Siamesifying the COVID-Net

Roberto Castro Sundin
rosun@kth.se

Tony Rönnqvist
tonyr@kth.se

Alejandro Sarmiento
lasg@kth.se

Simon Westberg
swestber@kth.se

Abstract

Image recognition and identification have, in recent years, become almost synonymous with deep neural networks. Previous success with these networks indicates that they are well suited as a disease diagnostic tool, and due to the recent SARS-CoV-2 virus outbreak several initiatives have surged with the aim to aid medical practitioners in the triage process by using X-ray images of the lungs to indicate whether a patient should receive treatment or not. However, a significant limitation is the low amount of available X-ray images of COVID-19 infected lungs. In this work, the authors propose a siamese network structure based on COVID-Net, with the intent of bypassing this limitation. The siamese network is compared to an equivalent single-net structure and is shown to achieve a higher over-all test accuracy, performing better on images of healthy lungs and non-COVID-19 pneumonia lungs, but interestingly achieving a worse accuracy on COVID-19 images. The final test accuracy of the siamesified network was 87 % while the corresponding accuracy for the single network was 81%. However, by analyzing the results from the single-net architecture through the explainability method GradCAM, the authors find indications that the model in some cases made predictions based on irrelevant image artifacts, and thus the actual accuracy of the networks might be lower.

1 Introduction

DEEP LEARNING techniques certainly have had a remarkable impact on many research areas in the image analysis field within the last decade; in particular, convolutional neural networks have become the de facto methodology for analyzing medical images. Use cases in the field include classification for disease diagnostic, lesion detection and segmentation, organ localization, image generation, and enhancement, among others [1]. With the use of deep learning algorithms, researchers have achieved dermatologist-level classifiers of skin cancer [2] and created both stand-alone and radiologist-aiding screeners for breast cancer [3, 4]. Current success in medical imaging suggests that it is possible to outperform human-level expertise in specific tasks. With this background, it seems reasonable to employ deep neural networks to assist medical experts for screening or diagnosing patients with suspected cases of the novel coronavirus disease (COVID-19) as early detection is key to reduce mortality rates [5]. In this work, we explore a siamese network structure, based on the recent COVID-Net [6] architecture. Training and evaluation are performed on a publicly available lung X-ray dataset from healthy patients, as well as from patients with confirmed cases of COVID-19 and other non-COVID-19 pneumonia [7, 8]. The results are compared to the performance achieved by a non-siamese implementation of the network, also based on the COVID-Net architecture. In addition we analyzed the network further using GradCAM. The justification behind the proposed architecture becomes reasonable since the amount of labeled data required to train regular artificial neural networks of decent performance is considerable. Hence, there exist two constraints: the lack of publicly available datasets of COVID-19 X-Ray or CT scans and the amount of useful data that can provide consistent performance. Siamese networks can perform relatively well under such constraints, as exemplified in [9], where it was used in a one-shot learning context.

The remainder of this work is organized as follows: related work is treated in the following subsection, after which the network architecture is covered in more detail in Section 2. Section 3 describes the data and experimental setup, as well as the proceeding results, followed by an analysis of the latter. Finally, concluding remarks are left in Section 4.

1.1 Related Work

Although the SARS-CoV-2 virus outbreak is a recent event, several initiatives can already be seen on the deep learning front, both concerning data retrieval as well as proposed network architectures. To robustly identify the virus, much focus has been put on using radiography imaging and computed tomography as these have been proven to be useful

Project report for DD2424

indicators of the disease [1]. Other efforts aim to utilize previous knowledge from large X-ray datasets on pneumonia and related lung diseases to fine-tune classifiers on different network architectures; accuracy on validation sets are shown to be larger than 90 % [10] .

COVID-19 Wang et al. [6] were one of the first to develop a tailor-made network architecture for COVID-19 identification called COVID-Net, and – perhaps even more critical – instructions on how to put together a dataset – appropriately named COVIDx by the authors – from publicly available X-ray images. Other exciting initiatives along the same lines include COVNet [11], which delivers promising results but, unfortunately, uses X-ray images that, at the time of writing, have not been made publicly available.

Siamese networks The siamese network structure is often attributed to Bromley et al. [12], where it was used for signature verification. After that, siamese networks have been applied successfully to a wide range of subjects: perhaps most famously in face verification such as DeepFace [13], but also in plant identification [14]; real-time object tracking [15]; image matching [16, 17]; and word spotting [18]. A careful treatment of the subject can be seen in [19].

2 Methods

2.1 Siamese Network Architecture

The siamese network structure is perhaps most easily explained as two ordinary neural networks working side-by-side, where the "siamese" part comes from the fact that the two networks are of the same structure and share the same weights. In a typical setup, the networks are combined at the end by incorporating them into some distance norm or similarity measure, as illustrated in Figure 1. The formulation of the training problem then transitions from the standard "*find the weights that minimize the prediction error*" into "*find the weights that minimize the distance between similar categories while keeping the distance between non-similar categories large*". Hence, siamese networks prove particularly helpful in situations where the categories are not known a priori or when no sensible metric exists, such as might be the case in face verification or image matching. Siamese networks have also proven successful in situations where available training examples are limited, such as for the Omniglot set[1] [20]. This particular property is what makes us believe that a siamese network structure might be useful for our purpose, since the available training examples for COVID-19 classification are few.

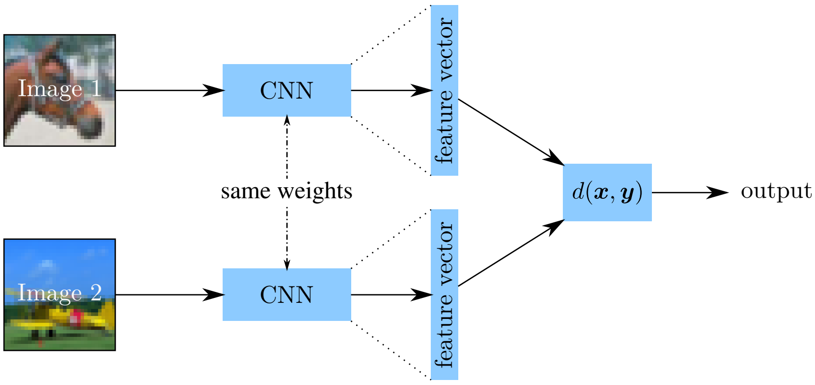


Figure 1: Typical structure of a siamese network. The function d denotes a similarity measure.

When it comes to the distance measure there are, of course, several candidates to choose from. In this work we will use a scaled version of the L_1-norm, so that the distance measure $d : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is given by

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = \sum_{i=1}^{N} \alpha_i |x_i - y_i|, \tag{1}$$

for a set of scaling weights $\{\alpha_i\}_{i=1}^N$, which consequently also have to be optimized during training. The purpose of these weights is to identify certain dimensions of extra interest. An interesting byproduct by using a norm and the structure in Figure 1, is that we automatically fulfill a symmetry condition, *i.e.*, switching places between the images does not change the output.

[1] The Omniglot set is referred to by the authors (Lake et al. [9]) as the "transpose" of MNIST because of its large set of categories and low amount of examples.

2.2 COVID-Net

Motivated by the current COVID-19 pandemic, several solutions based on deep learning have been developed around the world; however, most of these solutions are not open source. Therefore, COVID-Net was developed with this purpose, as a convolutional neural network specifically designed to assist in the detection of COVID-19 in chest X-ray images. This network was developed by Wong et al. [6] as part of Darwin AI with the assist of Gensynth, a generative synthesis platform aimed at automatically generating a set of compact and fast neural networks, provided an initial architecture. Hence, COVID-Net was initially created by human design, following best practices, and optimized by an AI. COVID-Net, as is, provides the capability of asserting if the presented chest X-Ray image is healthy, has a non-COVID-19, or a COVID-19 infection with a sensitivity $>$80 % [6]. The inner microarchitecture of some of the modules and parts of the network were designed with Gensynth and not made publicly available; therefore, it is not possible to replicate the exact model for open-source experimentation. However, the general structure of the architecture, as described by the authors, is as follows. The network is mainly composed of several computationally efficient PEPX modules (projection-expansion-projection-extension) which consist of [6]:

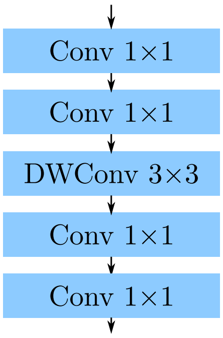


Figure 2: PEPX module

- **First-stage Projection:** 1×1 convolution to project data with a higher number of channels into a tensor with a lower channel dimension.
- **Expansion:** 1×1 convolution used to expand the number of channels, maps the features into a higher dimension, achieving more output channels than input channels.
- **DW Convolution:** 3×3 depth-wise convolution to preserve representation and learn key spatial features needed to reduce computational complexity.
- **Second-stage Projection:** 1×1 convolution to project data with a higher number of channels into a tensor with a lower number of channels.
- **Extension:** 1×1 convolution used to extend the number of channels in the output, used to produce the resulting features of the module.

As part of the design achieved by Gensynth, COVID-Net makes use of skip connections over all the architecture, to achieve a higher level of granularity and avoid degradation. The concept of skip connections is explored next.

2.2.1 ResNet

As the depth of neural networks increase, it becomes more difficult to train. Intuitively, increasing the depth of a neural network should help in learning better and with increased accuracy. However, in reality as the depth of a network increases, it has been observed that the accuracy gets saturated and decays whenever the model starts to converge, which ultimately translates into a higher training error and lower accuracy than that of shallower networks. This particular behaviour has been addressed as *degradation* and it may be attributed to the fact that not all architectures are easy to optimize properly or are prone to vanishing gradients. To alleviate this problem, residual learning was introduced in late 2015 by He et al. [21] as part of *residual networks*. The general idea is summarized in Figure 3, where skipping connections are also introduced. In normal neural networks, we try to learn a true output $\mathcal{H}(\boldsymbol{x})$, given an input $\boldsymbol{x}$, whereas residual networks instead try to approximate the residual $\mathcal{F}(\boldsymbol{x}) := \mathcal{H}(\boldsymbol{x}) - \boldsymbol{x}$ [2]. We can then reconstruct the true output $\mathcal{H}(\boldsymbol{x})$ from

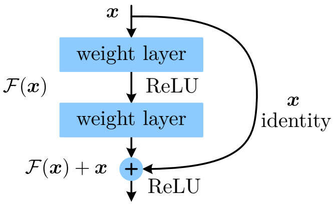

$\boldsymbol{x}$
weight layer
$\mathcal{F}(\boldsymbol{x})$
ReLU
weight layer
$\boldsymbol{x}$
identity
$\mathcal{F}(\boldsymbol{x}) + \boldsymbol{x}$
+
ReLU

Figure 3: Building block of ResNets.

$$\mathcal{H}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{x}) + \boldsymbol{x}. \tag{2}$$

Thanks to this reformulation of what the network should learn and by skipping n layers, it has been found that deeper neural networks are easier to train and should not achieve (in general) a higher training error than that of shallower networks [21]. This subsequently implies that gradients can be propagated faster to initial layers, which help with accelerating the learning and allowing deep architecture to be trained. As a side effect, the number of connections that can be skipped to achieve better results can be turned into a parameter that can be optimized. In the COVID-Net design for instance, this is taken into consideration.

2.3 Loss Function

The distance measure defined in Equation (1) will, for every image pair, return the output of whether the two images are close with respect to the inner representation of the network. The question that we want to answer is, however, if two images are of the same class or not. Following the same approach as in [9], we applied a sigmoid function on the

[2] Provided that $\mathcal{H}(\boldsymbol{x})$ and $\boldsymbol{x}$ are of the same dimension.

distance output, to produce a similarity output between 0 and 1. The output is thus binary, and consequently we used the binary cross entropy as loss function ℓ_{cr} given by

$$\ell_{\text{cr}}(\boldsymbol{y}, \boldsymbol{p}) = -\boldsymbol{y}^T \log(\boldsymbol{p}), \tag{3}$$

where $\boldsymbol{y}$ is a one-hot encoding vector and $\boldsymbol{p}$ the probability vector of whether two images belong to the same class or not.

3 Experiments

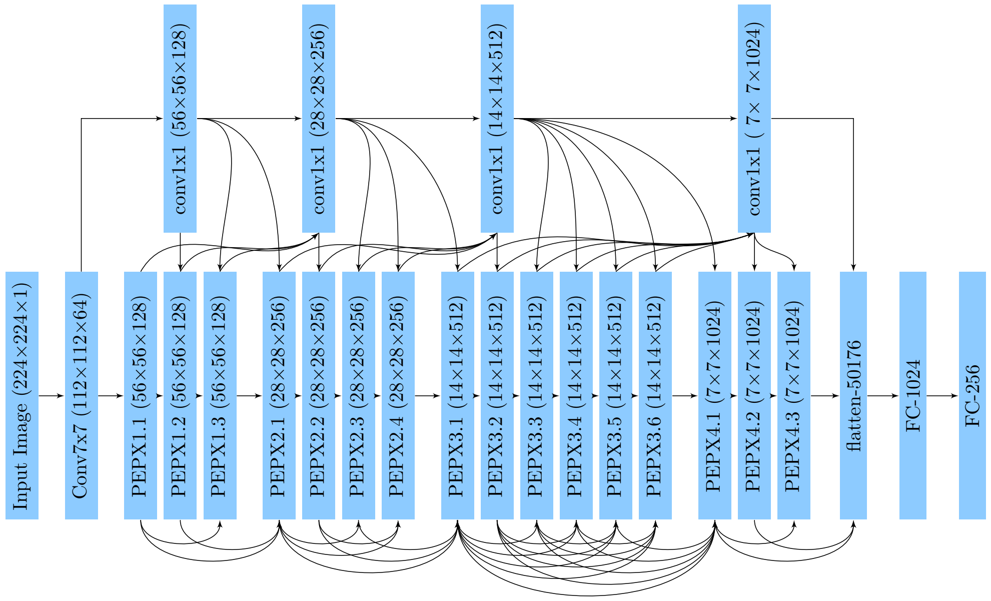


Figure 4: The architecture of our implementation of COVID-Net and similarly a component in the siamese network. ReLU activation is applied to all layers and down-sampling is done by 2×2 max-pooling.

In order to test our hypothesis regarding the siamese network structure, we benchmarked a "siamesified" implementation of COVID-Net against a standalone implementation of the same network – from now on referred to as *single-net* – whose architecture is illustrated in Figure 4.

3.1 Dataset

The experiments were performed on the dataset COVIDx, which consists of a collection of X-ray images from [7, 8] as well as from the *Figure 1 COVID-19 Chest X-ray Dataset Initiative* [22]. The dataset defines three classes; lungs infected with COVID-19, lungs infected with non-COVID-19 pneumonia, and healthy lungs. We used the scripts available in the original COVID-Net repository[3] for collecting the data. For validation data, we extracted 20% of the training data at random for each class. The distribution of samples for each class can be seen in Table 1.

The images were all pre-processed by

1. Converting to gray-scale.
2. Scaling down to 224×224 pixel images using the OpenCV inter-area interpolation.
3. Dividing all individual pixels by 255 to set the maximum pixel value to 1.
4. Normalizing by subtracting and dividing by the global mean and standard deviation of the training set, respectively.

For the siamese network we need pairs of images as input, so for each training epoch we extracted all COVID-19 images available for training and randomly sampled the same amount of images from the normal as well as the pneumonia dataset. This measure was taken to compensate for the imbalance between available data for each class. Due to the large amount of same and different class pairs that could be created, we limited the data-generation to 5000 pairs that were sampled at random during validation for each epoch, and increased to 12000 pairs when training the final network. An equal amount of same and different class pairs for each epoch was used.

[3] Linda Wang. `COVID-Net`. `https://github.com/lindawangg/COVID-Net`. 2020.

For training the single-net, a similar approach was used; we extracted all COVID-19 training images and randomly sampled the same amount of images from the normal as well as the pneumonia dataset each epoch, effectively using a balanced dataset of 537 images for each epoch.

Table 1: Distribution of samples for each class.

	COVID-19	**Normal**	**Pneumonia**
Training	179	6373	4361
Validation	44	1593	1090
Testing	31	885	594

3.2 Experimental Setup

As mentioned in section 2.2, the microarchitecture of each PEPX module was not available, so we opted for the following parameters for the number of filters in each PEPX module, based on the number of channels in the input `nf`, fulfilling the PEPX criterion:

$$\texttt{nf} \xrightarrow{\text{proj.}} 0.5 \cdot \texttt{nf} \xrightarrow{\text{exp.}} 0.75 \cdot \texttt{nf} \xrightarrow{\text{DW.}} 0.75 \cdot \texttt{nf} \xrightarrow{\text{proj.}} 0.5 \cdot \texttt{nf} \xrightarrow{\text{exp.}} \texttt{nf}_{\texttt{out}}, \tag{4}$$

where $\texttt{nf}_{\texttt{out}}$ is the number of channels in the output. Please refer to the source code for details[4]. Both networks were implemented as TensorFlow[5] models using the Keras[6] framework and were trained using backpropagation [23] with mini-batch gradient descent, and adaptive moment estimation (Adam) [24] for optimizing the learning rate during training. For the single-net, we added a last Softmax layer with three units to obtain accuracies for each class and we used cross entropy as the loss function. We also applied a dropout scheme on the fully connected layers for the single-net, using drop-out rates 0.5, 0.2, and 0.2.

We also made use of GradCAM [25] to produce a coarse heat map highlighting the essential regions of an image that corresponds to the prediction of the intended class. This was done in order to facilitate the understanding of the critical features of the image that the model uses to draw its conclusions.

3.2.1 Three-way Validation for Siamese Networks

Following the same methods as proposed by Koch et al. [20], we use a three-way validation method for the siamese network, where the class of an image with corresponding feature vector $\boldsymbol{x}$ is predicted by comparing it to three other validation images (one for each class), with feature vectors $\boldsymbol{x}^{(c)}$, $c = 1, 2, 3$, and choosing the class that maximizes the similarity.

Using the validation set, we created 129 such different three-way validation trials, and calculated the validation accuracy as the percentage of correct classifications over all 129 trials.

3.2.2 Testing the Siamese Network

Since the siamese network is trained to predict similarity between images, a relevant question is which images you should compare with when testing the network. Following a similar approach as in [19], we created a representation of each class by

1. Calculating the feature vectors outputted by the siamese network when run on every training image for each class.
2. Calculating the average of all feature vectors for each class.

This produces a feature vector template for each class, and when presented with a test image, we calculated the feature vector outputted by the network when run on the test image, and then obtained the similarity between the test image and each class template by using the distance measure (eq. (1)) and the sigmoid layer. Using these scores, the class could then finally be predicted as the one with the highest similarity.

3.3 Experimental Results

We first conducted a grid search for sensible values of initial learning rate η and batch size β, with $\eta \in \{10^{-3}, 10^{-4}, 10^{-5}\}$ and $\beta \in \{8, 16, 32\}$. For the siamese network, the grid search was performed on 5000 training pairs per epoch for 20 epochs, and for the single-net we used 100 epochs. The results of this search can be seen in Table 2.

[4] Source code is available at `https://gits-15.sys.kth.se/rosun/DD2424-Project`.

[5] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. `https://www.tensorflow.org`. 2015.

[6] François Chollet et al. *Keras*. `https://keras.io`. 2015.

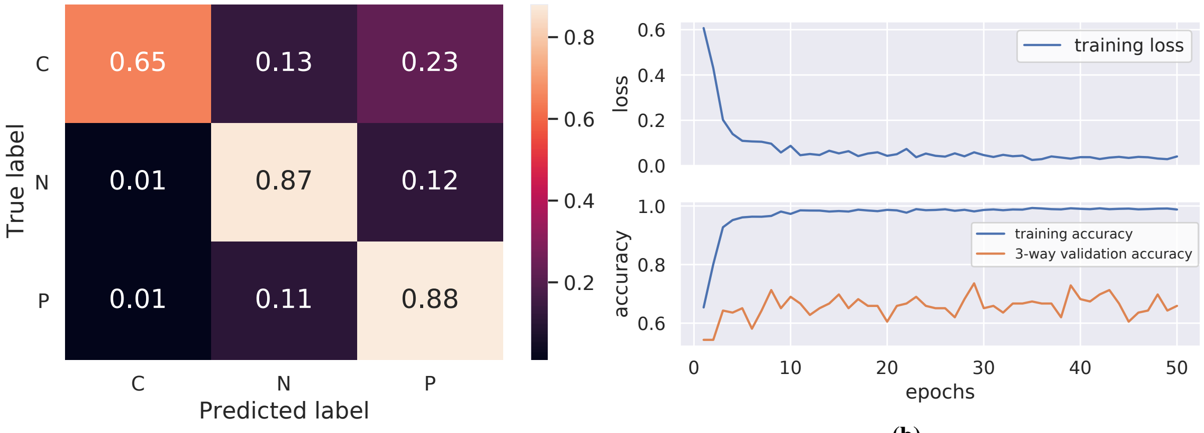
Table 2: Best values for hyperparameters η and β as well as the achieved maximum validation accuracy.

Network	best η	best β	Maximum validation accuracy
Siamese	1e-4	32	0.736
Single	1e-4	16	0.795

We then retrained both the siamese and the single-net using the best η and β found during the grid search. For this final training run, we used 12000 training pairs per epoch for 50 epochs for the siamese network, and the single-net was trained for 100 epochs. For each run, we saved the model that achieved the highest validation accuracy. The resulting training plots can be seen in Figures 5 and 6. Finally, we used the saved best performing networks to calculate the final test accuracy on the test set, shown in Table 3. The confusion matrices for the test data can be seen in Figures 5 and 6. One interesting thing to note is that the final test accuracy for the siamese net was about 13 p.p. higher than the best validation accuracy. This is probably because the three-way validation task may be harder than the final testing. Each trial in the three-way validation is only based on measuring the similarity between the validation image and three other images, while in the testing we measured the similarity between a test image and an average over all training feature vectors produced by the final network for each class.

Table 3: Final test accuracy after training for both networks.

	Final test accuracy
Siamese	0.87
Single	0.81



(a)

(b)

Figure 5: Results for the siamese network: **(a)** Confusion matrix for the test data, where C, N, P indicate COVID-19, Normal and Pneumonia, respectively; **(b)** Loss and 3-way validation accuracy on training and validation data.

3.3.1 GradCAM

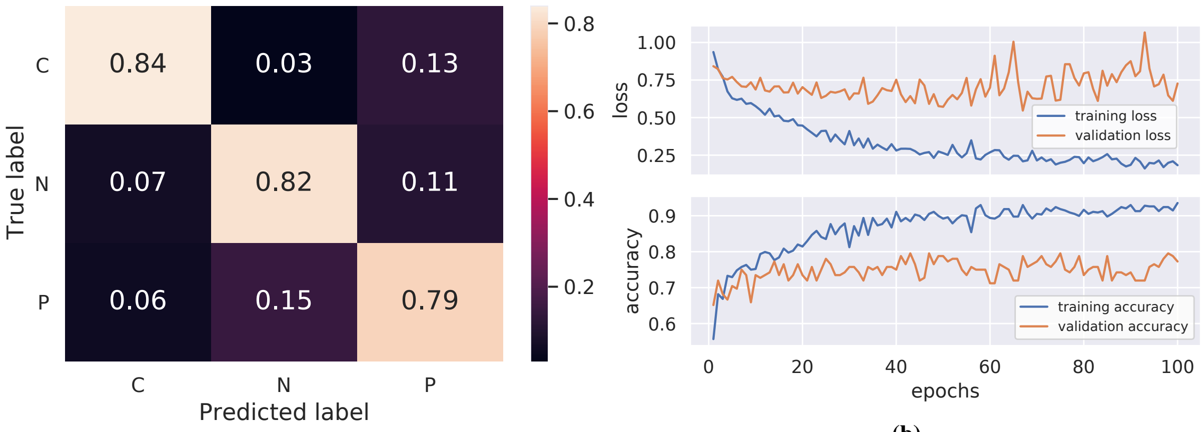
To check if the network had learned relevant features during training we employed GradCAM[7] on the test set to create different heat maps. This way we could see if the network had learned the features of interest during training. Due to time constraints, we used GradCAM only on the single-net.

The best performing single-net was then once again fed with some images from the test set, sampled at random, to produce heat maps. When analyzing the heat maps, the results were quite mixed; in some cases, the predictions seemed to be based on relevant parts of the input images, for example the lung region, as can be seen in Figure 7, while in other cases, the predictions seemed to be entirely based on irrelevant artifacts, as can be seen in Figure 8. To try and solve the issue of having predictions based on irrelevant artifacts, a second set of experiments were conducted.

Since the artifacts on the images were mostly present around the border, we decided to do additional pre-processing by cropping the images. We then retrained the single-net on the cropped images with the same hyper-parameters found during the validation phase. The result can be seen in Figure 7 and Figure 8.

Looking at Figure 8 we can see an improvement; more relevant features are highlighted, while when looking at Figure 7 the opposite is true. The results obtained during the GradCAM experiments for the COVID-19 class can be extended to the other classes, in the sense that similar behaviours were observed when classifying and running the

[7] Vitali Petsiuk. `keras-gradcam`. https://github.com/eclique/keras-gradcam. 2019.

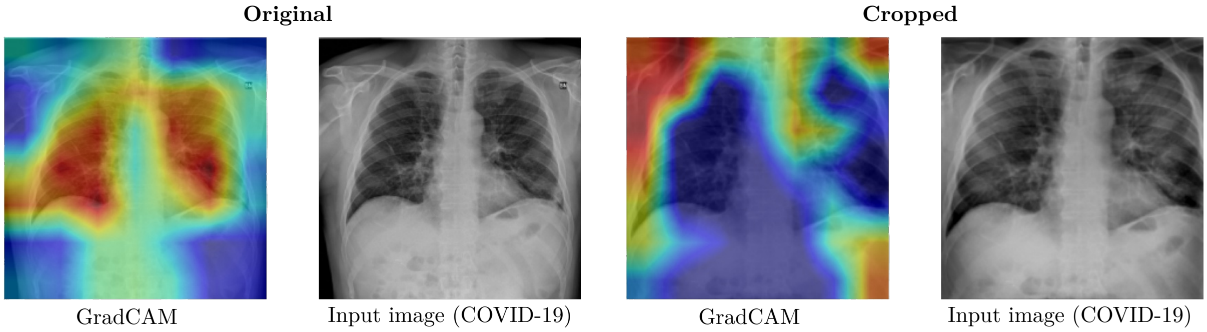


(a)

(b)

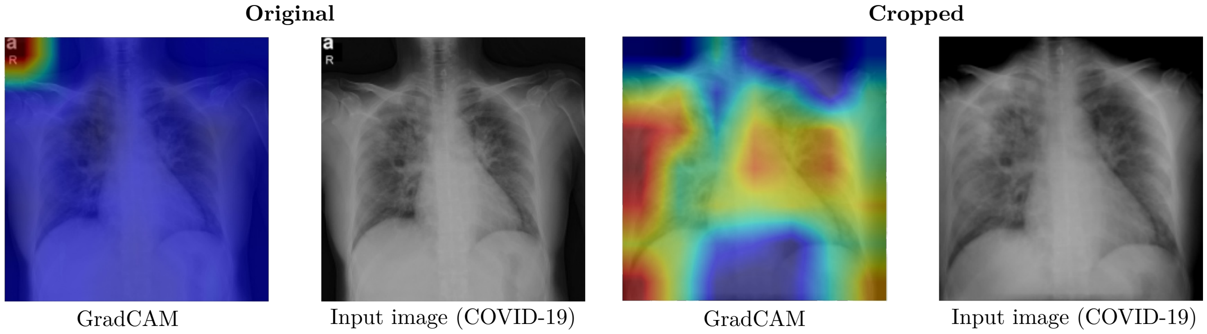
Figure 6: Results for the single-net: **(a)** Confusion matrix for the test data, where C, N, P indicate COVID-19, Normal and Pneumonia, respectively; **(b)** Loss and accuracy on training and validation data.

GradCAM analysis on Normal images and Pneumonia images. The test accuracy achieved for the single-net trained on cropped images was 78%, i.e. slightly lower than for the non-cropped images.



GradCAM   Input image (COVID-19)   GradCAM   Input image (COVID-19)

Figure 7: GradCAM result for two experiments on the same test X-Ray image



GradCAM   Input image (COVID-19)   GradCAM   Input image (COVID-19)

Figure 8: GradCAM result for two experiments on the same test X-Ray image

4 Conclusions

In this work, a siamese network based on the COVID-Net [6] was implemented. The intention of the siamese architecture was to overcome the shortage of publicly available X-Ray images of COVID-19, while achieving a network that can correctly identify three different classes: COVID-19, Normal, and Pneumonia. To validate the performance of the proposed method, a non-siamese network based on COVID-Net was implemented and used for comparison. The performance of both networks were then measured by calculating the accuracy achieved when making predictions on the final testing set. The siamese network achieved an accuracy of 87 %, while the single-net achieved an accuracy of 81 %. When looking at the distribution of accuracies over all classes, the single-net performed better on the COVID-19 class while the siamese network had better accuracy when classifying both healthy and pneumonia images. Both networks had a tendency to misclassify COVID-19 as pneumonia, which may be expected since the two diseases probably result in similar X-ray images.

As discussed in the previous section, the use of GradCAM allowed the verification of the relevant features being learned by the single-net. The GradCAM analysis revealed that in some cases, the classification is based on relevant features, but in other cases it is based on irrelevant artifacts near the image borders. The analysis demonstrates the importance of a clean dataset; extra care needs to be taken in the pre-processing of images to reduce undesired artifacts. Therefore, our obtained results may not be completely reliable.

We also investigated the performance of the single-net with some artifacts removed by cropping the images, which resulted in a similar test accuracy. When analyzing the GradCAM results on the single-net trained on cropped images, we saw that this increased the amount of relevant features learned in some cases, while in some cases the results were worse. Due to time constraints, a thorough investigation was not made.

To improve on our work, further experimentation is needed on a properly pre-processed dataset, using a more robust technique to address innate artifacts on the raw images. Furthermore, a more thorough hyperparameter search could be conducted, both for the learning rate, batch size, the amount of filters in each PEPX module, as well as the amount of dropout used on the single-net's dense layers, preferably with an increased dataset, as new X-ray images become available. Finally, a GradCAM analysis on the siamese network could be performed to get a better understanding of the difference in performance between the two networks.

Author Contributions: All authors contributed equally to this work.

References

[1] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (Dec. 2017), pp. 60–88.

[2] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (2017), pp. 115–118.

[3] Alejandro Rodriguez-Ruiz et al. "Stand-Alone Artificial Intelligence for Breast Cancer Detection in Mammography: Comparison With 101 Radiologists". In: *JNCI: Journal of the National Cancer Institute* 111.9 (Mar. 2019), pp. 916–922.

[4] N. Wu et al. "Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening". In: *IEEE Transactions on Medical Imaging* 39.4 (2020), pp. 1184–1194.

[5] Qin Sun et al. "Lower mortality of COVID-19 by early recognition and intervention: experience from Jiangsu Province". In: *Annals of Intensive Care* 10.1 (2020).

[6] Linda Wang et al. *COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images*. 2020.

[7] Joseph Paul Cohen et al. "COVID-19 image data collection". In: *arXiv 2003.11597* (2020). URL: https://github.com/ieee8023/covid-chestxray-dataset.

[8] Pranav Rajpurkar et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning". In: *arXiv preprint arXiv:1711.05225* (2017).

[9] Brenden Lake et al. "One shot learning of simple visual concepts". In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 33. 33. 2011.

[10] Khalid El Asnaoui et al. *Automated Methods for Detection and Classification Pneumonia based on X-Ray Images Using Deep Learning*. 2020.

[11] Lin Li et al. "Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct". In: *Radiology* (2020), p. 200905.

[12] Jane Bromley et al. "Signature Verification Using a "Siamese" Time Delay Neural Network". In: *International Journal of Pattern Recognition and Artificial Intelligence* 7.4 (1993), pp. 669–688.

[13] Y. Taigman et al. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 1701–1708.

[14] Zhi-Yong Gao et al. "Spatial-Structure Siamese Network for Plant Identification". In: *International Journal of Pattern Recognition and Artificial Intelligence* 32.11 (2018), p. 1850035.

[15] Yunhua Zhang et al. "Structured Siamese Network for Real-Time Visual Tracking". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 355–370.

[16] I. Melekhov et al. "Siamese network features for image matching". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. Dec. 2016, pp. 378–383.

[17] Yang Li et al. "Deep Discriminative Supervised Hashing via Siamese Network". In: *IEICE Transactions on Information and Systems* E100.D.12 (2017), pp. 3036–3040.

[18] B. K. Barakat et al. "Word Spotting Using Convolutional Siamese Network". In: *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. 2018, pp. 229–234.

[19] S Chopra et al. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE, 2005, 539–546 vol. 1.

[20] Gregory Koch et al. "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. Lille. 2015.

[21] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015.

[22] COVID-Net Team. *Figure 1 COVID-19 Chest X-ray Dataset Initiative*. https://github.com/agchung/Figure1-COVID-chestxray-dataset. 2020.

[23] David E. Rumelhart et al. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), p. 533.

[24] Diederik P. Kingma et al. *Adam: A Method for Stochastic Optimization*. 2014.

[25] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128 (2019), pp. 336–359.