Automatic Evaluation of the Lung Condition of COVID-19 Patients Using X-ray Images and Convolutional Neural Networks

Lorencin, Ivan; Baressi Šegota, Sandi; Anđelić, Nikola; Blagojević, Anđela; Šušteršić, Tijana; Protić, Alen; Arsenijević, Miloš; Ćabov, Tomislav; Filipović, Nenad; Car, Zlatan

Source / Izvornik: **Journal of Personalized Medicine, 2021, 11**

Journal article, Published version
Rad u časopisu, Objavljena verzija rada (izdavačev PDF)

https://doi.org/10.3390/jpm11010028

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:271:812063

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-03**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Dental Medicine

DIGITALNA KNJIŽNICA

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Journal of Personalized Medicine

MDPI

Article

Automatic Evaluation of the Lung Condition of COVID-19 Patients Using X-ray Images and Convolutional Neural Networks

Ivan Lorencin [1], Sandi Baressi Šegota [1], Nikola Anđelić [1], Anđela Blagojević [2,3], Tijana Šušteršič [2,3], Alen Protić [4,5], Miloš Arsenijević [6,7], Tomislav Ćabov [8], Nenad Filipović [2,3] and Zlatan Car [1,*]

1 Faculty of Engineering, University of Rijeka, Vukovarska 58, 51000 Rijeka, Croatia; ilorencin@riteh.hr (I.L.); sbaressisegota@riteh.hr (S.B.Š.); nandelic@riteh.hr (N.A.)
2 Faculty of Engineering, University of Kragujevac, Sestre Janjić, 34000 Kragujevac, Serbia; andjela.blagojevic@kg.ac.rs (A.B.); tijanas@kg.ac.rs (T.Š.); fica@kg.ac.rs (N.F.)
3 Bioengineering Research and Development Centre (BioIRC), Prvoslava Stojanovića 6, 34000 Kragujevac, Serbia
4 Clinical Hospital Centre, Rijeka, Krešimirova ul. 42, 51000 Rijeka, Croatia; alen.protic@uniri.hr
5 Faculty of Medicine, University of Rijeka, Ul. Braće Branchetta 20/1, 51000 Rijeka, Croatia
6 Clinical Centre Kragujevac, Zmaj Jovina 30, 34000 Kragujevac, Serbia; milos.arsenijevic83@gmail.com
7 Faculty of Medical Sciences, University of Kragujevac, Svetozara Markovića 69, 34000 Kragujevac, Serbia
8 Faculty of Dental Medicine, University of Rijeka, Krešimirova ul. 40, 51000 Rijeka, Croatia; tomislav.cabov@fdmri.uniri.hr
* Correspondence: car@riteh.hr

Abstract: COVID-19 represents one of the greatest challenges in modern history. Its impact is most noticeable in the health care system, mostly due to the accelerated and increased influx of patients with a more severe clinical picture. These facts are increasing the pressure on health systems. For this reason, the aim is to automate the process of diagnosis and treatment. The research presented in this article conducted an examination of the possibility of classifying the clinical picture of a patient using X-ray images and convolutional neural networks. The research was conducted on the dataset of 185 images that consists of four classes. Due to a lower amount of images, a data augmentation procedure was performed. In order to define the CNN architecture with highest classification performances, multiple CNNs were designed. Results show that the best classification performances can be achieved if ResNet152 is used. This CNN has achieved $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ up to 0.94, suggesting the possibility of applying CNN to the classification of the clinical picture of COVID-19 patients using an X-ray image of the lungs. When higher layers are frozen during the training procedure, higher $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ values are achieved. If ResNet152 is utilized, $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ values up to 0.96 are achieved if all layers except the last 12 are frozen during the training procedure.

Keywords: AlexNet; Convolutional Neural Network; COVID-19; ResNet; VGG-16

check for updates

Citation: Lorencin, I.; Baressi Šegota, S.; Anđelić, N.; Blagojević, A.; Šušteršič, T.; Protić, A.; Arsenijević, M.; Ćabov, T.; Filipović, N.; Car, Z. Automatic Evaluation of the Lung Condition of COVID-19 Patients Using X-ray Images and Convolutional Neural Networks. *J. Pers. Med.* **2021**, *11*, 28. https://doi.org/10.3390/jpm11010028

Received: 4 December 2020
Accepted: 27 December 2020
Published: 4 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Coronavirus disease 2019 (COVID-19) caused by Severe Acute Respiratory Syndrome virus 2 (SARS-CoV-2) is a viral, respiratory lung disease [1]. The spread of COVID-19 has been rapid, and it has affected the daily lives of millions across the globe. The dangers it poses are well-known [2], with the most important ones being its relatively high severity and mortality rate [3] and the strain it exhibits on the healthcare systems of countries worldwide [4,5]. Another problematic characteristic that COVID-19 exhibits is a wide variation in severity across the patients, which can cause issues for healthcare workers who wish to determine an appropriate individual treatment plan [6]. Early determination of the severity of COVID-19 may be vital in securing the needed resources—such as planning the location for hospitalization of the patient or respiratory aids in case they may be necessary.

There is a dire need for systems that will allow for the strain put on the resources of the healthcare systems, as well as healthcare workers, by allowing easier classification of patient case severity in early stages of hospitalization. Artificial intelligence (AI) techniques have already been proven to be a useful tool in the fight against COVID-19 [7,8], so the possibility exists of them being applied in this area as well. Existence of such algorithms may lower the strain on the potentially scarce resources, by allowing early planning and allocation. Additionally, they may provide decision support to overworked healthcare professionals.

Internet of Medical Things (IoMT) is a medical paradigm that allows for integration of modern technologies in the existing healthcare system [9]. The algorithms developed as a part of the presented research can be made available to health professionals using IoMT [10]. Models obtained using the described methodology can be integrated inside a pipeline system in which an X-ray image will automatically be processed using the developed models, and the predicted class of the patient whose image has been obtained will immediately be delivered to the medical professional examining the X-ray. Such automated diagnosis methods have already been applied in many studies, such as in histopathology [11], neurological disorders [12], urology [13], and retinology [14]. All the researchers agree that not only can such AI-based support systems provide an extremely precise diagnosis, but can also be integrated in automatic systems to provide assistance to medical experts in determining the correct diagnosis. The obtained models are suited for such an approach. While the training of the models is slow due to the backpropagation process, the classification (using forward propagation) is fast and computationally moderate [15,16], allowing for easy integration into existing in-hospital systems.

The machine learning diagnostic approach has been successfully applied to X-ray images a number of times in the past. For example, Lujan–Garcia et al. (2020) [17] demonstrated the application of CNNs for the detection of pneumonia using chest X-ray images using Xception CNN, which was pre-trained using a ImageNet dataset for initial values. The evaluation was performed using precision, recall, F1 score, and AUROC, with the achieved scores being 0.84, 0.99, 0.91, and 0.97, respectively. Kieu et al. (2020) [18] demonstrated the Multi-CNN approach to the detection of abnormalities on the chest X-ray images. The approach presented in the paper demonstrates the use of multiple CNNs to determine the class of the input image, with the hybrid system presented achieving an accuracy of 96%. Bullock et al. (2019) [19] presented XNet—a CNN solution designed for medical, X-ray image segmentation. The presented solution is suitable for small datasets, and achieves high scores (92% accuracy, F1 score of 0.92 and AUC of 0.98) on the used dataset. Takemiya et al. (2019) [20] demonstrated the use of R-CNNs (Region with Convolutional Neural Network) in the detection of pulmonary nodules from the images of chest X-ray images. The proposed method utilizes the Selective Search algorithm to determine the potential candidate regions of chest X-rays and applied the CNN to classify the selected regions into two classes—nodule opacities and non-nodule opacities. The presented approach achieved high classification accuracy. Another example is by Stirenko et al. (2018) [21], in which the authors applied the deep learning, CNN approach to the X-ray images of patients with tuberculosis. The CNN is applied to a small and non-balanced dataset with the goal of segmentation of chest X-ray images, allowing for classification of images with higher precision in comparison to non-segmented images. In combination with data augmentation techniques, the achieved results are better. Authors conclude that data augmentation and segmentation, combined with dataset stratification and removal of outliers, may provide better results in cases of small, poorly balanced datasets.

There was research that utilized transfer learning methodologies in order to recognize respiratory diseases from chest X-ray images. In [22], the authors proposed a transfer learning approach in order to recognize pneumonia from X-ray images. The proposed approach, based on utilization of ImageNet weights has resulted with high accuracy of pneumonia recognition (96.4%). Another transfer learning approach has been implemented

for pneumonia detection in [23]. By utilizing such an approach, a highly accurate multi-class classification can be achieved, with accuracy ranging from 93.3% to 98%.

Wong et al. [24] (2020) noted that radiographic findings do indicate positivity in COVID-19 patients, the conclusions of which are further supported by Orsi et al. [25] (2020) and Cozzi et al. [26] (2020). Borghesi and Maroldi [27] (2020) defined a scoring system for X-ray COVID-19 monitoring, concluding that there is a definite possibility of determining the severity of the disease through the observation of X-ray images. Research has been done in the application of AI in the detection of COVID-19 in patients. Recently, classification of patients for preliminary diagnosis has been done from cough samples. Authors Imran et al. [28] (2020) applied and implemented this into an app, called AI4COVID-19.

Bragazzi et al. [29] (2020) demonstrated the possible uses of information and communication technologies, artificial intelligence, and big data in order to handle the large amount of data that may be generated by the ongoing pandemic. Further reviews and comparisons of mathematical modeling, artificial intelligence, and datasets for the prediction were done by multiple authors, such as Mohamadou et al. [30] (2020), Raza [31] (2020), and Adly et al. [32] 2020. All aforementioned authors concluded the possibility of application of AI in the current and possibly forthcoming pandemics. Most promise in AI applications being applied in this field has been shown in the field of epidemiological spread. Zheng et al. [33] (2020) applied a hybrid model for a 14-day period prediction, Hazarika et al. [34] (2020) applied wavelet-coupled random vector functional neural networks, while Car et al. [35] (2020) applied a multilayer perceptron neural network for the goal of regressing the epidemiology curve components. Ye et al. [36] (2020) demonstrated a α-Satellite, AI-driven system for risk assessment at a community level. Authors demonstrate the usability of such a system in combat against COVID-19, as a system that displays risk index and the number of cases among all larger locations across the United States. Authors in [37] have proposed a method for forecasting the impact of COVID-19 on stock prices. The approach based on stationary wavelet transform and bidirectional long short-term memory has shown high estimation performances.

Still, a large amount of work was also done in the image classification and detection of COVID-19 in patients. Wang et al. [38] (2020) demonstrated the use of high-complexity convolutional neural networks in the application of COVID-19 diagnosis. Their COVID-Net custom architecture reached high sensitivity scores (above 90%) in the detection of the COVID-19 in comparison to other infections and a normal lung state. Narin et al. [39] (2020) also demonstrated a high-quality solution using deep convolutional neural networks on X-ray images. Through the application of five different architectures (ResNet50, ResNet101, ResNet152, Inception V3, and Inception-ResNetV2) high scores were achieved (accuracy 95% or higher) by the authors. Ozturk et al. [40] (2020) developed a classification network for classifying the inflammation, named DarkCovidNet. DarkCovidNet reached an impressive score in binary classification at 98.08% in the case of binary classification, but a significantly lower score for 87.02% for multi-label classification. In the presented case, a multi-label classification was conducted with the aim of differentiating X-ray images of the lungs of healthy patients, patients with COVID-19, and patients with pneumonia. Abdulaal et al. [41] (2020) demonstrated the AI-based prognostic model of COVID-19, achieving accuracy levels of 86.25% and AUC ROC 90.12% for UK patients.

There have been studies proposing a transfer-learning approach to COVID-19 diagnosis from X-ray images of the chest. The study presented in [42] used pre-trained CNNs in order to automatically recognize COVID-19 infection. Such an approach has enabled high classification performances with an accuracy level of up to 99%. The research presented in [43] proposed a similar approach in order to differentiate pneumonia from COVID-19 infection. Transfer learning has enabled higher classification accuracy with utilization of simpler CNN architecture, such as VGG-16.

While a lot of work suggests that neural networks may be used for the detection of COVID-19 infection, there is an apparent lack of work that tests the possibility of finding the severity of COVID-19 through patients' lung X-rays. Such an approach would allow for

automatic detection and prediction of case severity, allowing healthcare professionals to determine the appropriate approach and to leverage available resources in the treatment of that individual patient. Development of an AI basis for such a novel system is the goal of this paper. From a literature overview, it can be noticed that all presented research has been based on a binary classification of X-ray images (infected/not infected) or differentiating COVID-19 infection and other respiratory diseases.

To summarize the novelty, this article, unlike the articles presented in the literature review, deals with a multi-class classification of X-ray images of positive COVID-19 patients with the aim of estimating the clinical picture. All the examples have used a large number of images (larger than 1000) in the training and testing processes of the neural network. While the number of COVID-19 patients is high, data collection, especially in countries with lower quality healthcare systems, may be problematic due to the strain exhibited by the coronavirus. Because of this, it is important to test the possibility of algorithm development combined with data augmentation operations, which is the secondary goal of the presented research.

According to presented facts and the literature overview, the following questions arise:

- Is it possible to utilize CNN in order to classify COVID-19 patients according to X-ray images of lungs?
- Which CNN architecture achieves the highest classification performance?
- Which are the best-performing configurations in regards to the solver, number of iterations, and batch size?
- How do transfer learning and layer freezing influence the performances of the best configurations?

2. Dataset Construction

In this section, a brief description of the used dataset will be provided, together with examples of each class. Furthermore, a data augmentation technique will be presented. At the end, divisions in the training, validation, and testing sets will be presented.

2.1. Dataset Description

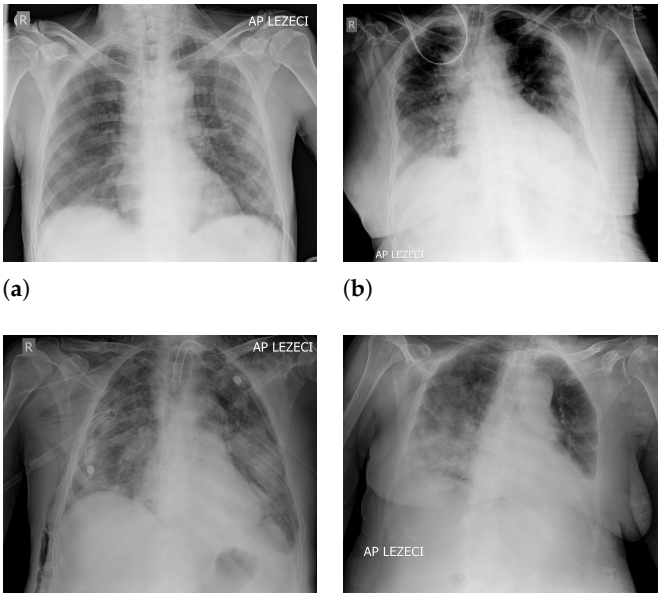
The dataset used in this research was obtained from the Clinical Centre in Kragujevac [44] and consists of 185 X-ray images that represent the lungs of 21 patients diagnosed with COVID-19. The dataset consists of 7 female and 18 male patients, and age of patients in the form of mean $\pm$ standard deviation was 58.9 ± 11.1 years. Images have been divided into four groups according to the clinical picture of the patient. Classification to a clinical picture was performed according to the clinical data that contained parameters such as:

- Clinical picture description;
- Physical examination;
- Laboratory examination; and
- X-ray finding.

According to the aforementioned division, images have been classified into classes:

- Mild clinical picture;
- Moderate clinical picture;
- Severe clinical picture; and
- Critical clinical picture.

An overview of image classes has been presented in Figure 1, where each class is represented with a X-ray image.



(**a**) (**b**) (**c**) (**d**)

Figure 1. Examples of X-ray images contained in the dataset: (**a**) A mild clinical picture; (**b**) moderate clinical picture; (**c**) severe clinical picture; and (**d**) critical clinical picture.

For the purposes of this research, X-ray images collected during treatment have been used to create a dataset. The dataset was created with respect to the clinical picture of the patient, where each X-ray image was classified to the appropriate class. Data distribution according to classes is presented in Figure 2.

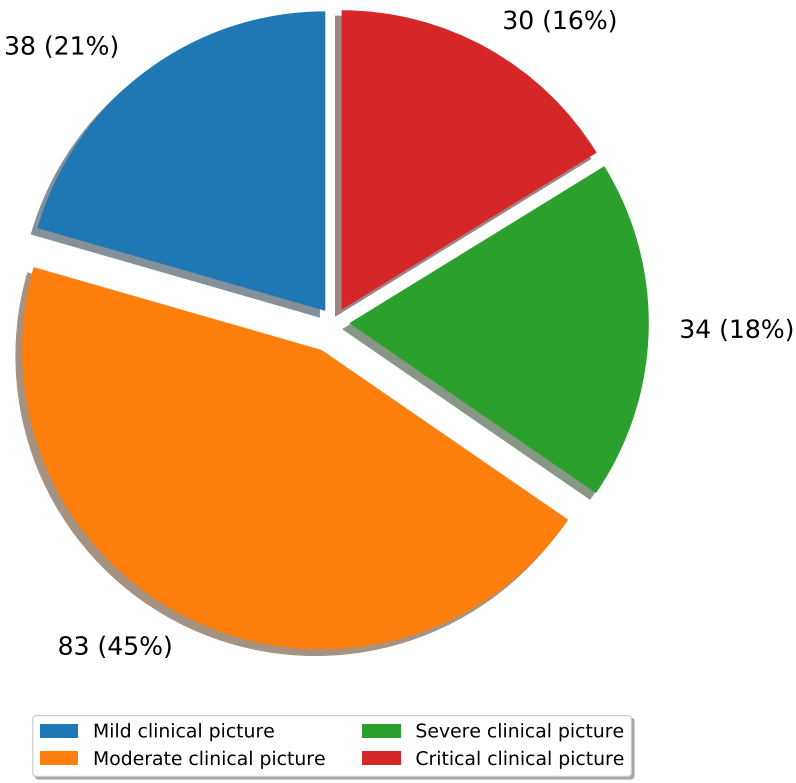


Figure 2. Overview of dataset distribution.

2.2. Description of Data Augmentation Technique and Resulting Dataset

Due to a small amount of images in the dataset, a process called augmentation has been utilized in order to increase the classification performances [45]. The augmentation procedure was performed with the aim of artificially increasing the training dataset [46], while the testing dataset remained the same. This procedure is often used in fields such as bio-medicine, due to the fact that a large amount of bio-medical data is often unavailable [47]. In this particular case, a set of geometrical operations was utilized in order to increase the dataset. The aforementioned geometrical operations are:

- 90 degree rotation around sagittal axis,
- 180 degree rotation around sagittal axis,
- 270 degree rotation around sagittal axis,
- 180 degree rotation around longitudinal axis,
- 180 degree rotation around longitudinal axis combined with 90 degree rotation around sagittal axis,
- 180 degree rotation around longitudinal axis combined with 180 degree rotation around sagittal axis, and
- 180 degree rotation around longitudinal axis combined with 270 degree rotation around sagittal axis.

In addition to the above list, brightness augmentation was also performed. All the images obtained by the geometrical transformations given above were further modified by multiplying all image pixel values with factors 0.8, 0.9, 1.1, and 1.2 in addition to the original brightness.

The 90-degree rotation presents an operation that rotates the original image (presented with Figure 3a) by 90 degrees in a clockwise direction around the sagittal axis, as presented in Figure 3b. Following the presented logic, rotations for 180 and 270 degrees were performed as well, as presented in Figure 3c,d. Images rotated by 90 and 270 degrees were resealed in order to have the same dimensions as the original image. Image generation by 180-degree rotation around the longitudinal axis was performed in such a way that the new image represented a mirrored projection of the original image, as presented in Figure 3e. The mirrored image was rotated around the sagittal axis, forming three new variations, as presented in Figure 3f–h. As the final approach to image augmentation, a process of multiplication of all image pixels with a certain factor is proposed. In this case, four different factors (0.8, 0.8, 1.1, and 1.2) were used. The described transformations have been presented on an original image with Figure 3i–l. It is important to notice that such transformations were applied on an augmented set that was created by using all described geometrical transformations. By using such an approach, the new augmented dataset was four times larger than the dataset created by using just geometrical transformations.

Only geometrical transformations and multiplication of all image pixels with a certain factor were used for data augmentation in order to keep the entire data of the image. Other techniques, such as scaling, could remove parts of the image, so they were considered inappropriate due to the nature of the problem, which observes the entire image as it is delivered from the hospital X-ray system.

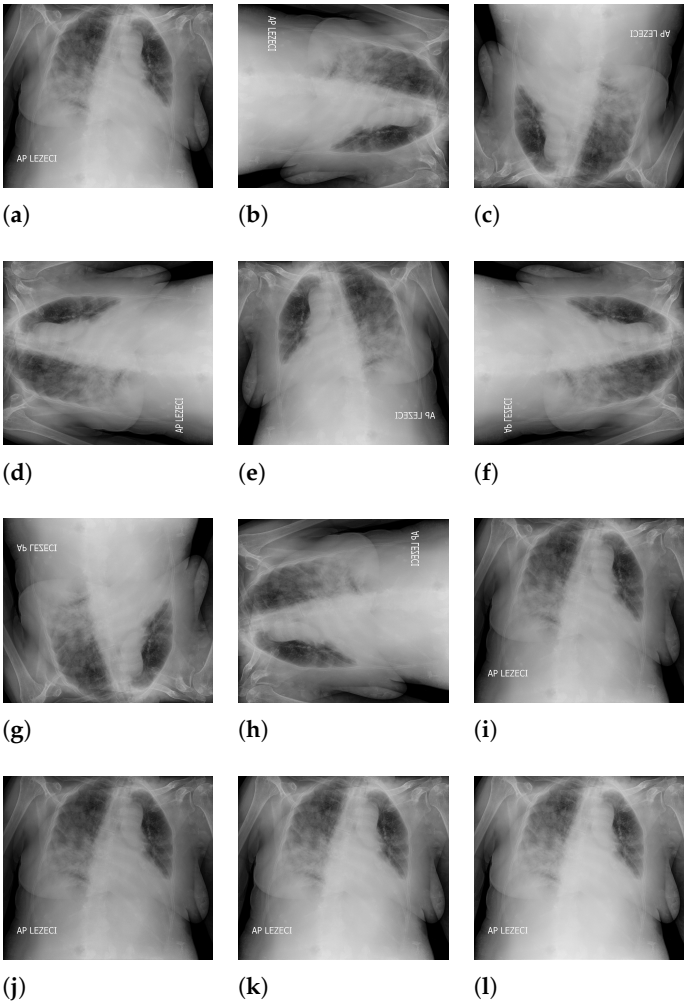


Figure 3. Overview of image augmentation procedure ((**a**): original image; (**b**): image rotated for 90 degrees around sagittal axis; (**c**): image rotated for 180 degrees around sagittal axis; (**d**): image rotated for 270 degrees around sagittal axis; (**e**): image rotated for 180 degree around longitudal axis; (**f**): image rotated for 180 degree around longitudal axis and rotated for 180 degree around sagittal axis; (**g**): image rotated for 180 degree around longitudal axis and rotated for 180 degree around sagittal axis; (**h**): image rotated for 180 degree around longitudal axis and rotated for 270 degree around sagittal axis; (**i**): image with pixels multiplied by a factor 0.8; (**j**): image with pixels multiplied by a factor 0.9; (**k**): image with pixels multiplied by a factor 1.1; (**l**): image with pixels multiplied by a factor 1.2).

By using the augmentation process described in the previous paragraphs, a new augmented dataset of 5400 images was constructed. The class distribution of the new set is presented in Figure 4a. It is important to notice that for the creation of the augmented dataset, only images contained in the original training set were used. In other words, images used for classifier testing were not used for the creation of the augmented set. According to the presented fact, the training set of 881 images was divided into training and validation sets in a 75:25 manner, as a ratio common in machine-learning practice. The presented sets were used for the training of CNNs, while the original testing set was used for the evaluation of their classification performances. The above-described dataset division is presented in Figure 4b.

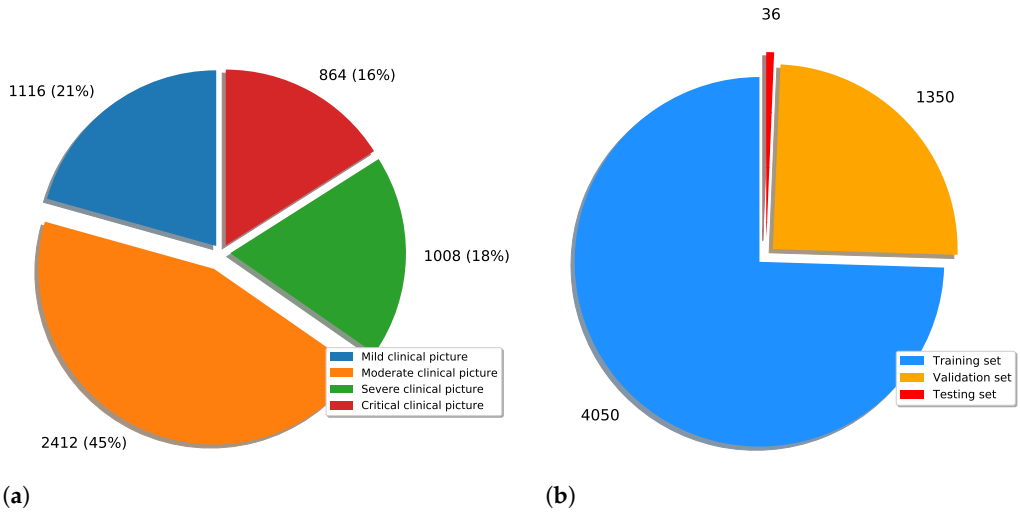


Figure 4. Representation of augmented dataset ((**a**): class distribution; (**b**): training-validation-testing division).

3. Description of Used Convolutional Neural Networks

In this subsection, an overview of CNN-based methods for image classification will be presented. The CNNs used in this research are, in fact, standard CNN architectures widely used for solving various computer vision and image recognition problems [48]. Such an algorithm, alongside its variations, is widely used for various tasks of medical image recognition [49]. For the case of this research, four different CNN architectures were used, and they are:

- AlexNet,
- VGG-16, and
- ResNet.

All of the above-listed CNN architectures have predefined layers and activation functions, while other hyper-parameters, such as batch size, solver, and number of epochs could be varied. The above-listed architectures were chosen due to the history of their high classification performances in similar problems. It has been shown that ResNet architectures have achieved high classification performances when used for multi-class classification of X-ray chest images [50]. Furthermore, ResNet architectures were used in various tasks of medical data classification ranging from tumor classification [51,52], trough recognition of respiratory diseases [53], to fracture diagnosis [54,55].

Extensive searches for the optimal solution through the hyper-parameter space can also be called the grid-search procedure. Variations of hyper-parameters used during the grid-search procedure for CNN-based models are presented in Table 1.

Table 1. An overview of CNN hyper-parameters used during the grid-search procedure.

Number of Epochs	Solver	Batch Size
1	Adam [56]	2
5	Adamax [57]	4
10	Nadam [58]	8
25	-	16
50	-	-
75	-	-
100	-	-
125	-	-
150	-	-
175	-	-
200	-	-

In order to determine the influence of overfitting, the number of epochs were varied with the aim of determining the number with the highest performances on the test dataset. With respect to theoretical knowledge, it can be defined that when training with a large number of epochs, the model is often over-fitted. For this reason, it is necessary to find the optimal number of training epochs [15]. Solvers used in this research were selected due to their performance on multiple multi-label datasets [59]. In the following paragraphs, a brief description and mathematical models will be provided for each solver.

Adam Solver

The Adam optimization algorithm represents one of the most-used algorithms for tasks of image recognition and computer vision. By using the Adam optimizer, weights are updated by following [56]:

$$w_t^i = w_{t-1}^i - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t, \tag{1}$$

where $\hat{m}_t$ is defined as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{2}$$

and $\hat{v}_t$ is defined as:

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}. \tag{3}$$

m_t is defined as a running average of the gradients, and it can be described with:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) G. \tag{4}$$

Furthermore, v_i is defined as the running average of squared gradients, or:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) G^2. \tag{5}$$

G can be defined with:

$$G = \nabla_w C(w_t), \tag{6}$$

where $C(arg)$ represents a cost function. Parameters of the Adam solver used in this research are presented in Table 2.

AdaMax Solver

AdaMax solver follows the logic similar to the Adam solver—in this case, the weights update was performed as [58]:

$$w_t^i = w_{t-1}^i - \frac{\eta}{v_t + \epsilon} \hat{m}_t, \tag{7}$$

where $\hat{m}_t$ is defined as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}. \tag{8}$$

Furthermore, v_t can be defined as:

$$v_t = max(\beta_2 v_{t-1}, |G_t|), \tag{9}$$

and m_t is defined as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) G. \tag{10}$$

As it is in the case of the Adam solver, parameters used in this research are presented in Table 2.

Nadam Solver

The third optimizer used in this research in Nadam. As with the AdaMax algorithm, Nadam is also based on Adam. Weights in these case updates are as [58]:

$$w_t^i = w_{t-1}^i - \frac{\eta}{\sqrt{v_t} + \epsilon} \tilde{m}_t, \tag{11}$$

where $\tilde{m}_t$ is defined with:

$$\tilde{m}_t = \beta_1^{t+1} \hat{m}_t + (1 - \beta_1^t) \hat{g}_t. \tag{12}$$

$\hat{m}_t$ and $\hat{g}_t$ are defined as:

$$\hat{m}_t = \frac{m_t}{\prod_{i=1}^{t} \beta_1} \tag{13}$$

and

$$\hat{g}_t = \frac{g_t}{\prod_{i=1}^{t} \beta_1}. \tag{14}$$

As it is in the case of the Adam and AdaMax optimizers, the parameters of the Nadam solver are presented in Table 2.

Table 2. Parameters of each optimizer used in this research.

Solver	η	β_1	β_2	ϵ
Adam	0.001	0.9	0.99	1×10^{-8}
Adamax	0.02	0.9	0.999	1×10^{-7}
Nadam	0.001	0.9	0.999	1×10^{-7}

The presented parameters will be used for training the CNNs, and the classification performances of all trained models will be evaluated by using the testing data set. In the following paragraphs, a brief overview of the used CNN architectures will be presented.

3.1. AlexNet

AlexNet represents one of the classical CNN architectures that are used for various tasks of image recognition and computer vision. This architecture is one of the first CNNs that are based on deeper configuration [60]. AlexNet won the ImageNet competition in 2012. The success of such a deep architecture has introduced a trend for designing even deeper CNNs that can be noticed today [61]. AlexNet is based on a configuration of nine layers, where the first five layers are convolutional and pooling layers, and the last four are fully connected layers [62]. The detailed description of AlexNet architecture in provided in Table 3.

Table 3. Description of AlexNet architecture (C—convolutional layer, P—Max pooling, FC—fully connected).

Layer	Type	Feature Map	Size	Kernel Size	Stride	Activation Function
Input	Image	1	$227 \times 227 \times 1$	-	-	-
1	C	96	$55 \times 55 \times 96$	11×11	4	ReLU
	P	96	$27 \times 27 \times 96$	3×3	2	-
2	C	256	$27 \times 27 \times 256$	5×5	1	ReLU
	P	256	$13 \times 13 \times 256$	3×3	2	-
3	C	384	$13 \times 13 \times 384$	3×3	1	ReLU
4	C	384	$13 \times 13 \times 384$	3×3	1	ReLU
5	C	256	$13 \times 13 \times 256$	3×3	1	ReLU
	P	256	$6 \times 6 \times 256$	3×3	2	-
6	FC	-	9216	-	-	ReLU
7	FC	-	4096	-	-	ReLU
8	FC	-	4096	-	-	ReLU
Output	FC	-	4	-	-	Softmax

3.2. VGG-16

The described trend of deeper CNN configuration resulted in improvements of the original AlexNet architecture. One of such architectures is VGG-16, presented in the following year. VGG-16 represents a deeper version of AlexNet, where the nine-layer configuration is replaced with a 16-layer configuration, from which the name is derived [63]. A main advantage of VGG-16 is the introduction of smaller kernels in convolutional layers, in comparison with AlexNet [64]. The detailed description of VGG-16 layers is provided in Table 4.

Table 4. Description of VGG 16 architecture (C—convolutional layer, P—Max pooling, FC—fully connected).

Layer	Type	Feature Map	Size	Kernel Size	Stride	Activation Function
Input	Image	1	224 × 224 × 1	-	-	-
1	2 × C	96	224 × 224 × 64	3 × 3	1	ReLU
	P	64	112 × 112 × 64	3 × 3	2	-
3	2 × C	128	112 × 112 × 128	3 × 3	1	ReLU
	P	256	56 × 56 × 128	3 × 3	2	-
5	2 × C	256	56 × 56 × 256	3 × 3	1	ReLU
	P	384	28 × 28 × 256	3 × 3	2	ReLU
7	3 × C	512	28 × 28 × 512	3 × 3	1	ReLU
	P	256	14 × 14 × 512	3 × 3	2	-
10	3 × C	512	14 × 14 × 512	3 × 3	1	ReLU
	P	512	7 × 7 × 512	3 × 3	2	-
13	FC	-	25,088	-	-	ReLU
14	FC	-	4096	-	-	ReLU
15	FC	-	4096	-	-	ReLU
Output	FC	-	4	-	-	Softmax

3.3. ResNet

According to the presented networks, the trend of designing deeper networks can be noticed [65]. This approach can be utilized to a certain level, due to the vanishing gradient problem [66]. It can be noticed that deeper configurations will have no significant improvements in terms of classification performances. Furthermore, in some cases, deeper CNNs can show lower classification performances than CNNs designed with a smaller number of layers. For these reasons, an approach based on residual blocks is proposed. The residual block represents a variation of a CNN layer, where a layer is bypassed with an identity connection [67]. The block scheme of such an approach is presented in Figure 5.

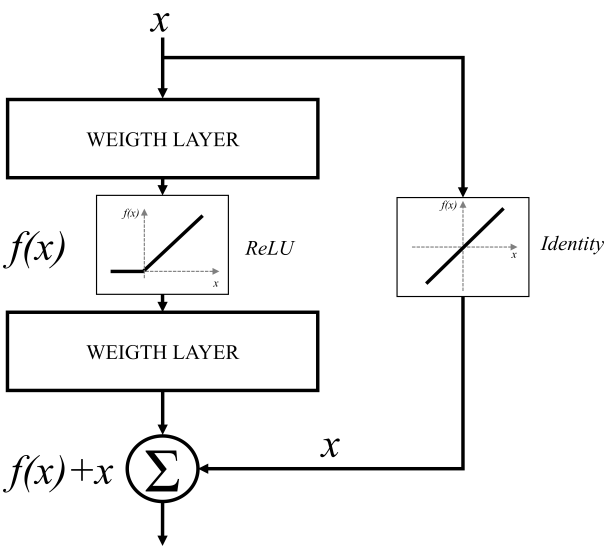


Figure 5. Schematic overview of a residual block.

By using the presented residual approach, significantly deeper networks could be used without the vanishing gradient problem. This characteristic is a consequence of identity bypass utilization because identity layers do not influence the CNN training procedure [68]. For these reasons, deeper CNNs designed with a residual block will not produce the higher error in comparison with shallower architectures. In other words, by stacking residual layers, significantly deeper architectures could be designed. For the case of this research, three different architectures based on the residual block will be used, and these are: ResNet50 [69], ResNet101 [70], and ResNet152 [71]. The aforementioned architectures are pre-defined ResNet architectures that are mainly used for image recognition and computer vision problems which require deeper CNN configurations.

4. Research Methodology

As presented in the previous sections, this research is based on a comparison of multiple methods of image recognition that will be used in order to estimate the severity of COVID-19 symptoms according to X-ray images of patients' lungs. All methods have been compared and evaluated from a standpoint of classification performances. In this case, $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$ are used.

4.1. Description of $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$

Image classifiers are evaluated using standard classification measures, such as the Area under the ROC curve (AUC). Such an approach is based on construction of the ROC curve by using a false-positive rate (FPR) and true-positive rate (TPR). TPR can be described as a ratio between the number of correct classifications in one class (A_C) and the sum of total members of that class. Such a number includes the number of correct classifications and the number of incorrect classifications (A_I). The aforementioned ratio can be defined as:

$$TPR = \frac{A_C}{A_C + A_I}. \tag{15}$$

On the other hand, FPR can be defined as a ratio of the number of incorrect classifications in the first class (B_I) and the total number of members of the second class ($B_I + B_C$). The aforementioned ratio can be written as:

$$FPR = \frac{B_I}{B_I + B_C}. \tag{16}$$

By using TPR and FPR, the ROC curve can be constructed and the AUC value can be determined. The challenge, in this case, lies in the fact that this measure is designed to evaluate the binary classifier. In the case of this research, the classification is performed in four classes. For this reason, a standard ROC-AUC procedure must be adapted to evaluate multi-class classification performances [49]. This approach is achieved by using $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$ measures.

4.1.1. $\overline{AUC_{micro}}$

The definition of $\overline{AUC_{micro}}$ is based on the calculation of TPR_{micro} and FPR_{micro}. TPR_{micro} can be calculated as a ratio between the number of correct classifications and the total number of samples. This relation can be written as:

$$\overline{TPR_{micro}} = \frac{A_C + B_C + C_C + D_C}{N}, \tag{17}$$

where A_C represents the number of correct classifications in the class A, B_C the number of correct classifications in the class B, C_C the number of correct classifications in the class C, and D_C the number of correct classifications in the class D, where N represents the total number of samples. Following a similar methodology, FPR_{micro} can be calculated as a

ratio between the total number of incorrect classifications and the total number of samples. According to the above-stated notation, this ratio can be written as:

$$FPR = \frac{N - (A_C + B_C + C_C + D_C)}{N}. \tag{18}$$

When the described $\overline{TPR_{micro}}$ and $\overline{FPT_{micro}}$ are used for ROC curve construction, the area underneath is called AUC_{micro}. This area represents a discrete micro-average value for the evaluation of multi-class classifier performances.

4.1.2. $\overline{AUC_{macro}}$

Similar to $\overline{AUC_{micro}}$, $\overline{AUC_{macro}}$ can be used for performance evaluation of a multi-class classifier. In this case, average TPR is calculated as an average of TPR values that represent individual classes. For example, the TPR value for the class A can be calculated as a ratio between the number of correct classifications in the class A (A_C) and the total number of class A members (N_A). Such a ratio can be written as:

$$TPR_A = \frac{A_C}{N_A}. \tag{19}$$

When the presented formalism is applied to all classes, $\overline{TPR_{macro}}$ can be calculated as follows:

$$TPR_{macro} = \frac{1}{M} \sum_{n=1}^{M} TRP_n, \tag{20}$$

where M represents the total number of classes. Following the presented procedure, $\overline{FPR_{macro}}$ can be calculated as an average of individual FPR values:

$$\overline{FPR_{macro}} = \frac{1}{M} \sum_{n=1}^{M} FRP_n, \tag{21}$$

where the individual value can be calculated as a ratio between the number of incorrectly classified images as members of a particular class and the total number of images that are members of the same class. By using these measures, $\overline{AUC_{macro}}$ can be calculated.

4.2. Overfitting Issue

Due to the large CNN models used in this research, it is necessary to include steps to overcome overfitting. Over-fitted CNN shows high classification performances on the training dataset, while the performances on the testing dataset are quite poor. In order to prevent overfitting, some steps must be taken. According to [58], there are several mechanisms used to overcome overfitting in image classifiers. The mechanisms used in this research are:

- Image augmentation; and
- Early stopping.

Image augmentation, as one of the key techniques for handling the overfitting issue, was addressed earlier in the article. In order to perform early stopping, an analysis regarding the change of $\overline{AUC_{micro}}$ and $\overline{AUC_{micro}}$ over the number of epochs was performed. Data obtained with this analysis will be used to determine the optimal number of training epochs for each CNN architecture. By using this approach, selected networks will be trained for the number of epochs which will allow for full training, while avoiding overfitting.

4.3. Freezing Layers

In order to increase classification performances of proposed networks, an approach of layers freezing during training procedure will be used. Such an approach will be performed on the CNN configurations that have already achieved the highest performances. The

procedure of freezing layers will be performed in an iterative manner from the bottom of the network towards the higher layers until the maximal classification performances are achieved. Such a procedure is selected in order to fine-tune only specific layers of a CNN architecture, pre-trained with ImageNet, while other layers remain frozen during a training procedure. By using such an approach, issues regarding unscientific datasets are overcome to some extent. An example of a freezing layers methodology is presented on a ResNet architecture in Figure 6, where the first, second, third, and half of the fourth block are frozen during training, while other layers remain unfrozen.

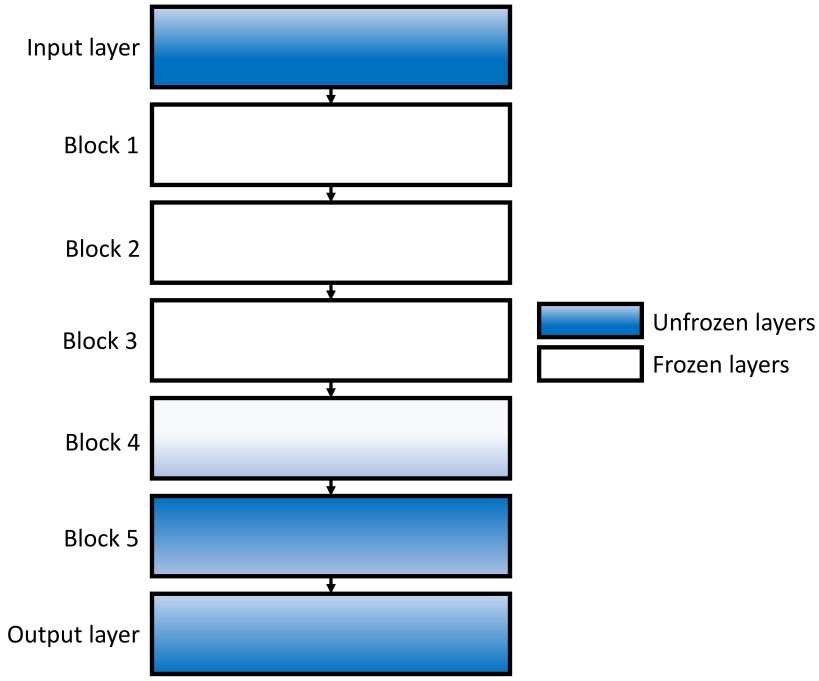


Figure 6. A schematic representation of freezing methodology on ResNet architecture.

4.4. Results Representation

In order to define the network that achieves the best classification performances, maximal $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$ achieved with all networks will be compared. As a first step, the influence of the number of epochs and the batch size on maximal $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$ will be examined. Furthermore, the configuration that produces the highest result will be presented for all CNNs. As a final step, all maximal $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$ values achieved with each CNN will be compared in order to determine the architecture with the highest classification performances. A schematic representation of the research methodology is presented in Figure 7.

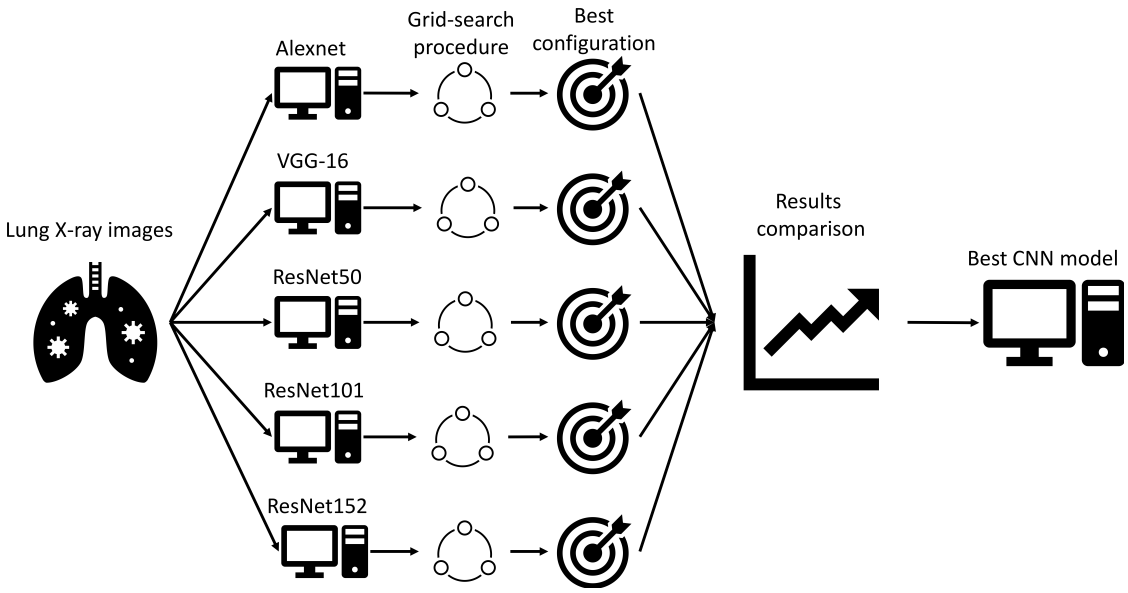


Figure 7. A schematic representation of the presented research methodology.

5. Results and Discussion

In this section, an overview of results achieved with each of the proposed CNN architectures will be presented. For each aforementioned architecture, diagrams that describe the change of maximal $\overline{AUC_{micro}}$, $\overline{AUC_{macro}}$ value in dependence of number of epochs and batch size will be provided. At the and of the section, a comparison of the achieved results will be presented and discussed.

5.1. Results Achieved with AlexNet

As the first of the results achieved with AlexNet architecture, the change of $\overline{AUC_{macro}}$ over the number of training epochs is presented in Figure 8. When the results are compared, it can be noticed that $\overline{AUC_{macro}}$ achieved its maximum at 50 and 75 training epochs, regardless of the solver utilized. Furthermore, it can be noticed that in the higher number of epochs, a significant fall of $\overline{AUC_{macro}}$ value occurs in the case of all solvers. Such a fall in classification performances could be recognized as a consequence of overfitting.

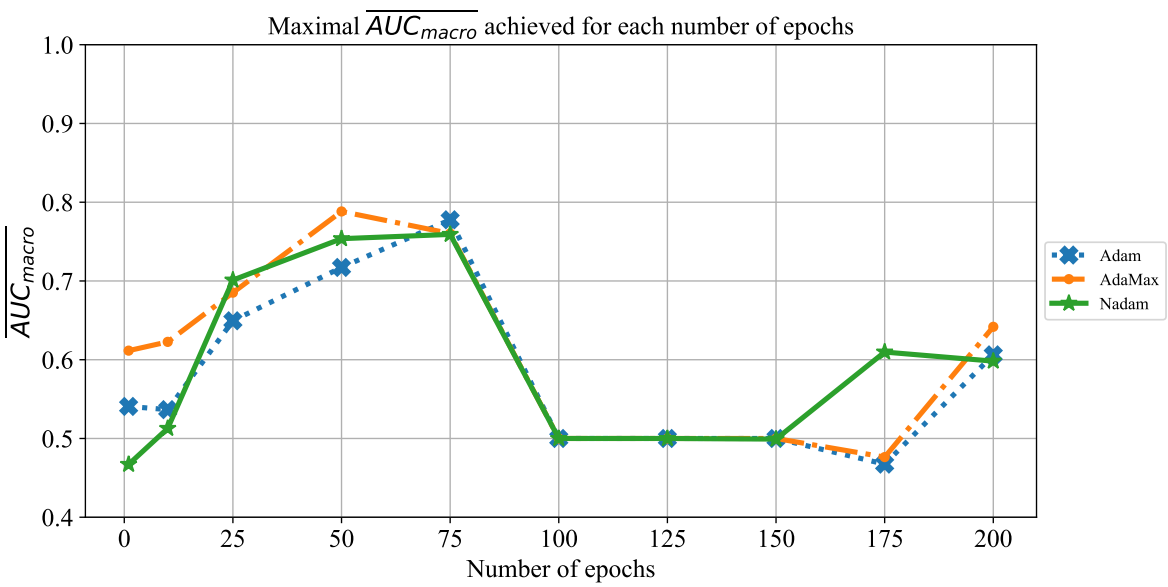


Figure 8. The change of maximal $\overline{AUC_{macro}}$ in dependence of the number of epochs for AlexNet.

The change of $\overline{AUC_{micro}}$ is presented in Figure 9, where a similar trend as in the case of $\overline{AUC_{micro}}$ can be noticed. In this case, maximal performances are also achieved with 50 and 75 consecutive training epochs. Furthermore, it can be noticed that $\overline{AUC_{micro}}$ values are, at the same point, slightly higher than $\overline{AUC_{macro}}$. The trend of overfitting on a larger number of epochs is also noticeable.

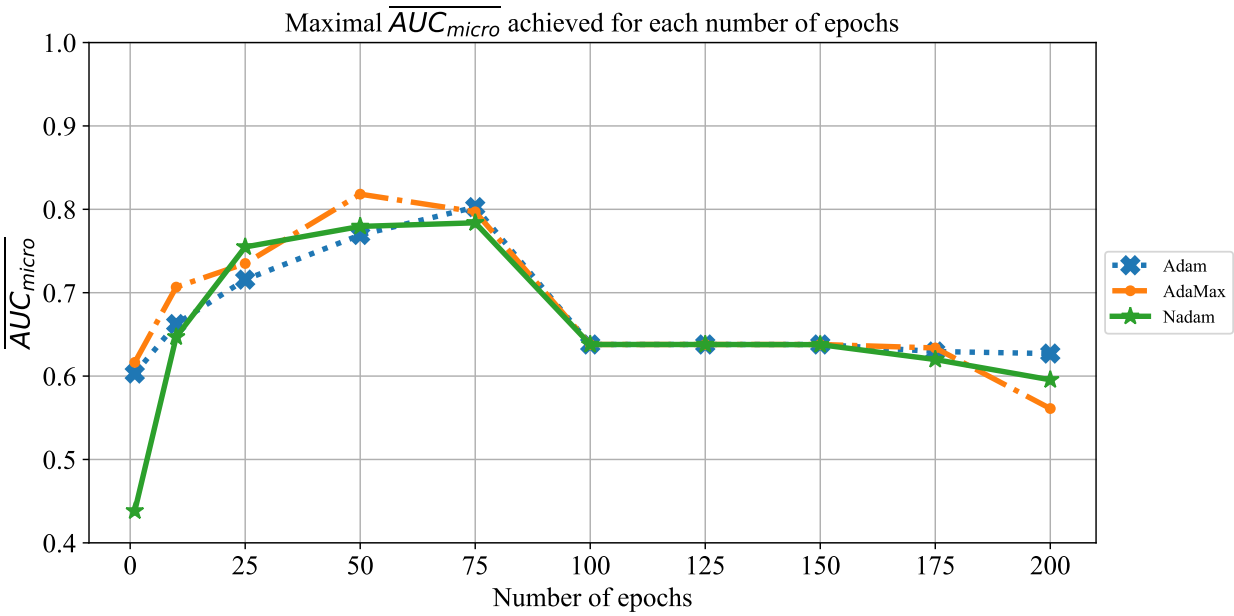


Figure 9. The change of maximal $\overline{AUC_{micro}}$ in dependence of the number of epochs for AlexNet.

When the influence of batch size on $\overline{AUC_{macro}}$ is observed, it can be noticed that there is no configuration that achieves a $\overline{AUC_{macro}}$ value higher than 0.8. This property is in correlation with the case described with Figure 8. It is interesting to notice a significant fall of $\overline{AUC_{macro}}$ values in the case when batches of size 16 are utilized. For this case, $\overline{AUC_{macro}}$ is set around a value of 0.7. This characteristic can be noticed for all three solvers utilized, as presented in Figure 10. Presented results are in correlation with previous knowledge regarding a regularizing effect of smaller batch sizes [72]. Such an approach has enabled overcoming of the overfitting issue.

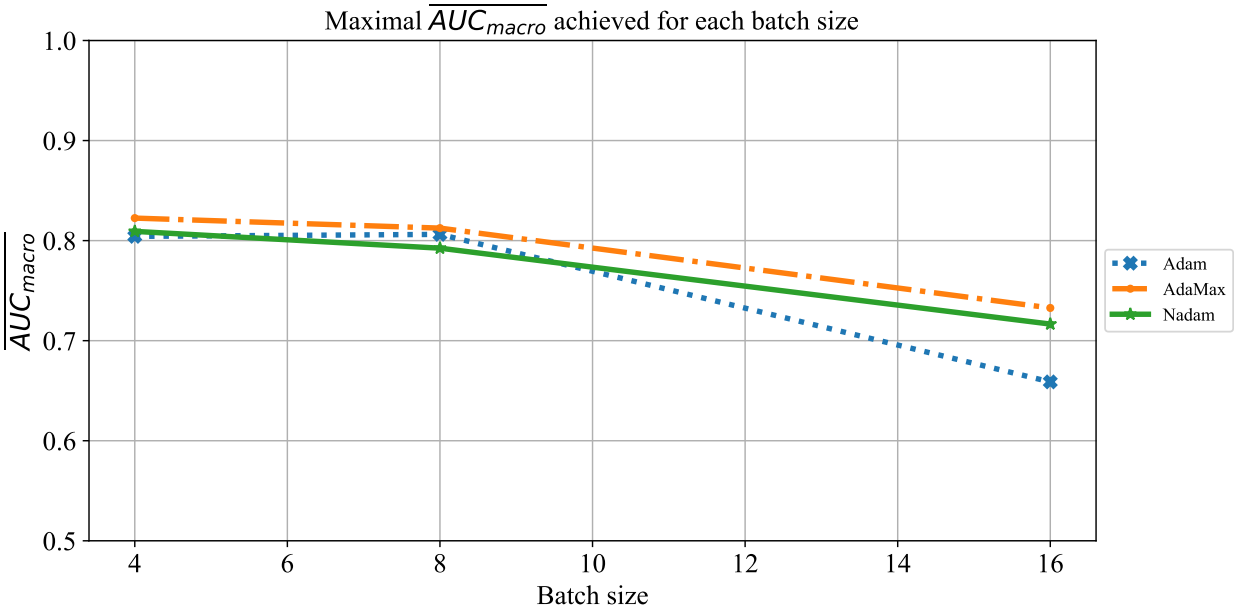


Figure 10. The change of maximal $\overline{AUC_{macro}}$ in dependence of batch size for AlexNet.

As the final evaluation of AlexNet's classification performances, the influence of batch size on $\overline{AUC_{micro}}$ will be observed. Similar to the case presented in Figure 9, an $\overline{AUC_{micro}}$ slightly higher than 0.8 was achieved if batches of four and eight were used. In the case of a batch size of 16, significantly lower $\overline{AUC_{micro}}$ around 0.7 was achieved. The described property can be noticed regardless of the solver utilized, as presented in Figure 11. These results are in correlation with results presented in the case of $\overline{AUC_{macro}}$.

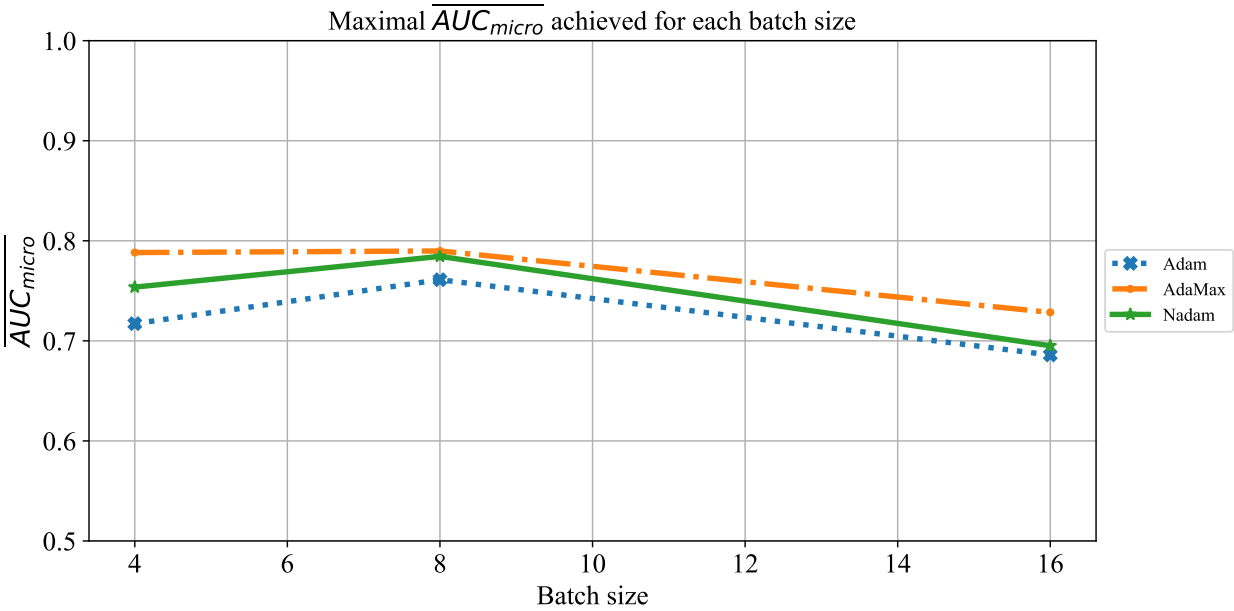


Figure 11. The change of maximal $\overline{AUC_{micro}}$ in dependence of batch size for AlexNet.

5.2. Results Achieved with VGG-16

The results, similar to the results achieved with AlexNet, are achieved with VGG-16, as presented in Figure 12. Maximal $\overline{AUC_{macro}}$ values are achieved when the network is trained for 50 and 75 epochs. On the other hand, a significant decrease of $\overline{AUC_{macro}}$ can be noticed when the network is trained for a larger number of epochs. These lower results are a consequence of overfitting on a larger number of epochs.

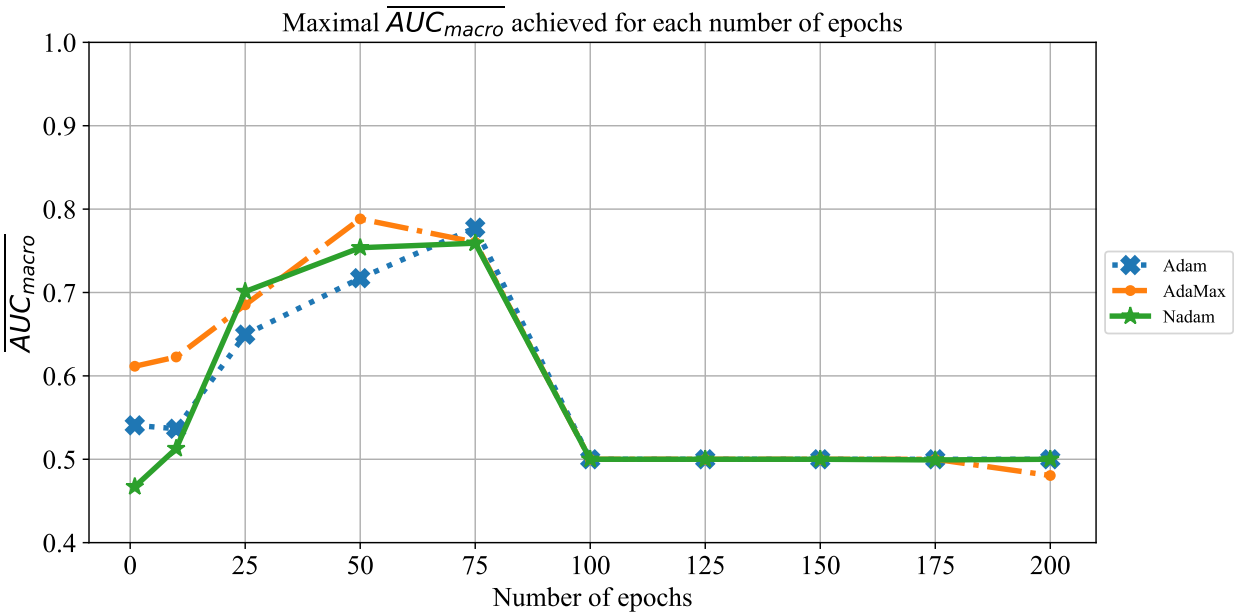


Figure 12. The change of maximal $\overline{AUC_{macro}}$ in dependence of number of epochs for VGG-16.

Similar behavior of $\overline{AUC_{micro}}$ is presented in Figure 13, where maximal performances could be noticed when the network was trained for 50 or 75 consecutive epochs. When the network was trained for a higher number of epochs, a significant fall of $\overline{AUC_{micro}}$ could be noticed. Such a result is a consequence of overfitting on the larger number of epochs.

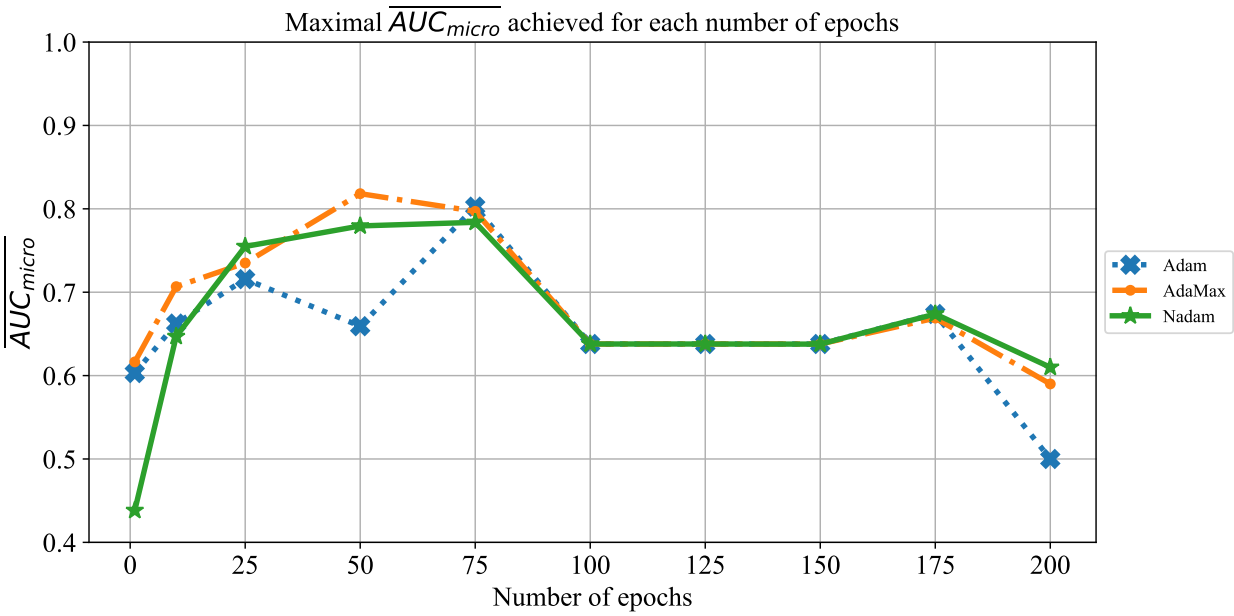


Figure 13. The change of maximal $\overline{AUC_{micro}}$ in dependence of number of epochs for VGG-16.

The influence of batch size on $\overline{AUC_{micro}}$ for the case of VGG-16 is presented in Figure 14. In the case of $\overline{AUC_{macro}}$, the maximal values are achieved when batches of four and eight are utilized, regardless of solver utilized. For the case of a batch size of 16, classification performances, with a value of 0.5, fall into the domain of the coin-flip classification. Such a result can be attributed to the regularization character of smaller batch-sizes and overfitting of larger batch-sizes.

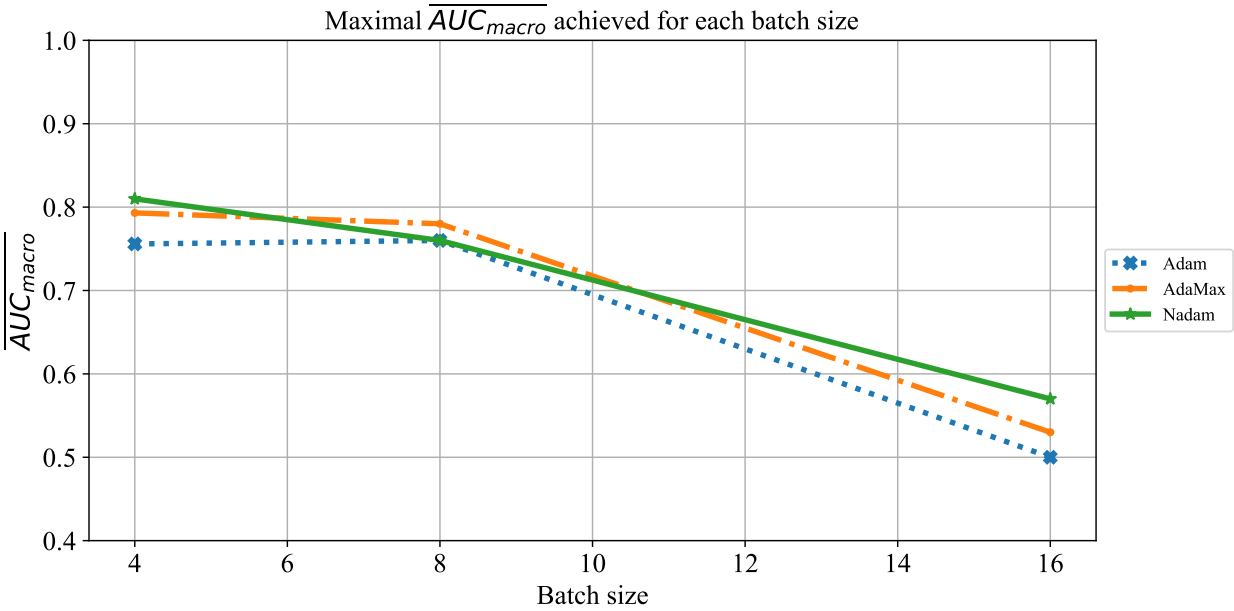


Figure 14. The change of maximal $\overline{AUC_{macro}}$ in dependence of batch size for VGG-16.

When $\overline{AUC_{micro}}$ is measured, it can be noticed that the highest values are achieved when batches of four and eight are used. In the case when larger batches of 16 are used, $\overline{AUC_{micro}}$ value is positioned around a value of 0.7, as presented in Figure 15. In this case, a gap between $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$ can also be noticed.

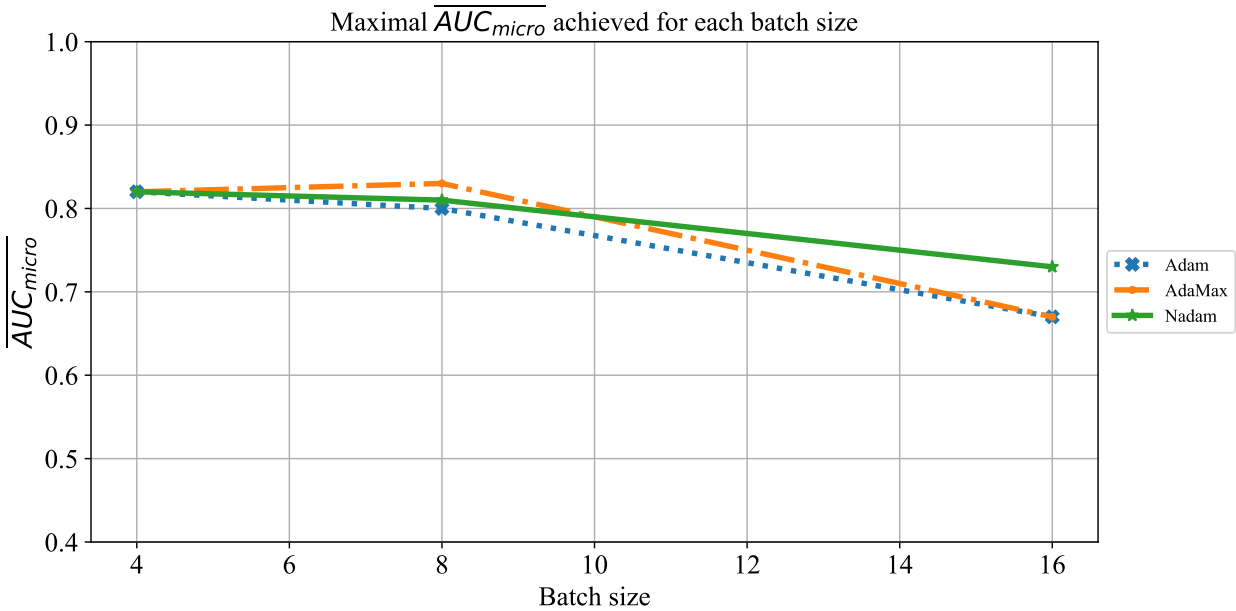


Figure 15. The change of maximal $\overline{AUC_{micro}}$ in dependence of batch size for VGG-16.

5.3. Results Achieved with ResNet Architectures

In the following sub-section, an overview of results achieved by using ResNet architectures will be presented. All results will be presented and described in a similar manner as in the case of AlexNet and VGG-16.

5.3.1. Results Achieved with ResNet50

The change of $\overline{AUC_{macro}}$ over the number of epochs is presented in Figure 16. From the presented results, it can be noticed that the maximal $\overline{AUC_{macro}}$ values are achieved when the network is trained for 100 epochs. This characteristic can be noticed only for the case of the Adam and Adamax solvers, while for the case of the Nadam solver, the maximal $\overline{AUC_{macro}}$ is achieved when the network is trained for 50 consecutive epochs. If the CNN is trained for a larger number of epochs, a significant drop of $\overline{AUC_{macro}}$ can be noticed. Such a result is pointing towards the fact that the overfitting issue occurs if ResNet50 is trained for a larger number of epochs.

Furthermore, when Figures 16 and 17 are observed, a similar trend can be noticed for the case of $\overline{AUC_{micro}}$. A significant drop of $\overline{AUC_{micro}}$ occurs if ResNet50 is trained for a higher number of consecutive epochs, while the $\overline{AUC_{micro}}$ value tops when the network is trained for 75 epochs with an Adam solver or 100 epochs for the AdaMax and Nadam solvers. The lower performances at the higher number of epochs are pointing towards overfitting.

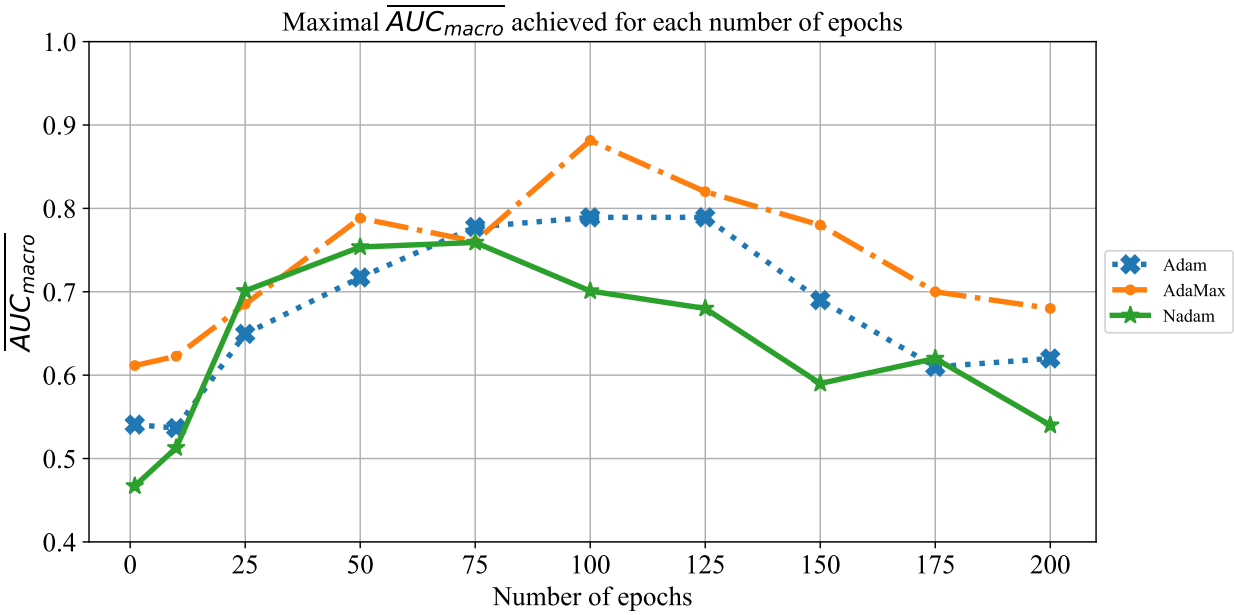


Figure 16. The change of maximal $\overline{AUC_{macro}}$ in dependence of number of epochs for ResNet50.

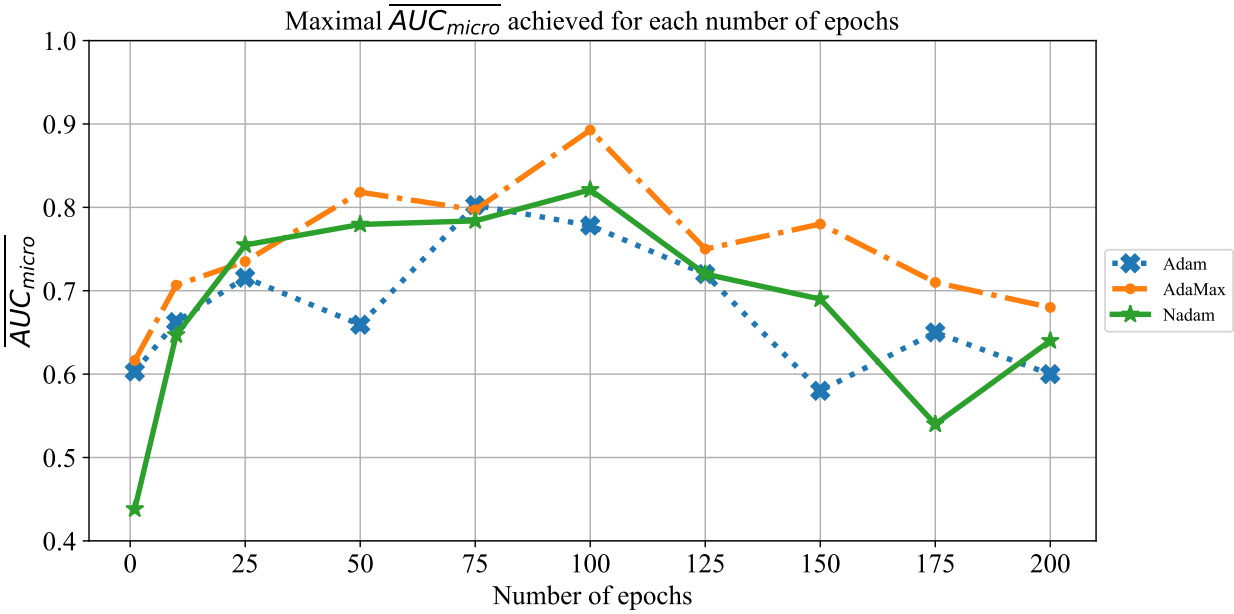


Figure 17. The change of maximal $\overline{AUC_{micro}}$ in dependence of the number of epochs for ResNet50.

When the influence of batch size on $\overline{AUC_{micro}}$ and $\overline{AUC_{macro}}$ is examined, the results presented in Figures 18 and 19 are achieved. It can be noticed that the highest results are achieved when larger batches of 16 are used. In this case, the maximal $\overline{AUC_{macro}}$ will go up to 0.9 only if the AdaMax solver is utilized. In the case of smaller batches, the $\overline{AUC_{macro}}$ values between 0.7 and 0.8 are achieved, regardless of solver utilized.

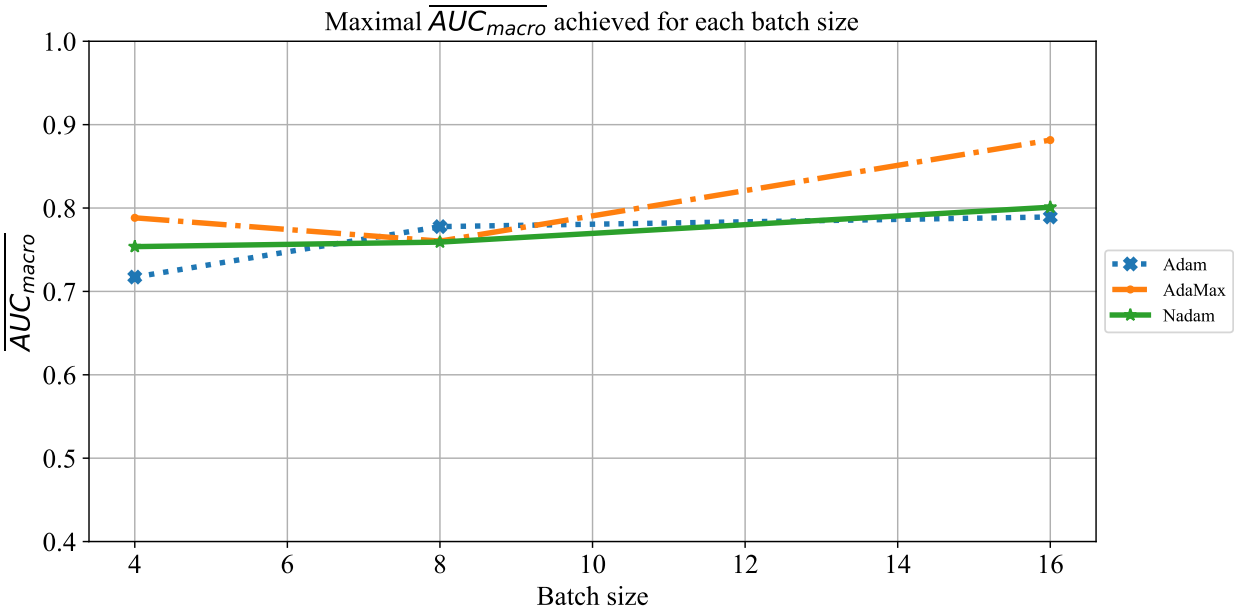


Figure 18. The change of maximal $\overline{AUC_{macro}}$ in dependence of batch size for ResNet50.

Similar conclusions could be drawn when $\overline{AUC_{micro}}$ values are compared. In this case, the only significant difference is a significant underperformance of networks trained with the Adam solver by using smaller batches, as presented in Figure 19.

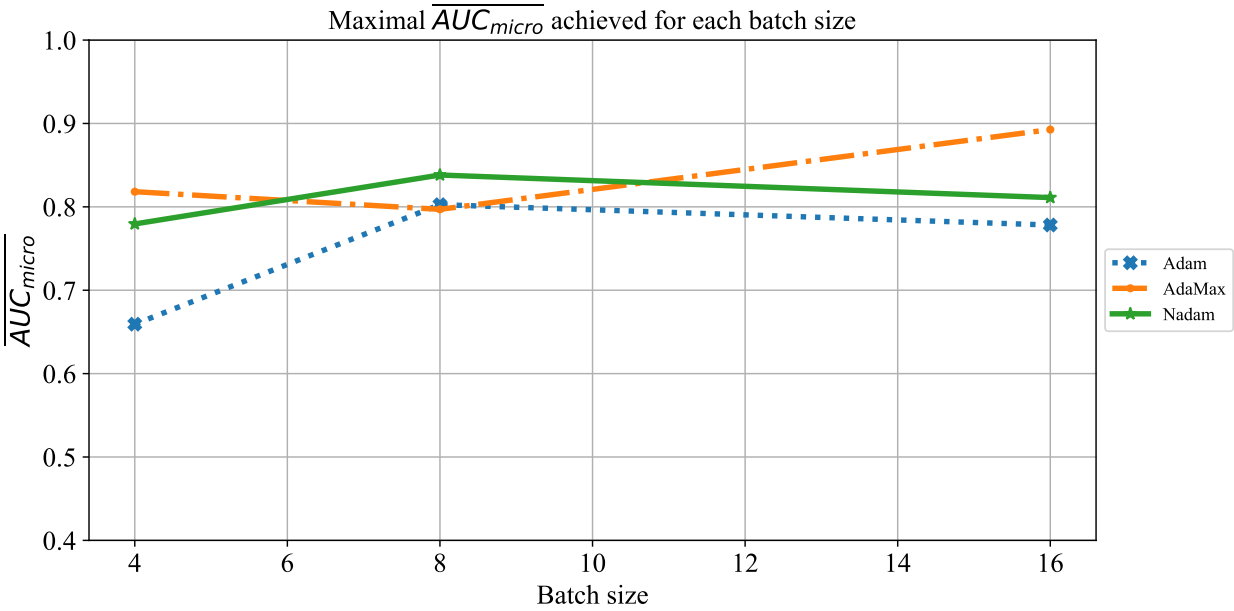


Figure 19. The change of maximal $\overline{AUC_{micro}}$ in dependence of batch size for ResNet50.

5.3.2. Results Achieved with ResNet101

The change of $\overline{AUC_{macro}}$ over the number of epochs achieved with ResNet101 is presented in Figure 20. From the presented results, it can be noticed that the highest performances are achieved when the CNN is trained for 150 epochs. Such a property can be noticed for all solvers, with an exception of Nadam, which has achieved similar results at 50 epochs. Furthermore, the significant drop of $\overline{AUC_{macro}}$ value can be noticed at a higher number of epochs. Such a fall can be attributed to overfitting.

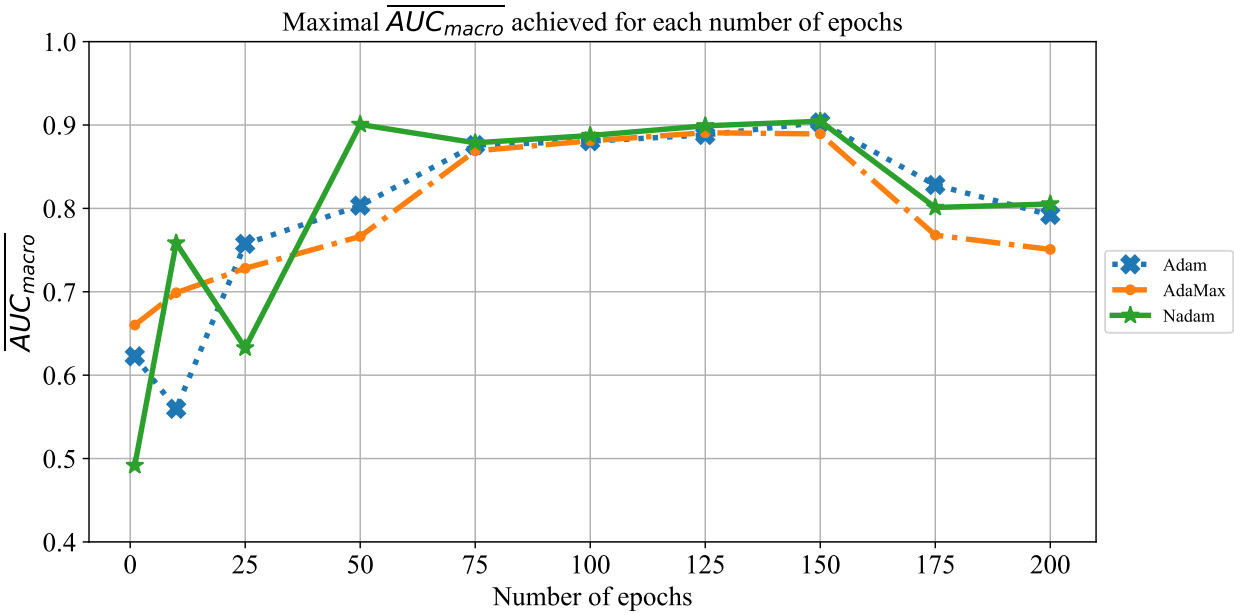


Figure 20. The change of maximal $\overline{AUC_{macro}}$ in dependence of number of epochs for ResNet101.

A similar trend is presented in Figure 21, where the change of $\overline{AUC_{micro}}$ value over the number of epochs is presented. As it is in the case of $\overline{AUC_{macro}}$, the maximal classification performances are achieved when CNNs are trained for 100, 125, and 150 consecutive epochs. Due to overfitting, significantly lower $\overline{AUC_{micro}}$ values are achieved when CNNs are trained for 175 and 200 epochs. The drop of $\overline{AUC_{micro}}$ value is noticeably deeper than in case of $\overline{AUC_{macro}}$.

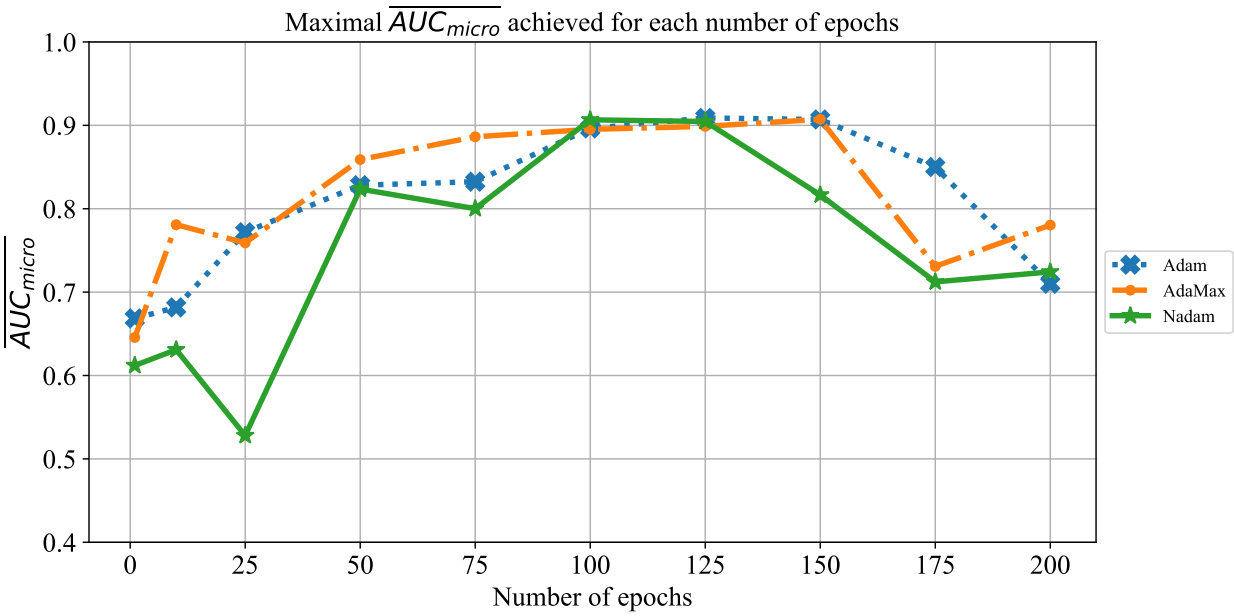


Figure 21. The change of maximal $\overline{AUC_{micro}}$ in dependence of number of epochs for ResNet101.

The influence of batch size on $\overline{AUC_{macro}}$ is presented in Figure 22. When the results are observed, it can be noticed that the highest classification performances are achieved when larger data batches are used during training of ResNet101. These characteristics are noticed only for the case of the AdaMax and Adam solvers. On the other hand, when the

Adam solver is used, no significant difference of $\overline{AUC_{macro}}$ is achieved when batches of 8 and 16 are used during training.

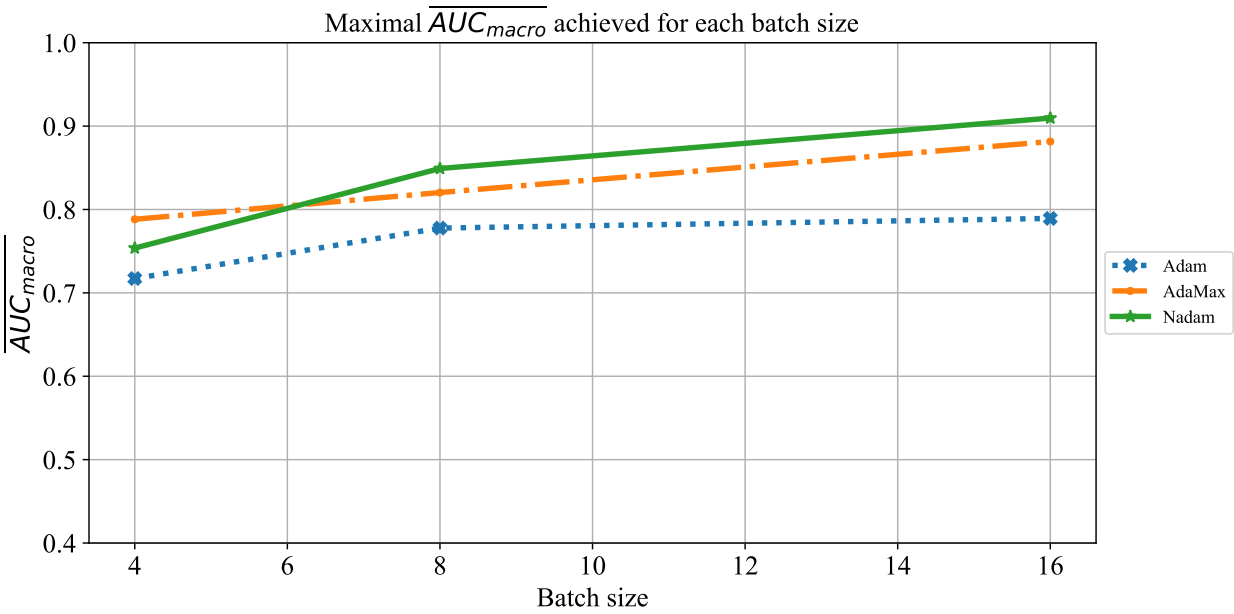


Figure 22. The change of maximal $\overline{AUC_{macro}}$ in dependence of batch size for ResNet101.

A similar conclusion could be reached if classification performances are evaluated by using $\overline{AUC_{micro}}$. It can be noticed that the highest $\overline{AUC_{macro}}$ values will be achieved if the network is trained by using larger batches, as presented in Figure 23.

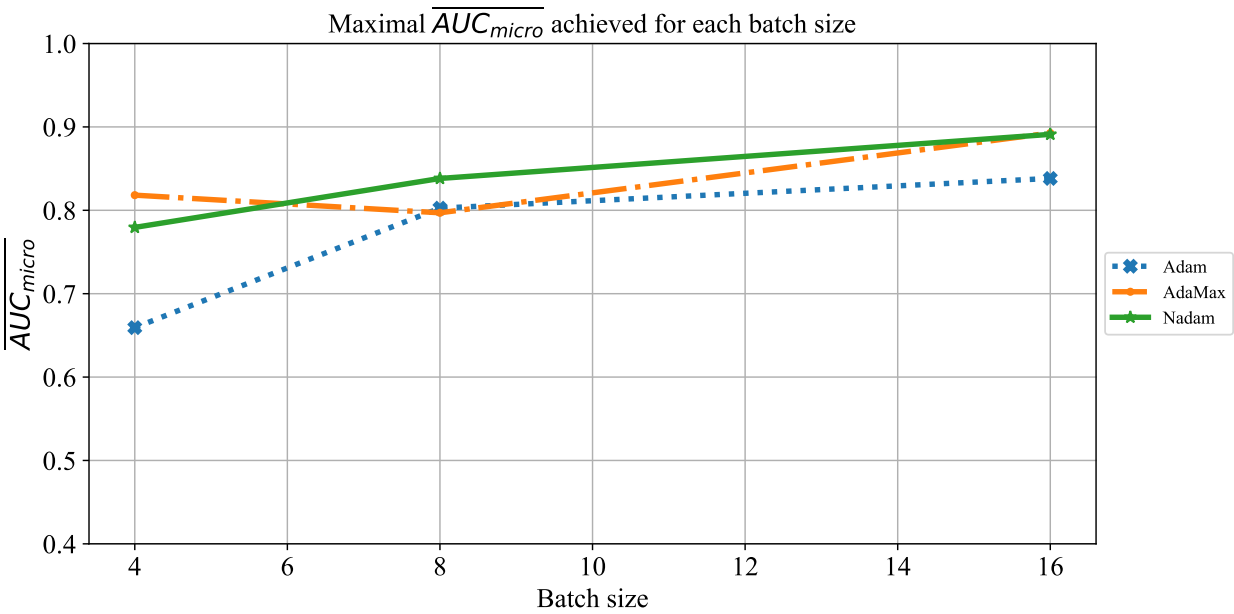


Figure 23. The change of maximal $\overline{AUC_{micro}}$ in dependence of batch size for ResNet101.

5.3.3. Results Achieved with ResNet152

The last CNN used in this research is ResNet152. The change of $\overline{AUC_{macro}}$ over number of epochs is presented in Figure 24. From the presented result, it can be noticed that the highest $\overline{AUC_{macro}}$ value is achieved when ResNet152 is trained for 125 epochs. Such a property can be noticed regardless of the solver utilized. An exception can be noticed in

the case of the AdaMax solver. In this case, $\overline{AUC_{macro}}$ values over 0.9 are also achieved when the network is trained for 75 epochs. Furthermore, an influence of overfitting can be noticed when the network is trained for a larger number of epochs. Due to this property, it is important to train the network for a lower number of consecutive epochs in order to prevent overfitting and, consequently, lower classification performances.

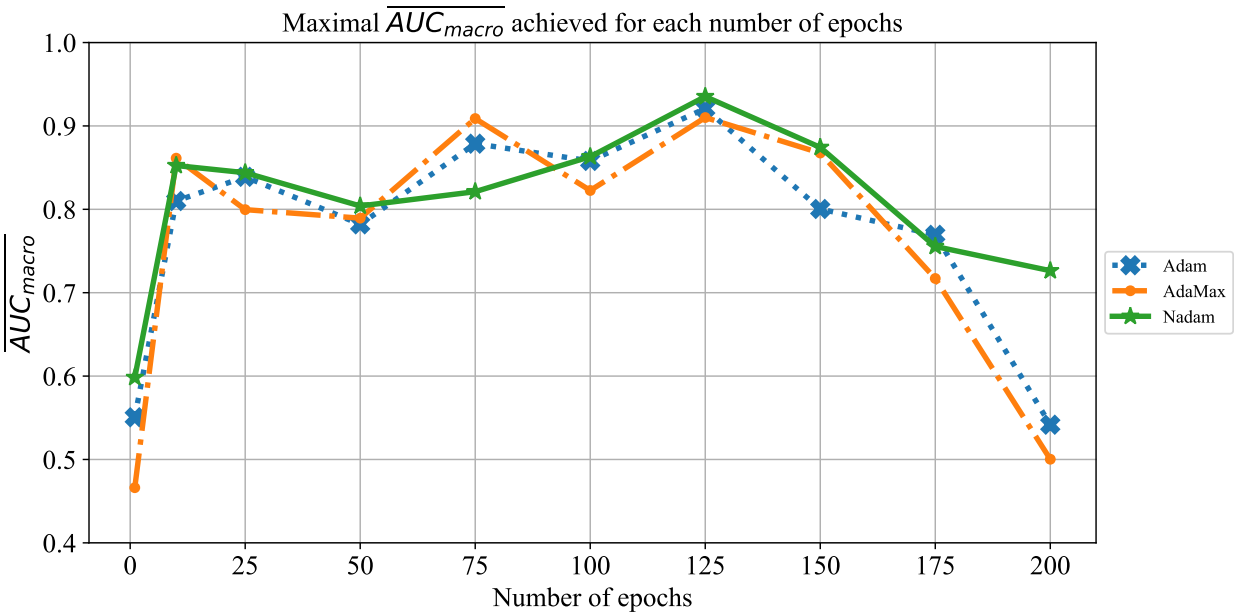


Figure 24. The change of maximal $\overline{AUC_{macro}}$ in dependence of number of epochs for ResNet152.

Similar results are achieved when the change of $\overline{AUC_{micro}}$ over different number of epochs is observed, as presented in Figure 25. In this case, the highest $\overline{AUC_{micro}}$ values are achieved when the network is trained for 100 and 125 consecutive epochs. It is important to notice that a significant fall of $\overline{AUC_{micro}}$ occurs when the CNN is trained for 175 and 200 consecutive epochs. Such a trend can be noticed regardless of solver utilized, and it points toward an occurrence of overfitting. Due to these results, it can be concluded that it is necessary to avoid training for such a large number of epochs in order to prevent overfitting and to achieve higher classification performances.

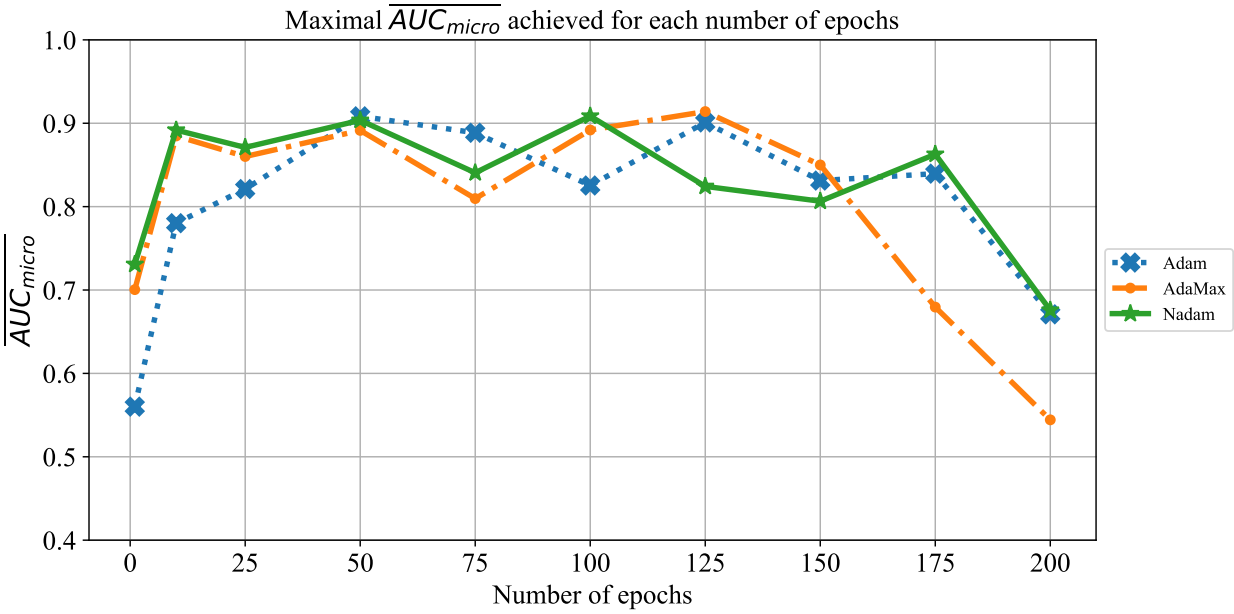


Figure 25. The change of maximal $\overline{AUC_{micro}}$ in dependence of number of epochs for ResNet152.

When the influence of batch size on $\overline{AUC_{macro}}$ is observed, it can be noticed that by using a larger batch size of 16, higher $\overline{AUC_{macro}}$ will be achieved. The significantly lower $\overline{AUC_{macro}}$ values are achieved when ResNet152 is trained by using smaller batches of four. This property can be noticed for the case of all three solvers, as presented in Figure 26.

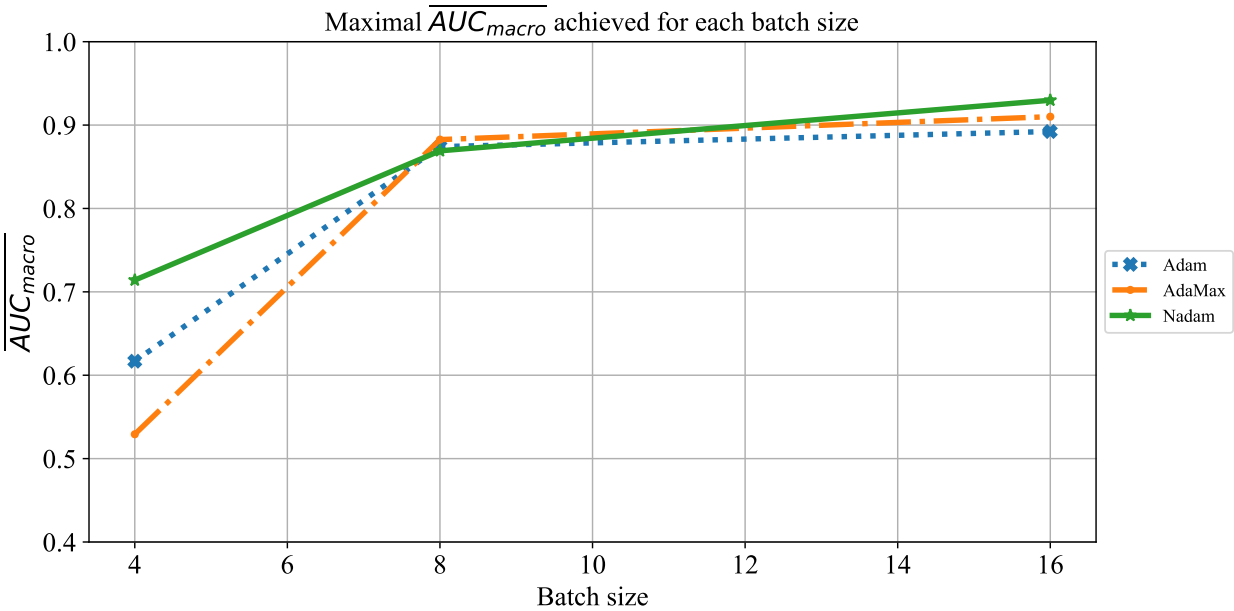


Figure 26. The change of maximal $\overline{AUC_{macro}}$ in dependence of batch size for ResNet152.

Similar results can be noticed when the influence of batch size on $\overline{AUC_{micro}}$ is observed. The only significant difference lies in the fact that for a batch size of four, somewhat higher values are achieved, as presented in Figure 27. Regardless of higher value, $\overline{AUC_{micro}}$, in this case, is still too low to be taken into consideration for practical application.

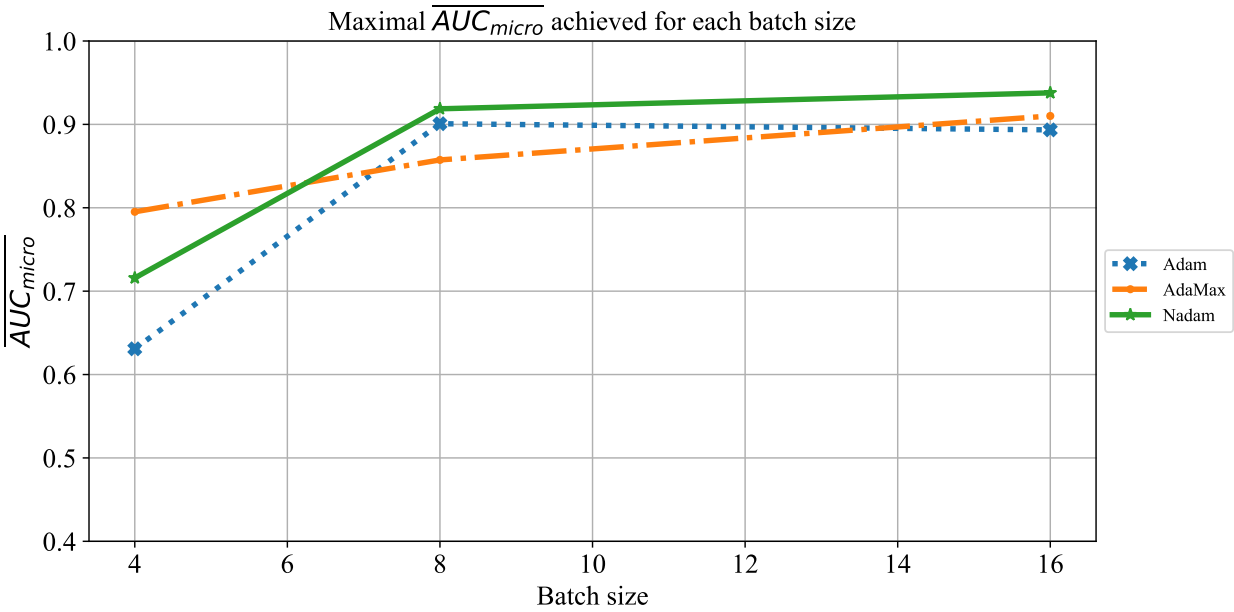


Figure 27. The change of maximal $\overline{AUC_{micro}}$ in dependence of batch size for ResNet152.

5.4. Comparison of Achieved Results

When the result achieved with all CNN architectures is compared, it can be noticed that in the case of AlexNet and VGG-16, the highest $\overline{AUC_{macro}}$ values are achieved if networks are trained by using smaller batches for a lower number of epochs. On the other hand, ResNet architectures show better performances when trained by using larger batches for a higher number of consecutive epochs. Configurations that have achieved the largest $\overline{AUC_{macro}}$ values are presented in Table 5.

Table 5. Overview of configurations that achieved highest $\overline{AUC_{macro}}$ for all CNN architectures.

Network	Number of Epochs	Batch Size	Solver
AlexNet	50	4	AdaMax
VGG-16	50	4	AdaMax
ResNet50	100	16	AdaMax
ResNet101	50	16	Nadam
ResNet152	125	16	Nadam

Similarly to the above-presented architectures, the highest $\overline{AUC_{micro}}$ values are achieved when AlexNet and VGG-16 are trained by using smaller batches for a lower number of consecutive epochs. Furthermore, for the case of ResNet architectures, the highest $\overline{AUC_{micro}}$ values are achieved when CNNs are trained by using larger batches for a larger number of epochs. The described configurations are presented in Table 6.

Table 6. Overview of configurations that achieved highest $\overline{AUC_{micro}}$ for all CNN architectures.

Network	Number of Epochs	Batch Size	Solver
AlexNet	50	4	AdaMax
VGG-16	50	4	AdaMax
ResNet50	100	16	AdaMax
ResNet101	100	16	Nadam
ResNet152	100	16	Nadam

Finally, when the highest $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ achieved with each CNN architecture are compared, it can be noticed that ResNet architectures are achieving dominantly higher classification performances. On the other hand, it can be noticed that by using deeper ResNet architectures, a rising trend of $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ is present, as presented in Figure 28. The achieved results are pointing to the conclusion that by using ResNet152 architecture, the highest $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ values of 0.93 and 0.94 are achieved. Given the results achieved, the possibility of using CNN for automatic classification of patients with COVID-19 with respect to lung status should be considered.

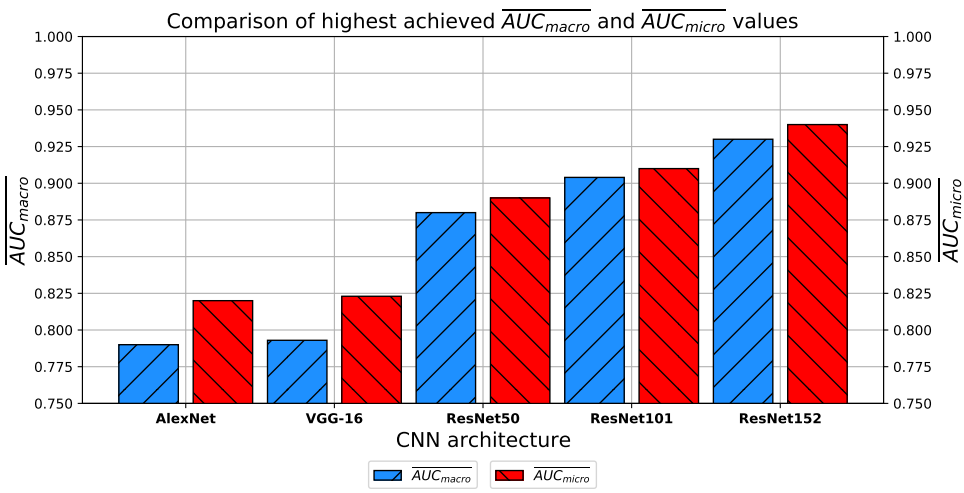


Figure 28. Comparison of highest $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ achieved with every CNN architecture.

Furthermore, when layer freezing is considered, it can be noticed that by freezing higher layers of CNNs during the training procedure, higher classification performances are achieved. The distribution of frozen and unfrozen layers is presented in Table 7 for each CNN architecture utilized.

Table 7. Representation of distribution of frozen and unfrozen layers with classification performances.

Network	Frozen Layers	Unfrozen Layers
AlexNet	1–5	6–9
VGG-16	1–12	13–16
ResNet50	1–42	43–50
ResNet101	1–92	93–101
ResNet152	1–139	140–152

When comparing the achieved classification performances with classification performances in the case when all layers are fine-tuned, it can be noticed that in the case of freezing layers, slightly higher performances are achieved with each of the proposed CNN architectures, as presented in Figure 29.

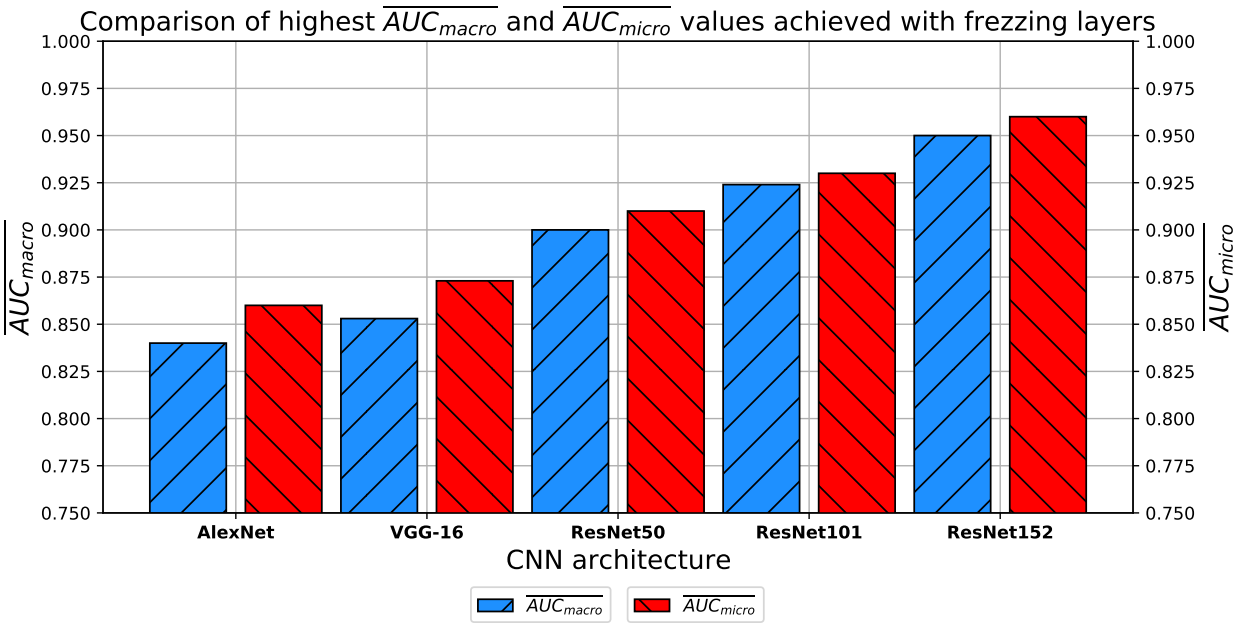


Figure 29. Comparison of highest $\overline{AUC_{macro}}$ and $\overline{AUC_{micro}}$ achieved with every CNN architecture and freezing layers.

Furthermore, it can be noticed how the order of the architectures from the one with the best classification performance to the one with the worst is the same as in the previous case.

When the results achieved with transfer learning are compared to the results achieved on similar problems, it can be noticed that higher classification performances are achieved if transfer learning is utilized. Such a correlation can be noticed when the achieved results are compared with the results of research dealing with both COVID-19 [42,43] and other respiratory issues [22,23]. These results are pointing towards the utilization of transfer learning in order to increase the accuracy of evaluation of the clinical picture of COVID-19 patients from X-ray lung images.

These results show that ResNet152, in combination with transfer learning, is the network that achieves the best results in the case of evaluation of the clinical picture of COVID-19 patients using X-ray lung images.

6. Conclusions

The results achieved with this research are pointing towards the conclusion that CNN-based architectures could be used in estimation of the clinical picture of a COVID-19 patient according to the X-ray lung images. It is important to notice that deep CNN architectures have the tendency to overfit when they are trained with a higher number of consecutive epochs. Due to this property, it is concluded that steps such as early stopping and image augmentation must be used in order to prevent overfitting. According to the presented results and stated research hypothesis, the following conclusions could be drawn:

- It is possible to utilize CNN for automatic classification of COVID-19 patients according to X-ray lung images;
- The best results are achieved if ResNet152 architecture is utilized;
- The best results are achieved if the aforementioned architecture is trained by using larger batches of data for an intermediate number of consecutive epochs by using Nadam solver; and
- It can be noticed that by utilization of transfer learning and freezing layers, higher classification performances are achieved.

Due to the presented results and conclusions, a possibility for utilization of such an algorithm in battle against COVID-19 and its application in clinical practice should be taken into account. The main limitation of this research was the small amount of X-ray images, which could be overcome, to some extent, by augmentation techniques, and another limitation was class imbalance. Regardless of the presented limitations, the presented approach has shown promising results which point to further research on a larger and more balanced data set.

Author Contributions: Conceptualization, I.L., S.B.Š., N.A., A.B., T.Š., A.P., M.A., T.Ć., N.F. and Z.C.; methodology, I.L. and S.B.Š.; software, I.L. and S.B.Š.; validation, N.A., A.B., T.Š. and Z.C.; formal analysis, A.P., M.A., T.Ć., N.F. and Z.C.; investigation, I.L, S.B.Š., A.B. and T.Š.; resources, T.Ć. and Z.C.; data curation, A.B., T.Š., M.A. and N.F.; writing–original draft preparation, I.L. and S.B.Š.; writing–review and editing, N.A., A.B., T.Š., A.P., M.A., T.Ć., N.F. and Z.C.; visualization, I.L.; supervision, N.F. and Z.C.; project administration, N.F. and Z.C.; funding acquisition, T.Ć., N.F. and Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This research has been (partly) supported by the CEEPUS network CIII-HR-0108, European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS), project CEKOM under the grant KK.01.2.2.03.0004, CEI project "COVIDAi" (305.6019-20) and University of Rijeka scientific grant uniri-tehnic-18-275-1447.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Milenković, D.A.; Dimić, D.S.; Avdović, E.H.; Marković, Z.S. Several coumarin derivatives and their Pd (II) complexes as potential inhibitors of the main protease of SARS-CoV-2, an in silico approach. *RSC Adv.* **2020**, *10*, 35099–35108. [CrossRef]
2. Spiegelhalter, D. Use of "normal" risk to improve understanding of dangers of covid-19. *BMJ* **2020**, *370*, m3259. [CrossRef] [PubMed]
3. Li, X.; Xu, S.; Yu, M.; Wang, K.; Tao, Y.; Zhou, Y.; Shi, J.; Zhou, M.; Wu, B.; Yang, Z.; et al. Risk factors for severity and mortality in adult COVID-19 inpatients in Wuhan. *J. Allergy Clin. Immunol.* **2020**, *146*, 110–118. [CrossRef] [PubMed]
4. Brindle, M.E.; Gawande, A. Managing COVID-19 in surgical systems. *Ann. Surg.* **2020**. [CrossRef]
5. Weissman, G.E.; Crane-Droesch, A.; Chivers, C.; Luong, T.; Hanish, A.; Levy, M.Z.; Lubken, J.; Becker, M.; Draugelis, M.E.; Anesi, G.L.; et al. Locally informed simulation to predict hospital capacity needs during the COVID-19 pandemic. *Ann. Intern. Med.* **2020**. [CrossRef]
6. Yıldırım, M.; Güler, A. COVID-19 severity, self-efficacy, knowledge, preventive behaviors, and mental health in Turkey. *Death Stud.* **2020**, 1–8. [CrossRef]
7. Vaishya, R.; Javaid, M.; Khan, I.H.; Haleem, A. Artificial Intelligence (AI) applications for COVID-19 pandemic. *Diabetes Metab. Syndr. Clin. Res. Rev.* **2020**, *14*, 337–339. [CrossRef]

8. Zhang, K.; Liu, X.; Shen, J.; Li, Z.; Sang, Y.; Wu, X.; Zha, Y.; Liang, W.; Wang, C.; Wang, K.; et al. Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography. *Cell* **2020**, *181*, 1423–1433.e11. [CrossRef]
9. Al-Turjman, F.; Zahmatkesh, H.; Mostarda, L. Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning. *IEEE Access* **2019**, *7*, 115749–115759. [CrossRef]
10. Guo, C.; Zhang, J.; Liu, Y.; Xie, Y.; Han, Z.; Yu, J. Recursion Enhanced Random Forest With an Improved Linear Model (RERF-ILM) for Heart Disease Detection on the Internet of Medical Things Platform. *IEEE Access* **2020**, *8*, 59247–59256. [CrossRef]
11. Kayser, K.; GĂśrtler, J.; Bogovac, M.; Bogovac, A.; Goldmann, T.; Vollmer, E.; Kayser, G. AI (artificial intelligence) in histopathology—From image analysis to automated diagnosis. *Folia Histochem. Cytobiol.* **2009**, *47*, 355–361. [CrossRef] [PubMed]
12. Raghavendra, U.; Acharya, U.R.; Adeli, H. Artificial intelligence techniques for automated diagnosis of neurological disorders. *Eur. Neurol.* **2019**, *82*, 41–64. [CrossRef] [PubMed]
13. Lorencin, I.; Anđelić, N.; Španjol, J.; Car, Z. Using multi-layer perceptron with Laplacian edge detector for bladder cancer diagnosis. *Artif. Intell. Med.* **2020**, *102*, 101746. [CrossRef] [PubMed]
14. Tan, Z.; Simkin, S.; Lai, C.; Dai, S. Deep learning algorithm for automated diagnosis of retinopathy of prematurity plus disease. *Transl. Vis. Sci. Technol.* **2019**, *8*, 23. [CrossRef] [PubMed]
15. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006.
16. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer Science & Business Media: Berlin, Germany, 2009.
17. Luján-García, J.E.; Yáñez-Márquez, C.; Villuendas-Rey, Y.; Camacho-Nieto, O. A Transfer Learning Method for Pneumonia Classification and Visualization. *Appl. Sci.* **2020**, *10*, 2908. [CrossRef]
18. Kieu, P.N.; Tran, H.S.; Le, T.H.; Le, T.; Nguyen, T.T. Applying multi-CNNs model for detecting abnormal problem on chest X-ray images. In Proceedings of the 2018 10th International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, Vietnam, 1–3 November 2018; pp. 300–305.
19. Bullock, J.; Cuesta-Lázaro, C.; Quera-Bofarull, A. XNet: A convolutional neural network (CNN) implementation for medical X-ray image segmentation suitable for small datasets. In *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 10953, p. 109531Z.
20. Takemiya, R.; Kido, S.; Hirano, Y.; Mabu, S. Detection of pulmonary nodules on chest X-ray images using R-CNN. In *International Forum on Medical Imaging in Asia 2019*; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 11050, p. 110500W.
21. Stirenko, S.; Kochura, Y.; Alienin, O.; Rokovyi, O.; Gordienko, Y.; Gang, P.; Zeng, W. Chest X-ray analysis of tuberculosis by deep learning with segmentation and augmentation. In Proceedings of the 2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO), Kiev, Ukraine, 24–26 April 2018; pp. 422–428.
22. Chouhan, V.; Singh, S.K.; Khamparia, A.; Gupta, D.; Tiwari, P.; Moreira, C.; Damaševičius, R.; De Albuquerque, V.H.C. A novel transfer learning based approach for pneumonia detection in chest X-ray images. *Appl. Sci.* **2020**, *10*, 559. [CrossRef]
23. Rahman, T.; Chowdhury, M.E.; Khandakar, A.; Islam, K.R.; Islam, K.F.; Mahbub, Z.B.; Kadir, M.A.; Kashem, S. Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray. *Appl. Sci.* **2020**, *10*, 3233. [CrossRef]
24. Wong, H.Y.F.; Lam, H.Y.S.; Fong, A.H.T.; Leung, S.T.; Chin, T.W.Y.; Lo, C.S.Y.; Lui, M.M.S.; Lee, J.C.Y.; Chiu, K.W.H.; Chung, T.; et al. Frequency and distribution of chest radiographic findings in COVID-19 positive patients. *Radiology* **2020**, *296*, 201160. [CrossRef]
25. Orsi, M.A.; Oliva, G.; Toluian, T.; Pittino, C.V.; Panzeri, M.; Cellina, M. Feasibility, reproducibility, and clinical validity of a quantitative chest X-ray assessment for COVID-19. *Am. J. Trop. Med. Hyg.* **2020**, *103*, 822–827. [CrossRef]
26. Cozzi, D.; Albanesi, M.; Cavigli, E.; Moroni, C.; Bindi, A.; Luvarà, S.; Lucarini, S.; Busoni, S.; Mazzoni, L.N.; Miele, V. Chest X-ray in new Coronavirus Disease 2019 (COVID-19) infection: Findings and correlation with clinical outcome. *Radiol. Med.* **2020**, *125*, 730–737. [CrossRef]
27. Borghesi, A.; Maroldi, R. COVID-19 outbreak in Italy: Experimental chest X-ray scoring system for quantifying and monitoring disease progression. *Radiol. Med.* **2020**, *125*, 509–513. [CrossRef] [PubMed]
28. Imran, A.; Posokhova, I.; Qureshi, H.N.; Masood, U.; Riaz, S.; Ali, K.; John, C.N.; Nabeel, M. AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. *arXiv* **2020**, arXiv:2004.01275.
29. Bragazzi, N.L.; Dai, H.; Damiani, G.; Behzadifar, M.; Martini, M.; Wu, J. How Big Data and Artificial Intelligence Can Help Better Manage the COVID-19 Pandemic. *Int. J. Environ. Res. Public Health* **2020**, *17*, 3176. [CrossRef]
30. Mohamadou, Y.; Halidou, A.; Kapen, P.T. A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. *Appl. Intell.* **2020**, *50*, 3913–3925. [CrossRef]
31. Raza, K. Artificial intelligence against COVID-19: A meta-analysis of current research. In *Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach*; Springer: Berlin, Germany, 2020; pp. 165–176.
32. Adly, A.S.; Adly, A.S.; Adly, M.S. Approaches based on artificial intelligence and the internet of intelligent things to prevent the spread of COVID-19: Scoping review. *J. Med. Internet Res.* **2020**, *22*, e19104. [CrossRef]

33. Zheng, N.; Du, S.; Wang, J.; Zhang, H.; Cui, W.; Kang, Z.; Yang, T.; Lou, B.; Chi, Y.; Long, H.; et al. Predicting covid-19 in china using hybrid AI model. *IEEE Trans. Cybern.* **2020**, *50*, 2891–2904. [CrossRef]
34. Hazarika, B.B.; Gupta, D. Modelling and forecasting of COVID-19 spread using wavelet-coupled random vector functional link networks. *Appl. Soft Comput.* **2020**, *96*, 106626. [CrossRef]
35. Car, Z.; Baressi Šegota, S.; Anđelić, N.; Lorencin, I.; Mrzljak, V. Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron. *Comput. Math. Methods Med.* **2020**, *2020*, 5714714. [CrossRef]
36. Ye, Y.; Hou, S.; Fan, Y.; Qian, Y.; Zhang, Y.; Sun, S.; Peng, Q.; Laparo, K. alpha-Satellite: An AI-driven System and Benchmark Datasets for Hierarchical Community-level Risk Assessment to Help Combat COVID-19. *arXiv* **2020**, arXiv:2003.12232.
37. Štifanić, D.; Musulin, J.; Miočević, A.; Baressi Šegota, S.; Šubić, R.; Car, Z. Impact of COVID-19 on Forecasting Stock Prices: An Integration of Stationary Wavelet Transform and Bidirectional Long Short-Term Memory. *Complexity* **2020**, *2020*, 1846926. [CrossRef]
38. 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. *Sci. Rep.* **2020**, *10*, 1–12. [CrossRef] [PubMed]
39. Narin, A.; Kaya, C.; Pamuk, Z. Automatic detection of coronavirus disease (covid-19) using X-ray images and deep convolutional neural networks. *arXiv* **2020**, arXiv:2003.10849.
40. Ozturk, T.; Talo, M.; Yildirim, E.A.; Baloglu, U.B.; Yildirim, O.; Acharya, U.R. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput. Biol. Med.* **2020**, *121*, 103792. [CrossRef] [PubMed]
41. Abdulaal, A.; Patel, A.; Charani, E.; Denny, S.; Mughal, N.; Moore, L. Prognostic modeling of COVID-19 using artificial intelligence in the United Kingdom: Model development and validation. *J. Med. Internet Res.* **2020**, *22*, e20259. [CrossRef] [PubMed]
42. Das, N.N.; Kumar, N.; Kaur, M.; Kumar, V.; Singh, D. Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays. *IRBM* **2020**. [CrossRef]
43. Rahaman, M.M.; Li, C.; Yao, Y.; Kulwa, F.; Rahman, M.A.; Wang, Q.; Qi, S.; Kong, F.; Zhu, X.; Zhao, X. Identification of COVID-19 samples from chest X-ray images using deep learning: A comparison of transfer learning approaches. *J. X-ray Sci. Technol.* **2020**, *28*, 821–839. [CrossRef]
44. Clinical Centre of Kragujevac. Available online: https://www.kc-kg.rs/ (accessed on 17 December 2020).
45. Bloice, M.D.; Stocker, C.; Holzinger, A. Augmentor: An image augmentation library for machine learning. *arXiv* **2017**, arXiv:1708.04680.
46. Farda, N.A.; Lai, J.Y.; Wang, J.C.; Lee, P.Y.; Liu, J.W.; Hsieh, I.H. Sanders classification of calcaneal fractures in CT images with deep learning and differential data augmentation techniques. *Injury* **2020**. [CrossRef]
47. Bloice, M.D.; Roth, P.M.; Holzinger, A. Biomedical image augmentation using Augmentor. *Bioinformatics* **2019**, *35*, 4522–4524. [CrossRef]
48. Agrawal, T.; Gupta, R.; Narayanan, S. On evaluating CNN representations for low resource medical image classification. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 1363–1367.
49. Lorencin, I.; Anđelić, N.; Šegota, S.B.; Musulin, J.; Štifanić, D.; Mrzljak, V.; Španjol, J.; Car, Z. Edge Detector-Based Hybrid Artificial Neural Network Models for Urinary Bladder Cancer Diagnosis. In *Enabling AI Applications in Data Science*; Springer: Berlin, Germany, 2020; pp. 225–245.
50. Baltruschat, I.M.; Nickisch, H.; Grass, M.; Knopp, T.; Saalbach, A. Comparison of deep learning approaches for multi-label chest X-ray classification. *Sci. Rep.* **2019**, *9*, 1–10. [CrossRef]
51. Lu, Z.; Bai, Y.; Chen, Y.; Su, C.; Lu, S.; Zhan, T.; Hong, X.; Wang, S. The Classification of Gliomas Based on a Pyramid Dilated Convolution ResNet Model. *Pattern Recognit. Lett.* **2020**, *133*, 173–179. [CrossRef]
52. Jiang, Y.; Chen, L.; Zhang, H.; Xiao, X. Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module. *PLoS ONE* **2019**, *14*, e0214587. [CrossRef] [PubMed]
53. Ma, Y.; Xu, X.; Yu, Q.; Zhang, Y.; Li, Y.; Zhao, J.; Wang, G. LungBRN: A Smart Digital Stethoscope for Detecting Respiratory Disease Using bi-ResNet Deep Learning Algorithm. In Proceedings of the 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), Nara, Japan, 17–19 October 2019; pp. 1–4.
54. Ning, D.; Liu, G.; Jiang, R.; Wang, C. Attention-based multi-scale transfer ResNet for skull fracture image classification. In Proceedings of the Fourth International Workshop on Pattern Recognition, Nanjing, China, 28–30 June 2019; Volume 11198, p. 111980D.
55. Thian, Y.L.; Li, Y.; Jagmohan, P.; Sia, D.; Chan, V.E.Y.; Tan, R.T. Convolutional neural networks for automated fracture detection and localization on wrist radiographs. *Radiol. Artif. Intell.* **2019**, *1*, e180001. [CrossRef]
56. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
57. Yi, D.; Ahn, J.; Ji, S. An Effective Optimization Method for Machine Learning Based on ADAM. *Appl. Sci.* **2020**, *10*, 1073. [CrossRef]
58. Kandel, I.; Castelli, M.; Popovič, A. Comparative Study of First Order Optimizers for Image Classification Using Convolutional Neural Networks on Histopathology Images. *J. Imaging* **2020**, *6*, 92. [CrossRef]

59. Dogo, E.; Afolabi, O.; Nwulu, N.; Twala, B.; Aigbavboa, C. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In Proceedings of the 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 21–22 December 2018; pp. 92–99.
60. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
61. Ballester, P.; Araujo, R.M. On the performance of GoogLeNet and AlexNet applied to sketches. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
62. Yan, L.; Yoshua, B.; Geoffrey, H. Deep learning. *Nature* **2015**, *521*, 436–444.
63. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
64. Qassim, H.; Verma, A.; Feinzimer, D. Compressed residual-VGG16 CNN model for big data places image recognition. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 169–175.
65. Canziani, A.; Paszke, A.; Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv* **2016**, arXiv:1605.07678.
66. Mhapsekar, M.; Mhapsekar, P.; Mhatre, A.; Sawant, V. Implementation of Residual Network (ResNet) for Devanagari Handwritten Character Recognition. In *Advanced Computing Technologies and Applications*; Springer: Berlin, Germany, 2020; pp. 137–148.
67. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
68. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2016; pp. 630–645.
69. Chu, Y.; Yue, X.; Yu, L.; Sergei, M.; Wang, Z. Automatic Image Captioning Based on ResNet50 and LSTM with Soft Attention. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8909458. [CrossRef]
70. Ghosal, P.; Nandanwar, L.; Kanchan, S.; Bhadra, A.; Chakraborty, J.; Nandi, D. Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network. In Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25–28 February 2019; pp. 1–6.
71. Khan, R.U.; Zhang, X.; Kumar, R.; Aboagye, E.O. Evaluating the performance of resnet model based on image recognition. In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, Chengdu, China, 12–14 March 2018; pp. 86–90.
72. Kandel, I.; Castelli, M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express* **2020**. [CrossRef]