COVID-19 and Pneumonia Chest X-Ray Classifier Web Application

Ananto Joyoadikusumo
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
ananto.joyoadikusumo@binus.ac.id

Brandon Scott Buana
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
brandon.buana@binus.ac.id

Fernando Liko Marchai
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
fernando.marchai@binus.ac.id

William Martin
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
william.martin001@binus.ac.id

Wilson Wongso
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
wilson.wongso001@binus.ac.id

***Abstract*—**

***Index Terms*—COVID-19, Pneumonia, Convolutional Neural Network, Computer Vision, Deep Learning**

I. Introduction

The COVID-19 pandemic may very soon come to a close with the development and distribution of vaccines. However, testing and diagnosis of the virus is still paramount in combating the calamity of the pandemic. Although most medical institutions have mostly relied on traditional methods of COVID-19 diagnosis (such as Reverse transcription-polymerase chain reaction or RT-PCR), several studies have indeed shown that chest radiography can also be a viable alternative. Features and abnormalities that distinguish chest x-rays of COVID-19 infected patients from uninfected patients can be identified and have been recommended as another diagnosis method [1]. Furthermore, early diagnosis using CXRs also takes less time and has better sensitivity compared to RT-PCR. It should also be noted that CT scans have higher sensitivity to COVID-19 detection compared to CXRs [2]. However, CXRs are much more readily available and affordable around the world, hence our focus in using them instead of the former.

Numerous researchers have experimented in using Deep Learning to diagnose COVID-19 using CXR images, and a large majority of them opted to use pre-trained Convolutional Neural Networks. Unfortunately, most of these studies often have different scales, sources of datasets and use different transfer learning architectures or methods. Our goal in this paper is to compare the various pre-trained models using a single large COVID-19 CXR dataset as a benchmark and find the best performing models. The results of the experiment can undoubtedly assist future studies pertaining COVID-19 CXR detection when selecting the best suited architectures for this task.

Afterwards, the model will be deployed as a web application using Laravel.

II. Related Works

A. The Integrity of Chest X-Ray Diagnoses

With regards to the topic of diagnosing COVID-19, the first factor that must be checked is whether or not using chest x-rays or CXRs is a viable method of diagnosing the virus. Fortunately, studies have shown that radiography results were able to identify specific features of COVID-19 patients [1], [2], thus making CXRs a viable alternative for diagnosis. Sarvamangala and Kulkarni [3] also confirm that professional radiologists were able to successfully diagnose COVID-19 infected patients through CXRs and subsequently recommend the use of neural networks to improve the robustness of COVID-19 detection.

B. COVID-19 Chest X-Ray Datasets

As with any projects related to Deep Learning, data and samples are of paramount importance. Cohen et al. [4] became one of the earliest pioneers in gathering a dataset for COVID-19 lung CXRs, accumulating over 500 CXR images. A later study also combined several CXR datasets with COVID-19 CXRs to allow for multiclass classification in neural network models [5]. Loey et al. [6] also ingeniously used GAN to generate COVID-19 CXR images to combat the lack of training data and improve the accuracies of CNNs.

C. Pre-trained CNN's Performance on Chest Radiography Classification

As we have established in the introduction, many studies have been conducted to find the best possible Deep Learning

models to classify COVID-19 CXR images. For instance, Chowdhury et al. [7] conducted an experiment by feeding a database of 3487 CXRs - consisting of normal, COVID-19 infected, and pneumonia-infected lung images - to a pre-trained CNN model. The model was able to obtain a very high accuracy ($\sim$99%), precision($\sim$99%), sensitivity ($\sim$98%), and specificity ($\sim$99%). Jain et al. [8] improved the research by using a larger dataset and multiple pre-trained CNN architectures (Xception, ResNeXt, Inception V3) yielding an accuracy of 97.97% for detecting COVID-19 infected CXRs. Similarly, other research [9]–[11] also tested the performance of various transfer learning models, although the studies have used different datasets.

D. CNN Architectures Constructed from Scratch

Some studies have also attempted to create their own architecture from scratch or modify existing pre-trained CNN models. For instance, Yoo et al. [12] integrated ResNet18 into a decision-tree classifier structure so that the CNN model will only conduct binary classifications instead of multiclass classifications. Another large-scale study [13] built a CNN architecture from scratch, called COVID-Net, and trained it using their own open-source dataset called COVIDx, which consists of almost 14000 CXR images. A study by Khan et al. [14] modified the Xception architecture and created the CoroNet CNN model. All these studies were able to obtain results that surpass the performance of traditional transfer learning models.

E. Chest X-Ray Enhancements

Image enhancement techniques have been experimented with in the hopes of improving the accuracy of classifying COVID-19 chest x-ray images. Degerli et al. [15] conducted a study with this objective in mind by generating the infection maps and subsequently compiled the largest dataset with 119316 CXR images to a pre-trained CNN model. Qi et al. [16] and Rahman et al. [17] have also attempted to use image enhancements techniques and were able to confirm that enhanced CXR images are able to boost the performance of CNN's when detecting COVID-19 infected lungs.

F. Improving Results with Hyperparameters

Like any other machine learning algorithm, there are hyperparameter values of the deep learning models which we need to tweak and tune on our own as it is not learned by the model during training. This must be performed on a separate validation subset of the dataset we chose, to ensure the generalizability of our model and to avoid overfitting to the training subset. Instead of doing this by hand, we can incrementally improve the hyperparameter choices in an agile manner, such as that of a scrum framework [18].

III. Methodology

A. Product Backlog

We begin by communicating with the product owner to discover the use cases and features of the web application.

Role	Task	Sprint	Size
Visitor	Register	1	5
User	Login	1	5
User	Upload Image	2	10
User	Classify Chest X-Ray Images	2	20
User	Find Nearest Hospitals	3	8
User	Learn More About Algorithm	3	3
User	Get Next Steps and Suggestions	3	3

B. Data Preperation and Preprocessing

For the benchmark dataset, we selected the COVID-19 Radiography Database [19], which contained 3616 COVID-19 images compiled from the Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 DATABASE [20], Novel CoronaVirus 2019 Dataset developed by Joseph Paul Cohen [4], and the BIMCV-COVID19+ dataset [21]. The dataset also contained 10192 normal CXR images obtained from the Chest X-Ray Images (pneumonia) database [22]. Lung opacity and Pneumonia infected CXRs were also present in the dataset, but of course, these are not of our interest. We split the dataset into training, validation, and test sets. The training set, which accounts for 80% of the dataset, contains 2892 COVID-19 images and 8154 normal images. Both the validation and test data contain 362 COVID-19 cases and 1019 normal CXR images. The dataset was selected as it already contained a sizable number of CXR images that, as we see later, were able to produce high-quality results without prolonging the training time of the models. Ideally, a larger dataset containing over 5000 covid CXRs would improve the reliability of the trained models exponentially. Unfortunately, this was not currently available. Another challenge that might pose a problem is the unbalanced dataset classes since there is over twice the number of normal images compared to COVID-infected images. However, even without using techniques to rebalance the dataset (such as resampling per se), all the models were still able to output outstanding results, as we will see later.

Before feeding the images into the model, it was first scaled down to 224 by 224 pixels. We also applied several transformations such as shearing and zooming (both with a 0.2 range), image rotation (up to 10º), and horizontal flipping of the image.

C. Model Selection

We selected 11 different models from the Keras module `keras.applications` (Xception, VGG16, VGG19, ResNet50, ResNet101, ResNet152, InceptionV3, Inception-ResNetV2, DenseNet121, DenseNet169, DenseNet201), all of which have been trained using the ImageNet dataset.

1) VGGNet: The VGG architectures proposed by K. Simonyan and A. Zisserman [23] were designed under the basis of increasing depth, reducing the number of parameters in the convolutional layers, and, hence, improve training time. This was possible due to the abundant use of max-pooling layers specifically placed after each convolutional block. VGGNet is also often characterized by the use of small yet effective 3x3

filters that is able to replicate the receptive fields of larger filters.

2) ResNet: ResNet architectures [24] were widely known for tackling the problem of simple mapping and vanishing gradients using identity connections, which prevents the change of computed gradient values during backpropagation. As a result, ResNets were able to get very deep, up to more than 100 layers, without compromising performance or accuracy. ResNets also utilizes batch normalization to adjust the input layer and mitigate covariate shifts commonly occurring in deep neural networks.

3) Inception: The idea behind the Inception Architectures [25], [26] is the implementation of different kernel sizes within the same layer. The optimal kernel size in an image classification task is often not very obvious to determine. Smaller kernels perform greatly for detecting area-specific features, while larger kernels are better suited for features scattered globally around the image. Therefore, Inception architectures use multiple inception modules (networks in networks) with convolutional layers of varying kernel sizes. Consequently, Inception architectures are not very deep and can get away with a small number of parameters without the expense of accuracy.

4) DenseNet: DenseNets [27] are characterized by the use of concatenation, whereby layers are connected in a feed-forward fashion. Therefore, each proceeding layer will receive an additional input, and essentially the entire feature map, from all preceding layers. Concatenation prevents the vanishing gradient problem, allows features and knowledge to be retained in the deeper layers, reduce the number of channels and parameter, and substantially increase computational efficiency.

D. Model Training

With regards to the training pipeline, we decided to unfreeze all the layers in the architectures since the ImageNet dataset does not contain any chest x-ray images. Therefore, allowing the model to learn the brand new features of a CXR image will hypothetically improve training speeds and accuracy. Each model is added with a Global Average Pooling layer to reduce the spatial dimensions of the tensors, and a 2-unit dense layer with a sigmoid function pertaining to the binary classification task that we are handling. Each model is then compiled using a binary cross-entropy loss function with the Adam optimizer. We also set a learning rate scheduler whereby the learning rate will stay constant for the first 5 epochs before decreasing exponentially. Early stopping with a 10 epoch patience is also implemented. Lastly, a 50 epoch limit to limit the training time. After the model is done training, the best validation accuracy, precision and recall are recorded. The test set is then fed into the model before recording the test accuracy, precision, and recall.

We repeated the training or experimentation process several times with different hyperparameters (learning rate and batch size). A combination of 1e-4, 1e-6 learning rate, and 16, 32 batch size is used.

TABLE I
BATCH SIZE 32, LEARNING RATE 1E-4

Model Name	Num of Epochs	Validation			Test		
		Acc	*Prec*	*Rec*	*Acc*	*Prec*	*Rec*
ResNet50	29	99.64	99.64	99.64	99.28	99.28	99.28
ResNet101	39	**100.0**	**100.0**	**100.0**	99.06	99.06	99.06
ResNet152	41	99.86	99.86	99.86	99.42	99.42	99.42
Xception	21	99.78	99.78	99.78	99.35	99.35	99.35
VGG16	28	99.13	99.2	99.13	99.20	99.28	99.20
VGG19	22	99.13	99.13	99.13	98.41	98.41	98.41
InceptionV3	40	99.71	99.71	99.71	99.57	99.57	99.57
InceptionResNetV2	25	99.57	99.49	99.57	99.42	99.35	99.49
DenseNet121	31	99.71	99.71	99.71	**99.64**	**99.64**	**99.64**
DenseNet169	25	99.64	99.64	99.64	99.49	99.57	99.49
DenseNet201	39	99.71	99.71	99.78	99.49	99.49	99.49

**The best result of each metric is bolded*

TABLE II
BATCH SIZE 16, LEARNING RATE 1E-4

Model Name	Num of Epochs	Validation			Test		
		Acc	*Prec*	*Rec*	*Acc*	*Prec*	*Rec*
ResNet50	42	99.71	99.71	99.71	99.71	99.71	99.78
ResNet101	50	**99.78**	**99.78**	**99.78**	**99.93**	**99.93**	**99.93**
ResNet152	19	98.48	98.55	98.48	92.83	92.71	92.98
Xception	20	99.64	99.64	99.64	99.57	99.57	99.57
VGG16	39	99.49	99.49	99.49	99.13	99.13	99.20
VGG19	36	99.57	99.57	99,57	99.28	99.28	99.28
InceptionV3	34	**99.78**	**99.78**	**99.78**	99.35	99.35	99.35
InceptionResNetV2	32	99.64	99.64	99.64	99.35	99.35	99.35
DenseNet121	30	99.49	99.49	99.49	99.64	99.64	99.64
DenseNet169	27	99.67	99.67	99.67	98.19	98.19	98.19
DenseNet201	43	99.64	99.64	99.64	99.49	99.42	99.49

**The best result of each metric is bolded*

IV. RESULTS

Observing the four result tables presented, most of the models seem to have an equal performance with regards to COVID-19 CXR classification. Both simple and complex CNN architecture does yield similar results. However, ResNet101 was able to consistently output the best results out of all the other pre-trained CNNs, although by only a small margin. In Table 1, for instance, ResNet101 was the only model to achieve 100% accuracy, precision, and recall. Other models, even with varying hyperparameters, were not able to replicate this result. With regards to the possibility of overfitting, we can observe from Table 1 that the ResNet101 architecture still performs incredibly well in the test set. The loss and accuracy graphs in Figure 1 also shows no sign of overfitting since test accuracy does not fall during the end of the training.

The ResNet101 architecture, however, has a relatively longer training time compared to models such as Xception (which converged very quickly with the 1e-4 learning rate). ResNet101 (Table 2) and InceptionResNetV2 (Table 4) were the only two models that reached the 50 epoch limit we set.

Some results that might seem to be anomalous, like the case of ResNet152 in Table 2, were already rechecked. After 3 different runs, the ResNet152 model was constantly converging at an unusual rate and was never able to perform well on the test set. The DenseNet201 model in Table 1, also experienced

TABLE III
BATCH SIZE 32, LEARNING RATE 1E-6

Model Name	Num of Epochs	Validation			Test		
		Acc	*Prec*	*Rec*	*Acc*	*Prec*	*Rec*
ResNet50	36	97.97	97.90	97.97	**98.19**	**98.19**	**98.19**
ResNet101	31	98.26	97.29	98.33	97.03	97.03	96.89
ResNet152	27	**98.41**	**98.19**	**98.41**	97.83	97.83	97.90
Xception	48	95.08	94.79	94.93	93.64	93.64	93.64
VGG16	47	97.18	97,17	96.89	96.52	96.66	96.52
VGG19	38	96.67	96.74	96.74	95.94	96.02	96.02
InceptionV3	38	95.15	95.08	95.08	94.86	94.80	95.08
InceptionResNetV2	36	95.22	95.01	95.08	93.56	93.50	93.70
DenseNet121	33	96.74	96.87	96.45	96.60	96.95	96.52
DenseNet169	47	97.61	97.54	97.47	97.25	97.60	97.10
DenseNet201	46	96.52	97.12	97.54	97.97	97.97	98.04

**The best result of each metric is bolded*

TABLE IV
BATCH SIZE 16, LEARNING RATE 1E-6

Model Name	Num of Epochs	Validation			Test		
		Acc	*Prec*	*Rec*	*Acc*	*Prec*	*Rec*
ResNet50	36	98.48	98.40	98.26	97.90	97.76	97.97
ResNet101	39	**98.77**	**98.48**	**98.70**	98.12	98.05	**98.26**
ResNet152	34	98.55	98.34	98.48	**98.19**	**98.11**	97.97
Xception	43	97.1	96.76	97.25	96.02	95.95	96.02
VGG16	48	97.47	97.25	97.25	96.67	96.53	96.81
VGG19	43	97.61	97.68	97.47	97.25	97.10	97.03
InceptionV3	29	95.73	95.59	95.73	94.13	94.19	93.99
InceptionResNetV2	50	96.16	95.96	96.23	94.71	94.92	94.64
DenseNet121	29	97.18	96.83	97.25	97.61	97.53	97.39
DenseNet169	27	97.54	97.39	97.18	97.83	97.68	97.76
DenseNet201	32	98.55	98.41	98.48	98.04	98.04	98.04

**The best result of each metric is bolded*

similar problems but was finally resolved on the third and final run, which is why the result does not look like an anomaly.

In terms of the performance of each family of CNN architecture, ResNet generally performs the best followed by DenseNet and VGG. Although Inception and Xception were able to perform up to par with the previous three models on certain tests, they had their fair share of underwhelming results.

From the results, it is clear that the 1e-4 learning rate is more suitable to be implemented as a hyperparameter in this task. Results in Tables 1 and 2, for instance, are generally better than the results in Tables 3 and 4. On the other hand, there is only a marginal difference between the results of using a batch size of 32 compared to 16 when using a learning rate of 1e-4. However, when a smaller learning rate of 1e-6 is used, batch size 16 does seem to yield slightly better results.

The best model is then deployed as a web application via Laravel where users can login, upload chest x-ray scan images, and retrieve the predictions. Additional resources such as the nearest hospital, next steps and suggestions, as well as learning more about the classification algorithm are similarly provided.

V. CONCLUSION

Out of all the established state-of-the-art pre-trained CNNs, ResNet, specifically ResNet101, has been proven to be the best-suited architecture for differentiating COVID-19 CXRs from healthy/normal CXRs. Our experiments also found that smaller or simpler architectures, such as Xception, were able to perform to a similar caliber as the more complex pre-trained CNNs, like DenseNet201, while benefitting extensively with the much faster training time. Therefore, these models can be a reasonable alternative if larger datasets were to be used or limitations of hardware are present. In the context of hyperparameters, larger learning rates were shown to significantly improve model performances in the context of our training pipeline.

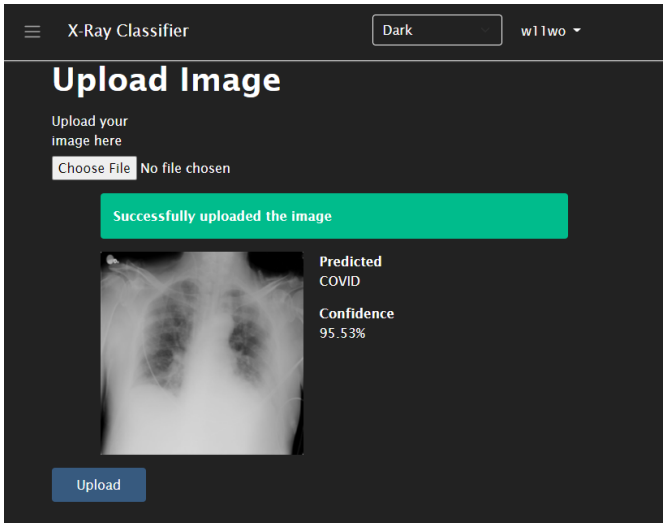

Fig. 1. Classifier Web Application.

REFERENCES

[1] A. Jacobi, M. Chung, A. Bernheim, and C. Eber, "Portable chest x-ray in coronavirus disease-19 (covid-19): A pictorial review," *Clinical imaging*, vol. 64, pp. 35–42, 2020.

[2] H. Y. F. Wong, H. Y. S. Lam, A. H.-T. Fong, S. T. Leung, T. W.-Y. Chin, C. S. Y. Lo, M. M.-S. Lui, J. C. Y. Lee, K. W.-H. Chiu, T. W.-H. Chung *et al.*, "Frequency and distribution of chest radiographic findings in patients positive for covid-19," *Radiology*, vol. 296, no. 2, pp. E72–E78, 2020.

[3] D. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary intelligence*, pp. 1–22, 2021.

[4] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "Covid-19 image data collection: Prospective predictions are the future," *arXiv preprint arXiv:2006.11988*, 2020.

[5] S. Tabik, A. Gómez-Ríos, J. L. Martín-Rodríguez, I. Sevillano-García, M. Rey-Area, D. Charte, E. Guirado, J.-L. Suárez, J. Luengo, M. Valero-González *et al.*, "Covidgr dataset and covid-sdnet methodology for predicting covid-19 based on chest x-ray images," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3595–3605, 2020.

[6] M. Loey, F. Smarandache, and N. E. M Khalifa, "Within the lack of chest covid-19 x-ray dataset: a novel detection model based on gan and deep transfer learning," *Symmetry*, vol. 12, no. 4, p. 651, 2020.

[7] M. E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. Al Emadi *et al.*, "Can ai help in screening viral and covid-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020.

[8] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, "Deep learning based detection and analysis of covid-19 on chest x-ray images," *Applied Intelligence*, vol. 51, no. 3, pp. 1690–1700, 2021.

[9] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. J. Soufi, "Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning," *Medical image analysis*, vol. 65, p. 101794, 2020.

[10] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, pp. 1–14, 2021.

[11] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network," *Applied Intelligence*, vol. 51, no. 2, pp. 854–864, 2021.

[12] S. H. Yoo, H. Geng, T. L. Chiu, S. K. Yu, D. C. Cho, J. Heo, M. S. Choi, I. H. Choi, C. Cung Van, N. V. Nhung *et al.*, "Deep learning-based decision-tree classifier for covid-19 diagnosis from chest x-ray imaging," *Frontiers in medicine*, vol. 7, p. 427, 2020.

[13] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific Reports*, vol. 10, no. 1, p. 19549, Nov 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-76550-z

[14] A. I. Khan, J. L. Shah, and M. M. Bhat, "Coronet: A deep neural network for detection and diagnosis of covid-19 from chest x-ray images," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105581, 2020.

[15] A. Degerli, M. Ahishali, M. Yamac, S. Kiranyaz, M. E. Chowdhury, K. Hameed, T. Hamid, R. Mazhar, and M. Gabbouj, "Covid-19 infection map generation and detection from chest x-ray images," *Health Information Science and Systems*, vol. 9, no. 1, pp. 1–16, 2021.

[16] X. Qi, L. G. Brown, D. J. Foran, J. Nosher, and I. Hacihaliloglu, "Chest x-ray image phase features for improved diagnosis of covid-19 using convolutional neural network," *International journal of computer assisted radiology and surgery*, vol. 16, no. 2, pp. 197–206, 2021.

[17] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan *et al.*, "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in biology and medicine*, vol. 132, p. 104319, 2021.

[18] M. S. Rahman, E. Rivera, F. Khomh, Y.-G. Guéhéneuc, and B. Lehnert, "Machine learning software engineering in practice: An industrial case study," 2019.

[19] T. Rahman, M. Chowdhury, and A. Khandakar, "COVID-19 Radiography Database," 03 2021. [Online]. Available: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[20] SIRM, "COVID-19: caso 115," 05 2021. [Online]. Available: https://sirm.org/category/senza-categoria/covid-19/

[21] M. de la Iglesia Vayá, J. M. Saborit, J. A. Montell, A. Pertusa, A. Bustos, M. Cazorla, J. Galant, X. Barber, D. Orozco-Beltrán, F. García-García, M. Caparrós, G. González, and J. M. Salinas, "Bimcv covid-19+: a large annotated dataset of rx and ct images from covid-19 patients," 2020.

[22] Radiological Society of North America, "RSNA Pneumonia Detection Challenge — Kaggle," 01 2022. [Online]. Available: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[27] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.