Deep Learning project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Chest X-Ray Images

Diogo Peixoto - 20210993
João Costa - 20211005
Gonçalo Gomes - 20211007
Gabriel Avezum - 20210663
Danilo Arfelli - 20211296

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Index

1. Introduction

Deep learning is a subset of machine learning, which is essentially based on artificial neural networks. These neural networks attempt to simulate the behavior of the human brain, allowing it to “learn” from large amounts of data. Nowadays it's being widely used in different fields such as self-driving cars, news aggregation, fraud detection, virtual assistants, visual recognition, language translations, medical image analysis and photo descriptions.

In this project the usage of deep learning methods is applied on medical imaging exams, chest X-rays are frequently requested by healthcare professionals to assess the presence of thoracic diseases, due to its low-cost and non-invasive nature. For this reason, providing a second opinion using deep learning methods is an attractive solution to increase the productivity and efficiency in the interpretation of these exams and give the possibility to clinicians to manage their time and resources.

The dataset was available in Kaggle [1] with chest images with normal status, virus pneumonia and bacteria pneumonia. In kaggle they build a deep learning model classifying the chest image in a binary outcome, normal status or pneumonia (virus or bacteria). This project tried to incorporate some ideas trying to go further as creating a multivariate outcome as normal status, virus pneumonia or bacterial pneumonia.

Illustrative Examples of Chest X-Rays in Patients with Pneumonia. The normal chest X-ray (left panel) depicts clear lungs without any areas of abnormal opacification in the image. Bacterial pneumonia (middle) typically exhibits a focal lobar consolidation, in this case in the right upper lobe (white arrows), whereas viral pneumonia (right) manifests with a more diffuse “interstitial” pattern in both lungs [1].

2. Data set

The data set source can be found in Kaggle [1] and it contains 5859 chest images. Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children’s Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients’ routine clinical care [1].

The images were uploaded to google drive and google colab was used to develop the model. The images data were split in train, test and validation. For the train data set we have 4000 images divided in 1000 normal status, 1000 Virus pneumonia and 2000 for bacteria pneumonia. For the test data set it is composed of 800 chest images divided in 300 normal status, 201 virus pneumonia and 300 for bacteria pneumonia. The validation data set is composed of 1056 images divided into 283 normal status, 294 virus pneumonia and 480 bacteria pneumonia. We can summarize the split data set in the table below.

	Train	Test	Val	Total
Normal	1000	300	283	1583
Virus	1001	201	294	1496
Bacteria	2000	300	480	2780
Total	4001	801	1057	5859

3. Model development

We have started with a simple model, with only two convolution layers and trained with images of size 150x150. The graph curves of loss and accuracy depicted in graph 1, in appendix, display *overfitting*. The training accuracy increases, until it reaches nearly 100% on epoch 12, whereas the validation accuracy stalls at around 70%. The validation loss increases linearly over time, whilst the training loss keeps decreasing until it reaches nearly 0.

Based on the above, the following model iteration has included the dropout technique to help mitigate overfitting. The dropout rate has been set to 0.50. The plots displayed in graph 2,in appendix, still show a similar behavior to the model number one described above, with the clear presence of overfitting. The only difference is the training time, with the training accuracy reaching nearly 100% on epoch 22 rather than on epoch 12.

Taking that into account and considering the small-medium training dataset size, the next model introduces the data augmentation technique , which is specific to computer vision problems. The idea behind is to create more training samples via a number of random image transformations, applied to the original ones, which include the following ones defined by ourselves:

- rotation_range=40, meaning the new images are rotated in a value between (0-40)
- width-shift range=0.30 sets the new images width as a fraction of the original ones
- zoom range=0.20 is for zooming inside pictures
- horizontal_flip=True, refers to flipping half the images horizontally

The model 3 resulting plots display a different output. Now both the training and validation loss decrease linearly over time. On top of that, the training accuracy reaches 75% after the 30 epochs, with similar values upon the validation accuracy. We might conclude the overfitting issue is sorted out.

The goal now is to change different hyperparameters and test different topologies to improve the results.
The first step is to identify the impact of reducing the size of the image that will be introduced, so the parameter target_size is changed to (80,80). The output of the model is presented in the graph 4 and comparing this to the previous model we can see that the validation is more stable and the performance didn't get worse, so this parameter will be changed.

Now we will try different topologies to understand how our data will perform with these alterations. The first alteration is to add one layer of convolution, so the new model will have a total of three convolutional layers with max pooling layers between them. Graph 5 has the output of the model and we can check that the train accuracy and loss presents a little improvement from the model with two layers, but the accuracy and loss for the validation presents a huge improvement. One other important information that we can extract from the graph is that the validation outperforms the train in almost all epochs.

One final topology to test will be to construct two connected convolution layers followed by a max pooling layer and repeat this one more time. So, our new model will be like Conv layer -> Conv layer -> Max pooling -> Conv layer -> Conv layer -> Max pooling. The results are presented in graph 6. The loss and accuracy are more volatile than the results presented by the graph 5, also the results are not better than the model with three layers. We decided to keep the topology presented in graph 6 because it is similar to the one presented in kaggle, so we believe that with more alterations in the parameter the results will get better.
—---- Gonçalo — modelos 7 a 10
Next, we decided to look deeper into the "color_mode" parameter, which is responsible for the coloring of the images that are presented to the network. Initially, and due to the fact that we are dealing with medical X-ray images, we decided to initialize this parameter with "grayscale". This choice was based on the belief that, in our case, the most relevant features of our images are not intensified by their coloration. Furthermore, the use of "rgb" would inevitably lead to an increase in dimensionality in our network due to the presence of 3 color input channels. Another aspect to consider is related to the consequent increase in the number of parameters to be evaluated, which is, as we will see later, directly related to the complexity and processing time of the network.
However, despite all these suspicions, we decided to test the previous topology with "rgb" because we considered that there was a possibility that some visual details would be lost when using "grayscale". The results of this model, presented in graph 7, were very similar to the previous one, with only a few small improvements in the accuracy of the validation set and a 5 minute increase in network processing, which confirmed our previous suspicions. Therefore, we made the decision to proceed with grayscale for the remaining topologies.

At this point in the Project, we decided that it would be important to reassess the data augmentation that we applied to our models. Early on, we realized the importance of this step in preventing our models from overfitting the training data and, consequently, not being able to generalize unseen data as well as intended. With this in mind, we decided to further our research on medical topics where Convolutional neural network topologies would have been applied. One of the papers found, related to the application of data augmentation in neural networks for covid-19 detection, made special emphasis on geometric transformation and the consequences of its application in x-ray images [2]. Through the results obtained, it

was clear to us that there was a strong possibility that geometrical data augmentation would not be beneficial for the performance of our models, due to the relatively static characteristics of the x-ray images we are analyzing.
Hence, we decided to reformulate our data augmentation process by eliminating parameters such as "rotation_range" and "horizontal_flip". We tested this change on the same topology outlined earlier and found that, although there was no substantial improvement in accuracy and processing time, our model was more stable (Graph 8), so we decided to continue not using geometrical augmentation.

Another method that we decided to adjust, was the dropout of our models. As we mentioned before, dropout is a regularization technique that tries to prevent neural networks from overfitting [3]. Throughout the topologies tested so far, with the exception of the first one, we always applied a constant rate of 0.5, which means that half of the layer's outputs are randomly ignored from the network. This was one of the measures we took in order to reduce the initial overfit. However, in the last models produced, we consider that the accuracy of the training set is a little lower than intended, which may be an indication that our models are not fully capturing the complexity of the images provided. For this, in the next setup, we decided to reduce the dropout to 0.2 and observe the results obtained (Graph 9). In this model, we were able to observe small improvements, in relation to the previous model, in what concerns the best result obtained, both in the training set and in the validation set. Thus, we decided to move on to the last topology tested, with this specific dropout value.

Throughout our pipeline, we always approach model architectures that we have created in order to better classify the three types of image classes present. However, in this last step, we decided to implement vgg16, a pre-trained convolutional neural network that was used to win the Imagenet competition in 2014. This neural network is much more complex at the structural level than those we have been testing, and is characterized by its enormous efficiency in image classification. When testing the model with vgg16, we apply a dense layer of 256 units, similar to what we have been doing so far, with a dropout of 0.2, based on the previous step. Analyzing the results obtained, we were able to verify a slight increase in accuracy compared to previous models, falling a little short of what was expected.
—----------

4. **Prediction**
5. **Conclusion**
6. **References**

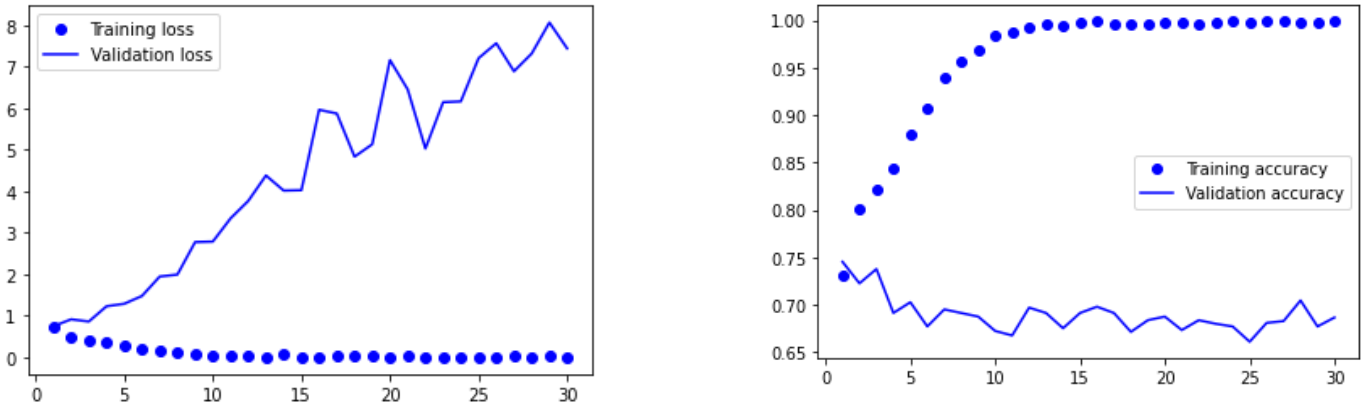
[1] https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

[2] https://www.frontiersin.org/articles/10.3389/fmed.2021.629134/full

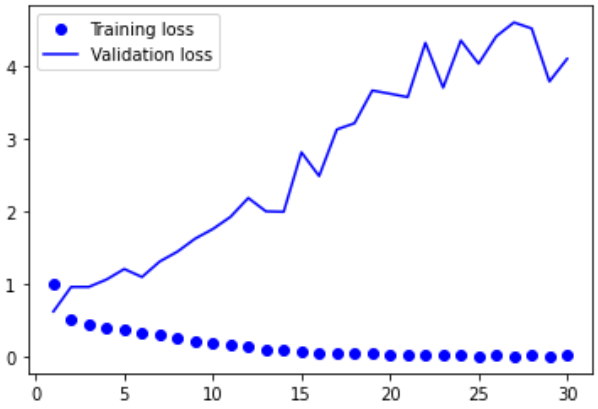
[3]https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/

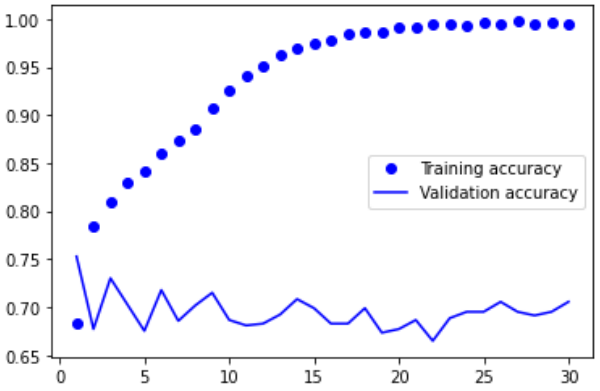
[4]https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

7. **Appendix**

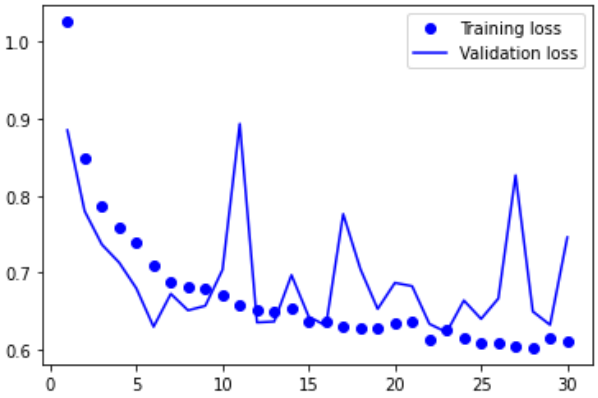


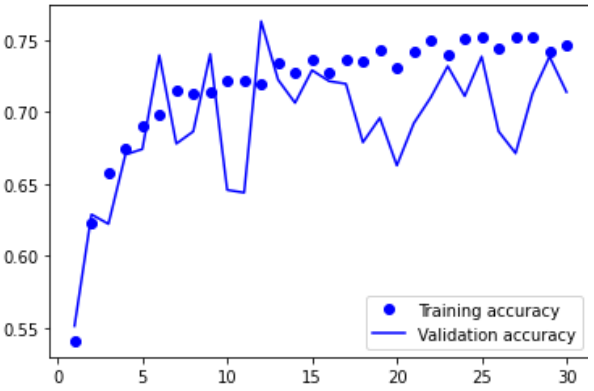
Graph 1: 2 convolution layers, (WITHOUT augmentation and WITHOUT dropout) and target size (150,150)



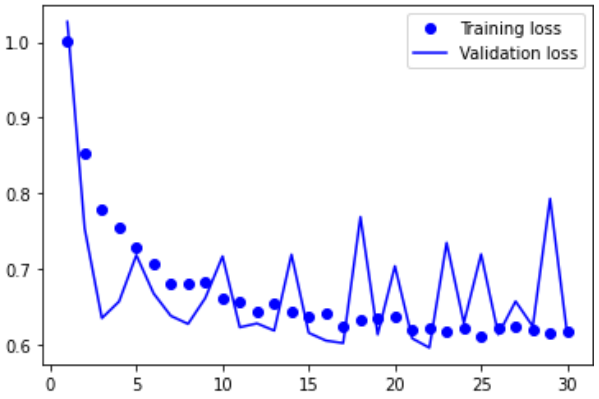


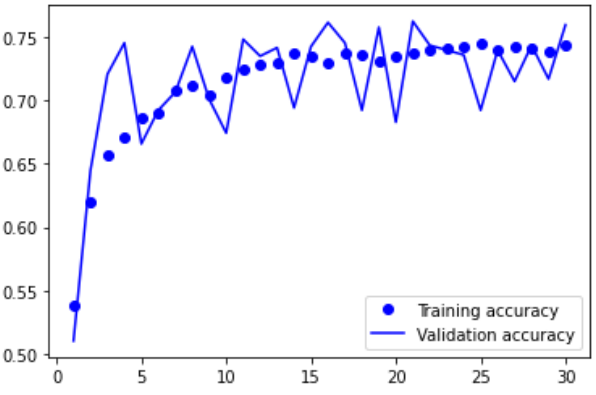
Graph 2- 2 convolution layers, (WITHOUT augmentation and WITH dropout-0.5) and target size (150,150)





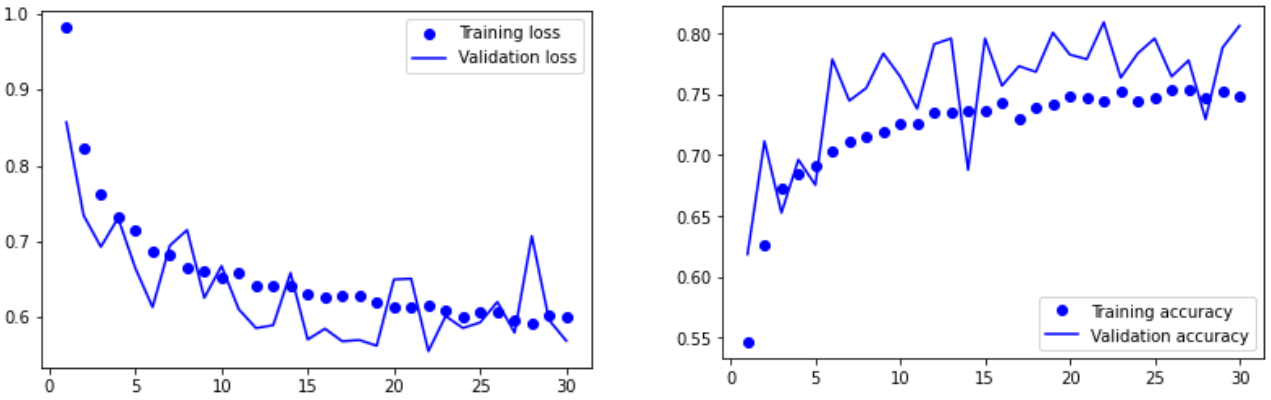
Graph 3:- 2 convolution layers, (WITH augmentation and WITH dropout-0.5) and target size (150,150)



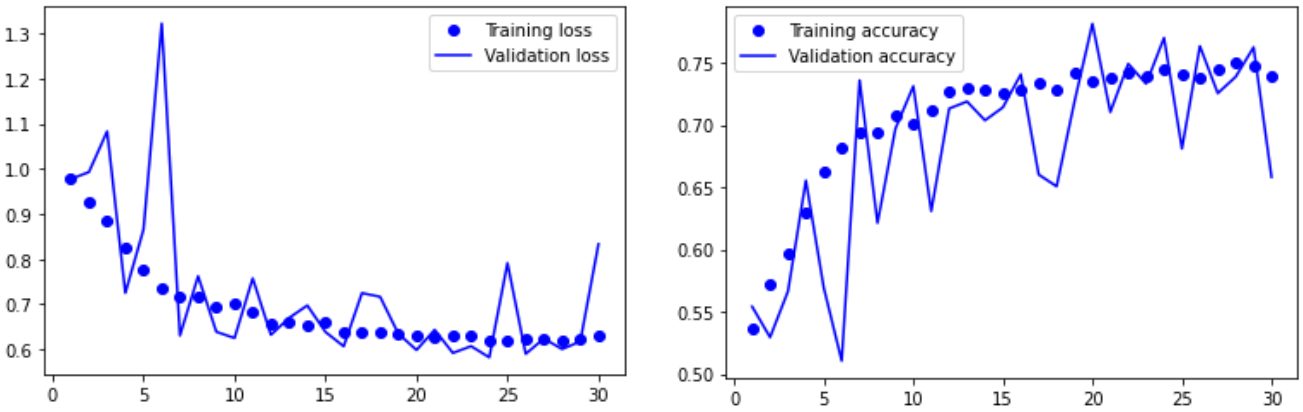


Graph 4: 2 convolution layers, (WITH augmentation and WITH dropout-0.5) and target size

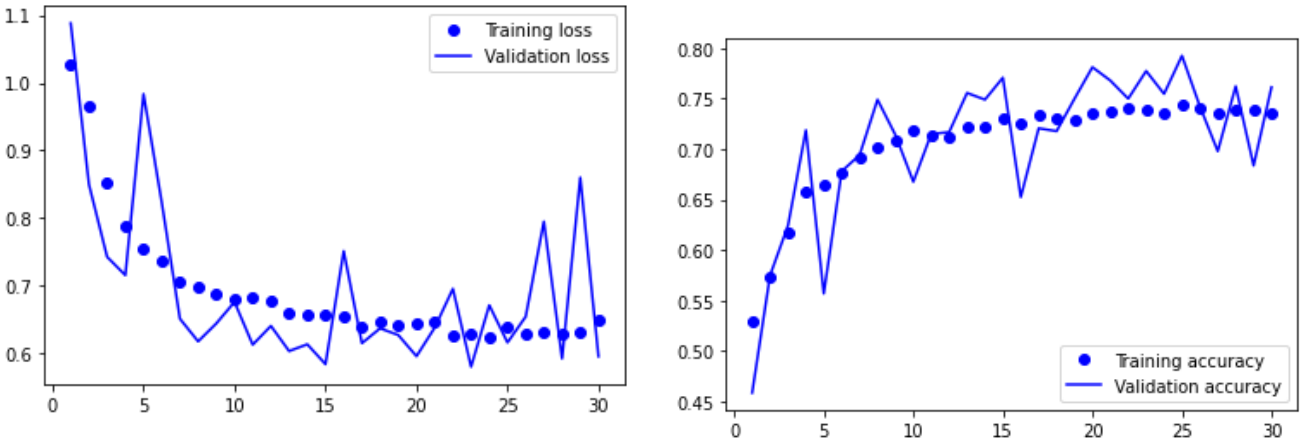
(80,80)



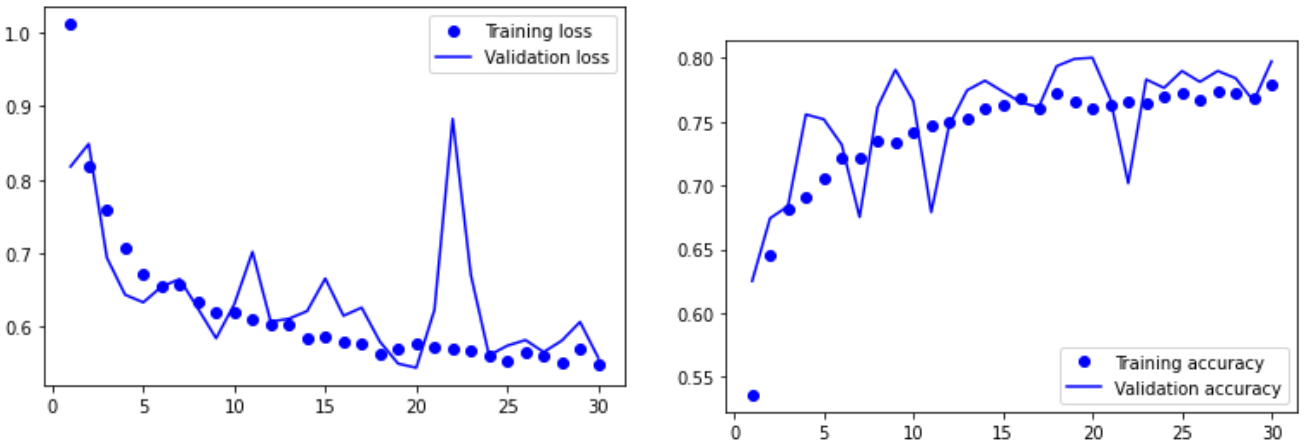
Graph 5 - 3 convolution layers, (WITH augmentation and WITH dropout-0.5) and target size (80,80)



Graph 6 - Conv ->Conv ->max pooling->Conv ->Conv ->max pooling->dense layers(WITH augmentation and WITH drop out 0.50) and target size (80x80)

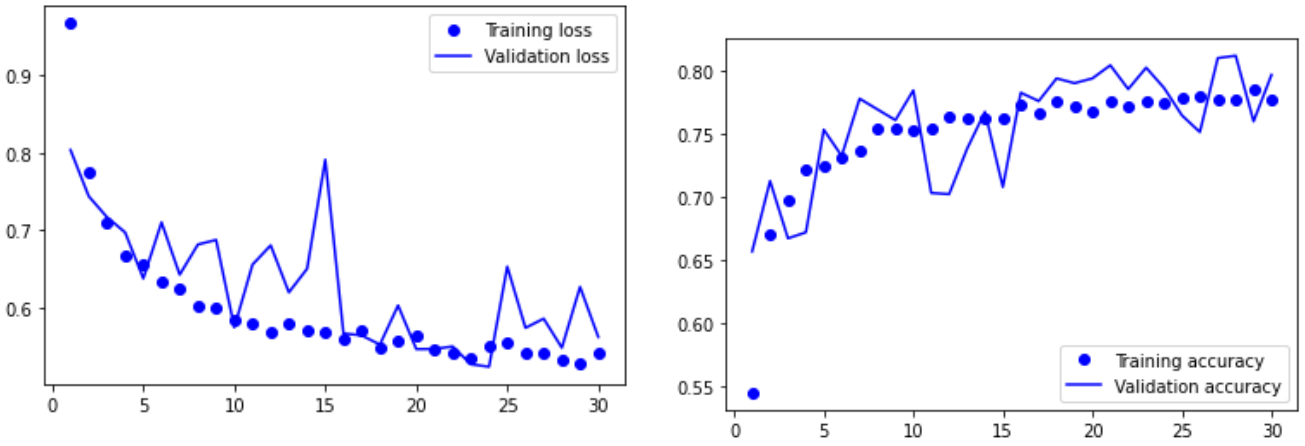


Graph 7 - Conv ->Conv ->max pooling->Conv ->Conv ->max pooling->dense layers(WITH augmentation and WITH drop out 0.50), target size (80x80) and with RGB

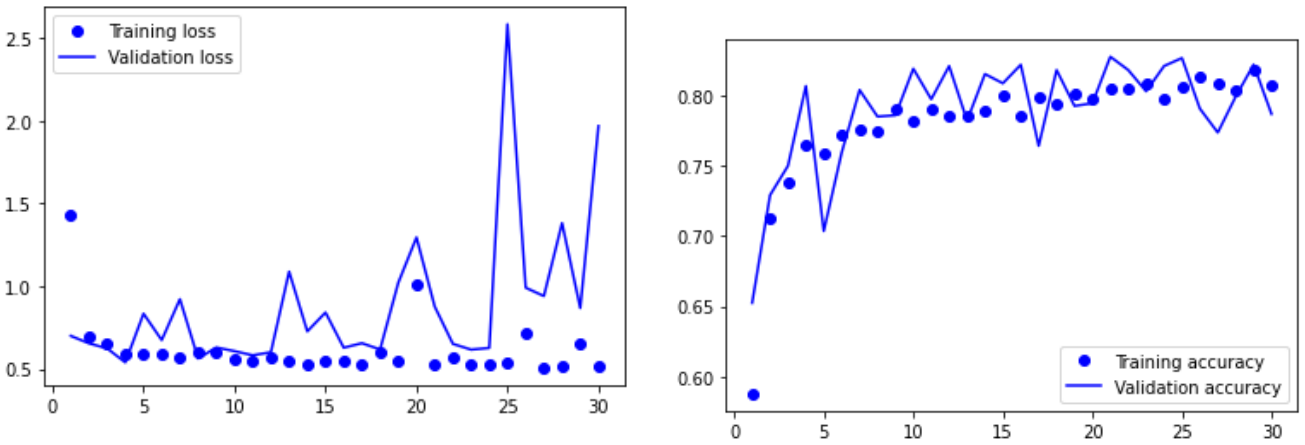


Graph 8 - Conv ->Conv ->max pooling->Conv ->Conv ->max pooling->dense layers(WITH augmentation but without geometrical transformation and WITH drop out 0.50), target size

(80x80) and without RGB



Graph 9 - Conv ->Conv ->max pooling->Conv ->Conv ->max pooling->dense layers(WITH augmentation but without geometrical transformation and WITH drop out 0.20), target size (80x80) and without RGB



Graph 10 - VGG16>dense layers(WITH augmentation but without geometrical transformation and WITH drop out 0.50), target size (80x80) and with RGB

		Predicted label			
		Bacteria	Normal	Virus	All
True Label	Bacteria	277	16	7	300
	Normal	0	300	0	300
	Virus	96	22	82	200
	All	373	338	89	800