Project: COVID-19 Assisted Diagnosis

Frie Van Bauwel[1] Lotte Van de Vreken[2] Marcin Jedrych[3] Xueting Li[4]

[1,2,3,4] Master of Science in statistical data analysis - computational statistics

I. Introduction

Within this project on medical image classification, two deep learning models are trained to classify X-ray images of lungs as COVID-positive or COVID-negative images. Further, the Grad-CAM visualization technique is used to be able to show on which areas of an X-ray image are mainly used by the algorithm to decide whether an image shows a person affected with COVID or a person not affected with COVID. This report describes first how the data are explored, preprocessed and augmented (section II). It then explains how a baseline model, using a convolutional network, was fitted (section III) and how transfer learning was used to make use of the pre-trained ResNetV2 model (section IV). Section V describes in which the Grad-CAM visualisation is used to understand why missclassifications happen in the best performing model trained in the previous tasks. After the conclusion (section VI), the report describes how this project was divided between all group members (section VII). The final section entails our use of generative AI during the project (section VIII).

II. Task 1: Data Exploration, Pre-Processing and Augmentation

As a first step, a data exploration was carried out. The dataset consists out of 1600 training images, 400 validation images and 200 test images. The amount of training images can be considered as quite limited, hence a risk of overfitting exists. Other potential challenges for automatic classification that become clear with having a first look at the images are the subtle differences in COVID versus normal chest X-rays, as well as the variability in brightness, contrast, and the exact part of the lungs shown on the X-ray between pictures. Both challenges are addressed by using normalization and augmentation techniques as described below. Based on the pixel statistics and label distribution, the datasets appear to be fairly uniformly divided. The mean values and standard deviations of the pixel values are comparable across the sets: training (mean = 0.53, std = 0.25), validation (mean = 0.55, std = 0.25), and test (mean = 0.56, std = 0.26). This finding indicates that the image intensity is consistent. The class distribution in all three sets also looks well balanced. There are no consistent differences in image quality between the COVID and normal X-ray images. However, it can be observed that some pictures have a much higher intensity distribution than others, creating the need for proper normalization. In some pictures, artefacts like wires could be seen, which can lead to lower model performance. To reduce the training time and memory usage, the pictures were downsampled from 299x299 to 128x128 resolution. This allows for faster runtimes without losing too much information. Additionally, it makes the picture more general thus reducing the risk that the model will overfit on small artefacts in the pictures. The images were normalized using dataset statistics: the mean and standard deviation of all pixels in the training, validation and test dataset were used as normalization statistic. The X-ray images contain black-and-white images thus all three image channels contain the same information and could be normalized using the same statistic. To augment the data, pictures were slightly shifted in width, rotated with a maximum range of 15 degrees and zoomed with a maximum factor of 0.2. These augmentation patterns were chosen because they represent still realistic X-ray variations, as patients can be positioned slightly more on the side, can be a bit closer or further from the imager or can be slightly tilted. Other augmentations such as vertical or horizontal flip would create unrealistic images that would never occur in real-life examples.

III. Task 2: Building the baseline model

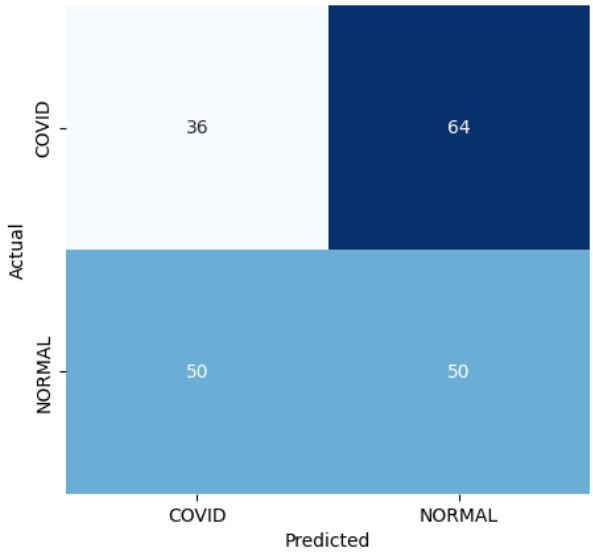

Fig. 1. Confusion matrix of the test data classified by the final baseline model.

IV. Task 3: Transfer Learning

Within task 3, a neural network is built using the transfer learning approach. In other words, a pretrained model is used as a basis to built a model, hereby making it possible to use a network with many layers even though a limited set

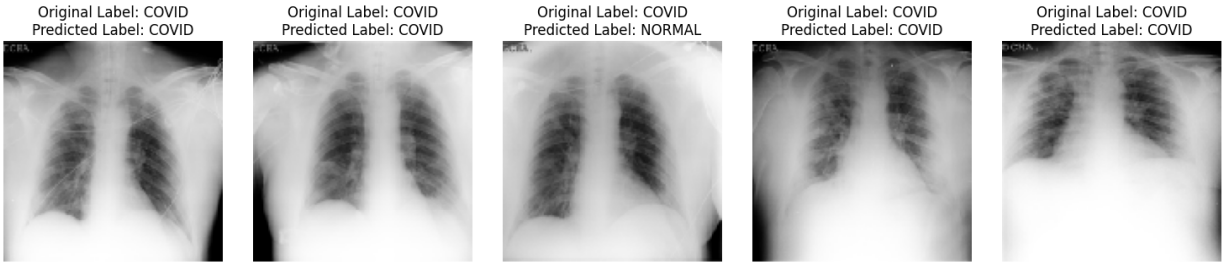

Fig. 2. Sample of test images with the real and predicted label, as classified by the final baseline model.

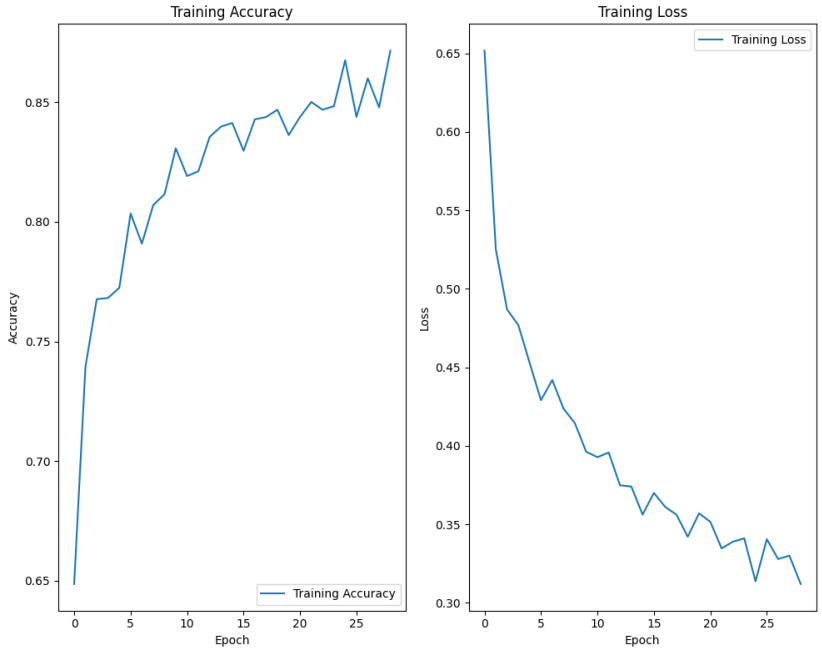

Fig. 3. The training curves of the loss and accuracy of the final baseline model where the model was trained based on the combination of validation and training data.

of training data is available. In this case, the ResNet50V2, a model consisting of 50 layers, is used as the pretrained model. After the ResNet50V2 model, four extra layers are added. First, a GlobalAveragePooling2D layer is added to reduce the spatial dimensions. Then, a 128-node dense layer with relu-activation is added in combination with a dropout layer to reduce risk of overfitting. A final single-node dense layer with sigmoid activation was used to get the binary output.

After setting the model architecture, the hyperparameters of the added layers are tuned using a general grid search. The tuned parameters consist of the batch size, learning, and dropout rates. Table XXXXX shows the XXranges/combinations that are used during the hypertuning process and indicated the optimal outcomes for each hyperparameter. While tuning the hyperparameters, the pretrained parameters in the ResNet50V2-layers remain unchanged. Early stopping based on the validation loss is used with a patience of five steps. After this tuning step, a model is trained during XX epochs based on the training and validation data. Another, final, model is trained in which the ResNet50V2-layers are unfrozen, hence fine-tuning the weights in these layers is possible. The training curves of the model before unfreezing the ResNet50V2-layers is visualized in figure 8, the training curves of the model after unfreezing is visualized in figure 7.

The training curves of both trained models within this task look smooth and have shape coming close to the ideal training curve. The model where the ResNet50V2 model parameters are finetuned shows a higher accuracy and lower loss (respectively X and X in the fully trained model) than the model without the extra finetuning (respectively X and X in the fully trained version). This hints towards better performance of the finetuned model, but might also indicate a bigger chance of overfitting on the training dataset (consisting of the provided training and validation set). As requested in the task, no training of this model was done using the validation set as holdout-samples, hence this is hard to judge. Comparing these curves to the training curve in task 2, XXXX

As a final step in this task, the finetuned model is eval-

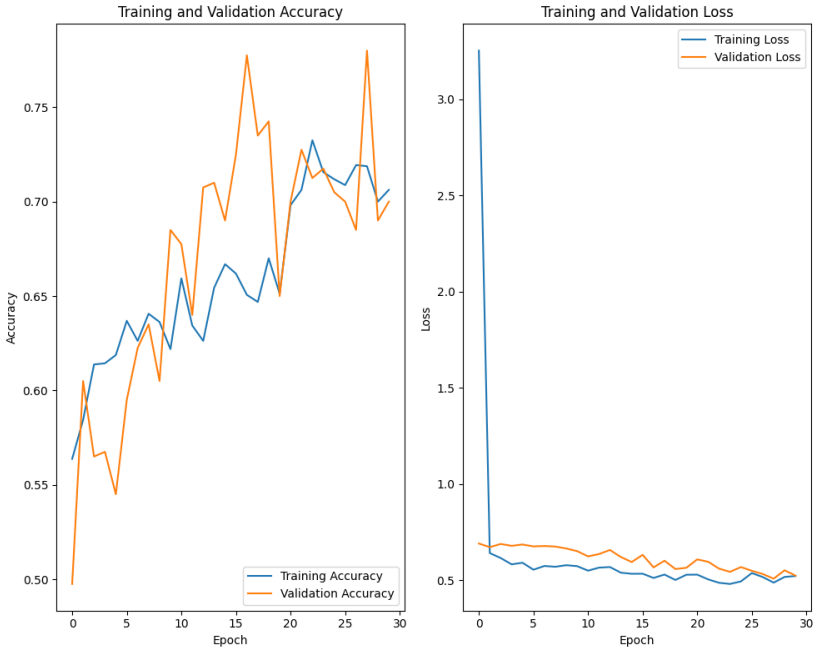

Fig. 4. The training and validation curves of the loss and accuracy of the initial baseline model before the hyperparameter tuning.

uated using the test set. Both the confusion matrix (5) and the calculated accuracy of XX shows a major improvement compared to the baseline model. This increase in performance is probably due to the transfer learning approach which caused a smoother learning curve that converged faster to decent results with only a limited increase in learning time, even with limited training data. However, still XX out of XX samples are misclassified as COVID and XX samples are misclassified as normal. This indicates a big difference in performance compared to the training data, which implies overfitting. Even though the added layers on top of the pretrained ResNet50V2 are meant to restrict this problem. One way to alleviate this overfitting is to introduce bigger transformations in the training set, or add more training data if possible. Alternatively, starting with a lower resolution can also help. Another way to improve the results is using a pretrained model specifically trained on medical images such as CheXNet instead of ResNet50V2 as a basis of the transfer learning.

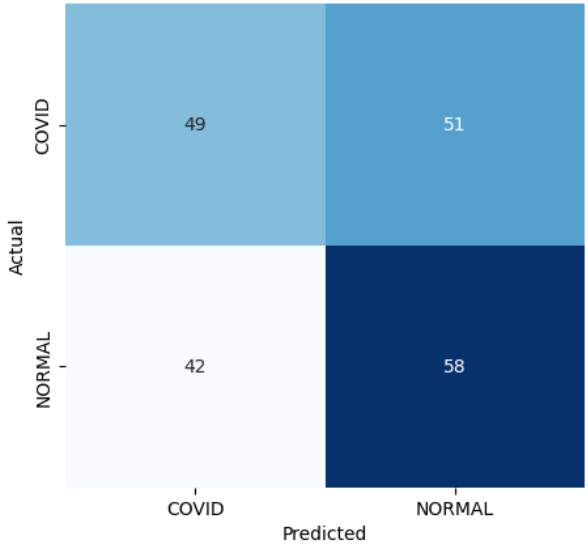

Fig. 5. Confusion matrix of the test data classified by the final transfer model (with tuned hyperparameters and fine-tuned weights of the ResNet50v2 model).

V. Task 4: Explainability through Grad-CAM

To better understand the decisions made by our COVID-19 classifier, we did a Gradient-weighted Class Activation Mapping (Grad-CAM). Grad-CAM is a technique that highlights the regions of an input image that contribute most strongly to a model's decision. It works by computing the gradient of a target class score with respect to the activations of the final convolutional layer. These gradients are averaged spatially to obtain importance weights, which are then combined with the activation maps to generate a heatmap that can be overlaid on the original image. Initially, the model's scalar output directly indicated a binary decision. However, Grad-CAM requires class-specific outputs. Therefore, we replaced the final layer with a dense layer producing two logits—one for each class (COVID and normal). After this adjustment, the model generated separate scores for both classes, with the predicted class corresponding to the higher score. As a result, the fundamental decision-making process of the model remained unchanged. Applying Grad-CAM to test samples showed that, when focusing on the COVID-19 class, the visualizations predominantly highlighted the lung regions—consistent with clinical expectations, given that lung opacities are a key

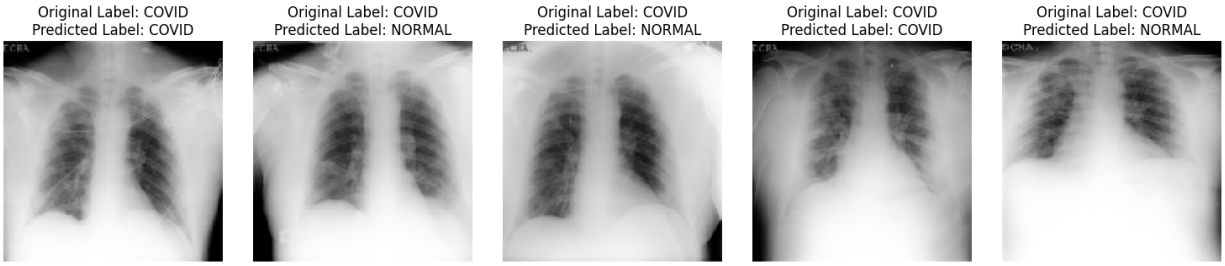

Fig. 6. Sample of test images with the real and predicted label, as classified by the final transfer model.

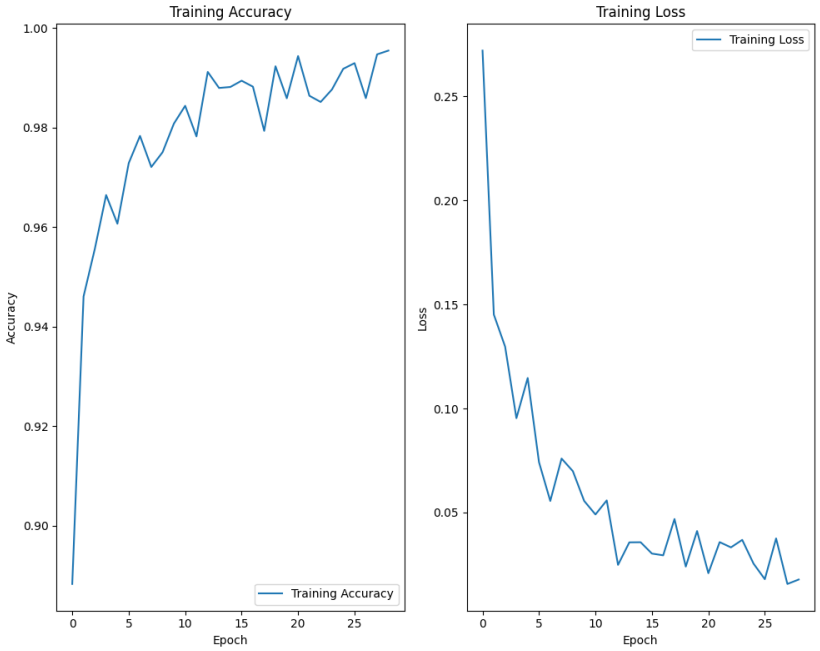

Fig. 7. The training curves of the loss and accuracy of the final transfer model with tuned hyperparameters and weights of the ResNet50v2 where the model was trained based on the combination of validation and training data.

indicator of COVID-19 on chest X-rays. However, Grad-CAM also revealed potential model biases: in some cases, activations appeared outside the lungs, often around rectangular artifacts. These findings highlight that the model may have learned spurious, non-disease-related features, which underlines the purpose of Grad-CAM in explaining and evaluating the model's behavior. Our Grad-CAM implementation follows the Keras official tutorial [**?**], adapting it to our model. Using TensorFlow's `GradientTape`, we computed the gradient of the selected class score with respect to the convolutional feature maps. The gradients were averaged over the spatial dimensions to weigh the importance of each feature channel, and the resulting weighted activation maps were aggregated to form the Grad-CAM heatmap. Finally, the heatmap was normalized between 0 and 1 for visualization.

Question 23 & 24

VI. Conclusions

VII. Author contributions and collaboration

In first instance, the tasks were divided as follows:

- Task 1: Marcin Jedrych and Frie Van Bauwel,
- Task 2: Marcin Jedrych and Xueting Li,
- Task 3: Lotte Van de Vreken and Frie Van Bauwel,
- Task 4: Marcin Jedrych and Lotte Van de Vreken.

According to this division, everyone made sure there was a first version of the code for their assigned task. Afterwards, questions and unclarities in each task were discussed, after which everyone checked and complemented the first draft of the code of each task. Task 4 was kept until the other tasks were almost done. For the text, first an answer to the questions was formulated by the assigned people to each task, after which everyone Read them and complemented where the need was felt. Some questions were discussed at length before an answer was formulated.

VIII. Use of Generative AI

Generative AI was used to fix bugs in the code for all tasks except task 1. It was also used to get a first version of some parts of the code to plot the images for all tasks except task 1.

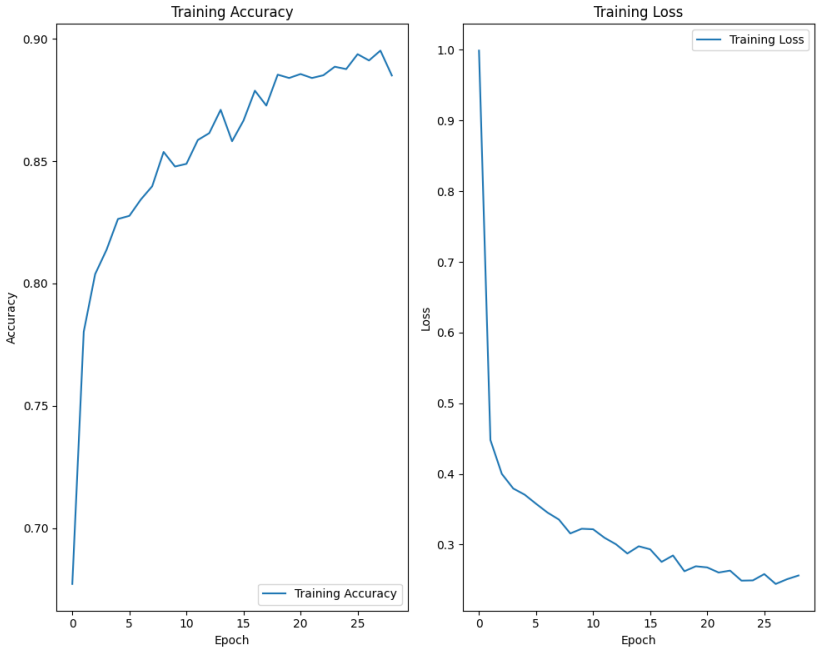

Fig. 8. The training curves of the loss and accuracy of the final transfer model with tuned hyperparameters, but the original weights of the ResNet50v2, where the model was trained based on the combination of validation and training data.

Suggested code was however always looked at critically and never taken over one-on-one.

A. *Figures and Tables*

a) Positioning Figures and Tables:

REFERENCES

Example References:

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, "Title of paper if known," unpublished.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

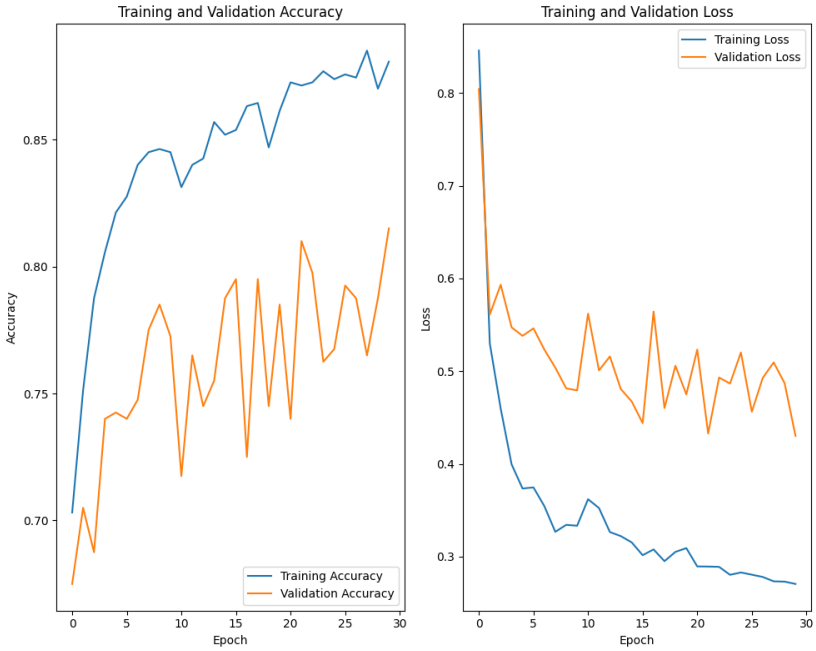

Fig. 9. The training and validation curves of the loss and accuracy of the initial transfer model.

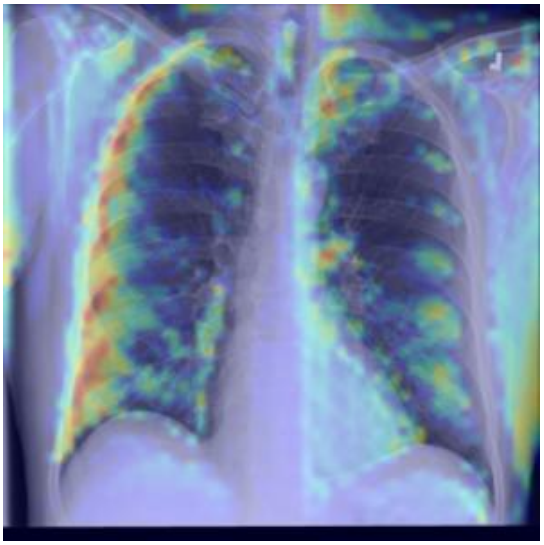

Fig. 10. Example Grad-CAM heatmap highlighting lung regions associated with a COVID-positive classification.