Univerzitet u Novom Sadu
Prirodno-matematički fakultet
Departman za matematiku i informatiku

Dušan Binić

Diagnosis of Pneumonia Based on X-rays of the Patient Using Machine Learning

Master Thesis

Supervisor:

Oskar Marko, PhD

2022, Novi Sad

Acknowledgements

First of all, I would like to thank my supervisor PhD Oskar Marko, for his tremendous patience, advice and ideas throughout the process. Then to my family who is giving me support, strength and faith to reach my goals. To Ljilja, who is always there for me and who always brings a smile to my face. To all my friends and colleagues who helped me to spend my student days in fun and success.

Contents

List of Figures

List of Tables

1 Abstract

Pneumonia is usually diagnosed by inspecting X-ray images of the chest. These images are made following international standards, always well aligned, monochromical, and with the same dimensions. Although these characteristics make the task perfectly suited for machine learning algorithms, the process of diagnosis is still almost exclusively manual. However, the Covid-19 pandemic caused a sudden huge increase in pneumonia cases and the number of X-rays produced, which increased the demand for an automated solution to assist human experts, followed by many research papers proposing neural network-based solutions. In this thesis, we will approach the problem using convolutional neural networks, and in specific, we will compare four suitable architectures: AlexNet, DenseNet, ResNet, and VGG. Final experiments show that even with slight adjustments of the original weights, we are able to produce models that achieve over 94% accuracy in pneumonia diagnosis. Pre-trained DenseNet and ResNet attained the highest performance with 97% and 99% accuracy in distinguishing pneumonia from normal cases.

2 Introduction

Technological development brings advancement in all fields of life. What used to be done manually is now slowly taken over by machines and computers. In this way, the possibility of error is reduced, and accuracy and productivity are increased. In this thesis, we will focus on how artificial intelligence has contributed to the development of medicine, primarily in the field of medical diagnostics. Motivated by the need for faster interpretation of radiography images we apply several convolutional neural networks on Chest X-Ray Images (CXR) images for the detection of pneumonia cases. All of them show promising results in terms of, both, accuracy and speed.

In recent years, medical diagnosis using deep learning models have demonstrated remarkable progress in recognizing different diseases, their detection, classification and characterization. Modern technology has enabled us to record and gather a large amount of data in the field of medicine and many others. Due to significant advances in image collection devices, the amount of data is tremendous, and therefore their processing is a great challenge. Accelerated growth of data collection requires the introduction of processes that allow us to process them automatically. Moreover, it is necessary to increase the number of staff, and the problem is the subjectivity of medical staff and the inability to make the same decision based on the same input. An alternative is to use machine learning techniques to automate the diagnosis process, but these traditional machine learning methods are not sufficient to solve complex problems. Combining high-performance computers and machine learning algorithms results in reliable information diagnosis. Deep learning will help make a diagnosis, but it can also help us research the ways of disease progression and find new methods of treatment to heal the patient faster and prevent unwanted outcomes. Making fast and correct diagnoses also helps medical professionals to effectively isolate different positive cases of pneumonia and reduce the resources allocated in further testing the negative cases.

Machine learning and artificial intelligence have achieved significant growth in previous periods and have become indispensable techniques used for medical image processing, computer-aided diagnostics, image interpretation, registration and segmentation, therapy determination, and efficient and effective information presentation. These techniques help doctors diagnose and predict the risk of the disease and prevent it in time, and of course, they help determine the appropriate therapy. While human experts still make final decisions, machine learning models are a good aid, and consulting their predictions helps to reduce possible errors.

AI based systems enhance the ability of physicians and researchers to analyze

the patterns by which disease occurs in patients. They consist of well-known algorithms such as support vector machine (SVM) [1] and neural network (NN) [2], but especially more specialized deep learning algorithms, such as convolutional neural network (CNN) [3], recurrent neural network (RNN) [4], long short-term memory (LSTM) [5], extreme learning model (ELM), generative adversarial networks (GAN), etc. Earlier algorithms could not process the natural image in its raw form, and expert knowledge was required to use them. However, algorithms that were later developed can work with raw data to learn independently and do not require expert knowledge to use. Although automated illness identification based on traditional approaches in medical imaging has demonstrated a high level of accuracy for decades, breakthroughs in machine learning techniques have resulted in a surge in deep learning. Deep learning algorithms have shown promising performance, as well as a speed-up, in various domains such as speech recognition, text recognition, lip-reading, computer diagnostics, facial recognition [13] and many more.

We are interested in building a model for automated detection of Covid-19-caused pneumonia. The model should take as an input X-ray of a chest (CXR images) region and classify it into one of two classes, positive for pneumonia and negative for healthy lungs. Chest computed tomography (CT), as a routine imaging tool for pneumonia diagnosis, is relatively easy to perform and can produce fast diagnosis. We can take as an example detection of Covid-19 caused pneumonia: even though there are several ways of detecting the presence of this virus, such as PCR test, the low sensitivity and accuracy of those tests might lead to the wrong conclusion and the patient may not receive appropriate treatment in time. That is, even if the PCR test is negative, positive CT features can still highly suggest of Covid-19. The aim of this work if to make this detection even faster and more accurate, and by doing so, we will not only improve the care the patient receive, but also aid the medical stuff in the current crisis. For this, we focus on a group of convolutional neural network models that are widely used in medical image processing.
We consider two tactics:

- Fine-tuning pre-trained models to take advantage of good generalization
- Fitting all the weights from scratch to build a narrowly-specialized classifier.

The rest of this thesis is organized as follows. Section 4 introduces the problem and network architectures of interest. Section 5 goes over technical details and derivation of the algorithms, while the Section 6 presents the data and experimental setup. In this section we also present and discusses the results of experiments conducted to evaluate the efficacy of the proposed models. Finally, we conclude the thesis with Section 7 with the conclusion and propose ideas for the future work.

3 Related work

Many studies has showed the potential of artificial intelligence as a tool for detecting pneumonia based on based on chest x-ray images. Lately, those studies are focused on Covid-19 cases. Some preliminary works that utilized AI-driven methodologies to assist metical stuff in the cases connected to Covid-19 can be found in the [6]. In their work, authors introduced new convolutional neural network tailored specifically for the detection of Covid-19 cases from chest X-ray images. This CNN is available to public, as well as dataset used in their study at [7]. COVIDx, an open access benchmark dataset that they generated consists of 13,975 CXR images across 13,870 patients. It is also important to note that COVIDx dataset continues to evolve as new patient cases are continuously added and are made available publicly on a regular basis, allowing further research and improvement of existing detection and classification models.

Work similar to ours can be found in [8], who implemented 3 out of 4 CNNs which we have included – AlexNet, VGGNet and ResNet, but on significantly larger dataset consisting of 108,379 CXR images, derived from the US National Institute of Health. With the use of transfer learning they were able to achieve 90.13% classification accuracy among four classes - normal, pneumonia, other disease and Covid-19.

In [9] autors generated custom dataset which is available to public at [10]. This dataset contains carefully picked images with couple of criteria, e.g all patients are above 18 years of age, data is equally distributed between male and female patients, demographic and clinical data was known, etc. Dataset also contains information whether pneumonia was caused by covid virus or bacteria, allowing researchers to test if their model is capable of recognizing different types of pneumonia, and with that, giving us a possibility to create a model specialized for the detection of Covid-19.

Recent literature demonstrates that it is possible for artificial intelligence to distinguish Covid-19 from other pneumonia cases with good accuracy [11] and as such, they could be readily used as assistance in the medical diagnosis. However, even though this and several comparative studies demonstrate good performance of AI in the given problem, it cannot replace an expert and should be adopted and integrated in clinical workflow as a decision support tool. This was illustrated in the work [12], in which the performance of radiologist and AI methods were compared. In their work, they have shown that radiologists were able to better detect pneumonia with the help of AI, confirming the use and applicability of these models. With the aid of artificial intelligence medical professionals are able to make decisions with much higher accuracy, or simply confirm their diagnose

within seconds.

4 Problem Description

We have all witnessed the events of 2020 when a global pandemic hit the planet. Although humanity is constantly preparing for similar situations, it is very difficult to react adequately when they happen. Covid-19, the virus that caused the outbreak, has caused a large number of instances of severe pneumonia. The rapid rate of spread has affected most health systems in the world, where the lack of medical staff and the fatigue of the employees were the leading cause of the inadequate care that the patients received. A huge number of X-ray images were collected, and medical doctors had less time to inspect each scan before giving a diagnosis. Moreover, wrong or incorrect diagnosis often means life or death.

Chest X-ray is becoming one of the most common medical diagnoses due to its non-invasiveness. It is an approach radiologist use to distinguish whether the patient has pneumonia or not, by looking for white spots or patches in the lungs (called infiltrates) that identify an infection. The number of chest X-rays has jumped sharply, but radiologists are still manually reading chest X-rays, which takes a lot of time for medical workers, primarily in processing the images. Traditionally, as a subfield of radiology that can provide a large amount of information from medical images, radionics facilitated the diagnosis of medical images before the era of artificial intelligence. With the development of deep learning, analyzing X-rays of the lungs becomes possible in everyday medical practice, allowing for much faster and possibly, more precise diagnosis. Furthermore, the misdiagnosis rate of Covid-19 is very high, and such wrong or incorrect diagnosis often means life or death.

This thesis will describe how machine learning can be utilized to analyze lung X-rays using a particular set of data. In specific, we will consider several state-of-the-art network architectures to build a high-quality model for predicting the Covid-19-caused pneumonia from X-ray images. We shall present the whole procedure starting form the image prepossessing, to building the model and finally comparing the results based on different metrics of success. The final outcome is to evaluate an automated model for accurate and rapid pneumonia screening based on CXR images.

All of the above is an excellent challenge for the medical profession and will influence the direction of medical development in the future. Modern technologies allow us to solve this problem and improve health care quality worldwide. With the improvements in machine learning and neural networks, this is becoming much easier and available to everyone. Deep learning technologies are being employed more frequently to improve clinical practice [14], and the list of instances is rising every day. We will not try to give a thorough overview of deep learning in medical

imaging. Instead, we will sketch out some of the landscape before diving into a more systematic discussion of deep learning in Magnetic Resonance Imaging (MRI) [15].

4.1 Convolutional Network for BioMedical Image Classification

Among machine learning, Convolutional Neural Network (CNN) is the leading deep learning tool. They are popularly used in different sub-fields of healthcare system due to their ability to extract features, even those that may be missed by human experts. The architecture of a model is an important aspect in increasing the performance of various applications. The CNN architecture has undergone a number of changes, such as structure reformulation, parameter optimizations, regularization, etc. On the other hand, it should be emphasized that the major improvement in CNN performance was primarily attributable to the restructuring of processing units and the development of new blocks. The utilization of network depth was used to perform the most novel developments in CNN designs [16]. We consider four CNN architectures commonly used in medical imaging problems and exhibit good performance for problems similar to ours.

AlexNet. AlexNet is well-known for its groundbreaking work in the fields of image identification and classification. AlexNet has eight layers, five of which are convolutional and three of which are fully connected. A single-pixel filter is used to perform these convolutions. The sigmoid function is substituted by the Rectified Linear Unit (ReLU) to calculate nonlinearity. Dropout layers are employed to solve the overfitting problem. The maximum pool is utilized to reduce the network [16]. Paper [25] applies transfer learning with AlexNet to achieve a 100% accuracy in predicting a pathological brain from the MRI images. A modification of the architecture is considered in [26] to improve the prediction over diabetic retinopathy images.

Visual geometry group (VGG). This architecture outperforms AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layers) with multiple 3×3 kernel-size filters one after the other and a stride of 1, which is designed for high accuracy in large-scale image recognition applications. In addition, VGG also adjusts network complexity by putting single-pixel convolutions in the midst of the convolutional layers. It learns a linear grouping of the feature maps that follow. After the convolutional layer, a max-pooling layer is added, and padding is used to keep the spatial resolution. VGG16 has 16 layers in the base model [16]. In [27] VGG is framed with Gaussian mixture models

and singular value decomposition to give an improved classification of diabetic retinopathy, and in [28] it is utilized for early detection of cancer-bearing cells from cytological images.

Residual neural network (ResNet). Deep residual networks were a breakthrough concept that allowed for the creation of far deeper networks (hundreds of layers as opposed to tens of layers). On the other hand, adding extra layers was found by numerous researchers to have a negative impact on the ultimate performance. Deeper networks often converge at a larger error rate than shallower networks, which the authors refer to as the degradation problem. The introduction of residual blocks, in which intermediary layers of a block learn a residual function with reference to the block input, is a solution to this deterioration problem. This residual function can be thought of as a refinement stage where we learn how to alter the input feature map for better quality features [16]. A pre-trained ResNet was applied in [29] to work with X-ray images of lungs and distinguish between bacterial, viral, and Covid-19-caused pneumonia. An adjusted version of ResNet is used in [30] to predict malign tissues in histopathological images.

Densely connected convolutional network (DenseNet). DenseNet was introduced to tackle the vanishing gradient problem. DenseNet improves cross-layer connectivity by employing a feed-forward approach to connect each layer to all other layers in the network. As a result, the feature maps from each preceding layer were used as input into the subsequent layers. As a result, the network has the ability to distinguish clearly between added and conserved data. On the other hand, DenseNet becomes parametrically expensive because of its narrow layer structure and the increasing amount of feature maps. The loss function's direct admittance of all layers to the gradients improves information flow across the network. Additionally, this has a regularizing effect, reducing overfitting on tasks and small training sets. DenseNets are said to achieve higher performance with less complexity when compared to ResNet models [16]. A paper [31] compares a pre-trained DenseNet with a modified version trained from scratch on classifying cellular images containing endometrial and colorectal cancer. In [31], different transfer-learning strategies with DenseNet are compared to build a model for detecting multiple sclerosis from the brain scan images.

5 Technologies

The models of interest in our work are CNNs. In this section, we will introduce the structure and main principles of work of the model. We first start with the preliminaries on perceptrons and feed-forward networks before getting into more details on the CNNs. Further, we address some common issues and the strategies to overcome them, and give some details about the PyThorch library that we used for the implementation.

5.1 Perceptron Learning

Neural Networks (NN) appeared in literature as early as the mid-XX century [17] with the idea to build a machine learning model mimicking the human brain. For this reason, the main building elements of NN are called neurons and are connected in a way that neighboring cells activate each other and build stronger connections if it helps in the decision-making process. Initial models actually consisted of a single neuron, and we call them perceptrons [17]. Perceptron was one of the first algorithms to solve the problem of linear classification and regression. Mathematically, the perceptron represents a scalar function of several variables, the values of which depend on the parameters that represent the weight coefficients of the linear discriminant function. The principle of work of perceptron is illustrated in Figure 5.1. Consider a regression problem, with an observation $\mathbf{x} \in \mathbb{R}^l$. The model applies weights $\mathbf{w} = [w_1, w_2, ..., w_l]^T$ and bias w_0 to obtain a new variable $z = \mathbf{w}^T\mathbf{x} + w_0$. Then the activation function $f(z) : \mathbb{R} \to \mathbb{R}$ is applied to forecast the final value. Choices of the function $f(\cdot)$ are discussed later in the text, and they might depend on the data and problem at hand. In the case of classification, we can use the same activation function as for regression, but with additional *clipping*, i.e., we apply a step function on top of the output of $f(z)$. In the case of a binary classification, when $\mathbf{x}$ needs to be assigned to either -1 or 1, we choose a threshold value α and make a decision:

$$\begin{aligned} f(z) > \alpha \to \mathbf{x} = 1, \\ f(z) < \alpha \to \mathbf{x} = -1. \end{aligned} \tag{5.1}$$

Activation Function. Let us now consider some of the well known choices for an activation function $f(\cdot)$. As mentioned above, the function is applied over an instance that is already weighted and with added bias: $z = \mathbf{w}^T\mathbf{x} + w_0$. An identity function is the simplest option [18]:

$$f(z) = z, \tag{5.2}$$

and gives an affine prediction hyperplane. Another popular choice is a hinge or ReLu function [18]:

$$f(z;\alpha) = \begin{cases} z & \text{if} \quad z > \alpha \\ 0 & \text{if} \quad z \leq \alpha \end{cases}, \tag{5.3}$$

where $\alpha \in \mathbb{R}$ is a pre-specified threshold value. Finally, a sigmoid function is also common in the literature since it gives a smooth S-shaped output:

$$f(z) = \frac{e^z}{e^z + 1}. \tag{5.4}$$

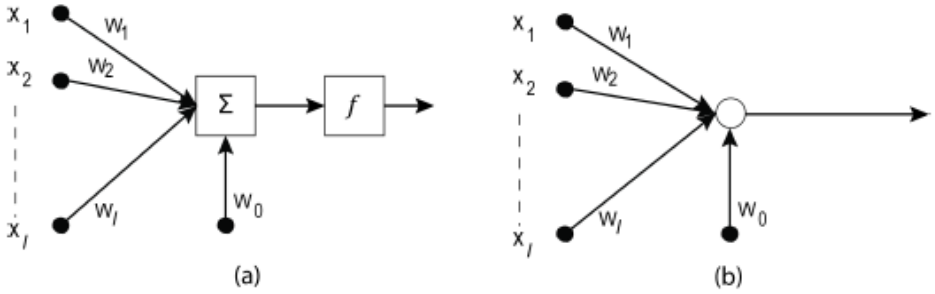

Figure 5.1: Perceptron - a basic network unit: a) a display with the collector and activation function; b) a simplified view [17].

Loss Function. In order to objectively compare different models (or a single model with different parameter values) we need to have a criterion to evaluate the goodness of fit. A function $L(\cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, constructed for such a purpose, is called a loss function (also called a cost function). This function is used in model training where after each iteration to evaluate the goodness of fit, and also to estimate a direction in which the model weights should change in order to further decrease the loss. The function takes a ground truth value $\hat{y}$ and a predicted value y as inputs. A simple exaple of a loss function is a squared error loss:

$$L(y, \hat{y}) = (y - \hat{y})^2. \tag{5.5}$$

A more popular choice in the case of perceptron learning is hinge loss given by

$$L(y, \hat{y}) = max(0, 1 - y\hat{y}). \tag{5.6}$$

A special case of this, when both y and $\hat{y}$ take only values 0 or 1 is a so called 0-1 loss:

$$L(y, \hat{y}) = \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & \text{if } y = \hat{y}. \end{cases} \tag{5.7}$$

A loss function does not have to be the same as the function used for measuring the final success of the algorithm (see Section 4.2), but it is rather preferable to chose a differential loss function to make computation of the derivatives possible.

5.2 Neural Networks (NN)

Due to its simplicity, a single perceptron cannot produce a complex model, but if a number of perceptrons are added up, they build a powerful model — neural network (NN) [2]. NNs are famous for their ability to make complex predictive models that generalize well, and there are several specialized types of networks. The most basic one is a Feed-Forward NN [19] that is used for a wide variety of applications. Another type, specialized for time-series data, are recurrent neural networks (RNN) [4] that further have a subgroup of Long Short Term Memory (LSTM) networks [5]. Finally, convolutional neural networks (CNN) are used in image and signal processing [3], and in this thesis, we specifically focus on this type of network. In this section we first give an introduction to a general principles of work of NNs, and in the next one we get into details about CNNs.

NNs consist of an input layer, one or more hidden layers, and an output layer. Figure 5.2 shows an example of a multi-layer feed-forward network. Units are the building blocks of each layer. The data attribute values passed with each training tuple correspond to the network's inputs. The inputs are transmitted through the input layers units, where they are weighted and sent to the hidden layer. The outputs of a hidden layer can then be fed into another hidden layer, and so on. Weights of the last hidden layer are sent into the output layer, which yields a prediction for a given data tuple. The network is called *feed-forward* because none of the weights cycle back to the input unit or previous layers. It is usually fully connected, in the sense that each unit gives input to the following forward layers units [19], as depicted in Figure 5.2. Still, sometimes it might be beneficial to remove some of the edges, as we will mention in next section, when talking about dropout regularization.

Each output unit receives a weighted sum of the outputs from the previous layer's units as input. The weighted input is applied to a nonlinear (activation) function, as described in Section 5.1. An important note is not to set all the activation functions to be linear because, in that case, a whole network can be equivalently modeled using only a single node, i.e., a perceptron model [17]. NNs use nonlinear regression from a statistical standpoint. Given enough hidden units and training data, multi-layer feed-forward networks can closely approximate any function [19].

In the regression models, the value yielded by an output layer is used directly as a prediction. On the other side, the class prediction demands additional processing of the given result. In the case of binary classification, the threshold value is set

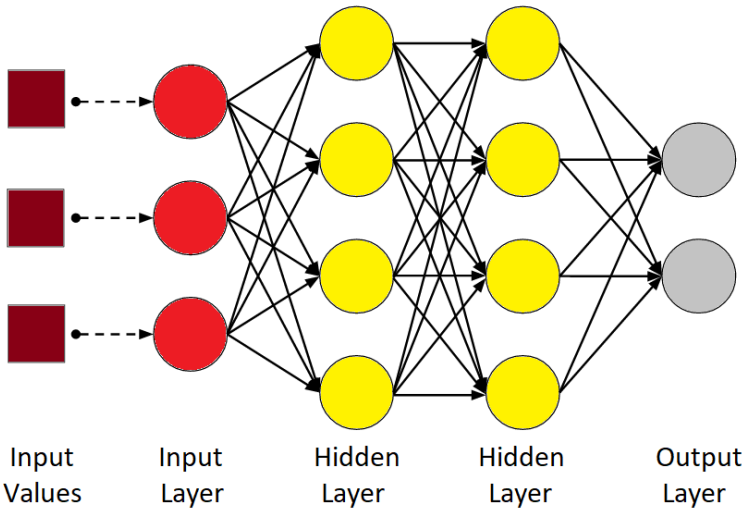

Figure 5.2: A fully-connected feed-forward neural network architecture.

to assign a value to one of the classes (see Equation 5.1). In the case of multi-class classification, it is convenient to have an output layer consisting of as many neurons as there are classes and assign a point to the class whose neuron yields the highest value — this is also known as a *softmax* layer.

5.2.1 Overfitting, Underfitting and Capacity

The main challenge of machine learning is to work well on new, previously unknown data. This process is called generalization. When we evaluate a model on a training set, we get an error called the training error — the model is refined by minimizing this quantity. On the other hand, we have a generalization error (i.e., test error), the expected value of the error at the new input (one that the model did not see before), that we want to be as low as possible. We sample the training set, then use it to pick parameters to reduce the error in the training set, and finally sample the test set. Two aspects that determine how effectively a machine learning algorithm will perform are 1.) making the training error small and 2.) making the gap between training and test error small. These two aspects lead to the two main machine learning issues: underfitting and overfitting. Underfitting is a case when the model cannot get a low enough error value on the training set. On the other side, overfitting is when the training error is low, but the difference between the training and test error is too big. By adjusting a model's capacity, we may have some intuition on whether it is more likely to be an overfit or an underfit. Low-capacity models should produce a more generalized decision but may struggle to fit the training set well, while high-capacity models can overfit by

memorizing training set attributes that are just not useful on the test set [20]. In the following lines, we introduce three widelly used approaches to fight overfitting in NNs: *training-test split, data augmentation* and *dropout*.

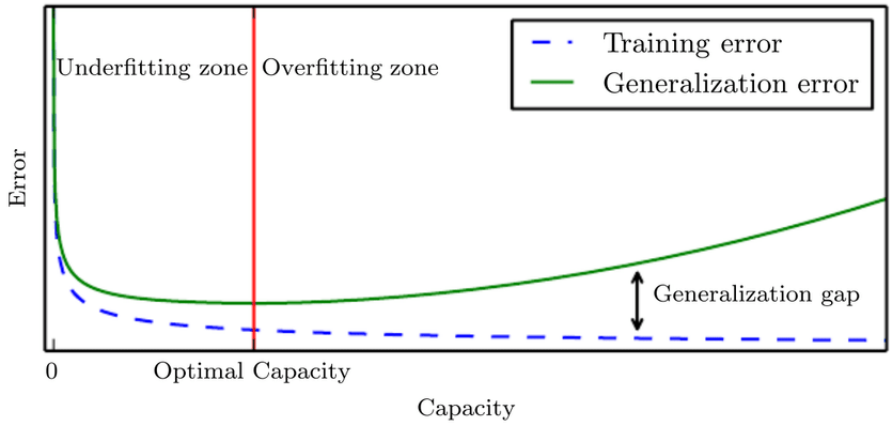

Figure 5.3: A typical relationship between training and generalization errors [20].

Train-Test Split. One strategy for measuring the performance of a machine learning algorithm while avoiding the overfitting is a train-test split (Figure 5.4). The train-test split can be used in any supervised learning technique, whether a classification or a regression task. As the name suggests, a dataset is split into two subsets — one for training and the other for testing. The training data is used to fit the model parameters, but it must not be used to evaluate the fit since that approach would highly value the models that simply memorized already seen instances. To see if the model generalizes well, we evaluate it on the unseen data — the test set. The error over the training data will always be lower than that of a test set, but we are interested in the relative difference between the two. As illustrated in Figure 5.3, initially, both errors are decreasing, but increasing the complexity of the model too much, we reduce training error on behalf of increasing test error. The optimal level of complexity is just before these two errors start diverging from one another.

Data Augmentation. The amount of data available often enhances the performance of deep learning networks, and as a rule, it is preferable to have a lot of data than a clever network structure. However, collecting new data is expensive in terms of both time and labor, so users often end up with relatively small datasets. Data augmentation is a method of artificially creating additional training data from the existing data. This is accomplished by using domain-specific approaches

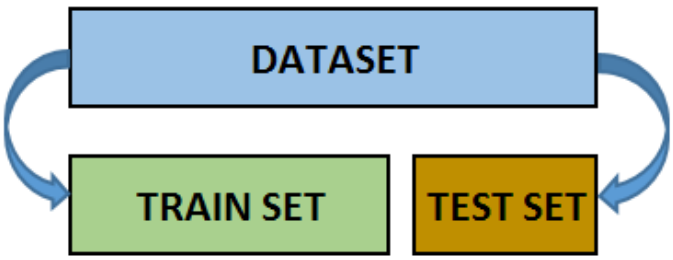

Figure 5.4: Train-Test Split

to transform examples from the training data into new and unique training examples. One of the areas where it is most commonly used (and also the case that is of most interest to us) is image data augmentation, which includes transforming images from the training dataset into altered copies that belong to the same class as the original image. Shifting, flipping, zooming, rotating, cropping, translating, etc., are all examples of applied transformations in the area of image modification. Images obtained in this way are different enough to stop a network from memorizing instances, yet they usually do not lose critical information for decision-making. This approach leads to a possibly infinite amount of images; however, it is important to keep in mind that an augmented data is relatively similar to the original, hence if the original data is not a representative sample, the augmentation alone will not help.

Dropout Regularization. Dropout is the most popular regularization method used in neural networks to prevent overfitting. A neuron is briefly dropped or inhibited with probability $p \in [0, 1]$ at each iteration during the training. This implies that all of this neuron's inputs and outputs will be disabled at a considered iteration. At each training step, the dropped-out neurons are resampled with probability p, so a dropped-out neuron at one step can become active at the next. The dropout rate, or hyperparameter p, is commonly a value around 0.5, which corresponds to 50% of the neurons being omitted. Although this approach does not seem intuitive, it avoids assigning too much importance to a single node or a small group of nodes in a network. If that happened, the network would suffer reduced robustness and generalization.

5.3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are the group of NNs that are specifically suitable for image processing applications [21]. The main component of CNNs is a convolution window [21] that allows an algorithm to observe pixels in a natural order and lead to a better understanding of an image. In the case of gray-scale (i.e., single-channel) data, we can observe an image as a matrix $M \in \mathbb{R}^{n,m}$, where each entry of a matrix is a single pixel. A convolution window is then a square submatrix of a fixed size $W \in \mathbb{R}^{n_0,n_0}$ that traverses a matrix M and applies a specific transformation over the regions it visits. A typical design of a window is to set zeros and ones at specific coordinates of W, and then sum the elements under the window that correspond to ones — such an example is depicted in Figure 5.5. A popular choice is a 3×3 window, where ones are aligned in the form of a cross or an x (Figure 5.5). The first case is used to discover vertical and horizontal patterns in the figure, and the second is for diagonal patterns. A 2×2 window with ones on the left (top) and zeros on the right (bottom) is used to find contours in the photo. There is another popular design of the window where it filters out only the largest element under it, and this is often addressed as *Max Pooling*. The idea of this filter is that the maximum values will point out to more important features of an image so we can reduce the size while keeping a high amount of the original information. Similar to this, one can use a minimum filter or average filter, although the last one did not show a lot of success in the applications.

Besides the window size n_0, the algorithm also takes an argument s_0, not necessarily equal to n_0. In the first step we apply a window W to the upper left corner of a matrix M and then need to shift to the right — the length of this step is exactly s_0. If $s_0 = n_0$ the visited regions of a matrix M will not overlap and we will traverse it in fewer steps than if $s_0 < n_0$.

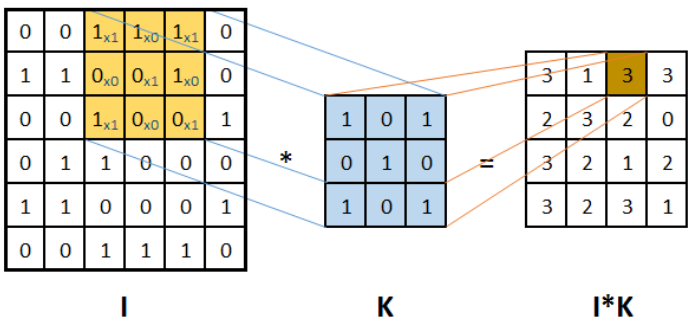

Figure 5.5: Example of a convolution window **K** applied to a matrix **I**

The final product of a convolution $M * W$ is a matrix $M_1 \in \mathbb{R}^{n_1,m_1}$, with $n_1 < n$ and $m_1 < m$. The dimensions are always lower than those of an original image because the convolution window cannot visit the edge elements. However, if we want to allow for this we can introduce a so called *padding* [21], which augments the initial matrix M so that the window can visit each component. This is usually done by adding zeros or duplicating the corresponding edge elements all around the border.

An example of a *max pooling* with window size $n_0 = 2$ and step $s_0 = 2$ is presented in Figure 5.6. Here we transform matrix $M_1 \in \mathbb{R}^{4,4}$ into $M_2 \in \mathbb{R}^{2,2}$.

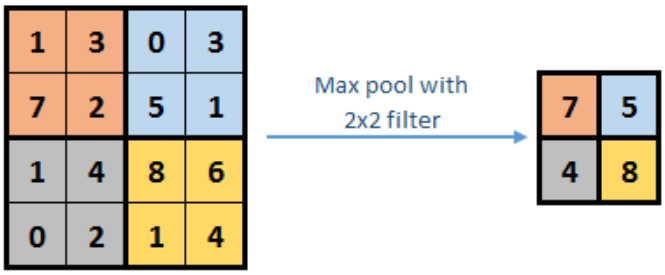

Figure 5.6: Max pooling with 2×2 convolution window and a step size 2.

A typical architecture of the CNN network is presented in Figure 5.7. We can see an input X-ray image, filtered using a convolutional pooling window over several network layers, until it yields a vectorized feature map. This vectorized map is then sent to a dense fully connected layer to produce an output. The output layer is a softmax activation that indicates the class with the highest final value (highlighted in red).

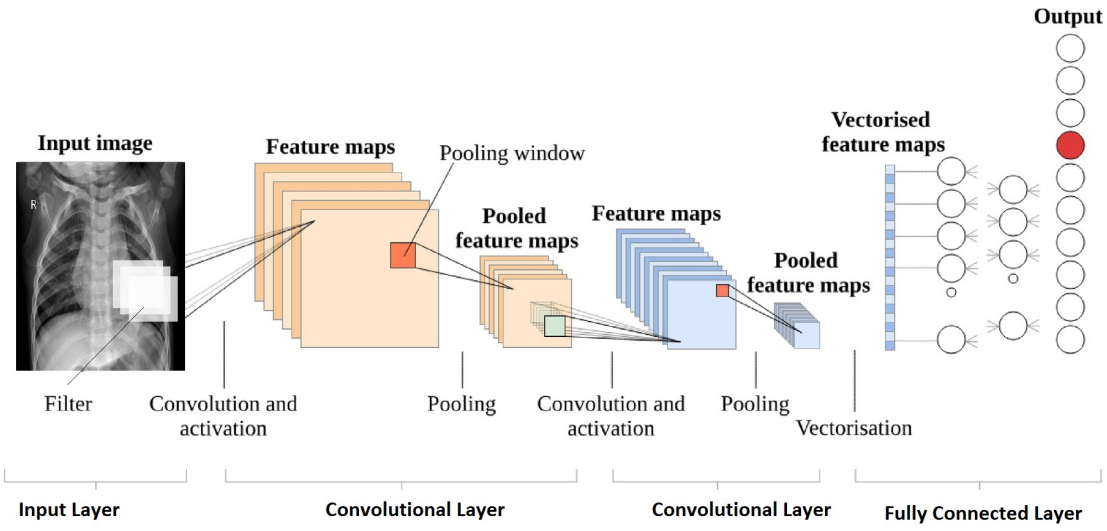

Figure 5.7: CNN Architecture [15].

5.4 Transfer Learning

Training NN models, especially CNN, is very demanding in terms of both time and computational resources. Even a simple CNN model might be impossible to train without a strong machine and GPU. This also depends on the number of model parameters and data size, but reducing any of the two reduces the performance as well. A way to overcome this is by using transfer learning [19]. Transfer learning is a machine learning technique in which models trained on a specific task is repurposed for a new task. Large companies and research centers build complex and powerful models every day, trained on terabytes of data. Many of these models solve common problems (e.g., face detection) and can be readily used for different purposes and new datasets. However, sometimes we would like to slightly adjust a model in order to solve a similar but not the same problem. In this case, it is possible to take a pre-trained model and keep initial layers unchanged, but then *unfreeze* last few layers and further train those weights on the data of interest — this easily changes the expertise field of the model (e.g., model trained to detect anomaly regions on brain scans can be adjusted to do a similar task on lung X-rays images). In some cases, even the structure needs to be adjusted, usually by changing only the output layer (e.g., when going from multiclass classification to binary classification). This principle is schematically presented in Figure 5.8.

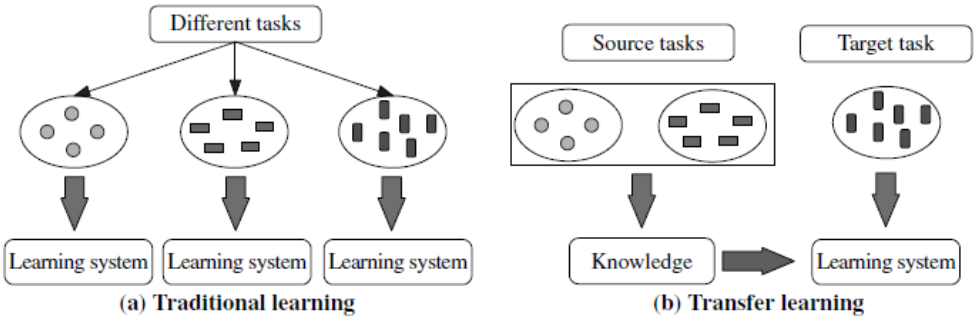

Figure 5.8: Traditional learning versus transfer learning. (a) For each classification challenge, traditional learning methods create a new classifier from scratch. (b) Transfer learning uses information from a source classifier to make a classifier for a new but similar task [19].

5.5 PyTorch

In this work we used PyTorch to build and train models and jupyter notebook with Python 3 for data manipulation and evaluation of obtained results. PyTorch is one of the widely used Machine learning librariesis (others being TensorFlow and Keras), open-source and highly popular in this field due to its simplicity and flexibility. It is an optimized tensor library primarily used for applications using GPUs and CPUs. To compute automatic differentiation, PyTorch uses the Autograd module. In a nutshell, a recorder keeps track of the actions and then replays them to create gradients. The forward pass performs data differentiation quickly, saving time in the development of neural networks. The optim module in PyTorch allows a user to define an optimizer that will automatically update weights. PyTorch allows to create several sorts of layers, including convolutional layers, recurrent layers, and linear layers, among others, thanks to its diverse modules. Project Jupyter is a notebook environment compatible with Python, R and Julia; it is user friendly and suitable for presenting the results in a concise way [22].

6 Data and Methods

As described in Introduction, the problem faced in this work is the classification of chest X-ray images into one of the two classes: 1. *Normal* or 2. *Pneumonia*. This section describes the data in more detail, introduces evaluation criteria, and finally compares the results of different Neural Network (NN) structures.

6.1 Data

We use a Chest X-Ray dataset published by Paulo Breviglieri [23] and publicly available at [24]. A dataset is already divided into three parts: training, validation, and test, each in a separate folder and with a moderately balanced number of samples between the two classes — *Normal* and *Pneumonia*. Due to this existing division, there is no need to additionally implement a train-test split for our models. Both training and validation sets contain 25% of *normal* images, while the rest 75% are the images with pneumonia cases. On the other hand, in the test set, this ratio is 37.5% to 62.5%. The total number of observations is 5,856, where the split per folder is as follows:

- Training observations: $4,192$ (1,082 normal cases, 3,110 pneumonia cases)
- Validation observations: 1,040 (267 normal cases, 773 pneumonia cases)
- Testing observations: 624 (234 normal cases, 390 pneumonia cases)

This division is presented in Figure 6.1.

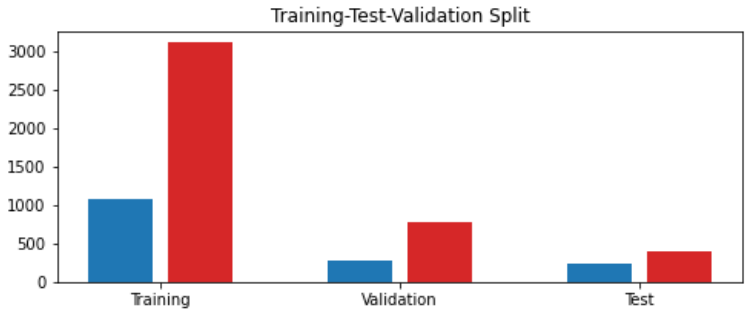

Figure 6.1: Cardinalities of training, validation and test sets. For each group blue bar represents a number of samples denoted as *Normal* and the red bar represents *Pneumonia* cases.

All the images are gray-scale and resized to 256×256 pixels in order to have a consistent input to NNs. A sample of training data from both classes is presented

in Figure 6.2. A decision if a patient has pneumonia or not is based on the opacity of the lungs in X-ray, as one can see from the data. Sometimes this difference is not very clear, or the inflammation is present only locally, making the decision hard even for human experts. In those cases, we might expect that our CNNs show a similar performance.

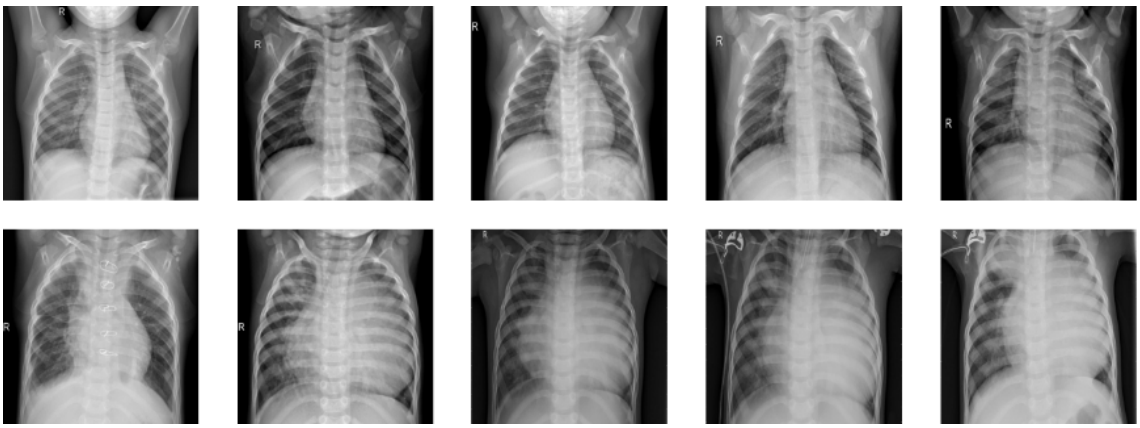

Figure 6.2: Random sample of images from the training dataset. The upper row shows *normal* cases while the bottom row shows *pneumonia* cases.

Our dataset is relatively small for deep NNs used in this work, and additionally, the training set is not perfectly balanced between classes (see Figure 6.1). To deal with these problems, we apply image data augmentation techniques. We need to be aware of the domain-specific characteristics of our datasets when using standard image augmenters. In the first place, notice that X-ray images are aligned so that the spine is in a vertical position; for this reason, we must not allow for a large rotation angle α when creating additional data, and we will keep it within $-10° < \alpha < 10°$. Additionally, we do not allow for the vertical flip for the same reason. This leaves us with a slight rotation angle α, horizontal flip, and a zoom-in/out factor β that we will also keep moderate since captured images are mostly tightly enclosing a chest region. Sample images with the augmented copies are shown in Figure 6.3.

6.2 Evaluation

During our work we have evaluated the performance of the models using several key parameters: accuracy, sensitivity, specificity, precision and F-score. The principal metric used in comparing classification models is accuracy. The formula for accuracy is given in equation (6.1) and represents a percentage of correct predictions over the total number of predictions. However, this is often not the best

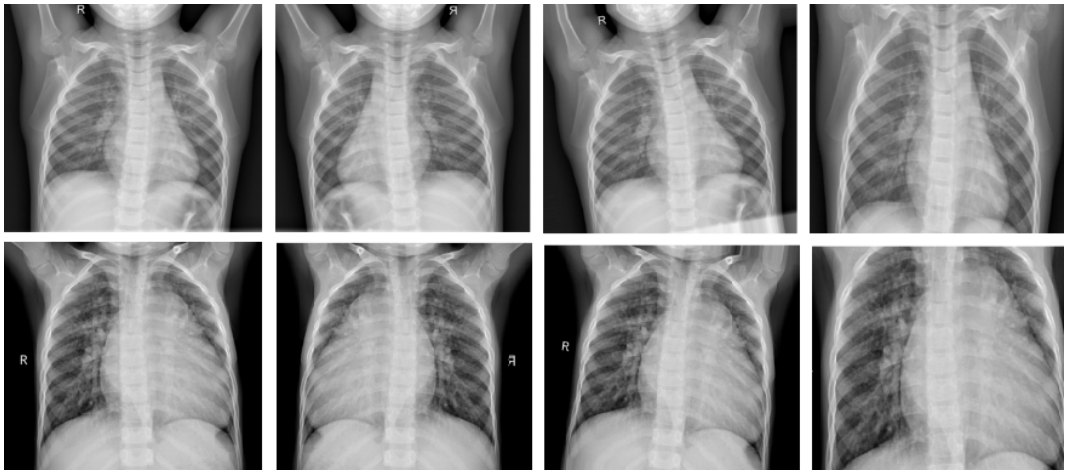

Figure 6.3: First column contains an original image from the training set, and the other columns are augmented variants of it.

indicator of the goodness of fit, especially in an unbalanced learning setting, so the additional metrics should be considered as well.During our work we have evaluated the performance of the models using several key parameters: accuracy, sensitivity, specificity, precision and F-score. Sensitivity and specificity refer to the ability of the model to detect positive and negative cases, respectively. However, as we are working on a medical prediction, the mistakes are expensive, but also they will not weigh equally depending on the nature of the mistake — it is less dangerous to miss-classify a patient as a pneumonia case than vice-versa. For that reason, we set accuracy as our main evaluation metric, as it takes into calculation both correct and wrong detections.

For binary classification, where classes are denoted with 0 and 1, we have four prediction terms: True Positives (TP) — an instance is predicted as class 1 when it really belongs to class 1; True Negatives (TN) — an instance is predicted as class 0 when it actually belongs to class 0; False Positives (FP) — an instance is predicted as class 1 when it belongs to class 0; False Negatives (FN) — an instance is predicted as class 0 when it belongs to class 1. False negatives are the most undesirable outcomes, as this happens when pneumonia is not detected by the model when it is present. This would worsen the patient's state or even lead to death. Using the above terms, we can define the following metrics [23]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6.1}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \tag{6.2}$$

Optimizer	Learning Rate	Batch Size	No. Epochs
Adam	0.01	16	5

Table 6.1: Parameters Setting for CNN architectures.

$$\text{Specificity} = \frac{TN}{TN + FP}, \tag{6.3}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{6.4}$$

$$\text{F-score} = 2\frac{Precision * Sensitivity}{Precision + Sensitivity}. \tag{6.5}$$

Sensitivity and Specificity are similar to accuracy, but regarding only a single class — Sensitivity gives a percentage of correctly predicted instances out of all positive valued ones, and Specificity gives a portion of correctly predicted cases out of all with a 0 label. Precision is similar to Sensitivity, but instead of providing the number of TP out of all positively valued test instances, it considers it out of all the instances that were predicted as positive, irrespective of its original label. Finally, F-score is obtained as a combination of Precision and Sensitivity. All five metrics range from 0, as the worst possible score value, to 1, as the best.

6.3 Experimental Setup

We consider four readily available NN architectures, AlexNet, DenseNet, ResNet and VGG, that we described earlier, in Section 4.1. For each of these structures we might consider two strategies:

1.) Re-initialize all the weights and train the model exclusively on our X-ray images data.
2.) Keep the pre-trained weights, except in the last two layers — these layers we train further with our data in order to make the network more specialized for this task.

The choice is, in fact, a trade-off between goodness of fit and learning cost. The first strategy should, in general, yield better prediction results, but it takes considerably more computational time, while the second one can be trained quickly and provide a moderately accurate solution.

Each of the CNNs that we use has a different architecture. The model parameters are kept the same across all the models and are given in Table 6.1. This helps to have a fair comparison of the performance. The models were trained for five epochs with a batch size of 16, a learning rate of 0.01, and Adam optimizer.

6.4 Results

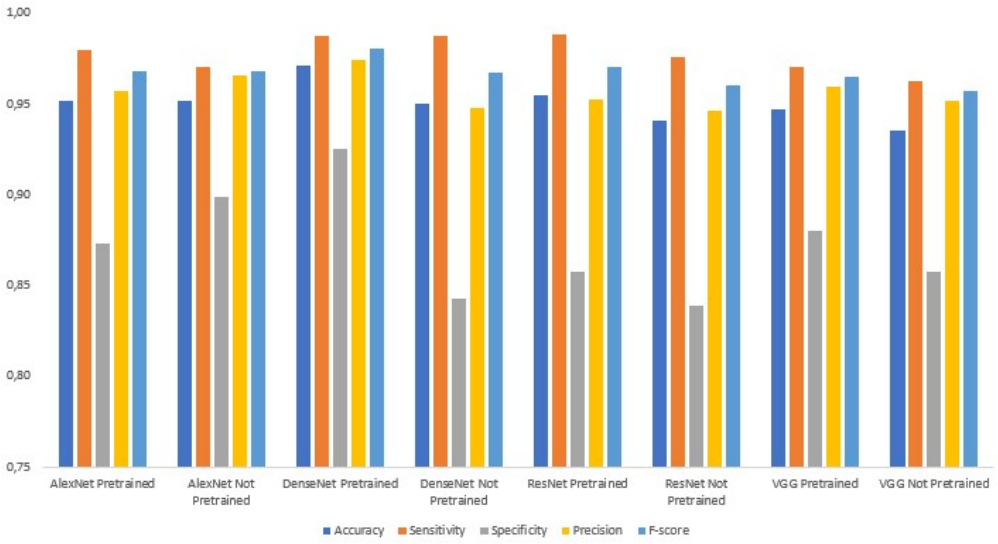

Figure 6.4: Results for different CNN architectures.

As mentioned before, for all of the four introduced CNNs, we train two cases — one in which models are learning all the weights from scratch, and the other one inheriting the original weights except for the last two layers, and this latter case we call pre-trained. Our model training demonstrates that all architectures are highly capable of distinguishing pneumonia. However, the models with the largest architecture achieved the highest results. The results are presented in Table 6.2 and Figure 6.4. In the table, we show in bold the highest value for each metric. The specificity value varies between 83% and 92% among tested models, while the sensitivity retains high value of approximately 99%. Following the definition of metrics in Section 6.2, this shows that the models are better in classifying CXR images with pneumonia, than those not containing the illness. Taking into account that both cases are equally important, we turn our focus to accuracy values of each model. From Table 6.2 and Figure 6.4 we can see that there are overall two winners — DenseNet and ResNet, both with the pre-trained weights. We already mentioned that accuracy is the main metric of interest for us, hence, even though all the models yield a high accuracy value, we can consider that a pre-trained DenseNet is an absolute winner for our problem, since it also yielded the highest sensitivity and specificity value among all models tested. This is shown at Figure 6.4, where we can notice that this CNN gives overall the tallest metric bars. The initial results show promise, subject to their possible replication on bigger and more diverse datasets.

CNN	Accuracy	Sensitivity	Specificity	Precision	F-Score
AlexNet PT	0.9519	0.9793	0.8727	0.9570	0.9680
AlexNet	0.9519	0.9702	0.8989	0.9653	0.9677
DenseNet PT	**0.9712**	0.9871	**0.9251**	**0.9745**	**0.9807**
DenseNet	0.9500	0.9871	0.8427	0.9478	0.9670
ResNet PT	0.9548	**0.9884**	0.8577	0.9526	0.9702
ResNet	0.9404	0.9754	0.8390	0.9460	0.9605
Vgg PT	0.9471	0.9702	0.8801	0.9591	0.9646
Vgg	0.9356	0.9625	0.8577	0.9514	0.9569

Table 6.2: Results for different CNN architectures. The abbreviation PT next to the name indicates that it is a pre-trained model.

In terms of accuracy, VGG without pretrained weights has lowest classification ability of approximately 93% in identifying both the positive and negative cases, as well as lowest F1-score of 95%. Lowest sensitivity value was found for VGG, in case where weights were not pre-trained. From these and from the obtained accuracy values we observe one interesting thing, which is that pre-trained models are usually slightly better than those trained from scratch. Earlier, we mentioned that models trained from scratch should generally perform better on a specific problem. However, these pre-trained models were originally fitted on much larger datasets and more epochs which gives them the advantage in this case.

To compare DenseNet and ResNet in more detail, we can look at the confusion matrices in Figure 6.5 and 6.6. Figure 6.5 shows the resulting confusion matrices in the case when all used weights are pre-trained. The difference is more obvious in the upper rows, where we have actual negatives, i.e. normal lung X-rays and it is in favor of DenseNet. DenseNet was able to predict 247 normal cases correctly, with 20 cases mistaken, which accounts to an error of 7.5%. On the other hand, that error is much higher for all other cases, with 11.9%, 12.7% and 14.2% for VGG, AlexNet and ResNet, respectively. Bottom rows actually have higher importance for us since it is where the X-ray cases with pneumonia are. As we mentioned, the highest cost in terms of health hazards, is in the case of false negatives. Here a pre-trained ResNet and DenseNet outperforms all the other architectures by a high margin (the similar false negative cases is found in DenseNet with new weights and equals 10 - see Figure 6.6). Pre-trained DenseNet and ResNet were able to correctly predict 763 i.e. 764 respectively out of 773 (approximately 99%) pneumonia cases. However, DenseNet is better in false positives cases (20 versus 38). Figure 6.6 demonstrates the capability of models when weights are learned all over. In this case, we can see higher numbers in false positive, as well as in false negative predictions.

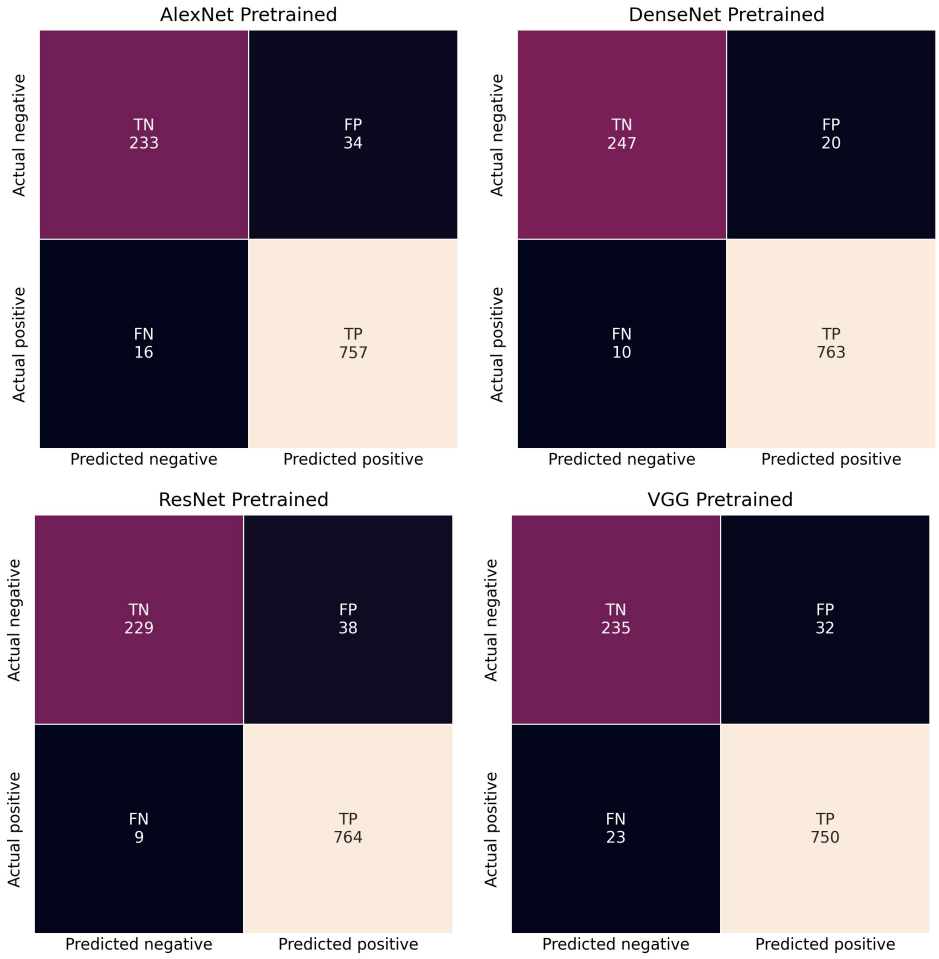

Figure 6.5: Confusion matrix for CNNs with pre-trained weights.

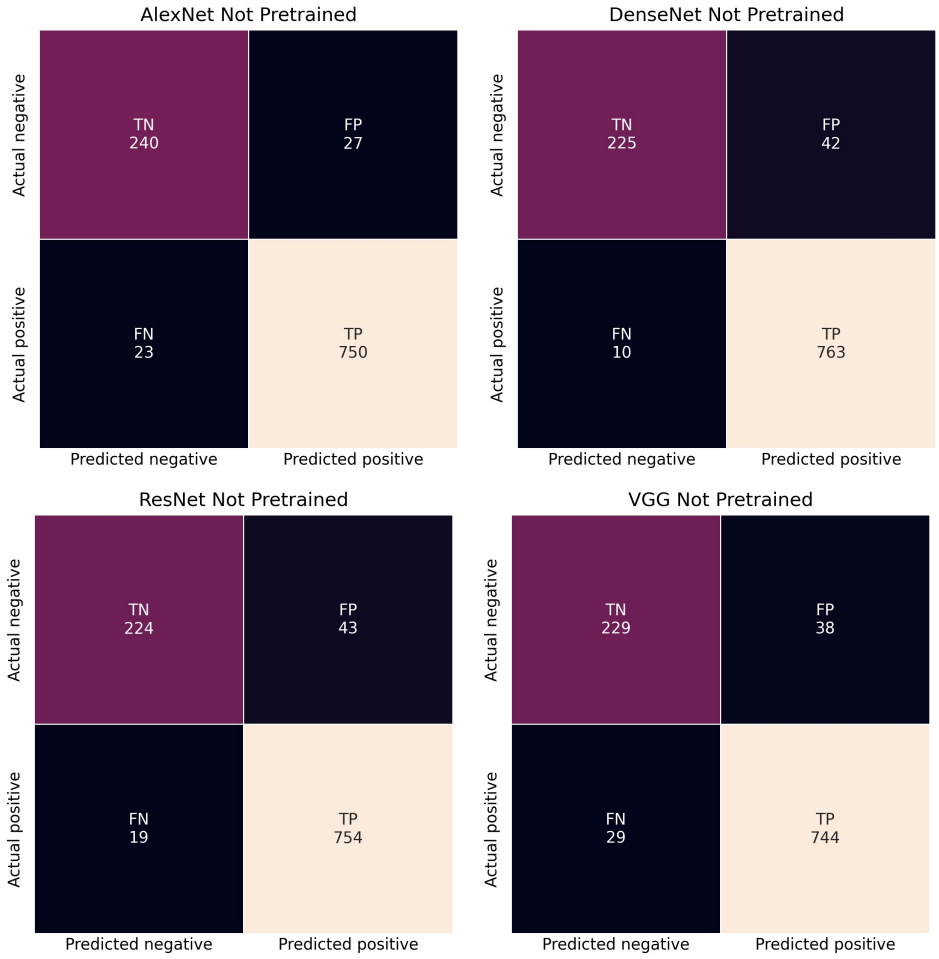

Figure 6.6: Confusion matrix for CNNs with new weights.

6.4.1 Comparison of models performances

Furthermore, we have compared performances of the models on the specific images to detect when the cases of false positive and false negative occurs. This is interesting to notice, taking into account that both DenseNet and ResNet achieved the highest overall accuracy. However, if we look deeper in how certain the model is, we can conclude that DenseNet performs much better, as it shows lower certainty in falsely classified and higher certainty in correct classifications. Since we are dealing with only two classes, those are represented with indices 0 and 1. Models return 0 when there is no pneumonia detected in an image i.e. normal or healthy image and returns 1 if it is classified as pneumonia case. Tables 6.3, 6.4 and 6.5 show model classifications for specific images with their respective estimations of classification sureness in percentages. Table 6.3 demonstrates performance of models when tested on normal images, in which case we expect 0 as an output. We notice that, in almost all cases, DenseNet retains the highest certainty in its classification abilities when the classification is correct. On the other hand, DenseNet retains the lowest numbers in false positive cases (example 1 and 2). We have also found that, in multiple cases, when other models falsely classified taken image as a pneumonia case, DenseNet outperformed them and detected no pneumonia. This is demonstrated in example 3 and 4, where DenseNet classified these images as class 0 with confidence level of 85.24% and 52.08%. This would mean that in the fourth example, it was 52.08% certain that it is normal CXR image, while 47.92% certain that it is a pneumonia case.

	Normal image							
	Example 1		Example 2		Example 3		Example 4	
	class	%	class	%	class	%	class	%
AlexNet	1	98.07	1	96.58	0	61.59	1	92.47
DenseNet	1	82.70	1	68.93	0	85.24	0	52.08
ResNet	1	84.19	1	70.79	1	87.49	1	52.09
VGG	1	89.50	1	98.98	1	59.23	1	93.99

Table 6.3: Performances of the models on normal images

Figure 6.7 shows the sample images from the test set which are falsely categorized as pneumonia cases, when actually belonging to the healthy persons. All of the tested models categorized image a) from Figure 6.7 (i.e normal example 1) as pneumonia with a very high certainty of over 80%. This can be attributed to the greater obscurity of the image and also larger opacity on the right side. That explains why all models recognized this CXR image as pneumonia case with high certainty of over 80%, with DenseNet, again, maintaining the lowest number. Similarly, image b), in the table referenced as normal example 2, contains increased

density of grey areas inside the lungs, which are usually indicators of ground glass opacity present in pneumonia cases. On the other hand, models showed different performances on images c) and d), referenced in table as normal example 3 and normal example 4, where in example 4 AlexNet and VGG classified this as pneumonia case with over 90% certainty, respectively, while DenseNet classified correctly and ResNet demonstrated lower sureness of only 52.09% of that being a pneumonia case and 47.91% that it is not.

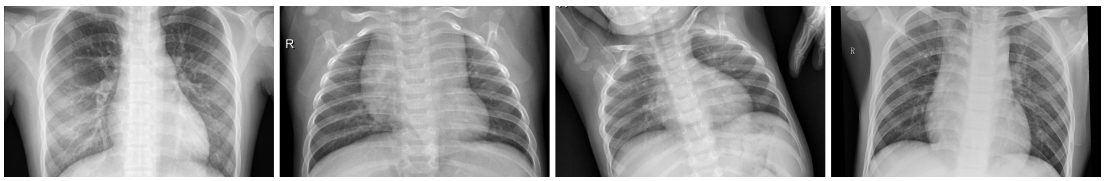

Figure 6.7: Normal image: a) Example 1, b) Example 2, c) Example 3, d) Example 4

	Pneumonia image							
	Example 1		Example 2		Example 3		Example 4	
	class	%	class	%	class	%	class	%
AlexNet	1	87.12	1	92.30	1	99.89	1	97.45
DenseNet	1	95.86	1	99.68	1	99.94	1	87.31
ResNet	1	94.82	1	99.96	1	99.91	1	99.45
VGG	0	70.57	1	91.30	1	99.72	1	68.96

Table 6.4: Performances of the models on pneumonia images

Table 6.4 shows the sureness of models in true positive detections, where examples 1 and 2 represent virus caused pneumonia, while examples 3 and 4 are bacterial pneumonia (see also Figure 6.8). In all cases, all models achieved high numbers, demonstrating high capability in detecting pneumonia when there in fact is one. Here, we have detected only one false classification of VGG in example 1. We can also notice that VGG achieved slightly lower results than other models even in true classification.

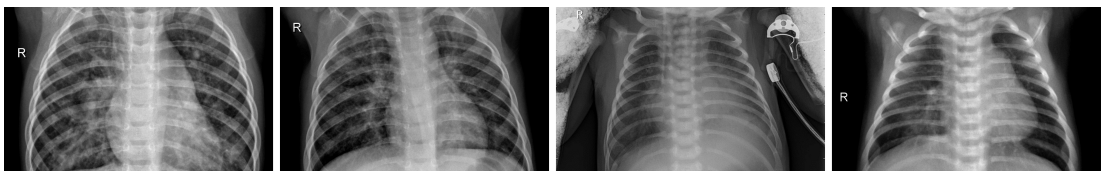

Figure 6.8: Pneumonia image: Examples 1-4

On the opposite side, Table 6.5 represents the most important case – false negative. As we explained in Section 6.2, this is the case when models fail to detect pneumonia, and these kind of mistakes can be crucial. Figure 6.9 shows the sample images of such cases. Similarly to previously discussed, in these images we can notice a smaller area of the lungs affected by ground glass opacity, which is a reason behind these mistakes. This can be attributed to a less serious case of pneumonia or an initial stage of an illness. From Table 6.5 we observe similar performance of models in false negative detections. However, we notice that all false classifications detected occurred in the case of bacterial pneumonia, while the models showed higher certainty in virus caused pneumonia. This would indicate that our methods are more suitable for, e.g. Covid19 caused pneumonia, than some other bacterial. Such evaluations should be further tested and discussed, which we leave for the future work.

	Pneumonia image							
	Example 5		Example 6		Example 7		Example 8	
	class	%	class	%	class	%	class	%
AlexNet	0	95.29	0	69.16	0	71.09	1	80.42
DenseNet	0	98.43	1	76.96	0	77.79	1	90.36
ResNet	0	98.01	1	78.52	0	57.50	0	57.03
VGG	0	98.31	0	65.82	1	62.87	0	69.05

Table 6.5: Performances of the models on pneumonia images

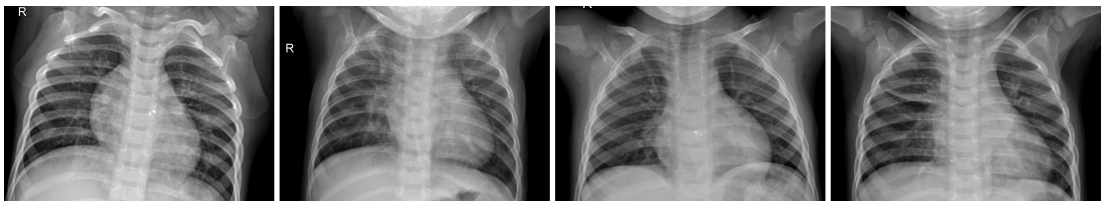

Figure 6.9: Pneumonia image: Examples 5-8

7 Conclusion and Future Work

As recent events increased the number of CXR images needed to scan, there is also a need for developing an alternative approach which is fast, cheap, simple and reliable. In this study, we propose using deep learning methods, more specific, we observe four CNN architectures that were originally trained for general-purpose image classification, and we adjusted them to specialize in detecting pneumonia in chest X-ray images. We assessed the performance of the network based on accuracy, sensitivity and specificity, but setting accuracy as main differentiator. The obtained results show very high accuracy for each CNN, which in both approaches with pre-trained or with new weights (it is always over 90%). The best performing models are DenseNet and ResNet (both with pre-trained weights) that achieve accuracy over 97% and 95% respectively. As such, they could be readily used as assistance in the medical diagnosis of pneumonia. It is important to mention that this relatively small dataset was trained for only five epochs; hence if we would significantly increase any of these two factors, the results would undoubtedly be even better. In the future, we hope to acquire larger dataset and test the models again, possible for higher number of epochs.

While the models tested in this study show high classification performance, several issues have emerged concerning their clinical applicability. The most common issue in this kind of problems is the data quality and quantity. The scarcity of the dataset is especially common while working with medical images due to data privacy. First of all, the images must be gathered from trustworthy source, afterwards the training set images must be carefully classified by an expert, deciding whether the patient has pneumonia or not. The majority of these medical datasets were derived from public repositories, they were aggregated from various sources and typically do not include metadata and associated clinical information that may allow researchers to verify its validity. What is more, such datasets exclude demographical data of the patient, restricting researchers to evaluate dependence of the results on the gender, age and lifestyle of the patient.

There are several directions that could be an interesting continuation of this work. One idea is to exclude a horizontal flip from the allowed transformations within the data augmentation procedure. This is motivated by the fact that the human hearth is always on the left side of the chest, which may result in higher fidelity of the input and further a better model. Additionally, it would be interesting to include over/under-sampling within the augmenter. A current dataset is unbalanced, and, although the ratio between two classes is not critically large, if we make them perfectly balanced, that might positively impact the resulting confusion matrices. Finally, recall that our models solve a binary classification

problem, only outputting the answer whether the patient has pneumonia or not. An improvement to the model would be to also include detection of the inflamed region. If the model is meant to assist human experts, bounding boxes around the inflamed areas would be beneficial as it would reduce the confirmation time. Futhermore, if the model also gives a confidence value for each prediction, eventually, the predictions with high enough confidence (say over 85%) could be trusted without a need for manual inspection, which would further reduce the need for human intervention.

Future work could also include testing these models on different types of pneumonia and discussing if they are able to distinguish between cases. As mentioned in Section 3, this would allow us to train a model specialized, for example, for detecting Covid-19 caused pneumonia, and therefore fasten the diagnose making and treatment of the patient. To specifically test the detection capability of the model in distinguishing Covid-19 from normal CXR, or among other types of pneumonia, one would have to introduce more than two classes for the classification process and train the models again.

Bibliography

[1] Ben-Hur A., Horn D., Siegelmann H., Vapnik V., Support vector clustering, *Journal of Machine Learning Research*, **2**:125–137, 2001.

[2] Hertz J., Palmer R., Krogh A., Introduction to the theory of neural computation, *Addison-Wesley Publishing Company*, **1**991.

[3] Valueva M.V., Nagornov N.N., Lyakhov P.A., Valuev G.V., Chervyakov N.I, Application of the residue number system to reduce hardware costs of the convolutional neural network implementation, *Mathematics and Computers in Simulation. Elsevier BV*, **1**77:232–243, 2020.

[4] Hinton G.E., Osindero S., Teh Y.W., A fast learning algorithm for deep belief nets, *Neural Computation*, **18**(2):1527-54, 2006.

[5] Felix Gers, Long Short-Term Memory in Recurrent Neural Networks, *Lausanne, EPFL*, 2001.

[6] Wang L., Lin Z.Q., Wong A., COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images, *Waterlo Artifcial Intelligence Institute, Canada*, 2020.

[7] https://github.com/lindawangg/COVID-Net

[8] Basu S., Mitra S., Saha N., Deep learning for screening covid-19 using chest x-ray images, *Symposium Series on Computational Intelligence*, 2020.

[9] Baltazer L.R., Manzanillo M.G., Gaudillo J., Vira E.D., Domingo M., Tiangco B., Albia J., Artificial intelligence on COVID-19 pneumonia detection using chest xray images, *Institute of Mathematical Sciences and Physics, University of the Philippines Los Baños, Philippines*, 2021.

[10] https://github.com/lpbaltazar/COVID-CXR-AI

[11] Li L., Qin L., Xu Z., et al., Artificial Intelligence Distinguishes COVID-19 from Community Acquired Pneumonia on Chest CT, *Department of Radiology, Wuhan, China*, 2020.

[12] Bai H.X., Wang R., Xiong Z., Hsieh B., Chang K., Halsey K., et al. Artificial intelligence augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of other origin at chest CT, *Radiology*, 2020.

[13] Razzak M.I., Naz S., Zaib A., Deep Learning for Medical Image Processing: Overview, Challenges and Future.

[14] Han Y., Chen C., Tewfik A., Ding Y., Peng Y., Pneumonia detection on Chest X-ray using Radiomic Features and Contrastive Learning, *Computer Science, Engineering*, 2021.

[15] Lundervold A., Lundervold A.S., An overview of deep learning in medical imaging focusing on MRI, **2**018.

[16] Alzubaidi L., Zhang J., Humaidi A., Al-Dujaili A., Duan Y., Al-Shamma O., Santamaría J., Fadhe M., Al-Amidie M., Farhan L., Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, **2**021.

[17] Vladimir Crnjojević, Prepoznavanje oblika za inženjere, *FTN Izdavaštvo, Novi Sad*, **2**014.

[18] Christopher M. Bishop, Pattern Recognition and Machine Learning, *Springer Science+Business Media*, **2**006.

[19] Han J., Kamber M., Pei J., Data Mining Concepts and Techniques, *Elsevier Inc.*, **2**012.

[20] Goodfellow I., Bengio Y., Courville A., Deep Learning, *MIT Press*, **2**015.

[21] Gonzalez R., Woods R., Digital Image Processing, *Pearson Education Limited*, **2**018.

[22] Francois Chollet, Deep Learning with Python, *Manning Publications Co.*, **2**018.

[23] Zhang D., Ren F., Li Y., Na L., Ma Y., Pneumonia Detection from Chest X-ray Images Based on Convolutional Neural Network *Electronics* , 2021.

[24] https://www.kaggle.com/datasets/pcbreviglieri/pneumonia-xray-images

[25] Lu, Siyuan and Lu, Zhihai and Zhang, Yu-Dong, Pathological brain detection based on AlexNet and transfer learning *Elsevier* , 2019.

[26] Shanthi, T and Sabeenian, RS, Modified Alexnet architecture for classification of diabetic retinopathy images *Elsevier* , 2019.

[27] Mateen, Muhammad and Wen, Junhao and Song, Sun and Huang, Zhouping, Fundus image classification using VGG-19 architecture with PCA and SVD *MDPI* , 2018.

[28] Guan, Qing and Wang, Yunjun and Ping, Bo and Li, Duanshu and Du, Jiajun and Qin, Yu and Lu, Hongtao and Wan, Xiaochun and Xiang, Jun, Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study *Ivyspring International Publisher* , 2019.

[29] Farooq, Muhammad and Hafeez, Abdul, Covid-resnet: A deep learning framework for screening of covid19 from radiographs *arXiv preprint arXiv:2003.14395* , 2020.

[30] Jiang, Yun and Chen, Li and Zhang, Hai and Xiao, Xiao, Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module *Public Library of Science San Francisco, CA USA* , 2019.

[31] Wang, Shui-Hua and Zhang, Yu-Dong, DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* , 2020.

Biography

Dusan Binic was born on March 10, 1996, in Kruševac. He finished elementary school "Aca Aleksić" in Aleksandrovac in 2011, after which he enrolled in high school "Sveti Trifun". During primary and secondary school, he competed for the cadet national team of Serbia in handball. Undergraduate studies in Applied Mathematics - module Mathematics of Finance at the Faculty of Sciences in Novi Sad in 2015, which ends in 2019 in January. In the same year, he enrolled in the master studies of Applied Mathematics - module Data Science at the same faculty. He was a participant in the Festival of Sciences in Novi Sad and Belgrade. He worked at the elementary school "Milos Crnjanski" in Novi Sad as a professor of mathematics. He is currently employed by Erste Bank in Novi Sad in the Strategic Risks Sector, Methods and Risk Models Department.

UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj:
RBR
Identifikacioni broj:
IBR
Tip dokumentacije: monografska dokumentacija
BF
Tip zapisa: tekstualni štampani materijal
TZ
Vrsta rada: Master rad
VR
Autor: Dušan Binić
AU
Mentor: dr Oskar Marko
MN
Naslov rada: Dijagnostika upale pluća na osnovu rendgenskih snimaka pacijenta upotrebom mašinskog učenja.
NR
Jezik publikacije: engleski
JP
Jezik izvoda: engleski
JI
Zemlja publikovanjan: Republika Srbija
ZP
Uže geografsko područje: Vojvodina
UGP
Godina: 2022.
GO
Izdavač: autorski reprint
IZ
Mesto i adresa: Novi Sad, Trg Dositeja Obradovića 4
MA
Fizički opis rada:7/43/31/5/17/
(broj poglavlja/broj strana /lit. citata/tabela/grafikona)
FO
Naučna oblast: matematika
NO
Naučna disciplina: primenjena matematika

ND
Predmetna odrednica/Ključne reči: konvolucija, neuralne mreže, klasifikacija
PO
UDK:
Čuva se: u biblioteci Departmana za matematiku i informatiku, Prirodno-matematičkog fakulteta, u Novom Sadu
ČU
Važna napomena:
VN
Izvod: Upala pluca se obično dijagnostikuje pregledom rendgenskih snimaka grudnog koša. Slike su napravljene po međunarodnim standardima, uvek dobro usklađene, monohromne i istih dimenzija. Iako ove karakteristike čine zadatak savršeno pogodan za algoritme mašinskog učenja, proces dijagnoze je i dalje skoro isključivo ručni. Međutim, pandemija COVID-19 izazvala je iznenada ogroman povecanje slučajeva upale pluca i broj proizvedenih rendgenskih zraka, koji je povecao potražnju za automatizovanim rešenjem za pomoc ljudskim stručnjacima, pracena mnogim istraživačkim radovima koji predlažu rešenja zasnovana na neuronskim mrežama. U ovoj tezi cemo pristupiti problemu koristeci konvolucione neuronske mreže, a konkretno, uporedićemo četiri odgovarajuce arhitekture: AlexNet, DenseNet, ResNet i VGG. Konačni eksperimenti pokazuju da čak i uz neznatna podešavanja originalnih težina, u stanju smo da proizvedemo modele koji postižu preko 94% tačnosti u dijagnozi pneumonije. Prethodno obučeni DenseNet i ResNet su postigli najviše performanse sa 97% i 99% tačnosti u razlikovanju pneumonije od normalnih slučajeva.
IZ
Datum prihvatanja teme od strane NN veća: 28.09.2021.
DP
Datum odbrane: 05.07.2022.
DO
Članovi komisije:
KO
Predsednik: prof. dr Srđan Škrbić, redovni profesor, Prirodno-matematički fakultet, Novi Sad
Mentor: dr Oskar Marko, naučni saradnik instituta BioSense u Novom Sadu
Član: dr Sanja Brdar, naučni saradnik instituta BioSense u Novom Sadu

UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCES
KEY WORD DOCUMENTATION

Accession number:
ANO
Identification number:
INO
Document type: monograph type
DT
Type of record: printed text
TR
Contents code: Master thesis
CC
Author: Dušan Binić
AU
Mentor: PhD Oskar Marko
MN
Title: Diagnosis of Pneumonia Based on X-rays of the Patient Using Machine Learning
TI
Language of text: English
LT
Language of abstract: English
LA
Country of publication: Republic of Serbia
CP
Locality of publication: Vojvodina
LP
Publication year: 2022.
PY
Publisher: author's reprint
PU
Publ. place: Novi Sad, Trg Dositeja Obradovića 4
PP
Physical description:7/43/31/5/17/
(chapters/pages/literature/tables/graphics)
PD
Scientific field: mathematics
SF
Scientific discipline: Applied mathematics

SD
Subject / Key words: convolution, neural network, classification
SKW
UC:
Holding data: Department of Mathematics and Informatics' Library, Faculty of Sciences, Novi Sad
HD
Note:
N
Abstract: Pneumonia is usually diagnosed by inspecting X-ray images of the chest. These images are made following international standards, always well aligned, monochromi- cal, and with the same dimensions. Although these characteristics make the task perfectly suited for machine learning algorithms, the process of diagnosis is still al- most exclusively manual. However, the COVID-19 pandemic caused a sudden huge increase in pneumonia cases and the number of X-rays produced, which increased the demand for an automated solution to assist human experts, followed by many research papers proposing neural network-based solutions. In this thesis, we will approach the problem using convolutional neural networks, and in specific, we will compare four suitable architectures: AlexNet, DenseNet, ResNet, and VGG. Final experiments show that even with slight adjustments of the original weights, we are able to produce models that achieve over 94% accuracy in pneumonia diagnosis. Pre-trained DenseNet and ResNet attained the highest performance with 97% and 99% accuracy in distinguishing pneumonia from normal cases.
AB
Accepted by the Scientific Board on: 28.09.2021.
ASB
Defended: 05.07.2022.
DE
Thesis defend board:
DB
Chair: Phd Srđan Škrbić, full professor, Faculty of science, Novi Sad
Mentor: PhD Oskar Marko, research associate at the BioSense institute in Novi Sad
Member: PhD Sanja Brdar, research associate at the BioSense institute in Novi Sad