CPSC 66 Final Report: Peumonia Detection from Chest X-ray Images using Convolutional Neural Networks

Jackson Brosgol JBROSGO1@SWARTHMORE.EDU
Alex Fischmann AFISCHM1@SWARTHMORE.EDU

Abstract

This project aims to address the challenge of detecting pneumonia in chest X-ray images through the application of Convolutional Neural Networks (CNNs) and transfer learning. Ultimately achieving an accuracy surpassing 95%, our study uses a dataset of 5,863 X-ray images sourced from Kaggle, categorized as Normal and Pneumonia. Leveraging TensorFlow for model implementation, we explore various transfer learning models and perform hyperparameter tuning to optimize CNN performance. Accompanying this is a discussion surrounding the nuanced challenges of selecting hyperparameter combinations that align with our objectives, navigating trade-offs inherent in the task at hand. The project also delves into the ethical considerations of deploying such a model in a medical context, scrutinizing potential consequences of false positives and false negatives. Besides the obvious goal of creating a robust model capable of accurate pneumonia detection, we also hope to contribute insights into model behavior and the broader implications of deploying machine learning algorithms in healthcare.

1. Introduction

Pneumonia, a respiratory infection in the lungs, affects millions of children a year. In fact, according to the World Health Organization, pneumonia was responsible for 14% of all deaths of children under 5 years old in 2019. (Organization, 2022) While pneumonia can be deadly if left untreated, the use of antibiotics and other treatments after identification of the illness can prevent this disease from becoming fatal for these vulnerable populations. Thus it is important to be able to diagnose pneumonia properly in patients.

CPSC 66 Machine Learning Proceedings, Swarthmore College, Fall 2023.

Pneumonia is often identified through the use of chest X-rays. This imaging is analyzed by doctors, who look for white spots in the lungs-called infiltrates-that identify an infection (Association; RSN). The goal of this project is to develop a machine learning network that can efficiently and reliably detect pneumonia in chest X-ray images of children.

Over the past decades, modern machine learning algorithms have greatly improved their capacity for pattern recognition in images, most notably Convolutional Neural Networks (CNNs) (O'Shea & Nash, 2015). Our work uses transfer learning and CNNs to detect the presence or absence of pneumonia in chest X-ray images.

2. Methods

2.1. Convolutional Neural Networks

In this project, we employed CNNs as the foundational architecture for the task of pneumonia detection in chest X-ray images. CNNs, designed specifically for image recognition tasks, leverage convolutional layers for hierarchical feature extraction, pooling layers for spatial reduction, and fully connected layers for classification.

2.2. Transfer Learning

A key element of our methodology involved harnessing the advantages of transfer learning with TensorFlow models, specifically MobileNetV2 (Ten, b), InceptionV3 (Ten, a), and Xception (Ten, c). These models are trained on images from ImageNet, a database over over 14 million images (Ima). Transfer learning allowed us to capitalize on pre-trained models, particularly benefiting from the stability of lower-level network layers when transitioning between datasets. In other words, because the low-level parts of CNNs are similar for the vast majority of image recognition tasks, we do not need to retrain them, reducing computational intensity of the task as well as decreasing the amount of data needed to sufficiently train the network.

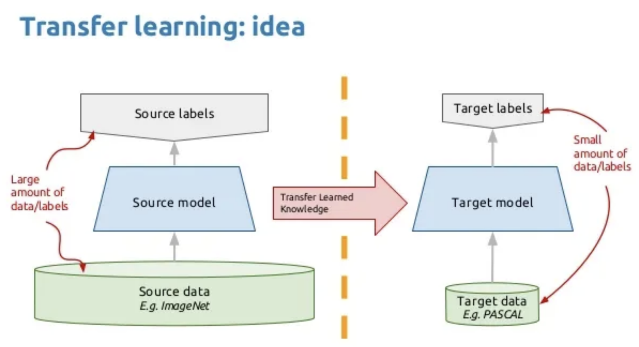

Figure 1. Diagram of Transfer Learning Model (Koehrsen, 2018)

As such, to implement transfer learning in our project, we froze the base layers of the model, then added additional layers on top, which we trained specifically for our task of detecting pneumonia in chest X-rays. *Figure 2* is a diagram of our final network's structure while using MobileNetV2 as the base model. We also introduced a global average pooling layer for spatial summarization and appended a fully connected layer with a sigmoid activation function for binary classification—distinguishing between pneumonia and normal cases. The structures for our networks that levereged other base models—InceptionV3 and Xception—had similar structures, occasionally differing in the shape of some hidden layers.

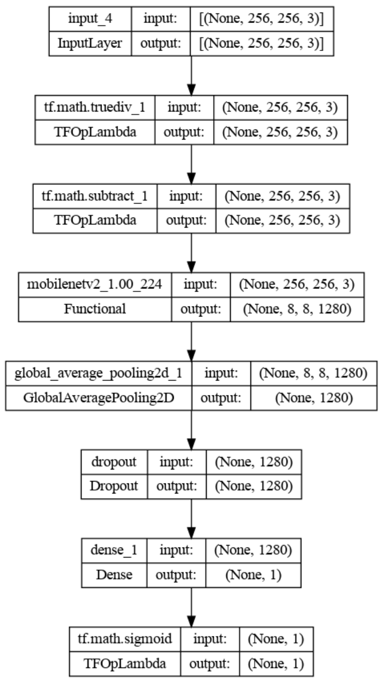
input_4 | input: | [(None, 256, 256, 3)]
InputLayer | output: | [(None, 256, 256, 3)]
tf.math.truediv_1 | input: | (None, 256, 256, 3)
TFOpLambda | output: | (None, 256, 256, 3)
tf.math.subtract_1 | input: | (None, 256, 256, 3)
TFOpLambda | output: | (None, 256, 256, 3)
mobilenetv2_1.00_224 | input: | (None, 256, 256, 3)
Functional | output: | (None, 8, 8, 1280)
global_average_pooling2d_1 | input: | (None, 8, 8, 1280)
GlobalAveragePooling2D | output: | (None, 1280)
dropout | input: | (None, 1280)
Dropout | output: | (None, 1280)
dense_1 | input: | (None, 1280)
Dense | output: | (None, 1)
tf.math.sigmoid | input: | (None, 1)
TFOpLambda | output: | (None, 1)

Figure 2. Network using MobileNetv2 as a base model

2.3. Workflow

Complementing our model architecture, we established a robust pipeline for pre-processing, training, testing, and analysis. One of the primary advantages of our pipeline was its ability for seamless re-execution, as reproducing experiments with ease was particularly beneficial for validating the results of our models, ensuring the reliability of our findings, and reproducing errors while debugging.

The general workflow of our pipeline was the following:

```
For each dataset: # raw data, shuffled data[1]
    For each base model:
        Pre-process data
        Build model # e.g. add pooling layer
        For each dropout value:
            Repeat 3 times:
                Train model
                Test model
                    Output CMs, accuracy, precision, recall
            Average performance metrics from the 3 trials
        Graph comparison of dropouts over all thresholds
```

3. Experiments

3.1. Data

Our data was sourced from Kaggle (Mooney, 2018) and was comprised of 5,863 images of Chest X-rays, 72% of which featured patients with pneumonia and 28% of which were images of non-infected lungs. The images were originally given to us in three folders: train (∼89% of total images), test (∼10% of total images), and validation (∼1% of total images). We first ran experiments using this train/test/validation split. Motivated by the results of this experiment, we then shuffled the data and created our own train (70% of total images), test (20% of total images), and validation (10% of total images) subsets. For pre-processing the data, we used the built-in Keras methods for a given model. This confirmed that the images were correctly formatted, and included steps such as ensuring that our values were within the correct range.

3.2. Analysis Methods

As we ran through each iteration of our experiments, our pipeline served as an important tool for hyperparameter testing by allowing us to run many variations of different combinations of hyperparamaters. Tuning involved analyzing the effect of dropout values (a hyperparameter to

[1]Raw data refers to the data directly from Kaggle with its provided training, validation, and testing datasets. The shuffled data refers to newly randomly constructed new training, validation, and testing datasets with a different split.

combat overfitting) and thresholds across all three models. We ran trials using dropout values of 0.0, 0.2, 0.5, and 0.7 and experimented with thresholds of 0.01, 0.1, .25, 0.5, 0.6, 0.75, 0.9, 0.95, and 0.99. As we trained, tested, and tuned the model, we outputted visualizations to help with our analysis, including:

1. Training vs. validation accuracy/loss during training

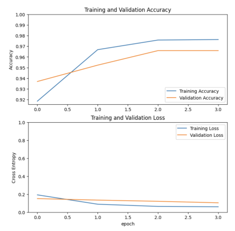

Figure 3. Example: mobilenetV2, dropout = 0.2

2. Confusion matrices for each threshold while testing a given model/dropout/threshold combination

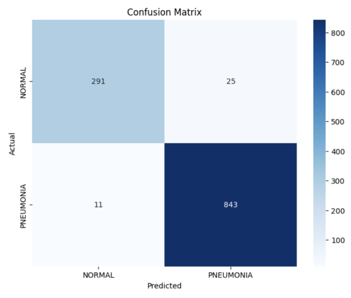

Figure 4. Example: mobilenetV2, dropout = 0.2, threshold = 0.5

3. The impact of different thresholds on accuracy, precision, and recall for a given model/dropout combination

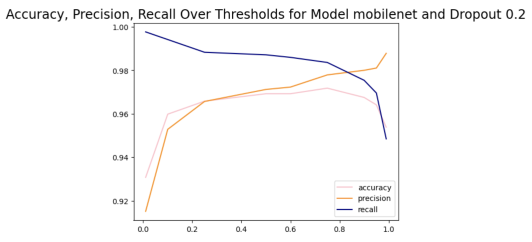

Figure 5. Example: mobilenetV2, dropout = 0.2

4. The comparison of different dropouts on accuracy, precision, and recall for a given model

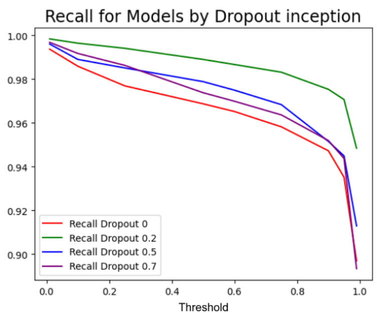

Figure 6. Example of recall curves for the Inception model

4. Results and Discussion

4.1. Impact of Shuffled Datasets

As highlighted earlier, we investigated the influence of altering the train/validation/test split and randomly allocating images to respective categories on our network. The graphs below illustrate comparative analysis between training and testing the MobilenetV2 base model on the original Kaggle dataset and the scenario where the assignment of data for training, validation, and testing was shuffled.

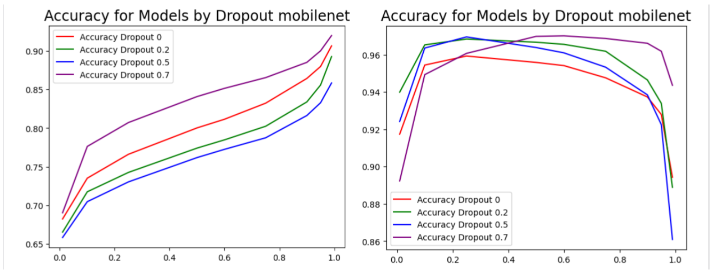

Figure 7. Shuffled verses Unshuffled Accuracy

It becomes evident that the results obtained from utilizing the unaltered Kaggle data exhibit an unusual pattern. Given our utilization of a sigmoid activation for the output node, the threshold serves as the critical value determining whether our model predicts pneumonia or normalcy. If the output value surpasses the threshold, the prediction is pneumonia; if it falls below, it predicts normal. The noteworthy observation that our network's accuracy increased with the threshold suggests an alarming level of confidence in positive predictions. To elucidate, a model trained and tested on this data achieved higher accuracy with a threshold of 0.9995 than with the default threshold of 0.5.

There are several plausible explanations for the observed output. Our initial hypothesis posits that there may be uniformity within the training data and uniformity within the testing data, but significant data differences between the former and later, contributing to this discrepancy. This divergence could be attributed to the origin of the X-rays, sourced from two distinct hospitals. It is conceivable that one hospital contributed more significantly to the training set, while the other was more prevalent in the testing set. Variations in the X-ray machines and sensors between these hospitals might influence characteristics such as the brightness of the images, introducing a potential source of disparity. This, coupled with the non-optimal train/validation/test split of 89% training data and 1% validation increases the potential for overfitting and larger variance in results on the testing set. In such a scenario, the model may become overly sensitive to the nuances of the training data, resulting in the observed outcome where the model exhibits an exceptionally high level of confidence in its positive pneumonia predictions.

As such, it should be clear that random assignment of X-ray images to training, validation, and testing categories on our network, coupled with a split adjustment to 70%, 10%, 20%, respectively, yields markedly superior results. This approach not only attains a higher accuracy (∼97% compared to ∼89%) but also does so consistently at relatively normal thresholds. Furthermore, under this methodology and at a threshold of 0.5, both precision and recall achieve ∼97% when using MobileNetV2 as the base model, whereas utilizing raw data from Kaggle introduces a significant disparity between precision and recall at the same threshold: ∼75% vs. ∼99%, respectively (precision and recall graphs not shown). Motivated by these outcomes, we choose to solely focus on the results from the shuffled data for the remaining analysis.

4.2. Impact of Different Dropouts

In order to identify whether the dropout value had a meaningful impact on the precision, recall and accuracy, we:

1. Created graphs to visually examine the given metrics' performance over different thresholds
2. Conducted paired t-tests of each dropout value against all other values (using an alpha of 0.10).

Our paired t-tests use each dropout value's average accuracy/precision/recall over three trials for thresholds between 0.25 and 0.75 (the thresholds most likely to achieve high performance).

It is also important to note that there is often a trade off between precision[2] and recall[3] for machine learning models. For the purpose of our analysis, we argue that having higher recall (an essential metric, especially for measuring false negatives) takes precedence in determining the best dropout; however, precision, though not our primary focus, remains an important metric, and a hyperparameter that boosts recall may be less optimal if it significantly compromises precision, even with the latter as the more valued metric.

[2]Precision: of the number of predicted pneumonia cases, what percentage were actually pneumonia

[3]Recall: of the number of positive pneumonia cases, what percentage does the model identify

4.2.1. Dropout Values for MobileNetV2

In the evaluation of MobileNetV2, we observe that dropout values of 0.2, 0.5, and 0.7 all exhibit statistically improved accuracy compared to 0.0, as evidenced by paired t-tests yielding p-values of 0.0017, 0.0042, and 0.05, respectively. While a dropout value of 0.7 is not statistically significant different than 0.2 and 0.5 in terms of accuracy, it's considerably lower precision does not justify its higher recall. Therefore, we contend that, for MobileNetV2, dropout values of 0.2 or 0.5 stand out as optimal among the tested range, but we cannot assert the superiority of one over the other in terms of accuracy/precision/recall with an alpha of 0.10.

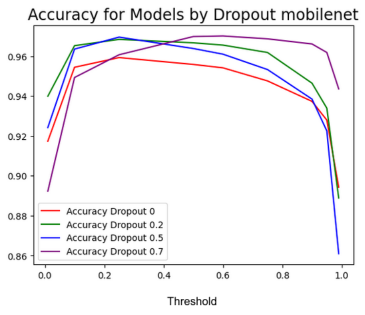

Figure 8. Accuracy of MobileNet

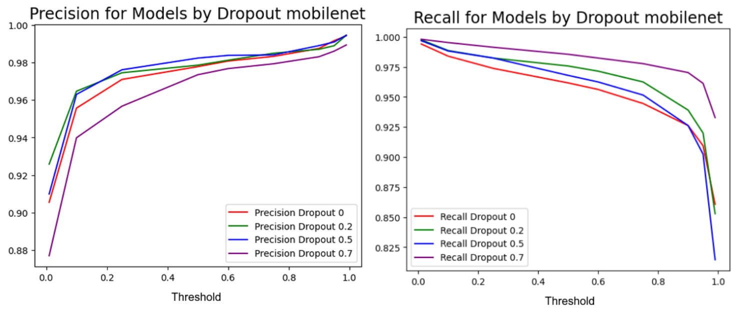

Figure 9. Precision and Recall of MobileNet

4.2.2. Dropout Values for InceptionV3

In the case of InceptionV3, a dropout value of 0.5 appears to be the most optimal among the tested values. It's accuracy is statistically significantly different than 0.0, 0.2, 0.7; the corresponding p-values are 0.0002, 0.077, 0.0084. While a dropout of 0.2 yields higher recall, this advantage is counterbalanced by a significant decrease in precision—an apt illustration of the precision/recall tradeoff discussed at the beginning of this section. This, coupled with observed higher accuracy of 0.5, solidifies its position as the preferred dropout value among those tested.

4.2.3. Dropout Values for XceptionV3

For XceptionV3, examining dropout values reveals that a value of 0.2 is the optimal choice among the tested values. When running t-tests on 0.2, 0.5, and 0.7, there were no statistically significant differences in accuracy observed. However, it is noteworthy that a dropout value of 0.2 demonstrates a statistically higher recall compared to 0.5 and 0.7 (with p-values of 0.01 and 0.06, respectively), all the while maintaining a precision score that is not substantially compromised.

4.2.4. Conclusions from Dropout Experiments

In all three models, the best performance tends to result from dropout values of 0.2 and 0.5. This suggests that dropout can be a helpful method for combating overfitting. However, too much dropout can prevent the model from learning enough information while training.

Finally, it is should be noted that we were limited in the number of distinct dropout values that we could test. For all three models, finding the value that maximizes accuracy while achieving the optimal balance between precision and recall is likely in between 0.2 and 0.5. Given more time, conducting additional trials within this range could be beneficial to ascertain the actual optimal dropout value. This expanded exploration could provide more nuanced insights into the model's performance across a broader spectrum of dropout values within the identified range.

See Appendix for accuracy, precision, and recall graphs.

4.3. Impact of Different Base Models

Table 1. Comparison of Base Models with Dropout = 0.2 and Threshold = 0.5

	MobileNet	Inception	Xception
Accuracy	0.9667	0.9581	**0.9681**
Precision	**0.9785**	0.9553	0.9713
Recall	0.9780	**0.9891**	0.9856

Table 2. Comparison of Base Models with Dropout = 0.5 and Threshold = 0.5

	MobileNet	Inception	Xception
Accuracy	0.9638	0.9658	**0.9678**
Precision	0.9823	0.9743	**0.9884**
Recall	0.9680	**0.9789**	0.9672

Much like the nuanced evaluation involved in determining the optimal dropout value, selecting the best base model for this task proves to be a complex decision, as it requires a careful consideration of various factors. As depicted in *Table 1* and *Table 2*, no model clearly distinguishes itself from the other two[4]. At a dropout of 0.2 and a threshold of 0.5, the highest scores for accuracy, precision, and recall are attributed to different base models. Meanwhile, at a dropout of 0.5, Xception emerges with the highest accuracy and precision but registers the lowest recall—often viewed as the most important metric in medical diagnosis where false negatives carry notable significance.

Furthermore, the models exhibit distinct responses to varying dropout values. To illustrate, in the case of MobileNet, a dropout value of 0.7 yields the highest recall and lowest precision across all thresholds; conversely, for Inception, it is a dropout value of 0.2 that results in the highest recall and lowest precision across all thresholds. This again underscores the complexity of selecting the optimal combination of hyperparameters—in this case the base model, dropout value, and threshold—necessitating a comprehensive cost-benefit analysis of many considerations.

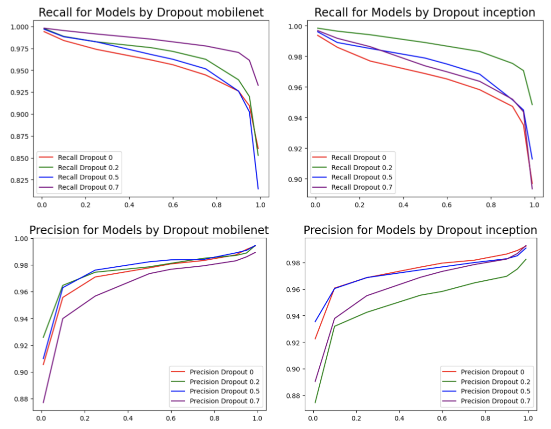

Figure 10. MobileNet verses Inception

To elucidate the previous points, not only is there no model that gives a clear advantage across performance metrics, but the lack of a consistent impact of dropout values on the base models makes it difficult to chose a combination of hyperparameters that best align with all of our objectives. Based on *Table 1* and *Table 2*, one could make the argument that Xception provides the best overall combination, but it is also reasonable to argue in favor of MobileNet and Inception. Further testing is certainly necessary if we wish to make a confident conclusion regarding the best base model

[4]The percentages shown in *Table 1* and *Table 2* were gathered from averaging the accuracy, precision, and recall from three trials of each model/dropout/threshold combo. The bolded numbers emphasize the largest percentage for each performance metric, but we did not test for a statistical difference between each model for a given performance metric.

for pneumonia detection.

5. Societal Implications of Model Deployment

It is important to note that deploying a model for automated pneumonia identification carries significant consequences for individuals' health and well-being. The errors made by such a model have the potential to be not only detrimental but, in some cases, life-threatening. Of particular concern is the exploration of whether our model is more prone to generating false positives or false negatives. Both types of misclassifications pose substantial risks; however, in the medical context, false negatives could be more severe as they might lead to patients with pneumonia being deprived of necessary treatment. As such, those who hope to use machine learning for pneumonia detection should be particularly concerned with the recall of the network. As we illustrated previously, different combinations of hyperparameters can produce higher recall scores (particularly lower thresholds and often lower dropouts). However, it is again important to note that optimizing hyperparameters to maximize recall tends to result in lower precision and accuracy scores. As such, it is important to find the proper balance that ensures the model is sensitive enough to detect true positive cases, minimizing the risk of false negatives, while still maintaining sufficient precision to avoid excessive false positives and maintaining overall accuracy in pneumonia identification.

Furthermore, the scope of our image dataset may not entirely mirror the broader population affected by pneumonia, as the dataset only consists of X-rays of children aged 1-5 from two specific hospitals. Recognizing this limitation prompts a deeper exploration into the generalizability of our model and the potential biases inherent in the dataset. This is especially true considering the meaningful difference in results from the data provided to us and the reshuffled data, suggesting that differences in the characteristics of the X-ray scans can impact how the model functions. Understanding these factors is essential for responsibly deploying machine learning algorithms in a medical context.

Our network has not undergone enough testing to assert that it can outperform doctors from an accuracy, precision, and recall perspective. Due to this, as well as the limited scope of our data, we recommend that, if deployed, this model be used in addition to other methods of diagnosis, rather then as a complete replacement.

In the event that a pneumonia detection model achieves a level of accuracy, precision, and recall suitable for medical diagnosis (i.e. outperforms doctors), it stands to impact both potential patients and doctors. Enhancing diagnostic efficiency and accuracy benefits patients by enabling swift treatment upon receiving test results. Doctors, utilizing the model, can allocate their time more effectively. However, the use of such a model may influence the doctor-patient relationship, as patients may perceive a lack of personalized care and consideration for symptoms from the model compared to their doctor. Finally, the advent of machine learning networks for medical diagnoses intersects with broader shifts in healthcare dynamics. The increasing trend of corporate acquisitions of medical practices, driven by profit maximization, introduces a new layer of complexity. As the need for specialized roles, such as radiologists, diminishes with the implementation of efficient models, it may alter the dynamics between doctors and their corporate overseers.

6. Conclusions

This project explored the effectiveness of Convolutional Neural Networks and transfer learning in classifying chest X-ray images to predict the presence of pneumonia. Our findings demonstrate the challenges inherent in pinpointing a singular optimal base model, dropout value, or threshold, given the diverse outcomes stemming from different combinations. The elusive definition of *optimal* is compounded by the precision/recall tradeoff commonly observed in these models. Nevertheless, our study demonstrates notable achievements, with accuracies, precisions, and recalls consistently exceeding 95% across various model/dropout/threshold combinations. This underscores the potential of Convolutional Neural Networks as a promising method for robust pneumonia detection from chest X-rays.

Future work might include further experimenting with dropout values that are closer in magnitude to the optimal dropouts among those tested for each model. Furthermore, as a result of the difference in results from Kaggle's train/test/validation split and our own reshuffled train/test/split, it may be advantageous to experiment with other splits using a method such as k-fold validation. Finally, further studies could experiment with different base models for transfer learning, as this work was limited to three different base models from TensorFlow's API.

Acknowledgments

Thank you to Professor Mitchell for his guidance on this project.

References

URL https://www.image-net.org/index.php.

Pneumonia. URL https://www.radiologyinfo.org/en/info/pneumonia.

Module: tf.keras.applications.inceptionv3, a. URL https://www.tensorflow.org/api_docs/python/tf/keras/applications/inception_v3/InceptionV3.

Module: tf.keras.applications.mobilenetv2, b. URL https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet_v2/MobileNetV2.

Module: tf.keras.applications.xception, c. URL https://www.tensorflow.org/api_docs/python/tf/keras/applications/xception.

d. URL https://www.tensorflow.org/tutorials/images/transfer_learning#continue_training_the_model.

Association, American Lung. Pneumonia symptoms and diagnosis. URL https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis.

Koehrsen, Will. Transfer learning with convolutional neural networks in pytorch, Nov 2018.

Mooney, Paul. Chest x-ray images (pneumonia), Mar 2018. URL https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia/data.

Organization, World Health. Pneumonia in children, 2022. URL https://www.who.int/news-room/fact-sheets/detail/pneumonia.

O'Shea, Keiron and Nash, Ryan. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

Appendix

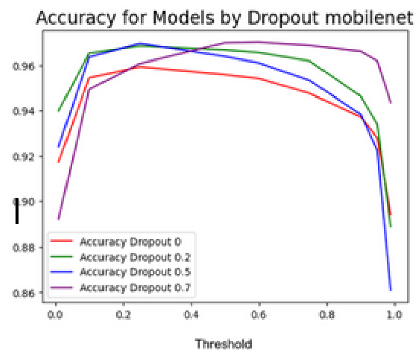

Figure 11. Accuracy of mobilenet

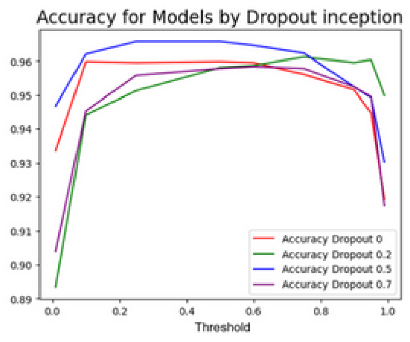

Figure 14. Accuracy of inception

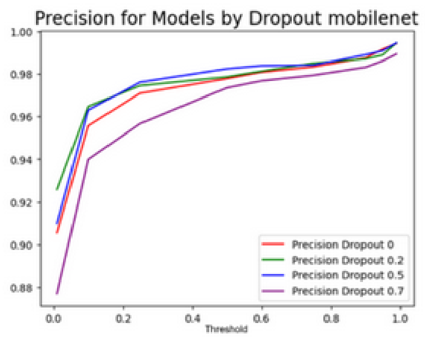

Figure 12. Precision of mobilenet

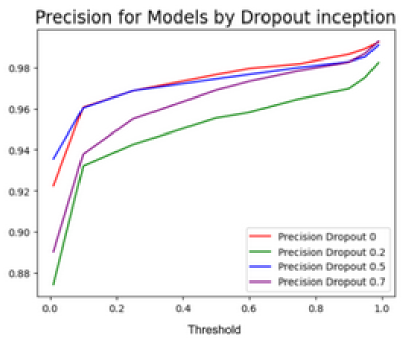

Figure 15. Precision of inception

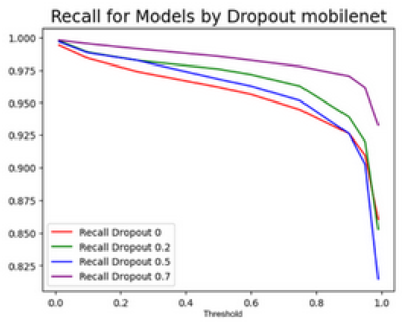

Figure 13. Recall of mobilenet

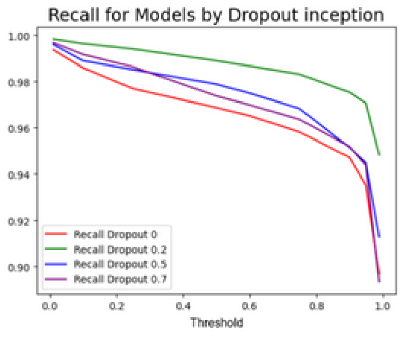

Figure 16. Recall of inception

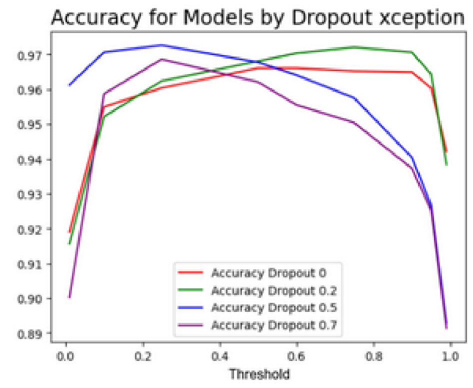

Figure 17. Accuracy of xception

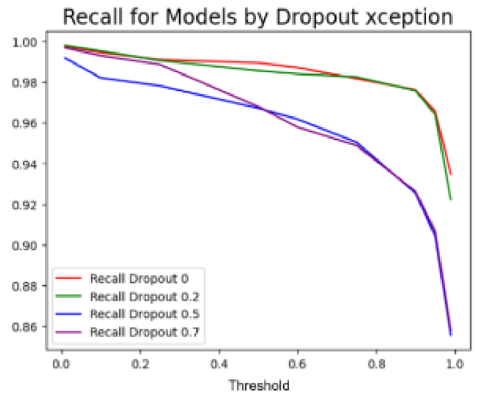

Figure 19. Recall of xception

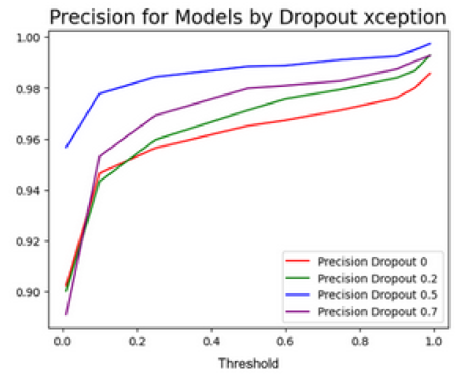

Figure 18. Precision of xception