Classification of Chest X-Ray Images with Deep Neural Networks for Detecting COVID-19

Inês Filipe

Instituto Superior Técnico

June 2022

Abstract

Chest X-ray imaging techniques are commonly used in patients to ascertain their status and diagnose respiratory or heart conditions. This work addresses the task of classifying accurately chest X-ray images for COVID-19 detection, differentiating them from both images from healthy patients and images from patients with pneumonia, experimenting with the Swin Transformer and the ConvNeXT architectures on a modified version of the COVIDx CXR-3 dataset. The results show that the task of classifying chest X-rays is complex. While the results obtained were relatively satisfactory, with an accuracy over 80%, there were no significant improvements in results between our baseline models, conventional convolutional neural networks such as ResNet50, VGG16, DenseNet121, and the studied models. The ConvNeXT model did provide a slight performance improvement in accuracy, macro precision, macro recall, weighted precision and weighted recall metrics, however we considered the difference to not be statistically significant to claim that the studied model presents itself as an improvement on conventional architectures for the task.

1 Introduction

The coronavirus disease (COVID-19) is a contagious infection caused by the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). This disease, firstly identified in December 2019, in Wuhan, China, has since spread worldwide, originating the ongoing pandemic.

Currently, there are a few tests available to diagnose COVID-19, the most notable one being the Reverse Transcription-Polymerase Chain Reaction (RT-PCR) test. However, for diagnosing the infection, the RT-PCR test is complex and expensive, and may be of limited availability. It is then necessary to find alternative diagnostic techniques to identify infected patients, especially in a disease with high infection rates.

Chest X-rays (CXR) are taken to ascertain the condition of the lungs, heart, and chest wall, and they are common diagnostic tools in respiratory diseases like influenza and pneumonia, at the same time also being relatively cheap. As such, with COVID-19 being a disease that affects the respiratory system and causes pneumonia, it is logical that chest X-rays can be used as part of clinical protocols for diagnosis: chest X-ray images are collected from patients suspected to have the coronavirus disease, analyzed by radiologists and, if evidence is found, other tests can be performed.

However, given that CXR of infected people have diverse characteristics, the diagnosis of COVID-19 requires expert radiologists to analyze the images. In this context, automatic image classification methods can be valuable for obtaining faster diagnosis, especially with the surge of infected patients. Modern methods based on deep neural networks are already successful at identifying pneumonia from X-ray images, rivalling expert radiologists.

Recent studies (Ilyas et al., 2020; Shi et al., 2021; Ulhaq et al., 2020) have reported the use of deep neural networks for diagnosing COVID-19, and although discriminating between COVID-19 and other types of pneumonia remains a challenging problem, automatic classification methods can be used to allocate resources more effectively (e.g. to prioritize the selection of images to be analyzed by expert radiologists, or to prioritize patients for RT-PCR testing), especially for understaffed and/or underfunded hospitals.

To address this issue, we aim to explore the use of transformer and deep convolutional neural network architectures to analyze chest X-ray images and discriminate between healthy patients, patients with COVID-19, and patients diagnosed with other types of pneumonia. This would be a valuable tool in situations where other tests are not readily available, and where radiologists would otherwise be overworked. To accomplish this, we will use deep learning techniques integrating state-of-the-art developments in transformer architectures, and convolutional models based on transformer architectures for computer vision. The models were implemented in Python, and the source code is available on GitHub[1].

[1] `https://github.com/inesfilipe/COVIDSwinConvNeXT`

Training and testing was performed with a publicly available dataset, the COVIDx CXR-3 (Wang and Wong, 2020) dataset, a database of thousands of CXR images of COVID-19 cases, along with healthy and viral pneumonia images.

This work is organized as follows: Section 2 reviews related work pertaining to image classification in the context of COVID-19 diagnosis. Section 3 details the models, introduces the dataset and the modifications performed, training methods and metrics employed. Sections 4 and 5 present and discuss the results obtained, respectively. Section 6 presents the conclusions of this project, and future work that can be performed.

2 Related Work

This section explores related work in the field of COVID-19 detection on chest X-rays. Firstly, it presents COVID-19 detection that leverages conventional neural network models, followed by works that created or adapted new models for this task.

2.1 Conventional Convolutional Neural Network Models for COVID-19 Detection

Since the beginning of the pandemic, there has been extensive work on techniques to automatically diagnose COVID-19 using CT scans and chest X-ray images. One of the approaches was to directly use known architectures for image classification, taking advantage of transfer learning, such as in Apostolopoulos and Mpesiana (2020), Narin et al. (2021), and Minaee et al. (2020).

Apostolopoulos and Mpesiana (2020) used networks that obtained satisfactory results with the ImageNet dataset: VGGNet (Simonyan and Zisserman, 2015), Inception (Szegedy et al., 2016), Inception-ResNet (Szegedy et al., 2016), Xception (Chollet, 2017), and MobileNet (Howard et al., 2017), with ReLU as activation function and a set amount of untrainable layers (layer cutoff), maintaining the weights from the original model trained with the ImageNet dataset.

Narin et al.'s work also evaluated CNNs with transfer learning for COVID-19 prediction. The CNNs used were ResNet50, ResNet101, ResNet152, Inceptionv3 and Inception-ResNetv2, also pre-trained with the ImageNet dataset, and optimizing the cross-entropy loss function with the ADAM optimizer (Kingma and Ba, 2017).

While Apostolopoulos and Mpesiana (2020) and Narin et al. (2021) used small datasets created from publicly available repositories (Cohen et al., 2020; Kermany et al., 2018; Wang et al., 2017), the work by Minaee et al. analyzes the performance of ResNet18, ResNet50 (as in Narin et al. (2021)), SqueezeNet (Iandola et al., 2016), and DenseNet161 for COVID-19 detection on the COVID-Xray-5k dataset, created by the authors, with chest X-ray images (of both COVID-19 and non-COVID-19 patients) from a publicly available repository (Cohen et al., 2020) and re-labeled by an expert radiologist, and non-COVID-19 images from the ChexPert (Irvin et al., 2019) dataset. This work also trains the models with the cross-entropy loss function and the ADAM optimizer, fine-tuning only the last layer of the networks was fine-tuned for the task.

Additionally, Minaee et al. employed a technique from Zeiler and Fergus (2013) to visualize the results of the prediction, by occluding squares of pixels in the original image, and verifying whether or not the model classified the X-ray as COVID-19 positive. If it no longer classified a sample as COVID-19 positive, the occluded pixels were considered relevant for COVID-19 detection, and irrelevant otherwise.

All of the studies mentioned suffer from the same flaw regarding the datasets: the datasets have a small number of samples, especially when compared to datasets such as the ImageNet, the number of COVID-19 samples was low compared to samples of other classes, and consequently the class imbalance was large. Such issues are expected due to the nature of the problem: it is not easy to obtain anonymized labeled chest X-ray images. Minaee et al. attempted to mitigate the issue with data augmentation, flipping, rotating, and distorting the images.

2.2 Recent Models Created or Adapted for COVID-19 Detection

One of the earliest models tailored for this task was the COVID-Net from Wang and Wong (2020), which makes use of what the authors call the PEPx module, and argue that it allows for great representational capacity while being computationally efficient. The module and the architecture were obtained by machine-driven exploration, starting with an initial prototype and using generative synthesis (Wong et al., 2018) to identify the optimal architecture. Like other models mentioned previously, the COVID-Net was pre-trained on the ImageNet dataset, and fine-tuned with ADAM optimizer on the COVIDx dataset, created by the authors by compiling multiple publicly available COVID-19 datasets. Additionally the authors applied data augmentation methods such as translation, rotation, horizontal flip, and intensity shift, and used balanced training batches to increase the probabability of images of different classes being present in each batch. Furthermore, the authors tried to address the problem of transparency and explainability of the results with GSInquire (Lin et al., 2019), an

explainability method where an inquisitor ($\mathscr{I}$) and a generator ($\mathscr{G}$) pair work together, with $\mathscr{G}$ generating new networks, that $\mathscr{I}$ analysing them.

Another architecture created for COVID-19 chest X-ray classification is CoroNet (Khan et al., 2020), i.e. a CNN based on Xception (Chollet, 2017) with modifications, which was first pre-trained on the ImageNet dataset, to avoid overfitting due to the small size of the training dataset, and fine-tuned with ADAM optimizer and 4-fold cross-validation on a combination of two publicly available datasets, one of them being (Cohen et al., 2020). To mitigate the effects of the class imbalance, the authors chose to perform random undersampling on the classes with larger number of samples.

Mamalakis et al. (2021) combined the ResNet and DenseNet architectures into a pipeline called DenResCov-19 to diagnose whether a patient has COVID-19, tuberculosis, pneumonia, or is healthy, first testing the performance of different variants of the architectures pre-trained on ImageNet with cross-entropy as loss function, followed by combining the models with the best performance and concatenating their outputs into an input for fully-connected layers using softmax for classification. The authors used three publicly available open-source collections of chest X-ray images (Kermany et al., 2018; Cohen et al., 2020; Jaeger et al., 2014) and created 4 different datasets, and explored the performance of DenResCov-19 on all 4 datasets. The images in this study also underwent pre-processing such as noise removal and normalization, and data augmentation techniques such as rotation, width shift, height shift, and ZCA whitening (Koivunen and Kostinski, 1999).

Le Dinh et al. (2022) tested 5 networks on a dataset that combined a modified version of the COVIDx CXR-3 dataset, an updated version of the COVIDx dataset by Wang and Wong (2020), and a pneumonia/healthy dataset by Kermany et al.: ResNet50 (He et al., 2015), DenseNet121 (Huang et al., 2018), Inception (Szegedy et al., 2015), Swin Transformer (Liu et al., 2021), and Hybrid EfficientNet-DOLG (Henkel, 2021). The authors employed early stopping for training, the ADAM optimizer, and data augmentation techniques, with the authors concluding that the Hybrid EfficientNet-DOLG was the best performing model of the set.

3 Methods

The goal of this thesis is to analyze chest X-ray images, and discern if they correspond to a healthy patient, a positive COVID-19 diagnosis, or a positive pneumonia diagnosis. For this purpose we evaluate the performance of some neural network architectures, and compare them to baselines to better assess if they present any improvements.

The neural networks we evaluate are a transformer based on the ViT (Vaswani et al., 2017), and a convolutional neural network with improvements based on characteristics presented by vision transformers. We expect that using more recent architectures and developments in computer vision, the results obtained will also be an improvement when compared to the baselines.

3.1 Baselines

When creating baselines, we chose conventional neural network models used in other studies for COVID-19 detection, namely ResNet (He et al., 2015), VGGNet (Simonyan and Zisserman, 2015), and DenseNet (Huang et al., 2018).

The ResNet variant we use is ResNet50, which consists of 4 stages, each with 3, 4, 6, 3 residual bottleneck blocks respectively.

The VGGNet variant we use is VGG16, which has 3 convolutional layers instead of 4 convolutional layers between pooling operations, thus having 3 less convolutional layers in total than the VGG19 suggested by Simonyan and Zisserman.

The DenseNet variant we use is DenseNet121, which is composed of 4 stages, each with a defined number of dense blocks (6, 12, 24, and 16, respectively).

The baseline models used ReLU (Nair and Hinton, 2010) as the activation function, softmax for classification and cross-entropy as loss function, and ADAM (Kingma and Ba, 2017) as optimizer.

3.2 Swin Transformer

The first model we experimented on is the Swin Transformer (Liu et al., 2021), a transformer architecture based on the original ViT (Vaswani et al., 2017). The authors sought to solve some issues of the original transformer architecture, such as the fact that all same-sized patches are inadequate for capturing objects of different sizes, and the quadratic complexity of the attention mechanism, by starting with smaller patches and merging them along the layers, and calculating the self-attention with a formula that reduces complexity to linear on the image size.

The authors present multiple variations of the architecture, and the one we will study is Swin-T, with $C = 96$, and where each of the 4 stages has $[2, 2, 6, 2]$ Swin transformer blocks, the patches are 4×4 pixels and the windows span 7×7 patches.

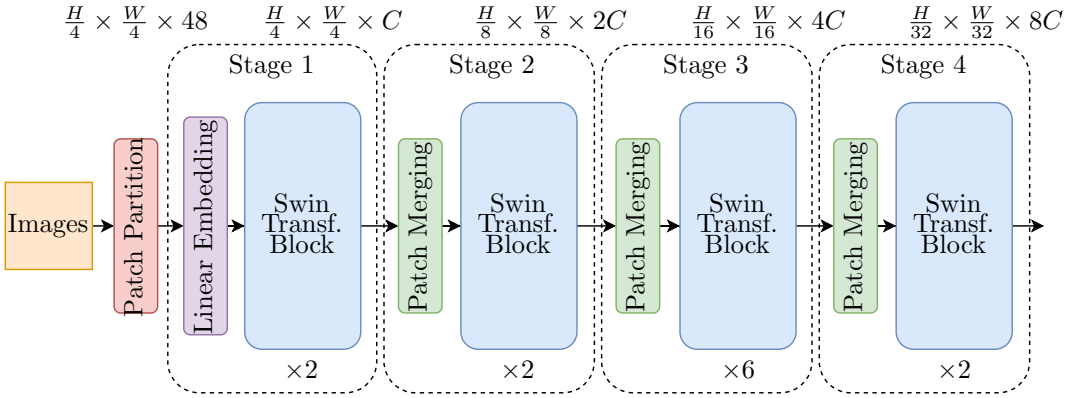

Figure 1: Swin Transformer Architecture (Swin-T variant).

Similarly to the baseline models, the transformer used softmax for classification and cross-entropy as loss function. However, instead of ReLU, the activation function used is GELU (Hendrycks and Gimpel, 2016), an activation function based on the cumulative distribution function of normal distribution, and the optimizer is ADAMW (Loshchilov and Hutter, 2017), a modified version of the ADAM optimizer that decouples the regularization in the ADAM optimizer from the step calculations.

3.3 ConvNeXT

The ConvNeXT (Liu et al., 2022) is a model created with a ResNet (i. e. ResNet50) as starting point, with modifications performed to improve the obtained results than both previous convolutional networks and vision transformers like Swin, coupling the modern approaches of the vision transformers with the natural advantage of convolutional neural networks' inductive biases.

The first modification the authors made to the model itself was alter the block ratio in each stage from $[3, 4, 6, 3]$ to $[3, 3, 9, 3]$. Then the ResNet stem cell was replaced by a "patchify" layer with a kernel with non-overlapping convolution. Furthermore, the residual block also suffered modifications: the convolution operations in the residual blocks were altered to depthwise convolutions, similar to the self-attention operations which are also calculated per channel, and the kernel size was increased; the bottleneck was inverted, with the block being now composed of a 7×7 depthwise convolutional layer, and two 1×1 convolutional layers; the normalization and activation functions were changed and their usage reduced - with Layer Normalization instead of Batch Normalization between the depthwise and 1×1 layer of a block and the GELU activation function between the two 1×1 layers, instead of ReLU activation.

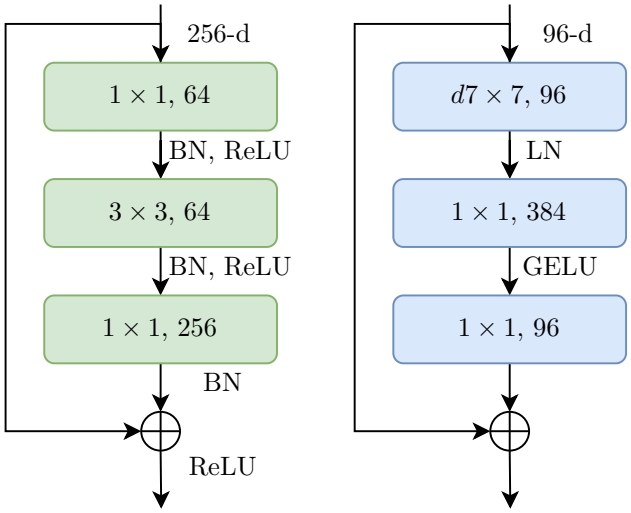

Figure 2: On the left, the ResNet bottleneck block. On the right, the ConvNeXT inverted bottleneck block.

The last changes performed by the authors are the addition of a convolution downsampling layer between

stages, simulating the dimension halving after patch merging in each Swin transformer stage.

Similarly to the Swin Transformer, the authors of ConvNeXT also present multiple versions of the model, and the one we will evaluate is ConvNeXT-T, with $C = (96, 192, 384, 768)$, and where each of the 4 stages has $[3, 3, 9, 3]$ residual blocks.

As in the Swin Transformer model, we used softmax for classification and cross-entropy as loss function. Additionally, we also used GELU as the activation function and ADAMW optimizer.

3.4 Training Methods

All of the models were pretrained on the ImageNet dataset, and fine-tuned with the COVIDx CXR-3 dataset (Wang and Wong, 2020). Fine-tuning was performed for all models for 20 epochs.

Additionally, we used data augmentation and pre-processing methods, namely rotation, horizontal flip, random drop, and normalization.

3.5 COVIDx CXR-3 Dataset

The main dataset that will be used for both training and testing the model is the COVIDx CXR-3 dataset Wang and Wong (2020), available on Kaggle. The dataset is comprised of 30,530 chest X-rays, combined from various sources, of COVID-19, pneumonia, and healthy patients.

	Train			**Test**		
Class	COVID-19	Pneumonia	Normal	COVID-19	Pneumonia	Normal
ActualMed	25	0	0	0	0	0
BIMCV-COVID19+	200	0	0	0	0	0
Cohen	270	52	0	0	0	0
Figure1	24	0	0	0	0	0
RICORD	896	0	0	200	0	0
RSNA	0	5503	8085	0	100	100
SIRM	943	0	0	0	0	0
Stony Brook	14132	0	0	0	0	0
Total/Class	16490	5555	8085	200	100	100
Total	30130			400		

Table 1: Dataset sample count per class and data source, with Wang and Wong (2020)'s original train/test split.

Le Dinh et al. (2022) performed experiments on a modified version of this dataset, and on their best performing model (Hybrid EfficientNet-DOLG) obtained at least 0.95 in macro and micro-averaged precision, recall, and F1-score.

Wang and Wong provide metadata containing patient ID, filename, class, and data source, in a pair of suggested train/test split files. However, after correct analysis, some images were incorrectly labelled, i.e. in COVID-19 datasets, the images were labelled as "positive" instead of "COVID-19". After re-labelling, we also remarked that the suggested train/test split was 98.69%/1.31%. We deemed that the test split was too small for our purposes, and reorganized the split to 89.14%/10.86%.

To balance the size of each split, the images from the Cohen, Ricord, and RNSA datasets were transferred in its entirety to the test split. Our thinking was that, if our models are able to correctly learn what differentiates images from different images, without relying on confounders, then it should be able to generalize to data from sources it hasn't seen while training. Additionally, we further transferred images from the RSNA dataset, until we achieved a train/test split close to 90/10.

3.6 Metrics

Evaluation methods are essential in any machine learning project. They not only allow us to assess a neural network's performance, but also allow us to compare different models, and decide which ones are more appropriate for certain tasks. Using different evaluation metrics is important as well, since different models will have different results for different metrics, and it is the combination of all metrics, or the combination of specific metrics that are more important for a task, that indicate which are the better models for the task.

One of the metrics to be considered is the accuracy of the model. Accuracy is the fraction of correct predictions made by a model. Formally, its definition for a multiclass model is:

	Train			Test		
Class	COVID-19	Pneumonia	Normal	COVID-19	Pneumonia	Normal
ActualMed	25	0	0	0	0	0
BIMCV-COVID19+	200	0	0	0	0	0
Cohen	0	0	0	270	52	0
Figure1	24	0	0	0	0	0
RICORD	0	0	0	1096	0	0
RSNA	0	4723	7166	0	880	1019
SIRM	943	0	0	0	0	0
Stony Brook	14132	0	0	0	0	0
Total/Class	15324	4723	7166	1366	932	1019
Total	27213			3317		

Table 2: Dataset sample count per class and data source, rearranged for our experiments.

$$\text{accuracy} = \frac{\sum_{i=1}^{k} \frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i}}{k}, \tag{1}$$

where k is the number of classes, tp_i is the number of true positive cases of class i detected by the model, tn_i is the number of true negative cases of class i detected, fp_i is the number of false positive cases of class i detected by the model, and fn_i is the number of false negative cases of class i detected.

Other metrics often used are precision (or positive predictive value), recall (or sensitivity), and F-score. Informally, for a given class, the precision corresponds to the amount of positive predictions that were correct, the recall corresponds to the number of actual positive cases that were detected, and the F-score is a combination of the precision and recall metrics. The F-score generally used is F_1, which is a harmonic mean of the precision and recall. The formulas for these metrics are

$$\text{precision}_i = \frac{tp_i}{tp_i + fp_i}, \tag{2}$$

$$\text{recall}_i = \frac{tp_i}{tp_i + fn_i}, \tag{3}$$

$$\mathrm{F}_1^i = \frac{2 \times \text{precision}_i \cdot \text{recall}_i}{\text{precision}_i \cdot \text{recall}_i} \tag{4}$$

where i is the class for which the metrics are being calculated, tp_i is the number of true positive cases of class i detected by the model, tn_i is the number of true negative cases of class i detected, fp_i is the number of false positive cases of class i detected by the model, and fn_i is the number of false negative cases of class i detected. It is also possible to combine these class metrics, as follows

$$\text{precision}_M = \frac{\sum_{i=1}^{k} \text{precision}_i}{k}, \tag{5}$$

$$\text{recall}_M = \frac{\sum_{i=1}^{k} \text{recall}_i}{k}, \tag{6}$$

$$\mathrm{F}_1^M = \frac{\sum_{i=1}^{k} \mathrm{F}_1^i}{k}, \tag{7}$$

where k is the number of classes.

Note that, according to the formulas provided, all of items have equal weight, which means classes with more elements have more influence in the metric. It is possible to calculate these metrics by giving the same importance to each class, independently of the number of elements each contains:

$$\text{precision}_{weighted} = \frac{\sum_{i=1}^{k} \#i \cdot \text{precision}_i}{\sum_{i=1}^{k} \#i}, \tag{8}$$

$$\text{recall}_{weighted} = \frac{\sum_{i=1}^{k} \#i \cdot \text{recall}_i}{\sum_{i=1}^{k} \#i}, \tag{9}$$

$$\text{F}_1^{weighted} = \frac{\sum_{i=1}^{k} \#i \cdot \text{F}_1^i}{\sum_{i=1}^{k} \#i}, \tag{10}$$

In the previous expressions, *weighted* means that this metric gives equal weight to each class, while a macro-level metric, i.e. a metric that gives equal weight to all items, is represented with an M, that we can see in the expressions 5, 6, and 7.

The aforementioned metrics are often calculated based on a confusion matrix, i.e. a table where the columns and rows correspond to the classes, and each column contains the total number of the actual samples belonging to a class, while each row contains the total number of samples predicted to belong to a class.

4 Experiments

This section introduces the results obtained in our experiments, starting with the training and test accuracy, followed by the confusion matrices and associated precision, recall and F1 scores per class, for the baseline and studied models. Finally, it addresses macro and weighted metrics of the baseline and studied models.

4.1 Training and Testing Accuracy

The first models we will analyze are the baselines: ResNet50, VGG16, and DenseNet121.

Model	**Train Accuracy (%)**	**Test Accuracy (%)**
ResNet50	95.5690	88.2122
VGG16	95.5625	88.0916
DenseNet121	95.6508	86.3732
Swin-T	95.0022	85.7401
ConvNeXT-T	96.8073	88.3328

Table 3: Accuracy results of baselines and studied models.

From Table 3, we can see that the results are not significantly different between each model, with the DenseNet having the best accuracy in the training set, and the ResNet having the best accuracy in the test set.

As for the studied models, the ConvNeXT has both the best training and testing accuracy overall, while the Swin Transformer has the lowest testing accuracy overall. On the test dataset, the DenseNet121 and Swin-T slightly underperform in accuracy, with a difference of 2-3% to the other models.

We also remark that the accuracy on the test set decreases by 7%-11% when compared to the training accuracy for all models.

4.2 Confusion Matrices, Precision, Recall, and F1 Score per Class

The confusion matrices (Table 4) provide insight on each baseline model's performance on the test dataset, especially when the values are converted into precision, recall, and F1 score for each class.

According to these results (Table 5), among the baselines we can say that, for the COVID-19 class, the DenseNet121 has greater precision, and the VGG16 has greater recall; for the Normal class, the ResNet50 has greater precision, while the DenseNet121 has greater recall; and for the Pneumonia class, the ResNet50 has greater precision and recall.

However, when compared to the studied models, the Swin Transformer has the greatest precision for the COVID-19 class, and the greatest recall for the Normal class. As for the ConvNeXT, it has larger recall for the pneumonia class than the ResNet50 model.

4.3 Macro and Weighted Metrics

The metrics per class, after averaged per class (macro) and per sample (weighted), can be used to evaluate the overall performance of the models.

ResNet50	COVID-19	Normal	Pneumonia
COVID-19	1140	45	181
Normal	6	989	24
Pneumonia	58	77	797

VGG16	COVID-19	Normal	Pneumonia
COVID-19	1126	43	197
Normal	6	965	48
Pneumonia	47	54	831

DenseNet121	COVID-19	Normal	Pneumonia
COVID-19	1069	74	223
Normal	11	996	12
Pneumonia	39	93	800

Swin-T	COVID-19	Normal	Pneumonia
COVID-19	1015	96	255
Normal	0	1006	13
Pneumonia	26	83	823

ConvNeXT-T	COVID-19	Normal	Pneumonia
COVID-19	1105	54	207
Normal	1	991	27
Pneumonia	42	56	834

Table 4: Confusion matrices of baselines and studied models.

		COVID-19	**Normal**	**Pneumonia**
ResNet50	**Precision**	0.94684	**0.83455**	**0.88716**
	Recall	0.89019	0.97056	0.92864
	F1	**0.79541**	0.85515	0.82420
VGG16	**Precision**	0.95505	0.82430	0.88487
	Recall	**0.90866**	0.94701	0.92744
	F1	0.77230	0.89163	0.82769
DenseNet121	**Precision**	0.95532	0.78258	0.86036
	Recall	0.85641	0.97743	0.91292
	F1	0.77295	0.85837	0.81342
Swin-T	**Precision**	**0.97502**	0.74305	0.84337
	Recall	0.84895	**0.98724**	0.91289
	F1	0.75435	0.88305	0.81364
ConvNeXT-T	**Precision**	0.96254	0.80893	0.87908
	Recall	0.90009	0.97252	**0.93491**
	F1	0.78090	**0.89485**	**0.83400**

Table 5: Precision, Recall, and F1 per class for each model.

	Ma. Precision	**W. Precision**	**Ma. Recall**	**W. Recall**	**Ma. F1**	**W. F1**
ResNet50	0.87748	0.88689	0.88675	**0.88212**	0.88000	**0.88221**
VGG16	**0.87867**	**0.88945**	**0.88765**	0.88092	**0.88000**	0.88188
DenseNet121	0.86156	0.87369	0.87279	0.86373	0.86224	0.86332
Swin-T	0.85944	0.87429	0.87111	0.85740	0.85663	0.85637
ConvNeXT-T	**0.88118**	**0.89232**	**0.89210**	**0.88333**	**0.88266**	**0.88356**

Table 6: Macro and Weighted Precision, Recall, and F1 per for each model.

The macro and weighted metrics (Table 6) for each baseline model show that VGG16 has larger macro and weighted precision, and macro recall, while the ResNet50 has larger weighted recall. When also analyzing the studied models, we can see that ConvNeXT has larger precision and recall than all of the other models in both macro and weighted averages, and consequently also a better F1 score.

For the purpose of classification in a medical context, we want to balance precision and recall, to avoid resources from being wasted and also assuring that the correct treatment is given to patients. In this case, despite the model with the best precision/recall varying per class, when averaging for all classes/samples, the model with the best metrics - accuracy, macro/weighted precision, macro/weighted recall - is the ConvNeXT model. However, the performance difference is not sufficient to declare that ConvNeXT is the better alternative among all without a doubt.

5 Discussion

This section analyzes and provides explanations regarding the results obtained. The first part addresses the dataset and associated issues. The second part analyzes the differences in performance between the baselines and the studied models. Finally, we address potential issues with training methods, namely number of training epochs and underfitting/overfitting.

5.1 Dataset and Associated Challenges

As mentioned previously, the COVIDx CXR-3 dataset is comprised of over 30,000 chest X-rays. Considering datasets used in other image classification challenges, such as ImageNet, the size of this dataset is rather small, even if it is not the smallest dataset when compared to datasets used in other tasks with potential medical applications, which are particularly difficult to obtain in the first place, due to patient privacy concerns.

A consequence of this is that there may not be enough data to for the models to learn properly what characteristics make it possible for radiologists to differentiate between an X-ray of a COVID-19 patient, an X-ray of a pneumonia patient, and an X-ray of a healthy patient. Additionally, there may be situations where a radiologist would have difficulty diagnosing an X-ray, and the only method to do so would be through an RT-PCR test. In situations such as these, while it would be significant if a model could outperform a certified radiologist, it is not expected for a network to do so.

The dataset does not provide information on the origin of the classification label - whether it is from diagnosis by certified radiologists, or the result of other means of diagnosis. It is also possible that the labelling method is inconsistent and was obtained with different methods, depending on the origin dataset of the chest X-ray.

It also does not provide information on ethnicity, age, or other characteristics of the patients that would make models trained on the dataset prone to bias (Cruz et al., 2021). As a consequence, it's difficult to consider those characteristics in training to mitigate any potential biases.

Following on the fact that the datasets come from different sources - different hospitals, different original resolutions, different X-ray machines - there's a probability that the results obtained are classifying the images based not on the diseases' characteristics, but confounders, such as hospital-specific information, or annotations, that are not removed by the transform methods while training. This is particularly significant when the model needs to differentiate between COVID-19 and Pneumonia/Normal X-rays, since the data for Pneumonia/Normal is only from one source that is different from all COVID-19 X-ray sources.

5.2 Baselines Versus Experimental Models

The baselines presented performed relatively well for the task. While the idea of adding attention mechanisms to models seems like a potential path for improvements in image classification, in practice, the Swin Transformer did not obtain significant improvements when compared to the baselines. Neither did the ConvNeXT, a convolutional network that attempts to combine features from both regular convolutional networks and transformer models. Each model has obtained better results than others in the different metrics analyzed in this thesis, but ultimately no significant differences in performance were found in either the baseline models, nor the models studied, that could signify an improvement in the classification task.

5.3 Epochs and Overfitting

The approach we used for training limited the number of epochs for all models to 20, to avoid overfitting on the training set. However, there are no guarantees that overfitting hasn't occurred before 20 epochs, or that the models would have suffered overfitting past 20 epochs. It is possible that the models would have performed better on the test set if training had stopped at an earlier epoch, or if the models had been allowed a larger number of training epochs.

6 Conclusions and Future Work

This work presented a comparison on various architectures for image classification for COVID-19 detection with chest X-rays. This section offers a final review of the main contributions provided by this work and suggests possible future avenues of exploration to improve the models' performance.

6.1 Contributions

This work provides a study on automatic image classification of chest X-rays with deep neural networks, determining whether a patient has COVID-19, pneumonia, or is healthy. We observed that both baselines and the studied models performed relatively well, with an accuracy over 80%, and similar results in the other evaluation metrics. However, there were no significant improvements in our studied models, Swin-T and ConvNeXT, when compared to our baseline models. The ConvNeXT model did provide a slight performance improvement in accuracy and macro/weighted metrics, however we considered the difference to not be statistically significant to claim that the studied model presents itself as an improvement on conventional architectures for the task.

6.2 Future Work

The obtained results shown in Section 4 highlight the difficulty of neural models in significantly improving their classification ability. There was no significant increase in the performance of the studied models, when compared to the baselines. As highlighted in Section 5, this may suggest that:

- The training methods used in this study were not adequate to ascertain whether the performance obtained is the best performance possible for the models on the used dataset;
- The dataset used, along with its modifications, is not sufficiently representative of the data distribution, which does not allow the models to learn the characteristics present in the images that correspond to the purported conditions, and generalize them to unseen data;
- The models studied and associated parameters are a limitation that do not show the full potential that the architectures possess in this task.

To address the first issue, one possible approach would be to split the training set further into a training and validation set, and evaluate when the models would be at risk of overfitting based on their performance on the validation set (Morgan and Bourlard, 1990; Reed, 1993; Prechelt, 2012), with the consequence that there will be less data available for training the models. Additionally, an early stopping criteria could be introduced where training would be stopped once performance on the validation set had decreased.

To address the second issue, possible solutions would be to either change the dataset used to one with further information on possible confounders or information on limitations that could affect the model's training (Cruz et al., 2021; Ahmed et al., 2021), or further analyze the current dataset, and adjust the dataset or training methods according to our findings.

The first option would be ideal, but does not seem feasible with the current publicly available datasets, which are, for the most part, either small-sized datasets i.e. the Cohen dataset (Cohen et al., 2020), or collections of various COVID-19/pneumonia datasets, such as the one we used in this thesis (Wang and Wong, 2020).

The second option would require considerable work, since we would have to analyze the sources of the collection for characteristics that could prove to be possible confounders. Additionally, we could employ the method in Minaee et al. (2020) of occluding parts of the images until the models no longer classified the image as a COVID-19 chest X-ray, or Grad-CAM (Selvaraju et al., 2019), to visualize if the models were classifying the samples based on information present in the relevant portion of the X-rays. However, there would be no guarantees that the areas relevant for the model are the areas relevant for diagnosis without the assistance of expert radiologists or a dataset with the relevant areas labeled.

As for the third issue, multiple approaches could be taken: one such approach would be to use larger variants of the architectures. We used the "T" variants of the Swin and ConvNeXT architectures, which are the smallest available. However, "B" and larger variants have obtained improved results on the ImageNet and other datasets, which could indicate they have a better classification ability compared to smaller versions. However, fine-tuning on larger variants will also require more resources.

Another approach would be to further modernize parameters associated with training, such as the optimizer. The optimizer used in this thesis for the proposed models was the ADAMW optimizer (Loshchilov and Hutter, 2017). More recent alternatives for optimizers have appeared, such as Sharpness-Aware Minimization (Foret et al., 2020) or SAM, which attempts to find the minimum loss in neighbourhoods with both low loss and low curvature (hence why its sharpness-aware), or the Surrogate Gap Guided Sharpness-Aware Minimization (Zhuang et al., 2022) or GSAM, an improvement on the SAM optimizer. Both optimizers have empirically shown improved model generalization on various datasets.

References

Ahmed, K. B., Hall, L. O., Goldgof, D. B., Goldgof, G. M., and Paul, R. (2021). Deep Learning Models May Spuriously Classify COVID-19 from X-ray Images Based on Confounders. *arXiv 2102.04300*.

Apostolopoulos, I. D. and Mpesiana, T. A. (2020). COVID-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *arXiv 2003.11617*.

Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv 1610.02357*.

Cohen, J. P., Morrison, P., and Dao, L. (2020). COVID-19 image data collection. *arXiv 2003.11597*.

Cruz, B. G. S., Bossa, M. N., Sölter, J., and Husch, A. D. (2021). Public Covid-19 X-ray datasets and their impact on model bias – A systematic review of a significant problem. *medRxiv 2021.02.15.21251775*.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. (2020). Sharpness-Aware Minimization for Efficiently Improving Generalization. *arXiv 2010.01412*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv 1512.03385*.

Hendrycks, D. and Gimpel, K. (2016). Gaussian Error Linear Units (GELUs). *arXiv 1606.08415*.

Henkel, C. (2021). Efficient large-scale image retrieval with deep feature orthogonality and Hybrid-Swin-Transformers. *arXiv 2110.03786*.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv 1704.04861*.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2018). Densely Connected Convolutional Networks. *arXiv 1608.06993*.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv 1602.07360*.

Ilyas, M., Rehman, H., and Nait-ali, A. (2020). Detection of COVID-19 From Chest X-ray Images Using Artificial Intelligence: An Early Review. *arXiv 2004.05436*.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., Seekins, J., Mong, D. A., Halabi, S. S., Sandberg, J. K., Jones, R., Larson, D. B., Langlotz, C. P., Patel, B. N., Lungren, M. P., and Ng, A. Y. (2019). CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. *arXiv 1901.07031*.

Jaeger, S., Candemir, S., Antani, S., Wáng, Y. X., Lu, P. X., and Thoma, G. (2014). Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative Imaging in Medicine and Surgery*, 4(6):475–477.

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., Shi, A., Zhang, R., Zheng, L., Hou, R., Shi, W., Fu, X., Duan, Y., Huu, V. A., Wen, C., Zhang, E. D., Zhang, C. L., Li, O., Wang, X., Singer, M. A., Sun, X., Xu, J., Tafreshi, A., Lewis, M. A., Xia, H., and Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5):1122–1131.

Khan, A. I., Shah, J. L., and Bhat, M. M. (2020). CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *arXiv 2004.04931*.

Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. *arXiv 1412.6980*.

Koivunen, A. C. and Kostinski, A. B. (1999). The Feasibility of Data Whitening to Improve Performance of Weather Radar. *Journal of Applied Meteorology*, 38(6):741–749.

Le Dinh, T., Lee, S.-H., Kwon, S.-G., and Kwon, K.-R. (2022). COVID-19 Chest X-ray Classification and Severity Assessment Using Convolutional and Transformer Neural Networks. *Applied Sciences*, 12(10).

Lin, Z. Q., Shafiee, M. J., Bochkarev, S., Jules, M. S., Wang, X. Y., and Wong, A. (2019). Do Explanations Reflect Decisions? A Machine-centric Strategy to Quantify the Performance of Explainability Algorithms. *arXiv 1910.07387*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv 2103.14030*.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A ConvNet for the 2020s. *arXiv 2201.03545*.

Loshchilov, I. and Hutter, F. (2017). Decoupled Weight Decay Regularization. *arXiv 1711.05101*.

Mamalakis, M., Swift, A. J., Vorselaars, B., Ray, S., Weeks, S., Ding, W., Clayton, R. H., Mackenzie, L. S., and Banerjee, A. (2021). DenResCov-19: A deep transfer learning network for robust automatic classification of COVID-19, pneumonia, and tuberculosis from X-rays. *arXiv 2104.04006*.

Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., and Soufi, G. J. (2020). Deep-COVID: Predicting COVID-19 From Chest X-Ray Images Using Deep Transfer Learning. *arXiv 2004.09363*.

Morgan, N. and Bourlard, H. (1990). *Generalization and Parameter Estimation in Feedforward Nets: Some Experiments*, page 630–637. Morgan Kaufmann Publishers Inc.

Nair, V. and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the International Conference on Machine Learning*.

Narin, A., Kaya, C., and Pamuk, Z. (2021). Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks. *arXiv 2003.10849*.

Prechelt, L. (2012). *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg.

Reed, R. (1993). Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *arXiv 1610.02391*.

Shi, F., Wang, J., Shi, J., Wu, Z., Wang, Q., Tang, Z., He, K., Shi, Y., and Shen, D. (2021). Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation, and Diagnosis for COVID-19. *IEEE Reviews in Biomedical Engineering*, 14:4–15.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv 1602.07261*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. *arXiv 1409.4842*.

Ulhaq, A., Khan, A., Gomes, D., and Paul, M. (2020). Computer Vision For COVID-19 Control: A Survey. *arXiv 2004.09420*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *arXiv 1706.03762*.

Wang, L. and Wong, A. (2020). COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. *arXiv 2003.09871*.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017). ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. *arXiv 1705.02315*.

Wong, A., Shafiee, M. J., Chwyl, B., and Li, F. (2018). FermiNets: Learning generative machines to generate efficient neural networks via generative synthesis. *arXiv 1809.05989*.

Zeiler, M. D. and Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. *arXiv 1311.2901*.

Zhuang, J., Gong, B., Yuan, L., Cui, Y., Adam, H., Dvornek, N., Tatikonda, S., Duncan, J., and Liu, T. (2022). Surrogate Gap Minimization Improves Sharpness-Aware Training. *arXiv 2203.08065*.