DD2424 Project Report

Detecting COVID-19 from X-ray imaging using SNN - Stacked Neural Networks.

Anirudh Seth (aniseth@kth.se) Magnus Pierrau (mpierrau@kth.se)
Resa Dadfar (resdaf@kth.se)

Spring 2020

Abstract

The global pandemic is affecting human lives in various ways and introducing social and economic problems. To fight against the spread of the virus, it is significant to have a practical yet precise diagnostic tool. Chest X-ray images are widely known as a low cost, highly available solution, however they require the assessment of a medical professional and may introduce personal or professional biases. In this project, we investigate and develop a multi class deep convolutional network to assist in the classification of chest X-ray images. To account for the lack of publicly available COVID19 data we explore and perform comparative analysis of various data augmentation techniques and their applicability on X-ray images.

Finally we apply Grad-CAM to highlight the regions of high importance and improve the explainability of our deep neural network. We find that standard data augmentation techniques as well as advanced methods, such as image segmentation, do not improve the results. Whether this is a result of removing biases or a lacking implementation is uncertain.

Contents

1 Introduction

As the COVID-19 virus continues to spread throughout the world, the entire research community has been scrambling to respond to the different challenges caused by the pandemic. The AI-community has risen to the task of aiding medical staff by developing machine learning based diagnostic tools to assist with diagnosing possible COVID-19 cases.

In this project we have studied papers in which different convolutional neural networks (CNN), both multi-label and binary, have been developed and trained using transfer learning to classify X-ray images of lungs. We implement a CNN using transfer learning from a pre-trained VGG16 model and apply this to a dataset of X-ray images with lungs from healthy, and pneumonia and COVID19 positive patients. We evaluate the performance using several metrics and compare these to some similar and recent articles on the same task.

The major challenge of this task is the small amount of COVID-19 positive lung image samples, which prompts us to investigate data augmentation techniques as well as to consider data class imbalance and data bias issues.

When using a machine learning tool to help diagnose patients, it is of high importance that the medical staff is able to interpret the results of the model, and we therefore also investigate the visualization technique Grad-CAM [9] to improve the explainability of our model.

2 Dataset

As we are in the beginning of COVID-19 pandemic, a comprehensive dataset of COVID-19 positive X-ray images is not yet available to the researcher community. We collected and 15160 images from [12], [11] and [13] and combined them into one large dataset, as recommended in [8]. The dataset was then shuffled and then randomly divided (with specified proportions for each class) into training, validation and test sets.The resulting distribution is summarized in table 1.

Set	Normal	Pneumonia	COVID-19	Sum	%
Training	6373	4352	192	10917	72
Validation	1593	1088	48	2729	18
Test	885	604	26	1515	10
Sum	8851	6044	266	15161	100
%	58.4	39.9	1.7	100	

Table 1: Distribution of the X-ray images used in the project

A significant class imbalance is highlighted in table 1, where COVID-19 images make up only 1.7% of the entire dataset. We therefore implement and investigate data balancing and augmentation strategies to address the issue.

To compensate the class imbalance and the scarcity of COVID-19 positive images, a custom data loader was implemented that randomly up-samples COVID-19 images when generating a batch of training data. The rate of upsampling can be specified to the data generator as a hyperparameter. This helps to reduce the risk of overfitting the prominent classes, i.e Normal and Pneumonia.

3 Methodology

3.1 Model Architecture

Stacked Neural Networks (S-NN) can leverage transfer learning from multiple publicly available pretrained neural networks. They have been shown to achieve higher accuracies [1] and generate better features with faster training times. The intuition behind transfer learning for image classification follows from the fact that a model which is trained on a large and general enough dataset, can serve as a good starting point to describe general features of any image.

We used the pretrained **VGG16** neural network proposed by K. Simonyan and A. Zisserman [14], visualized in Figure 1, trained on around 15 million labeled high-resolution images from the ImageNet

database, as the base model. It consists of 16 convolutional layers and is very appealing because of its very uniform Architecture and proven performance. The fully connected top layers were replaced by our own neural network (Figure 2)

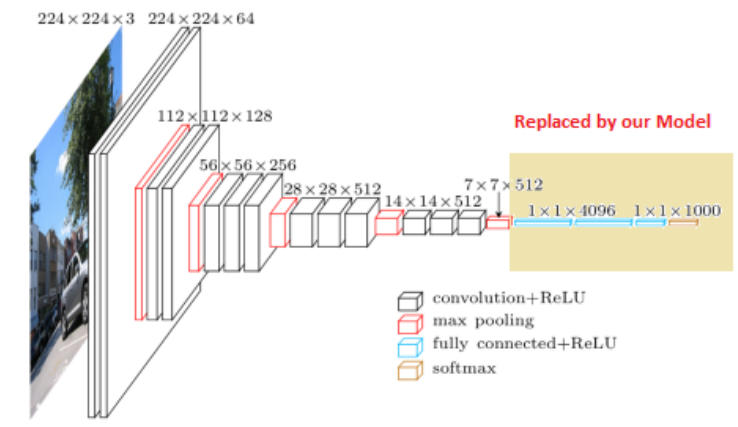


Figure 1: Transfer Learning from VGG16.

The gradients for VGG16 layers were frozen with the aim of assisting transfer learning. An additional block of layers was stacked on top to fine-tune the network specifically for X-ray images. The final model is visualized in Figure 2.

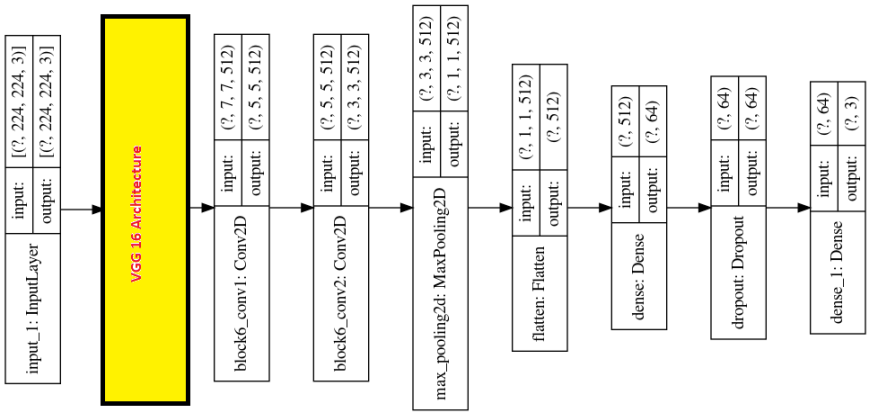


Figure 2: Model architecture implemented for the project.

3.2 Data augmentation

Before training any model we pre-process the data by various means of data augmentation. Data augmentation encompasses a suite of techniques that can enhance the size and qualities of training datasets such that better and more robust Deep Learning models can be built using them.

We considered some standard data augmentation techniques (Section 3.2.1), as well as devising an additional, advanced data augmentation process. This process, inspired by [3], consists of histogram equalization (Section 3.2.2), image segmentation using a U-net CNN (Section 3.2.3) and image normalization.

Two examples of X-ray images before and after applying standard and advanced data augmentation are shown in Figure 4.

3.2.1 Standard data augmentation

We implemented some commonly used techniques as specified in table 2 in order to enhance our training data. We also considered applying horizontal flips, but as the human organs are asymmetrical, and horizontal flips primarily assist when there are no assumptions on data asymmetry, this would likely not improve the performance of our network [10].

Figures 4b and 4e show two examples of the resulting images after randomly applying the augmentations from table 2.

Transformation	Value/range
Rotation	0 - 10°
Width shift	0 - 10%
Height shift	0 - 10%
Zoom	85 - 115%

Table 2: Image augmentations applied to training data.

3.2.2 Histogram equalization

Since the dataset consists of X-rays from multiple sources, it is important to consider various biases that may not be immediately apparent. Different sources may use different settings in their X-ray routines causing varying image brightness and contrasts. In order to eliminate this issue (for example all Pneumonia patients having similar brightness/contrast in the images), we consider Histogram equalization [18]. This method provides enhanced image contrast, without any information loss, by distributing the intensities over the image pixel histogram so that the region with lower contrast gain a higher contrast.

3.2.3 Image segmentation by U-net

Histogram equalization eliminates some biases, but there is still a risk that biases may be introduced into the dataset by artefacts in images, for example watermarks, annotations or medical equipment. Using only the original X-ray image for training introduces the risk that the network will learn features from the images that are not related to the quality of the lungs, but rather a feature of the image itself. We therefore choose to use image segmentation to crop out the important part of the images - the lungs.

Image segmentation is a large challenge in itself and convolutional neural networks can be, and have been, trained for this purpose.

To achieve this we use a CNN with a specific architechture, called the U-net [5]. The U-net is a state of the art model and has been widely implemented in various fields. Our implementation follows the adaption from [6], which is trained and tested on the JSRT [15] and Montgomery [16] X-ray datasets, using ADAM optimizer and binary cross-entropy loss. As our main purpose of this project is to train a CNN to classify COVID19 patients, we used a pre-trained model, which achieves IoU (Intersection over Union) and Dice scores between 0.95-0.99 on the test dataset.

For our data we are interested in extracting the lungs from the X-ray image, meaning that we need semantic segmentation. Each pixel in the image is thus assigned a predicted class by the U-net and all pixels corresponding to the same class can then be extracted to form a mask, which can be merged with the image to create a cropped image. Note that we are not interested in instance segmentation as we do not discriminate between left and right lung - both are simply classified as "lung".

The U-net architecture, visualized in Figure 3, consists of two paths; the contracting and the expansive path. The former contracts the image into a feature map, and the latter creates a high-resolution segmentation map. In the contracting path we repeatedly apply unpadded 3×3 convolutional layers followed by a ReLU activation layer and a 2×2 max pooling layer with stride 2 for downsampling. In the expansive path the feature map is upsampled by a 2×2 up-convolution, which is concatenated with the high resolution copy of the features from the corresponding contracting path. This is followed by a 3×3 convolution and a ReLU activation layer. Finally there is a 1×1 convolutional layer that maps each feature vector to the desired number of classes (in our case two - "lung" and "non-lung").

The copying of the high resolution features for the contracting path allows the network to propagate context information to the higher resolution layers [5].

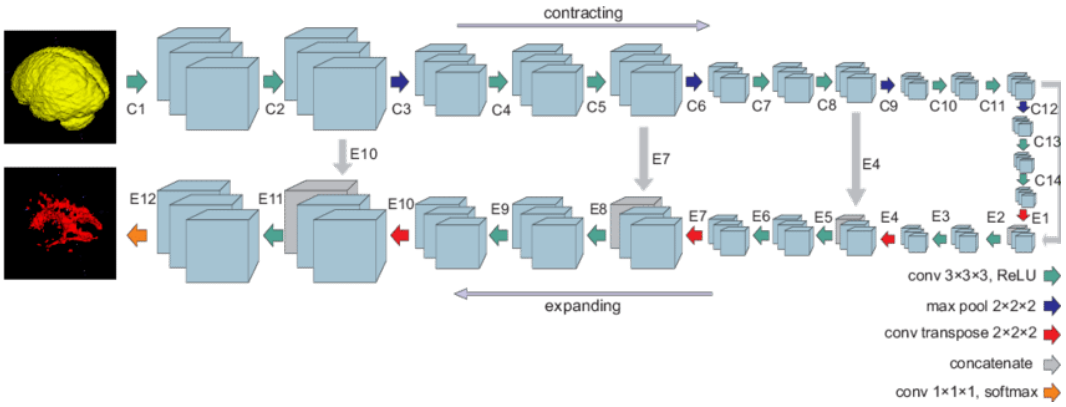
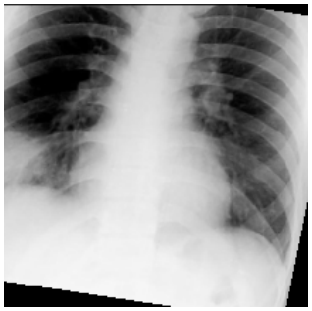


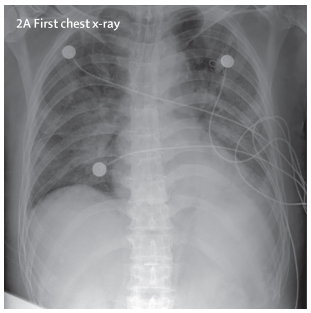
Figure 3: U-net architecture [17].

(a) Example 1 original image.

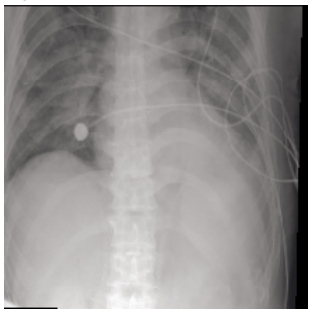


(b) Example 1 after standard data augmentations.

(c) Example 1 after advanced data augmentation.



(d) Example 2 original image.



(e) Example 2 after standard data augmentations.

(f) Example 2 after advanced data augmentation.

Figure 4: Example images of data before and after pre-processing. The two rows shows X-ray images of two different patients before any data augmentation is applied (column 1), after standard data augmentations according to table 2 are applied (column 2) and after applying advanced data augmentations, i.e. image segmentation, histogram equalization and image normalization (column 3).

As seen in Figures 4c and 4f, the application of histogram equalization evens out the contrast in the images causing artefacts such as medical devices and image labeling to be greatly reduced or completely removed, eliminating these factors as a source of bias.

Evaluating the performance of the U-net in the lung segmentation task is hard since we neither have the true masks to compute IoU or Dice metrics, nor are pulmonary experts. However, after comparing a number of resulting samples we consider the results acceptable, although a more thorough evaluation would have been preferable.

3.3 Metrics

As performance metrics we consider the accuracy, loss, precision, recall (sensitivity), F_1-score and F_2-score. The definitions of these metrics can be found in 7.1 in the appendix.

The F_1 score is the harmonic mean of the precision and the recall. It is a weighted average which considers both recall and precision. Since we have an unbalanced class distribution, this is a good metric, as it captures both measures of interest in one number.

The F_2 score is, just as the F_1 score, an instance of a F-measure, but one where Recall is weighted twice as much as Precision. This metric is relevant to us as we consider false negatives to be of a greater concern than false positives.

3.4 Experimental setup

The model was implemented using Tensorflow on Google Cloud Platform hosting a virtual machine with Linux Debian 9 OS and one Nvidia Tesla P4 GPU. We performed three experiments according to the table below:

1. **Experiment 1:** Training and validating on original images without data augmentation.
2. **Experiment 2:** Training with segmented images without standard augmentation techniques and validating on original images.
3. **Experiment 3:** Training on segmented images augmented with standard techniques as specified in Section 3.2.1 and validating on original images.

The experiments were performed on the same model architecture with the parameters specified in table 3.

Parameter	Value
Optimizer	ADAM
ETA	0.001 with decay 0.001/10
Loss	Categorical Cross Entropy
Metrics	Accuracy , Loss
Epochs	50
Batch Size	100

Table 3: Parameters used to for training the model.

As stated before we apply upsampling to increase the amount of COVID-19 images in the test set. The level of upsampling was set to 30% COVID-19 images, for all experimental setups.

3.5 Visual Explanations using Grad-CAM

Grad-CAM (Gradient-weighted Class Activation Mapping) [9], is a method which can be used for visualizing the regions of image that are "important" for predictions. It uses the gradients of any target concept, flowing into a convolutional layer to produce a coarse localization map highlighting these important regions. The method can be broken to the following three steps.

1. For any class c, we first compute the gradient of the score for class c, y^c (before the softmax), with respect to feature map activation's A^k of any convolutional layer, $\frac{y^c}{A^2}$.
2. The gradients are then global-average-pooled over the width and height dimensions (indexed by i and j respectively) to obtain the neuron importance weights α_k^c.
3. The final step is to perform a weighted combination of forward activation maps, and follow it by a ReLU to obtain the coarse heatmap.

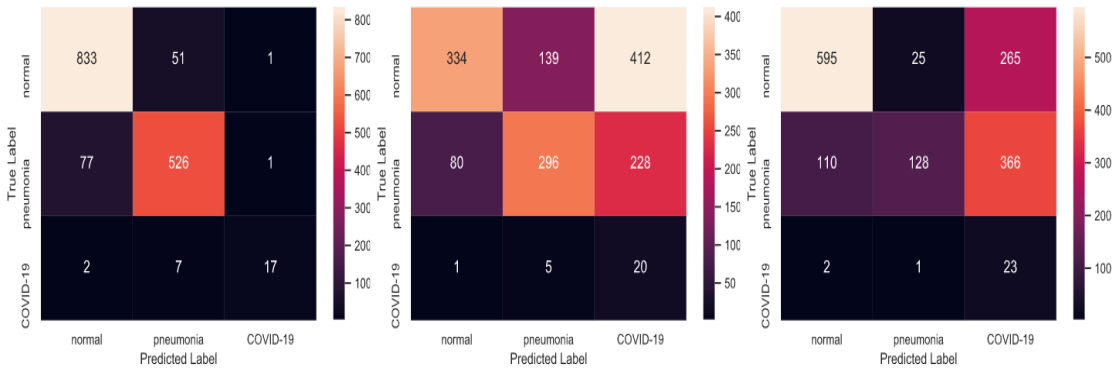
$$L^c_{\text{Grad-CAM}} = \text{ReLU}\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}} \tag{1}$$

The activation is applied since we are only interested in the features that have a positive influence on the class of interest.

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \tag{2}$$

Several previous works [7] have shown that deeper representations in a CNN capture the best high-level constructs. In our implementation we used the last convolutional layer i.e `block6_conv2 (Conv2D)` (Figure :2) since it is expected to to have the best trade off between high-level semantics and detailed spatial information.The results form our implementation are presented and analysed in the subsequent sections.

4 Analysis and Results



(a) (b) (c)

Figure 5: Confusion matrices for experiments 1 (a), 2 (b) and 3 (c).

By considering Figure 5 we get a clear visualization of the classifications of the various setups. Immediately we note that the initial experimental setup, where no image segmentation or data augmentation is performed, seems to produce the best results. As seen in table 4, the overall accuracy is 91 % for experiment 1, while it is 43 and 49% for experiment 2 and 3 respectively. As we are specifically interested in the classification of COVID-19 cases, we note that recall is only 65% for experiment 1, due to the 9 misclassifications, while the precision is 89%.

In our case a false positive leads to a person being ordered to self-isolate even though he or she is free from COVID-19, which is a relatively low price to pay to be on the safe side. A false negative, on the other hand, is something we wish to avoid - we don't want to risk an infected person being misclassified as healthy and risk him or her to infect more people. This could also lead to early disregard of specific countermeasures aimed at averting serious illness, allowing the virus to progress and cause more harm to the lungs of the patient.

We will therefore allow a larger precision, while aiming to keep the recall as low as possible and trying to achieve an acceptable overall accuracy.

The performance becomes significantly worse as we segment the images in experiment 2. The recall for COVID-19 classification increases somewhat to 77%, but at the cost of a dramatic decrease in specificity down to only 3%.

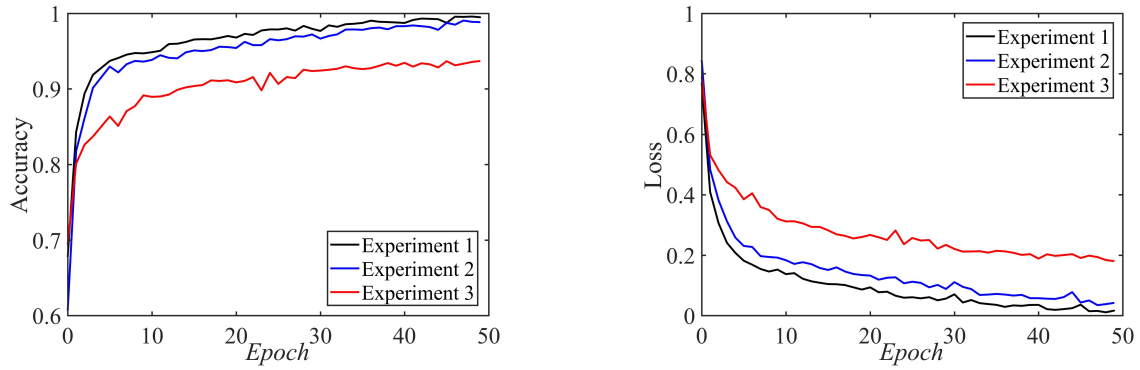
We note that the precision is very low for COVID-19 class in experiment 2 and 3. Inspecting the confusion matrices in figures 5b and 5c reveals that a large nubmber of normal and pneunomia classed images were erroneously classified as COVID-19. This indicates that the model is overfitting on COVID-19 data. Simultaneously we note that in experiment 2 a large number of normal images were

classified as pneumonia, causing specificity to decrease for this class as well. While we did argue that we are more interested in recall than precision, these results would lead to 95% of tested subjects to be ordered to self-isolation, which is neither desirable nor feasible.

	Experiment 1				**Experiment 2**				**Experiment 3**			
	Precision	Recall	F1 Score	F2 Score	Precision	Recall	F1 Score	F2 Score	Precision	Recall	F1 Score	F2 Score
Normal	0.91	0.94	0.93	0.93	0.8	0.38	0.51	0.42	0.84	0.67	0.75	0.7
Pneumonia	0.9	0.87	0.89	0.88	0.67	0.49	0.57	0.51	0.83	0.21	0.34	0.25
COVID-19	0.89	0.65	0.76	0.69	0.03	0.77	0.06	0.13	0.04	0.88	0.07	0.15
Accuracy	0.91				0.43				0.49			

Table 4: Four selected metrics are presented for the three experimental setups as well as the total accuracy for each experiment.

Figures 6a and 6b show the accuracy and loss function for all the three experiments. After 40 epochs, the accuracy of the for Exp. 1 and Exp. 2 cases approaches to 98.7%. However, the standard augmentation case (Exp. 3) needs more iterations to reach the same level of accuracy. The same trend is observed for the evolution of loss function. While the loss functions for experiments 1 and 2 converges a small value, experiment 3 is still decreasing after 50 epoch.



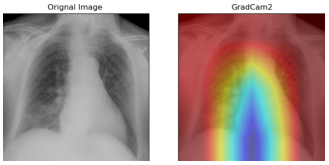
(a) Accuracy per epoch

(b) Loss per epoch

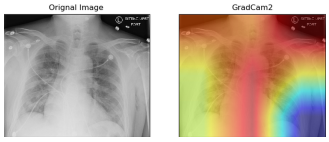
Figure 6: Accuracy and loss on training set during test for the three experimental setups.

To visualize the characteristic features/regions utilized by the model for classification, we employed the concept of Gradient Class Activation Map (Grad-CAM) as explained in Section 3.5. The algorithm was applied to a randomly selected subset of images from the test set. Figure 7 shows the results for one COVID-19 X-ray image for each experiment. The critical regions with the highest attention are highlighted in red and the ones with the least in blue. For Experiment 1 both the lungs are given high attention.In general, regions near or within the lungs are localized. This correlates with effects of the virus on the lungs but needs to be validated by experts.

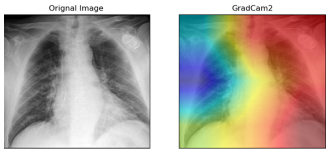
In general, Grad-CAM localizes locally on areas of importance, whereas for our results the heat map indicates a large area used for classification. This is in line with the low accuracy of experiment 2 and 3, as the network failed to find relevant features of the images to base classification on. However, a more local heatmap was expected for the first experimental setup, as the setup showed reasonable results on metrics.



(a) Experiment 1



(b) Experiment 2



(c) Experiment 3

Figure 7: Grad-Cam applied on randomly selected COVID X-ray images from the test set.(Original image on the left, GradCam overlayed heatmap on the right)

5 Discussion

The first model was trained on the original X-Ray images without any augmentation techniques to serve as the baseline model. It leverages transfer learning from a trained network and produced satisfactory results as presented above. Our initial hypothesis of achieving better results by applying augmentation techniques presented in Sections 3.2.1, 3.2.2 and 3.2.3 did not come to fruition, as expected.

In the initial runs we applied histogram equalization to our data, but were surprised to see that the performance decreased. One explanation might be that X-ray images do not lend themselves well to the method, as some fine detail information might be decreased in contrast, and thus not learnt as strongly by the network. The decreased performance could also be explained by initial segmentation producing white padding around images which might have affected training. When this error was removed performance increased. However this was in conjunction with the removal of histogram equalization, and so we cannot say for sure what the increased performance was due to - removal of paddings or histogram equalization. Unfortunately we did not run tests to compare normal data without histogram equalization to the same data with the augmentation applied. This would have given us a clear indication whether this was indeed the case.

The results from the Grad-CAM implementation seem to highlight the lungs but do not indicate specific areas within the lungs. The correctness of the heatmap visualization requires external validation by experts. Lastly, publicly available COVID-19 data is limited, which is a bottleneck for our experiments. The model architecture and hyper parameters can be further tuned once the data becomes available.

6 Conclusion and future work

We have found that it is possible to train a multi-class deep convolutional network to classify X-ray images accurately. We investigated various data augmentation techniques and found that recall can be improved, but at the cost of overall accuracy and precision. In the end the setup that provided the best performance was the one trained on the original full scale images, without any data augmentations. This is surprising, and leads us to suspect that the implementation is not without fault, as data augmentation techniques have been applied successfully and with increased performance in other works of high regard, such as [3]. As stated in the same paper however, data augmentation such as histogram equalization and image segmentation are important tools to battle biases in the data, and our results have not led us to believe that they are wrong, but that further investigation is necessary.

Future works would also include experiments with and without various data augmentation techniques to gauge its impact on the performance, including histogram equalization. Furthermore, hyperparameter tuning and further improvement upon the GradCAM and comparisons with other visualization techniques such as ACE or GradCAM++ are natural proceedings.

References

[1] Mohammadi, Milad, and Subhasis Das. “SNN: Stacked Neural Networks.” ArXiv:1605.08512 [Cs], May 2016. arXiv.org, http://arxiv.org/abs/1605.08512.

[2] Ali N, Ceren K, Ziynet P. “Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks”. CSCV 2020

[3] Enzo T et al., “Unveiling COVID-19 from Chest X-ray with deep learning: a hurdles race with small data”, 2020 (April 11th)

[4] Jianpeng Z. et al., “COVID-19 Screening on Chest X-ray Images Using Deep Learning based Anomaly Detection”, 2020 (22 Mar)

[5] Olaf R., Philipp F., Thomas B. "U-Net: Convolutional Networks for Biomedical Image Segmentation", 2015. arXiv:1505.04597 [cs.CV]

[6] Vitali B. `https://github.com/imlab-uiip/lung-segmentation-2d` Boston University, CS Dpt.

[7] Grad-cam++: Improved Visual Explanations For Deep Convolutional Networks Sarkar-Howlader- Balasubramanian-Vineeth N- Aditya - https://arxiv.org/abs/1710.11063

[8] Linda W., Zhong Q., Alexander W. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images, 2020, 2003.09871

[9] Rs, Ramprasaath Cogswell, Michael Das, Abhishek Vedantam, Ramakrishna Parikh, Devi Batra, Dhruv. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. 618-626. 10.1109/ICCV.2017.74.

[10] Paolo G. *Hands-On Neural Networks with TensorFlow 2.0*. Packt Pub. Ltd. 2019.

[11] Linda Wang Alexander Wong Zhong Qiu Lin Paul McInnis and Audrey Chung DarwinAI Corp., Canada and Vision and Image Processing Research Group, University of Waterloo, Canada `https://github.com/agchung/Figure1-COVID-chestxray-dataset`

[12] Joseph Paul Cohen and Paul Morrison and Lan Dao COVID-19 image data collection, arXiv:2003.11597, 2020 `https://github.com/ieee8023/covid-chestxray-dataset`

[13] RSNA Pneumonia Detection Challenge: overview. Kaggle website. www.kaggle.com/c/rsna-pneumonia-detection-challenge. Accessed July 17, 2019 `https://nihcc.app.box.com/v/ChestXray-NIHCC/file/220660789610`

[14] Karim, M., Döhmen, T., Rebholz-Schuhmann, D., Decker, S., Cochez, M., Beyan, O. (2020). Deepcovidexplainer: Explainable covid-19 predictions based on chest x-ray images. arXiv preprint arXiv:2004.04582.
Read More: https://www.ajronline.org/doi/abs/10.2214/AJR.19.21512

[15] Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K., Matsui, M., Fujita, H., Kodera, Y., Doi, K. *Development of a Digital Image Database for Chest Radiographs With and Without a Lung Nodule* AJR 2000 174:1, 71-74 `http://db.jsrt.or.jp/eng-04.php`

[16] Jaeger S, Candemir S, Antani S, Wáng YX, Lu PX, Thoma G. *Two public chest X-ray datasets for computer-aided screening of pulmonary diseases.* Quant Imaging Med Surg. 2014;4(6):475-477. doi:10.3978/j.issn.2223-4292.2014.11.20 `https://lhncbc.nlm.nih.gov/publication/pub9931`

[17] Salehi, Seyed Sadegh Erdogmus, Deniz Gholipour, Ali. (2017). Tversky loss function for image segmentation using 3D fully convolutional deep networks.

[18] Russ, The Image Processing Handbook: Fourth Edition, CRC 2002 ISBN 0-8493-2532-3

7 Appendix

7.1 Metrics

Using the following definitions we can further define the metrics that we consider.

- **True Positive (TP)** - Number of correctly classified COVID-positive cases
- **True Negative (TN)** - Number of correctly classified COVID-negative cases
- **False Positive (FP)** - Number of incorrectly classified COVID-positive cases
- **False Negative (FN)** - Number of incorrectly classified COVID-negative cases

$$
\begin{aligned}
\text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\
\text{Precision} &= \frac{TP}{TP + FP} \\
\text{Recall} &= \frac{TP}{TP + FN} \\
\text{F}_1 &= 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \\
\text{F}_2 &= (1 + 2^2) \cdot \frac{\text{Recall} \cdot \text{Precision}}{2^2 \cdot \text{Precision} + \text{Recall}} \\
&= 5 \cdot \frac{\text{Recall} \cdot \text{Precision}}{4 \cdot \text{Precision} + \text{Recall}}
\end{aligned}
$$