

# Pneumonia Detection Using Deep Learning Based on Convolutional Neural Network

Luka Račić, Tomo Popović, *Senior Member, IEEE*, Stevan Čakić, Stevan Šandi

**Abstract**— Artificial intelligence has found its use in various fields during the course of its development, especially in recent years with the enormous increase in available data. Its main task is to assist making better, faster and more reliable decisions. Artificial intelligence and machine learning are increasingly finding their application in medicine. This is especially true for medical fields that utilize various types of biomedical images and where diagnostic procedures rely on collecting and processing a large number of digital images. The application of machine learning in processing of medical images helps with consistency and boosts accuracy in reporting. This paper describes the use of machine learning algorithms to process chest X-ray images in order to support the decision-making process in determining the correct diagnosis. Specifically, the research is focused on the use of deep learning algorithm based on convolutional neural network in order to build a processing model. This model has the task to help with a classification problem that is detecting whether a chest X-ray shows changes consistent with pneumonia or not, and classifying the X-ray images in two groups depending on the detection results.

**Index Terms**— artificial intelligence; convolutional neural network; deep learning; image processing, machine learning; pneumonia detection.

## I. INTRODUCTION

Artificial intelligence (AI) was recognized as an academic discipline as early as in 1950s, but it was not widely explored by scientific community due to its limited practical feasibility for a long time [1]. In last couple of decades, because of increasing availability of processing power and emergence of big data, AI became a focal point of research and business discussions. AI went through various phases in its development, often regarded as “Seasons of AI”. Until recently the best-known season of AI, was the period from the 1970s to the 2000s and the majority of this period is referred to as the “AI Winter” [2]. One of the main reasons why AI could not develop at a fast pace during this period is

\*Research supported in part by the EuroCC project, grant agreement 951732 EuroCC-H2020-JTI-EuroHPC-2019-2.

Luka Račić is with Faculty for Information Systems and Technologies, University of Donja Gorica, Oktoih 1, 81000 Podgorica, Montenegro (e-mail: luka.racic@udg.edu.me).

Tomo Popović is with Faculty for Information Systems and Technologies, University of Donja Gorica, Oktoih 1, 81000 Podgorica, Montenegro (e-mail: tomo.popovic@udg.edu.me).

Stevan Čakić is with Faculty for Information Systems and Technologies, University of Donja Gorica, Oktoih 1, 81000 Podgorica, Montenegro (e-mail: stevan.cakic@udg.edu.me).

Stevan Šandi is with Faculty for Information Systems and Technologies, University of Donja Gorica, Oktoih 1, 81000 Podgorica, Montenegro (e-mail: stevan.sandi@udg.edu.me).

the fact that computers did not have sufficient processing power to handle the work required [3]. The comeback of AI started when IBM developed their chess playing program, Deep Blue, which was able to beat world champion Gary Kasparov in 1997 [1]. In the following years AI started to develop rapidly, which resulted in development of new fields such as Machine Learning (ML) and Deep Learning (DL). In contrast to ML, DL uses a set number of traits and requires human input, and can be trained to classify data on its own [2].

Despite the fact that they were first described in the 1940s, Artificial Neural Networks (ANN) made a big comeback in 2016 when Google’s program AlphaGo, based on DL, managed to beat the world champion in the board game Go [2]. This is considered to be one of the biggest milestones and successes of AI. For comparison in chess at opening there are 20 possible moves, but 361 in Go. This success further pushed AI research and ANN and DL algorithms are used for image recognition, speech recognition, sensor data processing, etc. It is no surprise that with its recent development AI, especially DL, and their application in medicine is growing rapidly, because of the ability of CNNs to successfully classify images [4,5]. Today AI is used in various medical fields, such as: gastroenterology [6], radiology [7], cardiology [8], endoscopy [2]. AI and ML are usually used in the fields that have the ability to build a large database of medical data, typically in a form of digital images, that can be used later for training models. The use of AI in support tools for processing medical images has been suggested in order to improve accuracy and consistency, and time efficiency in reporting [2].

In this paper, we have focused on pneumonia and the use of CNN based algorithm to process chest X-ray images. Pneumonia is an infection of the lower part of the respiratory tract in which the airways and lung parenchym are affected, and the alveolar spaces are consolidated. The annual incidence of pneumonia in children under the age of 5 in Europe is 34-40 per 1000. In developing countries, pneumonia is the leading cause of child mortality. X-rays of the lungs are very useful in diagnosing respiratory disorders and diseases in children [9]. Pneumonia is also detected in hospitalized patients infected with 2019-nCoV coronavirus [10], which further motivated us to explore and research this problem at the time of worldwide pandemic.

## II. MATERIALS AND METHODS

### A. Images of Chest X-Rays, The Dataset

The dataset used for this research is provided by Guangzhou Women and Children's Medical Center, Guangzhou and is openly available on Kaggle [11]. Before the analysis all the bad quality X-rays have been removed by the experts at the Medical Center. The rest has been classified by three experts in the field of radiology [11].

The dataset contains 5856 images of chest X-rays (JPEG). It is divided into three folders, named train, val and test, that are used as training, validation and testing data. The original dataset has only 16 images in the validation folder. For the purpose of experiments in this research, an 80/10/10 split has been performed. That means that 80% of the images is used as training data, 10% as validation data and 10% as test data. Therefore, the train folder for this experiment contains 4684 images, val folder 586 images and test folder 586 images. Each of these three folders contains two subfolders containing images diagnosed as pneumonia or normal. The subfolder names represent the data labels.

The images are high-quality and come in various sizes, but they were subsequently resized for training the model. Because there were more images of X-rays labeled as "Pneumonia" in the training dataset, in order to increase the number of training examples of "Normal" images, the data augmentation was used (techniques such as rotation, zoom, width and height shift). A couple of examples of X-ray images, with their corresponding labels are shown in Fig. 1.

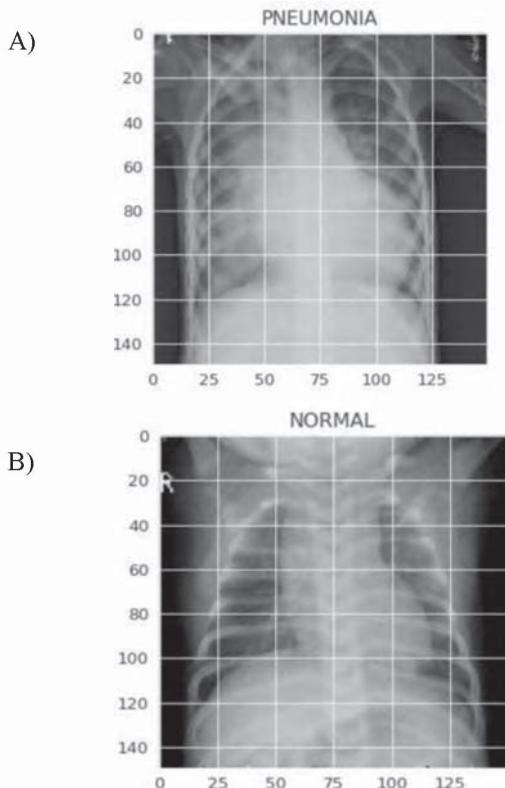


Fig. 1. The dataset illustration: A) a chest X-ray labeled as pneumonia, and B) a chest X-ray labeled as normal

### B. Pre-processing of Images, Dataset Preparation

Most of the time, the first step before building a model is the preprocessing of the imported data. The original images are RGB, but for the purpose of this experiment, they were imported as grayscale and resized to 200x200 pixels. After that, the pixel intensity values were normalized by dividing the pixel values with 255. This way the pixels in the image are represented with floating point numbers between 0 and 1, rather than integer numbers in the range 0 to 255. This should positively affect the performance of CNN [12].

As stated before, the data augmentation has been performed, because of the disbalance in the number of training examples of images showing pneumonia versus those that were normal. This was done because the degree of model overfitting is determined by both its power and the amount of training it receives, providing a CNN with more training examples of images diagnosed as normal, in order to reduce overfitting. In situations when there is no more available data of certain type it can be artificially created by zooming, asymmetrically cropping or rotating input images[12]. All of this can be performed with Keras's preprocessing tools [13].

### C. Convolutional Neural Network and Tools Selection

The main tools used in this example were: Numpy, Pandas, Keras, Jupyter notebook, matplotlib and seaborn [14]. The training and testing of the model were performed locally, on a PC with the following hardware: AMD Ryzen 5 3600 CPU, Nvidia GeForce RTX 2060 SUPER GPU and 16GB of RAM at 3200MHz. Training the model would take up to 90 minutes.

The image classification was done with the use of a CNN-based machine learning algorithm. The CNN is a class of deep learning neural networks. CNNs represent a huge breakthrough in image recognition and classification, where they are most commonly used. They consist of few layers, such as: input, output and between them there are hidden layers. These hidden layers do the most work in terms of calculation. Convolutional layers can be found inside of the hidden layers [13]. Hidden layers can also consist of pooling layers and fully connected layers.

The most important building block of a CNN is the convolutional layer: each neuron in the convolutional layer is only connect to a small number of neurons (receptive field) in the next convolutional layer [13]. Such a design allows the network structure to focus on a small low-level feature in the first hidden layer, and then aggregate them into higher-level features in the next hidden layer, and so on [13]. This design is one the main reasons why CNNs are successfully used in image recognition.

The main use of pooling layers is to reduce the size of the input image without losing any important information. This is done to reduce the computational cost, and reduce memory usage. This process also reduces the number of parameters, which reduces the risk of overfitting.

Figure 2 shows a max pooling layer, which is the most common type of pooling layer [12,13]. The max pooling has been used in this project as well.

The activation function used in our experiments was the ReLU activation function. ReLU behaves very good in deep neural networks, mostly because it does not saturate for positive values and because it is fast to compute. There are some variations of the ReLU activation function such as leaky ReLU, parametric leaky ReLU and SELU [13]. In addition, in the model creation, a dropout method was used to boost the performance. The dropout is a technique where some neurons are randomly shut down and are not used for that iteration. Just by simply adding a dropout, the network can get a 1-2% accuracy boost.

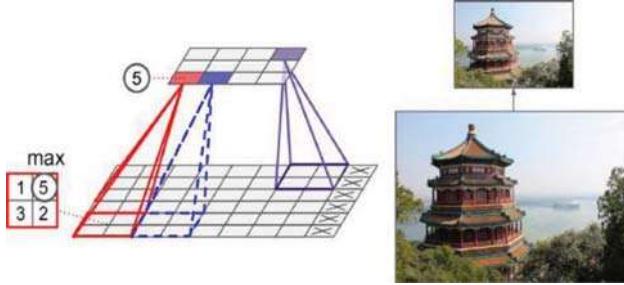


Fig. 2. Illustration of the max pooling layer ( $2 \times 2$  pooling kernel, stride 2, no padding) [13]

The architecture of the CNN used for this research is depicted in Figures 3 and 4. Figure 3 shows the printout of the model summary of the CNN model used for the experiments discussed in this paper. The details on the architecture of the CNN implementation are illustrated in Fig. 4.

Similar approaches were used in [15] and [16]. In [16], the same dataset was used, but the authors used the original split of the data into training, validation and test subsets.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 200, 200, 64)	1792
batch_normalization_6 (Batch Normalization)	(None, 200, 200, 64)	256
max_pooling2d_6 (MaxPooling2D)	(None, 100, 100, 64)	0
conv2d_7 (Conv2D)	(None, 100, 100, 128)	73956
dropout_5 (Dropout)	(None, 100, 100, 128)	0
batch_normalization_7 (Batch Normalization)	(None, 100, 100, 128)	512
max_pooling2d_7 (MaxPooling2D)	(None, 50, 50, 128)	0
conv2d_8 (Conv2D)	(None, 50, 50, 128)	147584
batch_normalization_8 (Batch Normalization)	(None, 50, 50, 128)	512
max_pooling2d_8 (MaxPooling2D)	(None, 25, 25, 128)	0
conv2d_9 (Conv2D)	(None, 25, 25, 256)	295168
dropout_6 (Dropout)	(None, 25, 25, 256)	0
batch_normalization_9 (Batch Normalization)	(None, 25, 25, 256)	1024
max_pooling2d_9 (MaxPooling2D)	(None, 13, 13, 256)	0
conv2d_10 (Conv2D)	(None, 13, 13, 512)	1180160
dropout_7 (Dropout)	(None, 13, 13, 512)	0
batch_normalization_10 (Batch Normalization)	(None, 13, 13, 512)	2048
max_pooling2d_10 (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_2 (Flatten)	(None, 25088)	0
dense_3 (Dense)	(None, 256)	6422784
dropout_8 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 3)	773
Total params: 8,126,467		
Trainable params: 8,124,281		
Non-trainable params: 2,176		

Fig. 3. The model summary printout of the CNN used in the experiment

### III. RESULTS AND DISCUSSION

Evaluation of the results can be done by analyzing the various parameters such as training accuracy and loss, validation accuracy and loss and of course model accuracy.

The progress of the training and validation accuracy and loss during time is shown in the Figures 5 and 6. They show that over time, both the training and validation accuracy were getting better, especially after the 12th epoch.

Figure 6 shows that the validation loss has spikes in the last 30 epochs, especially at epochs 198, but at the end it comes to a good level of near 0. The loss is a way to analyze how well the model is doing on both the training and validation set. It calculates how well the model is doing on each example in these sets and calculates the sum of errors made on them.

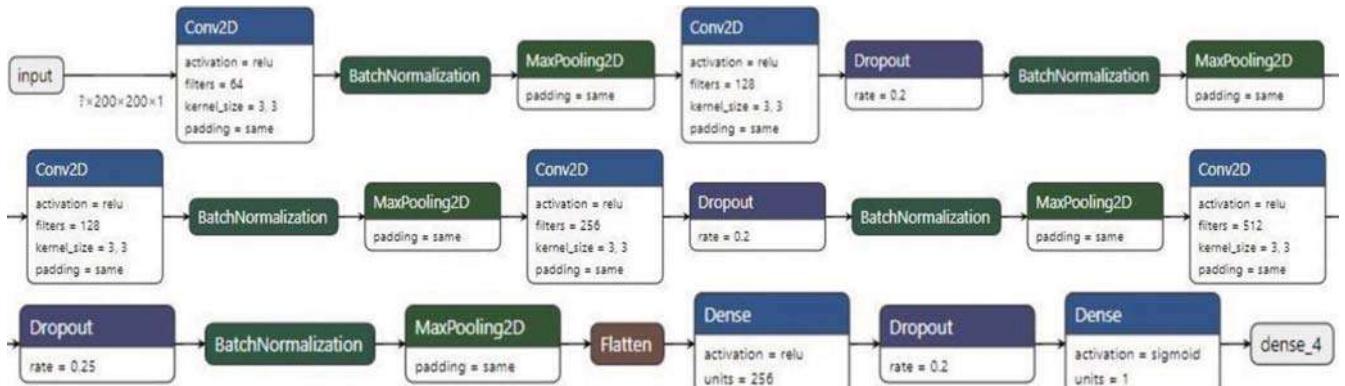


Fig. 4. Architecture of the CNN used for this example

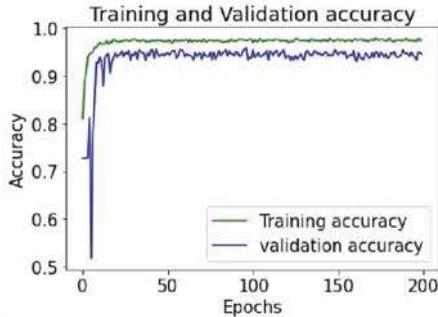


Fig. 5. Training and validation accuracy over time

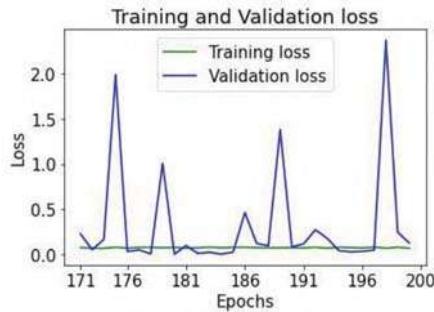


Fig. 6. Training and validation loss in the last 30 of 200 epochs

One additional way of representing the results of the model is to build a confusion matrix [13]. The Y-axis of the confusion matrix holds the predicted values, while the X-axis holds the true values. The confusion matrix for our latest experiment is illustrated in Fig. 7. With the trained model, 334 out of 381 were accurately predicted as images of X-rays with pneumonia, while 187 out of 205 were accurately predicted as X-rays without pneumonia. This gives us a model accuracy of 88.90%, which is comparable to the results in [15,16].

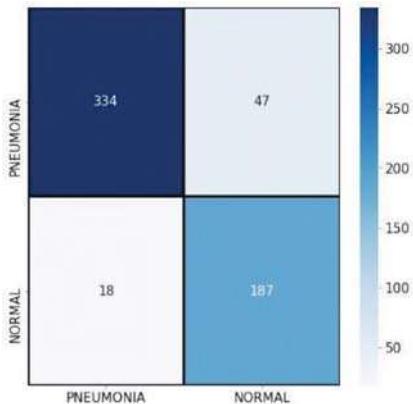


Fig. 7. Confusion matrix of the prediction model performance evaluation

#### IV. CONCLUSION

This paper describes the use of deep learning in order to classify digital images of chest X-rays according to presence or absence of changes consistent with pneumonia. The implementation was based on CNN model using Python programming and scientific tools. Initial experiments show

promising results, but more research is needed.

Even though the model accuracy is relatively high, nearly 90%, there is a possibility of overfitting due to the size of the dataset. Also, the 90% accuracy means that the prediction model could potentially be used as a decision support tool, but there is still much work to be done. The proper diagnosis of any kind of disease still requires the involvement and presence of medical specialists. In order to build a good and reliable disease classification model, it is very important to gather as much data as possible.

Further research steps will include experimenting with various preprocessing and CNN configurations, data augmentation techniques, as well as using additional X-ray datasets with additional data labels showing other pathologies.

#### ACKNOWLEDGMENT

This research is supported in part by the EuroCC project, grant agreement grant agreement 951732 EuroCC-H2020-JTI-EuroHPC-2019-2.

#### REFERENCES

1. M. Haenlein, A. Kaplan, "A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence", California Management Review, 61(4), 2019, pp 5-14
2. V. Kaul, S. Enslin, S.A. Gross, "The history of artificial intelligence in medicine", Gastrointestinal Endoscopy, In Press, 2020.
3. M. Minsky, S.A. Papert, "Perceptrons: An Introduction to Computational Geometry", MIT Press, Cambridge, MA, 1969.
4. A. Esteva et al., "A guide to deep learning in healthcare", Nature Medicine, 25(1), 2019, pp 24–29.
5. D. Ravi et al., "Deep Learning for Health Informatics.", IEEE Journal of Biomedical and Health Informatics, 21(1), 2017, pp. 4-21.
6. Y.J. Yang, C.S. Bang, "Application of artificial intelligence in gastroenterology", World Journal of Gastroenterology, 2019(25), 2019, pp 1666-1683.
7. A. Hosny, C. Parmar, J. Quackenbush, L.H. Schwartz, H.J.W.L. Aerts, "Artificial intelligence in radiology", Nature Reviews. Cancer, 18(8), 2018, pp 500–510.
8. K.W. Johnson, et al., "Artificial Intelligence in Cardiology", Journal of the American College of Cardiology, 71(23), 2018, pp 2668–2679.
9. V.N. Perišić, B. Janković, "Pedrijatrija za studente medicine", Medicinski fakultet, Univerziteta u Beogradu, Beograd, 2010.
10. N. Chen et al., "Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study", The Lancet, 395(10223), 2020, pp 507-513.
11. D. Kermany, K. Zhang, M. Goldbaum, "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, v2, 2018. Accessed July 2020. [Online]. Available: <http://dx.doi.org/10.17632/rscbjr9sj.2>
12. R.C. González, R.E. Woods. "Digital Image Processing", Prentice Hall, 2007, pp 85.
13. A. Geron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", O'Reilly Media, Inc., Canada, 2019.
14. Jupyter Notebook: Project Jupyter. Accessed July 2020. [Online]. Available: <http://jupyter.org>
15. D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, A. Mittal, "Pneumonia Detection Using CNN based Feature Extraction," 2019 IEEE International Conf. on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2019, pp 1-7.
16. V. Sirish Kaushik et al., "Pneumonia Detection Using Convolutional Neural Networks (CNNs)", In: Singh P., Pawłowski W., Tanwar S., Kumar N., Rodrigues J., Obaidat M. (eds) Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019). Lecture Notes in Networks and Systems, vol 121. Springer, Singapore.