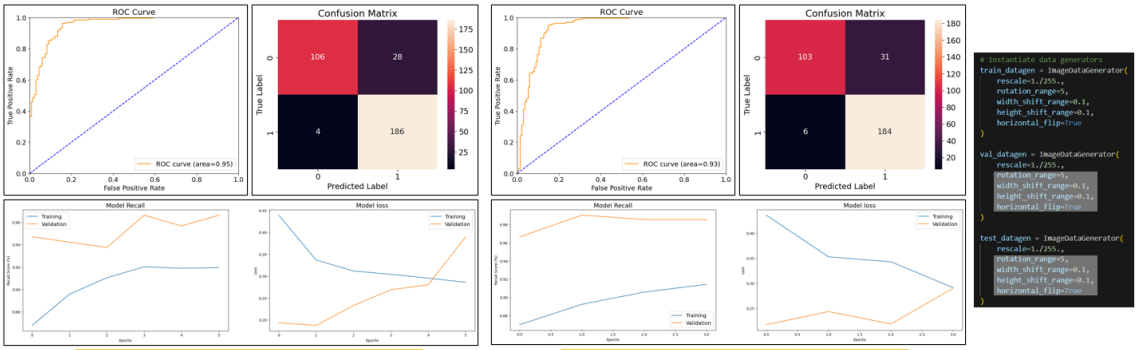
Table 6: Selected Hyperparameters for Keras Tuner

	Image Size	Neurons 1	Neurons 2
Best	448 x 448	32	224

The RandomSearch function from Keras Tuner returns a combination of hyperparameters at random and performs training with these. Depending on the settings in the function, this approach might require a tremendous amount of computational resources in order to return the best possible combinations. Due to limited resources available in this study, only three different combinations of hyperparameters were evaluated. On the flip side, this also demonstrates the value of the Box-Behnken Design, which helps to narrow down the range of hyperparameters before proceeding to randomized search tuning. This could save on computational resources while still being able to identify reasonably good hyperparameters.

Full Stage Data Augmentation

Full-stage data augmentation involves applying augmentation to validation and test image data, rather than just training data. Contrary to other studies, the overall results with full-stage data augmentation actually perform worse than without it. In other studies, CIFAR-10 and CIFAR-100 datasets were used, which consist of colour (RGB) images and much more diversified image categories. The type and magnitude of augmentations used could also influence the outcome. However, it's noteworthy that the validation loss (val_loss) along the epochs is generally lower (better) than without full-stage data augmentation.



Accuracy	Precision	Recall	F1
0.901235	0.869159	0.978947	0.920792

Without full stage data augmentation

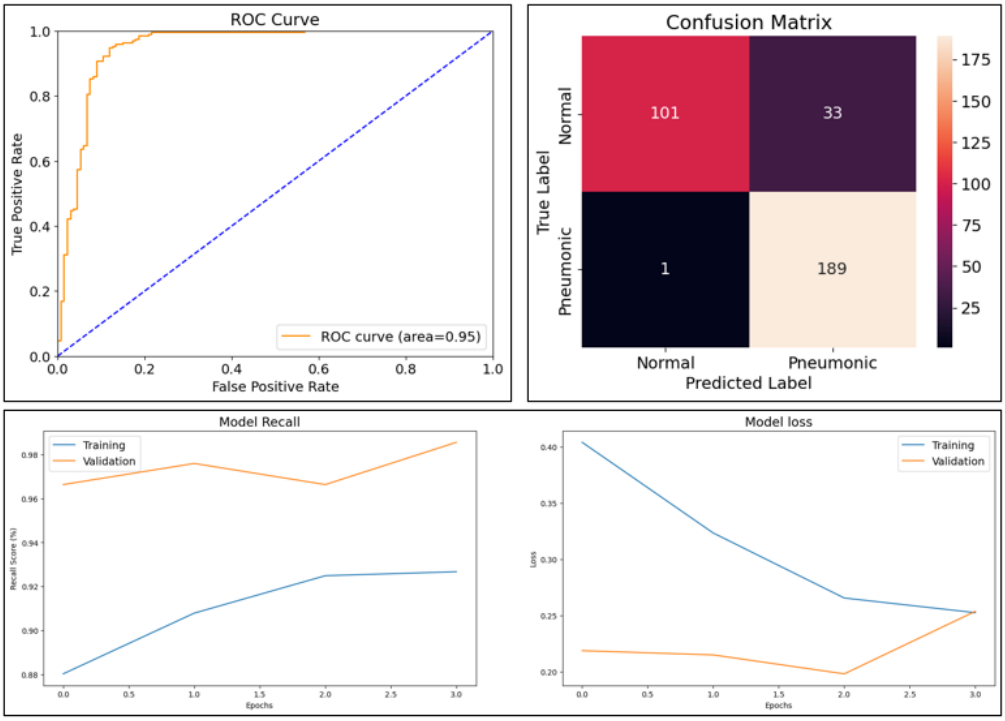
Accuracy	Precision	Recall	F1
0.885802	0.855814	0.968421	0.908642

With full stage data augmentation

Figure 15: Results with and without Full-Stage Data Augmentation

Outcomes: Finalized Model & Discussions

The model managed to achieve an AUC score of 95% and performs reliably on test images. The goal is to minimize False Negatives (Normal) to reduce the risk of misdiagnosing true positive cases. The best recall score was already achieved at the first epoch with minimal loss. Different hyperparameters were attempted, and all combinations achieved the best recall score in less than five epochs. As a trade-off, False Positives would increase as False Negatives reduce. This is probably acceptable depending on the medications and treatments given to patients who are misclassified as 'positive'. The recall and False Negative rates can be further reduced by adjusting the classification threshold (default = 0.5), if needed. The recall score achieved by the model is 99.47%.



Accuracy	Precision	Recall	F1
0.895062	0.851351	0.994737	0.917476

Figure 16: Results of the Finalized Model

Implementation

With a chest X-ray image classification model, the model can be implemented in various ways to benefit the healthcare sector. One common method is to implement the model as a web application using web frameworks such as Flask and Django, which allows users to upload chest X-ray images and receive prediction results online. Another popular method is to deploy the deep learning model within a mobile application. The advantages of this method include the ability to make predictions offline, lower costs (without server maintenance or web hosting), and likely mobile-friendliness. Users will be able to input chest X-ray images through a GUI and receive predictions from the mobile application.

Data answer

The model achieves a recall score as high as 99.47% and an AUC score of 0.95 on test image data, indicating good classification performance while having very low misclassifications on false negatives. At the same time, the accuracy and precision scores are 89.51% and 85.14% respectively, which are also indications of good model prediction reliability.

Business answer

With its good and reliable prediction performance, the developed model meets the need for fast and accurate pneumonia diagnosis with chest X-ray images. This model development addresses the staff shortage in the medical field [3], as well as significantly reducing human interpretation error. [4]

Response to stakeholders

This project and the developed model are open to discussion and review by certified medical professionals, in order to further fine-tune the model to meet real-world application standards, so that it can be applied in many use cases.

End-to-end solution

A simple mobile application has been created using CustomTkinter to deploy the deep learning model, and its functionalities were demonstrated during the presentation. This application, with its simple and intuitive GUI, allows users to quickly input chest X-ray images and receive predictions in a matter of seconds, which will be practical in medical scenarios where speed and time are crucial.

References

[1] H. J. Koo, S. Lim, J. Choe, S. H. Choi, H. Sung, and K. H. Do, “Radiographic and CT Features of Viral Pneumonia,” RadioGraphics: A Review Publication of the Radiological Society of North America, Inc, vol. 38, no. 3, pp. 719-739, Published Online, May 1, 2018. [Online]. Available: https://doi.org/10.1148/rg.2018170048

[2] “How to Read a Chest Xray II: Pneumonia,” Medchrome, June 7, 2015. [Online]. Available: https://medchrome.com/mbbs-exams/how-to-read-a-chest-xray-pneumonia/. [Accessed: 18- March- 2024].

[3] “Artificial Intelligence in Radiology,” Siemens Healthineers, [Online]. Available: https://www.siemens-healthineers.com/medical-imaging/digital-transformation-of-radiology/ai-in-radiology. [Accessed: 18- March- 2024].

[4] L. Berlin, “Faster Reporting Speed and Interpretation Errors: Conjecture, Evidence, and Malpractice Implications,” Journal of the American College of Radiology, vol. 12, no. 9, pp. 894-896, 2015.

[5] L. L. Plesner, F. C. Müller, J. D. Nybing, L. C. Laustrup, F. Rasmussen, O. W. Nielsen, M. Boesen, and M. B. Andersen, “Autonomous Chest Radiograph Reporting Using AI: Estimation of Clinical Impact,” Radiology, vol. 307, no. 3, e222268, Mar. 2023. [Online]. Available: https://doi.org/10.1148/radiol.222268.

[6] F. Pesapane, M. Codari, and F. Sardanelli, “Artificial intelligence in medical imaging: threat or opportunity? Radiologists again at the forefront of innovation in medicine,” European Radiology Experimental, vol. 2, no. 35, 2018.

[7] Z. Campos-Lopez, J. Diaz-Roman, B. Mederos-Madrazo, N. Gordillo-Castillo, J. Cota-Ruiz, and J. Mejia-Muñoz, “Identification of Pneumonia with X-ray Images Using Deep Transfer Learning,” in XLVI Mexican Conference on Biomedical Engineering, IFMBE Proceedings, vol. 96, pp. 32-40, Published Online, Oct. 26, 2023. [Online]. Available: https://doi.org/10.1007/978-3-031-46933-6_4

[8] C. F. Sabottke and B. M. Spieler, “The Effect of Image Resolution on Deep Learning in Radiography,” Radiology: Artificial Intelligence, vol. 2, no. 1, Published Online, Jan. 22, 2020. [Online]. Available: https://doi.org/10.1148/ryai.2019190015

[9] Q. Zheng, M. Yang, X. Tian, N. Jiang, and D. Wang, “A Full Stage Data Augmentation Method in Deep Convolutional Neural Network for Natural Image Classification,” Discrete Dynamics in Nature and Society, vol. 2020, Article ID: 4706576, Published Online, Jan. 11, 2020. [Online]. Available: https://doi.org/10.1155/2020/4706576

[10] M. Nagaraju, P. Chawla, and N. Kumar, “Performance improvement of Deep Learning Models using image augmentation techniques,” in 1197: Advances in Soft Computing Techniques for Visual Information-based Systems, Multimedia Tools and Applications, vol. 81, pp. 9177-9200, Published Online, Jan. 24, 2022. [Online]. Available: https://doi.org/10.1007/s11042-021-11869-x

[11] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” Journal of Big Data, vol. 6, Article number: 60, Published Online, July 6, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0