https://doi.org/10.3311/PPee.21616

Periodica Polytechnica Electrical Engineering and Computer Science, 67(3), pp. 345–354, 2023

Two Approaches for Detecting Pneumonia from Chest X-ray Images

Neural Network vs Kolmogorov Complexity

Andrey Pechnikov[1], Nikolai Bogdanov[2], Anthony Nwohiri[3*], Ijeoma Nwohiri[4]

[1] Institute of Applied Mathematical Research of the Karelian Research Centre, Russian Academy of Sciences, 11 Pushkinskaya street, 185000 Petrozavodsk, Russian Federation
[2] Faculty of Applied Mathematics & Control Processes, St. Petersburg State University, 7-9 Universitetskaya Embankment, 199034 St Petersburg, Russian Federation
[3] Department of Computer Sciences, University of Lagos, University Road, Akoka, Yaba, 101017 Lagos, Nigeria
[4] Federal Medical Centre, Railway Compound, 2 Ondo St., Ebute Metta, 101245 Lagos, Nigeria
* Corresponding author, e-mail: anwohiri@unilag.edu.ng

Received: 29 November 2022, Accepted: 23 March 2023, Published online: 10 May 2023

Abstract

Pneumonia is an infection that inflames the air sacs in the lungs. It remains the leading cause of death in children aged <5 years. This acute respiratory infection kills over 150,000 newborns yearly. We present two approaches for detecting pneumonic lungs. Both involve chest X-ray (CXR) image classification. The first approach is based on convolutional neural networks (CNN). The second approach, proposed by us, uses the theoretical notion of Kolmogorov complexity (KC), which introduces the normalized compression distance (NCD) – a way of measuring similarities between objects of different nature, such as images. The respective algorithms are described, software implementation details are presented. Experiments were conducted to enable us to choose optimal parameter values that would facilitate accurate pneumonia detection. The two procedures showed high classification quality. This convincingly indicates they were accurate in differentiating the chest X-rays. Though a known fact, the CNN approach was confirmed to be more efficient when dealing with a larger training dataset. On the other hand, the NCD-KC technique was shown to be more efficient when handling a small number of classified images. A more sensitive and more accurate pneumonia diagnosing technique that combines the strengths of both approaches is found to be feasible.

Keywords

chest X-ray imaging, convolutional neural network, image processing, normalized compression distance, pneumonia

1 Introduction

Pneumonia is a form of acute respiratory infection that specifically affects the lungs. It remains a major cause of childhood mortality and morbidity globally. It is the single biggest infectious cause of mortality in adults and children [1, 2]. Viral and bacterial pneumonia have similar symptoms. This acute respiratory infection has claimed the lives of 2.5 million, including 672,000 children, in 2019. It accounted for 14% of all deaths of children under 5 years old but 22% of all deaths in children aged 1 to 5 [3].

Early and accurate diagnosis of pneumonia by polymerase chain reaction (PCR)-based techniques [4] is difficult. Though PCR methods are known for high sensitivity and good reproducibility, and they produce results in a few hours, they come with strict requirements in terms of analysis technique and personnel skills.

A chest CT scan may be used to detect pneumonia that may be more difficult to see (even to a professional radiologist) on a plain X-ray [5]. Moreover, not all medical institutions have the required CT facilities. Though radiography is more common at healthcare facilities because of the wider availability of X-ray machines, the process introduces unavoidable "noise" several times and has lower spatial resolution [6].

Since early diagnosis of pneumonia is crucial to ensure curative treatment and increase survival rates, it is highly necessary to develop computer programs that would facilitate early and accurate detection by chest radiography.

In this paper, we present two approaches to accurate detection of pneumonia in CXR images, which can be utilized in the real world to treat pneumonia. The first

Cite this article as: Pechnikov, A., Bogdanov, N., Nwohiri, A., Nwohiri, I. "Two Approaches for Detecting Pneumonia from Chest X-ray Images: Neural Network vs Kolmogorov Complexity", Periodica Polytechnica Electrical Engineering and Computer Science, 67(3), pp. 345–354, 2023. https://doi.org/10.3311/PPee.21616

one, which has been actively studied recently, is based on CNN. The capabilities of this approach have been demonstrated [7–9]. It has been noted that classification accuracy strongly depends on sample size – the higher the size, the better the classification. Special focus is also placed on image preprocessing.

The second approach, proposed by us, is grounded on the KC-based NCD [10, 11]. The normalized compression distance uses the theoretical notion of the Kolmogorov complexity (KC) to measure the "length" of the compressed version of a file, using a real-world compression program [12–15]. Image classification through KC is not a new idea [16]. Findings from a comparison of these two approaches could be useful when developing a robust pneumonia diagnosis application in the future.

Artificial intelligence, machine learning, deep learning, neural networks, etc. have gained so much attention in the last few years. However, other effective detection and classification approaches must not be excluded and forgotten. Here, the KC (an approach invented a long time ago) is a good example; it performs better than a neural network when the amount of data is relatively small. In such a case (small amount of data), there is nothing to train a neural network on, whereas the Kolmogorov method can still perform because it does not require training. So, the motivation here is that sometimes, it is worth remembering the experience of predecessors, and not forgetting them.

Section 2 presents some related works. After Section 2, we describe the images used in the work, their source, and the basic steps taken in processing them for the study. Section 3.2.1 (CNN approach) and Section 3.2.2 (KC approach) present the basic theoretical aspects of each of the two algorithms and the most important ingredients in their software implementation. Section 4 features the description and results of experiments, comparison of both approaches, and the main strengths of each. We draw some conclusions and provide practical recommendations in Section 5.

2 Related work

Ranjan et al. [17] developed a deep learning-based approach using a customized deep learning model (VGG-16). Their target was to facilitate pneumonia detection and improve diagnosis accuracy. The proposed visual geometry group model was trained on 5856 CXR images, comprising that of both healthy individuals and pneumonia patients. The model achieved 98.28% accuracy, 0.98 precision, 0.97 recall, and 0.976 F1 score. However, there are often CNN architectures available today that perform better than the visual group geometry VGG16 model, and transfer learning can be a challenging task due to the vanishing gradient problem.

A CNN model for classifying positive and negative pneumonia data from X-ray images was demonstrated by Stephen et al. [18]. Unlike other traditional techniques, it was designed from scratch to retrieve features from a given CXR and categorize it to determine if a person is infected with pneumonia or not. Due to the lack of a large pneumonia dataset for this classification task, the authors used data augmentation algorithms to improve the validation and classification accuracy of the CNN model. The study had limited access to radiological data. Training of the model with data from patients and nonpatients in different parts of the world could make significant improvements.

Swapna et al. [19] deployed deep learning networks of CNN and CNN-LSTM (LSTM = long short term memory) combination to automatically detect diabetes. This was done by analyzing heart rate variability (HRV) signals obtained from ECG signals. With a 5-fold cross-validation, CNN gave an accuracy of 93.6% while CNN-LSTM combination gave the maximum accuracy of 95.1%. From results obtained, it can be inferred that the proposed deep learning architecture did not learn the complete patterns associated with diabetes and non-diabetes data. The main reason may be due to the fact that the size of the HRV data used is less than that needed to be pushed as input to deep learning networks.

Reading CXR images is a difficult and challenging task, and it is prone to subjective variability. A computer-aided diagnosis system for pneumonia detection through CXR images has been developed [20]. Deep transfer learning was used to handle data scarcity. An ensemble of three CNN-based models was designed. A weight vector was formed by fusing the scores of four standard evaluation metrics. The method was evaluated on two datasets using a 5-fold cross-validation scheme. The approach reached 98.81% and 86.85% accuracy and 98.80% and 87.02% sensitivity. These results outperformed those obtained by futuristic methods. Moreover, the method showed better performance than the widely used ensemble techniques. However, the ensemble framework was unable to predict correctly in some instances. This may be down to contrast enhancement of the images or other pre-processing steps that were aimed at improving image quality. Another possible solution is the use of segmentation of the image before classification to enable the CNN models to achieve improved feature extraction. In addition, since

three CNN models were required to train the proposed ensemble, the computation cost was higher than that of the CNN baselines considered in the study.

Manickam et al. [21] designed a deep learning (DL) method using two optimizers, Adam and Stochastic Gradient Descent. The method used to classify pneumonia patients was trained on a benchmark dataset of CXR images. Transfer learning approach was used, and three pre-trained architectures were adopted, namely ResNet50, InceptionV3 and InceptionResNetV2. Performances of the pre-trained models were compared with other CNN models using various metrics. The proposed model achieved 93.06% accuracy, 88.97% precision, 96.78% Recall rate and 92.71% F1-score. These figures were higher than those of the other models. For ResNets, detection of errors becomes difficult in a deeper network.

A pneumonia diagnosis model, trained on CXR images, was proposed by Hashmi et al. [22]. Image data augmentation was used to increase the training dataset. Transfer learning was deployed while training the models. The model was evaluated and statistically validated to overfitting and generalization errors. Different scores were computed to check the performance. A 98.14% test accuracy and 99.71 AUC score were achieved on the test data. However, the authors were unable to localize the specific parts of the lung affected by pneumonia.

De and Chakraborty [23] proposed a disease detection system that could be used by healthcare specialists to detect liver disorders, hepatitis, heart disease, diabetes, and chronic kidney disease. The authors used Adaboost Classifier to detect diseases. This ML algorithm is able to identify referred diseases in the detection system with 100% accuracy, precision, and recall. AdaBoost uses a progressively learning boosting technique. Therefore, it needs a high-quality dataset. Noisy data and outliers have to be avoided before adopting an Adaboost algorithm.

Chest x-rays of 23,954 individuals were analyzed by Qin et al. [24] for tuberculosis detection. The images were independently interpreted by a group of three radiologists and five AI algorithms. All the AI algorithms significantly outperformed the radiologists, reducing the number of Xpert MTB/RIF tests required by 50%, while maintaining a >90% sensitivity. The algorithms performed worse among older age groups (>60 years) and people with a tuberculosis history. The work demonstrated that AI algorithms can be highly accurate and serve as useful triage tools for detecting tuberculosis; they can outperform humans. The study had several limitations. Due to logistic and budgetary constraints, the authors did not use culture as the reference standard, meaning that some people with Xpert-negative, culture-positive tuberculosis might have incorrectly been labelled as not having tuberculosis. Due to the small number of asymptomatic individuals, the authors did not stratify by symptoms and analyze by symptom subgroup. No HIV testing was done because Bangladesh has a low HIV prevalence [24]. Children from the study population were excluded, even though some of the algorithms included were licensed for use in younger age groups. Additionally, only one brand of x-ray machine was used in this study due to procurement constraints. Lastly, the study was not conducted prospectively, and the authors did not collect implementation data. These decisions limit the generalizability of the findings.

3 Materials and methods

3.1 X-ray images and image preprocessing

X-ray images used in this work were obtained from the Kaggle dataset [25]. We retrieved 5,856 X-ray images in .jpeg format as separate files; 1,583 of them were X-ray images of healthy people, while the remaining 4,273 were those of people with pneumonia. In the Kaggle dataset, the images are classified and labeled as Normal and Pneumonia. Some of the X-ray images are shown in Fig. 1. The left image belongs to a person who does not have pneumonia, while the right one is that of a pneumonia patient.

Both detection algorithms were implemented in a dedicated Python integrated development environment Pycharm, using Keras [26], an open-source software library that provides a Python interface for artificial neural networks.

Chest radiographs are black and white raster images and are of different sizes in Kaggle. That is why the images need to be first processed into some matrix (vector) form, which is used in both of our methods. To do this, the images are divided into pixels. Both for neural networks and for KC, the number of pixels varies in width and height depending on the experiments performed. The image with the minimum number of pixels has 16,384 pixels (128 by width and 128 by height) and the maximum is 1,048,576 pixels

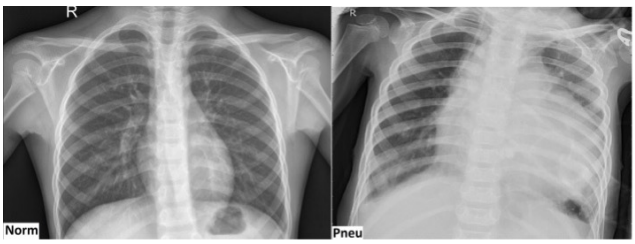

Fig. 1 X-ray images

(1,024 by width and 1,024 by height). Each pixel color is represented by an integer from 0 to 255. In grayscale, 0 is black and 255 is white. The .jpeg images were converted to vector representation using the OpenCV-Python package [27], which is often used for such purposes.

To work with the CNN-based program, we needed to have three non-intersecting sets: a training set *T*, a verification (or validation) set *V*, and a test set *D*. *T* is intended for training the network itself, *V* is used to set the classifier parameters; in our case, it was used to control the training progress. *D* is used to test the model. Each set is divided into two non-intersecting subsets: images of healthy people and images of pneumonia patients.

To artificially expand the size of *T*, real-time image data augmentation was performed using the ImageDataGenerator class [28]. The essence of this method is that before the next training epoch, each image from the initial set *T* undergoes a series of transformations – rotation/shift by a small random angle/shift over the images and then the image scale is multiplied by a small random multiplier (zoom in/zoom out). These manipulations ensure that the neural network does not encounter the same image from *T* twice during training. This significantly improves the generalization capability of the neural network.

The KC-based program required two sets of X-rays: a set of images *Y*, which we called the control set, and the test set *X*. Accordingly, *Y* and *X* were also split into two non-intersecting subsets.

The sizes of *T, V, D, Y* and *X* and their subsets vary depending on the purpose of the training and/or experiment.

3.2 Pneumonia detection approaches

Image classification is crucial in computer-aided diagnosis. Medical image analysis involves feature extraction and representation, feature selection that will be used for classification, and feature and image classification. Here we look at two image classification approaches - based on CNN and on KC.

3.2.1 Convolutional neural network

CNN was first described by Lecun et al. [29] as a specialized type of neural network. It is one of the most popular and effective models for image recognition problems. When it comes to accuracy, CNNs blow competition out of the water.

A CNN consists of a step-by-step transition from specific features of an image to more abstract details, then to even more abstract ones, and so on up to extraction of high-level concepts. These abstract features form the so-called feature maps (matrices). The network self-adapts and produces the required hierarchy of abstract features (feature map sequences), discarding unimportant details and picking essential ones.

In general, a CNN is composed of multiple building blocks (layers). The original image is fed to the input layer. The signal then passes through a sequence of convolutional layers where the actual convolution and subsampling alternate. Formally, convolution is a specific operation, which involves multiplying the original matrix by a smaller matrix called the convolution kernel. The kernel sort of "moves" over the original matrix from left to right and from top to bottom, computing, in each position, the scalar product of the filter and the part of the original matrix on which it is superimposed. The resulting number is stored as the corresponding element of the result. The convolution layer is an application of the convolution operation on outputs from the previous layer; convolution kernel weights are trainable parameters.

The matrix kernel dimension and the shift step are chosen depending on a number of conditions [30]. We used layers with 32, 64, 128 and 256 kernels, the same kernel size (3 × 3) and a shift step of 1. The scalar result of each convolution falls under a trigger function – a nonlinear function that is usually embedded in the convolution layer. Typically, the activation function "squeezes" the result to the desired constraints. For example, the ReLU (Rectified Linear Unit) function returns 0 if the input is negative, and the number itself if it is positive. We used the ReLU function because our experiments achieved good neural network results here for classification of radiographic images of pneumonia patients. A model that uses the rectified linear unit is easier to train and often achieves better performance [31]. Unlike other trigger functions, the main catch here is that the ReLU function does not activate all the neurons at the same time.

We used the Dropout regularization technique [32]. The essence here is that in one or more layers of a neural network, each neuron can be shut down (the output of this neuron is 0 regardless of the input signals) with a given probability *p*. The state of a neuron (on/off) is determined at each training iteration. When neurons are randomly removed from the network during training, others will have to chip in and handle the representation required to make predictions for the missing neurons. It is an efficient way of performing model averaging. The adaptation prevents all the neurons from converging to the same goal [33, 34]. This simulates a large number of networks

with different network structure and, in turn, make nodes in the network generally more resistant to inputs to prevent the network from overfitting by preventing complex co-adaptations on training data [35].

Pooling is used to reduce the dimension of the matrix (feature map). The initial matrix is divided into blocks and some function is calculated for each of them. For example, a maximum function is used where a block is replaced by a single value of the maximum element of that block.

Alternation of layers allows to create feature maps from previous ones; on each next layer, the map reduces in size but increases in number. Usually, after passing through several layers, the feature map is gradually transformed into a set of vectors or even scalars. So, since the matrix (feature map) degenerates, we get hundreds of such scalars (feature maps).

A sequence of convolutional layers (convolution and pooling) gives as output a set of terminal feature maps. In turn, the set is fed to the input of an ordinary fully connected neural network, which can also consist of several layers. This construction is sometimes referred to by the generic term "hidden layer". After the hidden layer, the signal is fed to the output layer, whose outputs form the result (network's response to the input).

A number of standard methods have been developed for CNN training. Among the methods is the backpropagation method, which is quite often used. In a mathematical sense, it is an iterative gradient algorithm that is used to minimize neural network error and obtain the desired output. For this purpose, an evaluation function that depends on the output signals and their required (known) values is designed. For example, the least squares method can be used. To minimize the estimation function, the weights should be changed after each training example, "moving" in the multidimensional weight space in the opposite direction to the gradient pointing to the direction of the largest error increase.

To tackle the problem of pneumonia detection in X-ray images, we developed a CNN with the following structure: an input layer, four convolutional layers, a hidden layer, and an output layer. The structure is depicted in Fig. 2 using Python and Keras terminology. The first block contains *Conv2D*, which is a convolutional 2D layer that takes a 150 × 150 pixel matrix as input and outputs 32 matrices of the same dimension. *Stride* denotes how many steps we are moving in each step in convolution. It is one by default. *Padding* describes the addition of empty pixels around the edges of an image. It can increase the

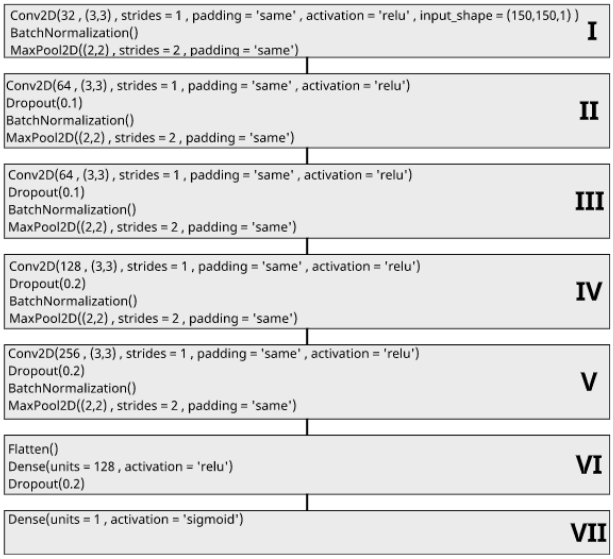

Fig. 2 CNN architecture and main parameters

height and width of the output. This is often used to make the output the same height and width as the input to prevent unwanted shrinking of the output. It also ensures that all pixels are used with equal frequency. The ReLU function is used as an activation function. Here *input_shape* is an additional argument containing the dimension of the input image: the first two components define the height and width of the input image, while the third component defines the number of color channels (1 since the image is black and white). *MaxPool2D* is a subsampling layer. The *BatchNormalization* operation implements the batch normalization method proposed by Ioffe and Szegedy [36] in 2015. This technique makes the training of neural networks faster and more stable.

The second block takes as input 32 formed matrices, convolves, pools, and normalizes them. The output consists of 64 matrices. Here, *Dropout* is performed at probability $p = 0.1$, as in the third block, and $p = 0.2$ in subsequent blocks. The third block outputs as many feature maps as it receives. However, the pooling layer additionally reduces the dimension of feature maps fed to the output, which subsequently affects reduction of the dimension of the vector fed to the input of the fully connected section of the network four-fold. The fourth and fifth blocks are almost similar to the third block, but their outputs are 128 and 256 matrices, respectively, and the *Dropout* operation, as mentioned above, is performed with 0.2 probability. The sixth block implements a fully connected neural network (hidden layer) consisting, in turn, of several layers. The *Flatten* layer converts 2D into 1D data. In our case, it forms a vector of 2,560 elements from

256 5 × 5 matrices. The *Dense* layer outputs a vector of 128 elements using the ReLU function. The seventh block is the output layer that forms the result. The sigmoid function is used as an activation function. It calculates the values of the sigmoidal function of argument x with Eq. (1):

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{1}$$

The neural network was trained using the RMSprop optimizer, a gradient-based optimization technique used in training neural networks [26]. Chosen as the loss function is binary cross-entropy, implemented in the Keras library as *binary_crossentropy* and having the following form:

$$\text{Loss}\left(y_j, y_j^*\right) = y_j * \log\left(y_j^*\right) - \left(1 - y_j\right) * \log\left(1 - y_j^*\right).$$

For n observations, $j \in 1 \ldots n$, x_j is the result (output signal) of the j-th observation, while y_j is the required (known) value. The neural network response for object x_j is a single real number y_j^*, which is the probability that object x_j belongs to class 1; the probability that x_j belongs to class 2 is $1 - y_j^*$.

The accuracy metric was used to evaluate the training quality. Accuracy is the ratio of correctly classified objects to the total number of objects.

In the training of neural networks, the term *epoch* is used to denote a repetitive process that consists of feeding all examples from the training set as input, updating the neural network weights after each iteration, and, possibly, evaluating the training quality on the validation set. The classification performance of the CNN is measured after each epoch and the validation results can influence training hyperparameters (e.g., learning rate, optimal number of epochs, etc.).

In our case, we had to perform a series of experiments for different training and validation sets and different number of epochs. As a basic example, we trained our CNN over 40 epochs for a training set T containing 6,375 processed images (2,500 images of healthy subjects and 3,875 images of pneumonia patients) using a validation set V (200: 100 and 100 images, respectively). After each training iteration on T, we subjected V to a quality assessment. The accuracy function (as a function of epoch for both sets) obtained is shown in Fig. 3.

Fig. 3 shows that training accuracy on the validation set reaches its maximum at the 20th training epoch, after which it slightly falls, probably indicating overfitting in the network. So, it makes sense to stop training after the 20th epoch.

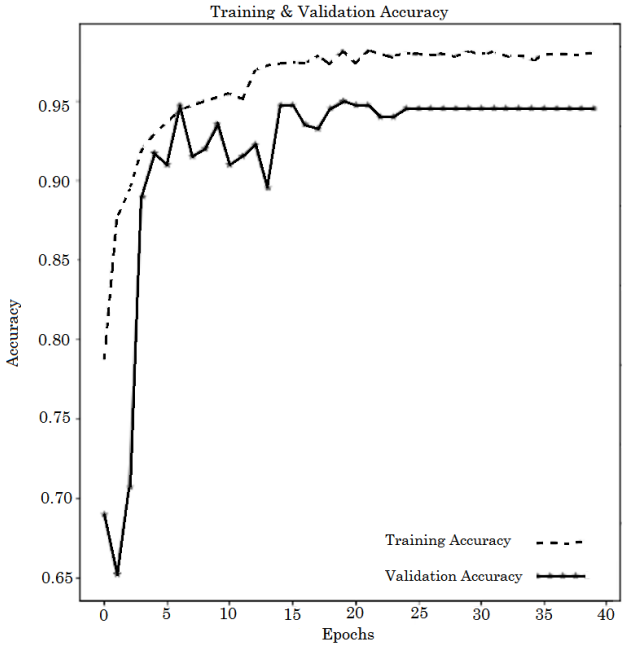

Fig. 3 Accuracy function for sets T and V

3.2.2 Kolmogorov complexity

The Kolmogorov complexity (KC) of an object, such as an image, is the length of the shortest computer program (in a specific programming language) that produces the object as output. Also known as algorithmic complexity, KC is a measure of the computational resources needed to specify the object.

The next two paragraphs present the definition of the Kolmogorov complexity as given exactly by Shen et al. [13] in the authors' English translation.

Let an arbitrary computable partial self-mapping D (from a set of binary strings Ξ) be the description method or decompressor. D is computable if there exists an algorithm that applies to strings from the D definition domain and only to these words; $D(x)$ is the result of applying the algorithm to string x. If $D(y) = x$, y is said to be a description of x under the D description method. For each D description method, we define complexity with respect to that description method by assuming it to be equal to the length l of the shortest description, see Eq. (2).

$$KSD(x) = \min\left\{l(y) \middle| D(y) = x\right\} \tag{2}$$

The minimum of the empty set is said to be ∞. Description method D_1 is said to be no worse than description method D_2 if there exists a constant c such that $KS_{D1}(x) \leq KS_{D2}(x) + c$ for all x. A particular description %& %' method is optimal if it is no worse than any other description method.

Now, we define a (not necessarily optimal) description method and denote by $K(x)$ the complexity of x with respect to this description method. Hereafter, we will consider it equal to the number of bits in the compressed version of x; archiver programs will serve as our D. This approach is used, for example, by Cilibrasi and Vitanyi [10], who stated that "*... the Kolmogorov complexity of a file is essentially the length of the ultimate compressed version of the file*" [10:p.1528].

Let y be another binary word. Following the method proposed by Cilibrasi and Vitanyi [10], we denote by $K(x|y)$ the minimum number of bits needed to reconstruct x from y. For any pair of x and y, we can define the normalized compression distance as

$$NCD(x,y)=\frac{\max\{K(x|y),K(y|x)\}}{\max\{K(x),K(y)\}}.$$

It is proved that $NCD(x, y)$ is symmetric, the identity and triangle axioms are also satisfied. The algorithmic information symmetry theorem is proved by Li and Vitányi [14]. Its consequence – the approximate equality $K(x|y) \approx K(yx) - K(y)$, where yx represents the concatenation of y and x. Therefore, given that in practice, $K(xy) \approx K(yx)$, the normalized compression distance can be approximately calculated as

$$NCD(x,y)=\frac{K(xy)-\min\{K(x),K(y)\}}{\max\{K(x),K(y)\}}.$$

Let there be a "test" set of images $Y = Y_{Pneu} \cup Y_{Norm}$, where Y_{Pneu} is a subset of lung images with signs of pneumonia, and Y_{Norm} is a subset of images without pneumonia signs; $|Y_{Pneu}| = |Y_{Norm}| = M$. There is also a test set X whose images should be classified as healthy or as pneumonic; $|X| = N$. Each element of Y and X is a file with the corresponding chest image. We denote the elements of set Y_{Pneu} as Y^i_{Pneu}, elements of set Y_{Norm} as Y^i_{Norm} and elements of set X as x_i.

The basic idea of the algorithm is that for each tested element of X, the distance to Y_{Pneu} and Y_{Norm} is calculated, after which the image is recognized as pneumonic or non-pneumonic based on the criterion of minimum distance to the corresponding set. The distance from an element to a set can be determined in different ways. We used the minimum of NCD distances from the tested element of X to each element of Y_{Pneu} and Y_{Norm}.

We assume there is a converter software that converts the contents of an image-containing file into a binary file. We also assume there is an archiver program that can compress the contents of this binary file. So, we denote the conversion operation by $Conv$, and the compression operation by $Arch$. Sequential execution of these two operations on image-containing file x is denoted by $Arch(Conv(x))$. Accordingly, the Kolmogorov complexity $K(x)$ of file x is equal to the size of the file resulting from the $Arch(Conv(x))$ operation, that is, $K(x) = Size(Arch(Conv(x)))$.

Let us denote file concatenation operation with the symbol ⊕. Given the introduced concepts and definitions, in general terms, the chest X-ray image classification algorithm for pneumonia diagnosis is presented in Algorithm 1.

In Algorithm 1, chest images were assigned to a particular class based on the minimum distance among the minimum normalized compression distances to each class. We tried

Algorithm 1 Classification algorithm for pneumonia diagnosis

//Calculating K for the control set:

for $\forall y^i_{Pneu} \in Y_{Pneu}$, **do**

$K(y^i_{Pneu}) = Size(Arch(Conv(y^i_{Pneu})))$;

end for

for $\forall y^i_{Norm} \in Y_{Norm}$, **do**

compute $K(y^i_{Norm}) = Size(Arch(Conv(y^i_{Norm})))$;

end for

//The main step of the algorithm (classification of elements of the tested set):

for $\forall x_i \in X$ **do**

compute $K(x_i) = Size(Arch(Conv(x_i)))$;

end for

for $\forall y^j_{Pneu} \in Y_{Pneu}$, **do**

perform $x_i \oplus y^j_{Pneu}$;

compute $K(x_i \oplus y^j_{Pneu}) = Size(Arch(Conv(x_i \oplus y^j_{Pneu})))$;

compute $NCD(x_i, y^j_{Pneu}) = \frac{K(x_i \oplus y^j_{Pneu}) - \min\{K(x_i), K(y^j_{Pneu})\}}{\max\{K(x_i), K(y^j_{Pneu})\}}$;

end for

for $\forall y^j_{Norm} \in Y_{Norm}$, **do**

perform $x_i \oplus y^j_{Norm}$;

compute $K(x_i \oplus y^j_{Norm}) = Size(Arch(Conv(x_i \oplus y^j_{Norm})))$;

compute $NCD(x_i, y^j_{Norm}) = \frac{K(x_i \oplus y^j_{Norm}) - \min\{K(x_i), K(y^j_{Norm})\}}{\max\{K(x_i), K(y^j_{Norm})\}}$

find $sign_{Pneu} = \min_{j=1...M} NCD(x_i, y^j_{Pneu})$ and

$sign_{Norm} = \min_{j=1...M} NCD(x_i, y^j_{Norm})$

end for

if $sign_{Pneu} \le sign_{Norm}$, **then**

assign x_i to the pneumonia class.

else

assign x_i to the non-pneumonia class

end

other criteria, e.g., minimum of the mean, but the result was worse. The image files were archived using 7z, an open-source file archiver. Other archiver programs were tested, e.g. WinRAR archiver, but they did not perform better.

4 Results and analysis

Binary classification results are quite often evaluated using a confusion matrix (error matrix) [37]. We have presented our matrix in Table 1.

In these notations, *precision*, which tells us how many of the correctly predicted cases actually turned out to be positive, is described by Eq. (3):

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3}$$

Recall, presented under Eq. (4), shows how many of the actual positive cases we were able to predict correctly with our model.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

F-measure, shown in Eq. (5), represents the harmonic mean of precision and recall. This measure allows to optimize the two metrics simultaneously.

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

We performed a series of experiments for the developed CNN and for the program implementing the KC-based algorithm. Our goal was to compare the two programs. So, the same randomly generated test set was used for each of them. Below, we present the most significant results obtained.

For CNN, the dimension of training set *T* was taken as variable parameters. Table 2 shows the values for the case of 20 training epochs for the network presented in Fig. 2.

Table 1 Confusion matrix

	True classification	
	Pneumonia	No pneumonia
Predicted result	TP:FN	TN:FP

Note: TP (true positive), image is correctly classified as pneumonic;
FP (false positive), image is incorrectly classified as pneumonic;
TN (true negative), image is correctly classified as non-pneumonic;
FN (false negative), image is incorrectly classified as non-pneumonic.

Table 2 Experimental results and metrics for CNN

T	norm	pneumo	precision	recall	F-measure
200 + 200	0:50	50:0	0.500	1.000	0.667
250 + 250	12:38	49:1	0.563	0.980	0.715
300 + 300	46:4	45:5	0.918	0.900	0.909
400 + 400	49:1	46:4	0.979	0.920	0.948

The network was retrained with the same initialization weights and other pseudorandom values as the first time. Column *T* shows the number of images of healthy subjects plus (+) that of pneumonia patients in the training set. Test set D contains 50 images of healthy people (we use the notation "*norm*" to denote them) and 50 images of pneumonia patients ("*pneumo*"). The validation set *V* for CNN in all cases contained 100 images of healthy people and 100 images of pneumonia patients.

For the KC-based algorithm, image segmentation into pixels was our variable parameter. The "pixels" column in Table 3 lists the pixel width and height partitioning values. Control set *Y* contains 100 images of pneumonia patients and 100 images of healthy subjects. The main step of the KC algorithm requires time to classify each element of the test set. The time increases significantly as the dimension of *Y* increases. Classification quality does not significantly improve when the dimension is increased beyond a set of 100 images of pneumonia patients and 100 images of healthy people. Set *D* for CNN is used as test set *X*.

Obviously, as the training set increases, the CNN metrics improve. For the KC-based algorithm, the picture is somewhat different: increasing the number of pixels in the image partition leads to better results up to a certain value, after which deterioration sets in. In our case, the best results were obtained for 512 × 512 pixels. Note that on the same test set, these results are between 250 + 250 and 300 + 300 for CNN, better than the former and worse than the latter.

As stated in the introduction, neural networks today represent a promising direction for solving the problem of pneumonia detection in CXR images. Our study has confirmed this and has also supported the fact that classification accuracy strongly depends on sample size.

Meanwhile, the proposed KC-based approach opens up the possibility for more accurate classification in cases where the dataset of classified images is small. However, when the training data is large enough, the KC-based algorithm becomes inferior to CNN, albeit slightly, on all metrics. This is evident in the second row of Table 2. Here, test set *X* is equal to *D*, while control set *Y* contains 100 images of pneumonia patients and 100 images of healthy subjects.

Table 3 Experimental results and metrics for the KC-based algorithm

pixels	norm	pneumo	precision	recall	F-measure
128 × 128	0:50	50:0	0.500	1.000	0.667
350 × 350	9:41	50:0	0.549	1.000	0.709
512 × 512	42:8	44:6	0.846	0.880	0.863
800 × 800	50:0	22:38	1.000	0.367	0.537

5 Conclusion

This paper has considered two fundamentally different approaches to the classification of chest X-ray images for the purpose of diagnosing pneumonia. The first approach, which is widely used today, is based on convolutional neural networks. The second approach adopts the Kolmogorov complexity theory, whereby the normalized compression distance is used to classify and measure "similarities" between the images. Both approaches were presented, their respective algorithms were described, and software implementation details were provided.

Experiments conducted enabled us to choose parameter values such that the problem (accurate pneumonia detection from X-ray images) could be successfully and efficiently solved. The CNN demonstrated high classification quality when the training set had ≥300 images of healthy subjects and ≥300 images of patients with pneumonia. For a smaller number of images (<300), classification quality falls significantly. To our knowledge, there is no rigorous characterization of the sample complexity of learning a CNN. The optimal sample size required to effectively train a CNN model has not yet been ascertained [38]. However, the Kolmogorov complexity algorithm shows rather high classification quality for 100 classified images of healthy people and 100 classified images of pneumonia patients, something that was unattainable by CNN.

All this suggests that CNN is superior when handling a fairly large training set. The KC approach can tackle the same problem much better but for a smaller number of classified images. Our findings offer a promising outlook for the development of more sensitive and accurate pneumonia detection methods that integrate the merits of both approaches.

Acknowledgement

We would like to thank the anonymous reviewers immensely for their careful reading of our manuscript, for taking the time and effort necessary to review the manuscript. We sincerely appreciate all their many insightful comments and suggestions, which helped us to improve the quality of the manuscript.

References

[1] Zar, H. J., Andronikou, S., Nicol, M. P. "Advances in the diagnosis of pneumonia in children", British Medical Journal, 358, j2739, 2017. https://doi.org/10.1136/bmj.j2739

[2] World Health Organization "Pneumonia in children", World Health Organization, 11 November 2022. [online] Available at: https://www.who.int/news-room/fact-sheets/detail/pneumonia [Accessed: 20 November 2022]

[3] Every Breath Counts Coalition "World Pneumonia Day 2021", Stop Pneumonia, [online] Available at: https://stoppneumonia.org/latest/world-pneumonia-day/ [Accessed: 20 August 2022]

[4] Bartlett, J. M. S., Stirling, D. "A short history of the polymerase chain reaction", In: PCR Protocols, Humana Press, 2003, pp. 3–6. ISBN 978-0-89603-642-0 https://doi.org/10.1385/1-59259-384-4:3

[5] Walsh, S. L. F., Calandriello, L., Silva, M., Sverzellati, N. "Deep learning for classifying fibrotic lung disease on high-resolution computed tomography: a case-cohort study", The Lancet Respiratory Medicine, 6(11), pp. 837–845, 2018. https://doi.org/10.1016/S2213-2600(18)30286-8

[6] Wang, Z., Xiao, Y., Li, Y., Zhang, J., Lu, F., Hou, M., Liu, X. "Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays", Pattern Recognition, 110, 107613, 2021. https://doi.org/10.1016/j.patcog.2020.107613

[7] Rahman, T., Chowdhury, M. E. H., 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", Applied Sciences, 10(9), 3233, 2020. https://doi.org/10.3390/app10093233

[8] Imran, A. "Training a CNN to detect Pneumonia", DataDrivenInvestor, 12 February 2019. [online] Available at: https://medium.com/datadriveninvestor/training-a-cnn-to-detect-pneumonia-c42a44101deb [Accessed: 29 June 2022]

[9] Chhikara, P., Singh, P., Gupta, P., Bhatia, T. "Deep convolutional neural network with transfer learning for detecting pneumonia on chest X-rays", In: Jain, L. C., Virvou, M., Piuri, V., Balas, V. E. (eds.) Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals, Springer, 2020, pp. 155–168. ISBN 978-981-15-0338-2 https://doi.org/10.1007/978-981-15-0339-9_13

[10] Cilibrasi, R., Vitanyi, P. M. B. "Clustering by compression", IEEE Transactions on Information Theory, 51(4), pp. 1523–1545, 2005. https://doi.org/10.1109/TIT.2005.844059

[11] Li, M., Chen, X., Li, X., Ma, B., Vitanyi, P. M. B. "The similarity metric", IEEE Transactions on Information Theory, 50(12), pp. 3250–3264, 2004. https://doi.org/10.1109/TIT.2004.838101

[12] Kolmogorov, A. N. "Три подхода к определениio понятия "количествоинформации"" (Threeapproachestothedefinitionofthe concept "quantity of information"), Problemy Peredachi Informatsii, 1(1), pp. 3–11, 1965. [online] Available at: https://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=ppi&paperid=68&option_lang=eng [Accessed: 28 November 2022]

[13] Shen, A., Uspensky, V. A., Vereshchagin, N. "Kolmogorov complexity and algorithmic randomness", Mathematical Surveys and Monographs, 2017. ISBN 978-1-4704-3182-2 https://doi.org/10.1090/surv/220

[14] Li, M., Vitányi, P. "An Introduction to Kolmogorov Complexity and Its Applications", Springer, 2008. ISBN 978-1-4899-8445-6 https://doi.org/10.1007/978-0-387-49820-1

[15] Pechnikov, A. A., Nwohiri, A. M. "Kolmogorov Complexity-Based Similarity Measures to Website Classification Problems: Leveraging Normalized Compression Distance", In: International Conference Dedicated to the Memory of Professor Vladimir Zubov, Saint Petersburg, Russia, 2020, pp. 351–358. ISBN 978-3-030-87965-5
https://doi.org/10.1007/978-3-030-87966-2_39

[16] Quispe-Ayala, M. R., Asalde-Alvarez, K., Roman-Gonzalez, A. "Image classification using data compression techniques", In: 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel, Eilat, Israel, 2010, pp. 000349–000353. ISBN 978-1-4244-8681-6
https://doi.org/10.1109/EEEI.2010.5662206

[17] Ranjan, A., Kumar, C., Gupta, R. K., Misra, R. "Transfer Learning Based Approach for Pneumonia Detection Using Customized VGG16 Deep Learning Model", In: Misra, R., Kesswani, N., Rajarajan, M., Veeravalli, B., Patel, A. (eds.) Internet of Things and Connected Technologies, Springer, 2022, pp. 17–28. ISBN 978-3-030-94506-0
https://doi.org/10.1007/978-3-030-94507-7_2

[18] Stephen, O., Sain, M., Maduh, U. J., Jeong, D.-U. "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare", Journal of Healthcare Engineering, 2019, 4180949, 2019.
https://doi.org/10.1155/2019/4180949

[19] Swapna, G., Soman, K. P., Vinayakumar, R. "Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals", Procedia Computer Science, 132, pp. 1253–1262, 2018.
https://doi.org/10.1016/j.procs.2018.05.041

[20] Kundu, R., Das, R., Geem, Z. W., Han, G.-T., Sarkar, R. "Pneumonia detection in chest X-ray images using an ensemble of deep learning models", PLoS ONE, 16(9), e0256630, 2021.
https://doi.org/10.1371/journal.pone.0256630

[21] Manickam, A., Jiang, J., Zhou, Y., Sagar, A., Soundrapandiyan, R., Samuel, R. D. J. "Automated pneumonia detection on chest X- ray images: A deep learning approach with different optimizers and transfer learning architectures", Measurement, 184, 109953, 2021.
https://doi.org/10.1016/j.measurement.2021.109953

[22] Hashmi, M. F., Katiyar, S., Hashmi, A. W., Keskar, A. G. "Pneumonia detection in chest X-ray images using compound scaled deep learning model", Automatika: Journal for Control, Measurement, Electronics, Computing and Communications, 62(3–4), pp. 397–406, 2021.
https://doi.org/10.1080/00051144.2021.1973297

[23] De, S., Chakraborty, B. "Disease Detection System (DDS) Using Machine Learning Technique", In: Jain V., Chatterjee, J. M. (eds.) Machine Learning with Health Care Perspective: Machine Learning and Healthcare, Springer, 2020, pp. 107–132. ISBN 978-3-030-40849-7
https://doi.org/10.1007/978-3-030-40850-3_6

[24] Qin, Z. Z., Ahmed, S., Sarker, M. S., Paul, K., Adel, A. S. S., Naheyan, T., Barrett, R., Banu, S., Creswell, J. "Tuberculosis detection from chest x-rays for triaging in a high tuberculosis-burden setting: an evaluation of five artificial intelligence algorithms", The Lancet Digital Health, 3(9), pp. e543–e554, 2021.
https://doi.org/10.1016/S2589-7500(21)00116-3

[25] Mooney, P. "Chest X-Ray Images (Pneumonia)", Kaggle, 24 March 2018. [online] Available at: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia [Accessed: 30 October 2021]

[26] Keras "Keras: the Python deep learning API: RMSprop", [online] Available at: https://keras.io/api/optimizers/rmsprop [Accessed: 29 October 2022]

[27] Pyton Software Foundation (PyPI) "Opencv-python 4.5.5.64", [computer program] Available at: https://pypi.org/project/opencv-python [Accessed: 27 November 2021]

[28] Shijie, J., Ping, W., Peiyi, J., Siping, H. "Research on data augmentation for image classification based on convolution neural networks", In: 2017 Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 4165–4170. ISBN 978-1-5386-3525-4
https://doi.org/10.1109/CAC.2017.8243510

[29] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. "Gradient-Based Learning Applied to Document Recognition", Proceedings of the IEEE, 86(11), pp. 2278–2324, 1998.
https://doi.org/10.1109/5.726791

[30] Chollet, F. "Deep learning with python", Manning Publications, 2017. ISBN 9781617294433

[31] Brownlee, J. "A Gentle Introduction to the Rectified Linear Unit (ReLU)", Machine Learning Mastery: Making Developers Awesome at Machine Learning, 20 August 2020. [online] Available at: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/ [Accessed: 27 November 2022]

[32] Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., Fergus, R. "Regularization of Neural Networks using DropConnect", Proceedings of the 30th International Conference on Machine Learning, PMLR, 28(3), pp. 1058–1066, 2013.

[33] Hertz, J., Krogh, A., Palmer, R. G. "Introduction to the Theory of Neural Computation", Addison Wesley Pub. Co., 1991. ISBN 0-201-51560-1

[34] Bonnin, R. "Machine Learning for Developers: uplift your regular applications with the power of statistics, analytics, and machine learning", Packt, 2017. ISBN 9781786466969

[35] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. "Dropout: a simple way to prevent neural networks from overfitting", The Journal of Machine Learning Research, 15(56), pp. 1929–1958, 2014.

[36] Ioffe, S., Szegedy, C. "Batch normalization: Accelerating deep network training by reducing internal covariate shift", In: 32nd International Conference on Machine Learning (ICML 2015), Lile, France, 2015, pp. 448–456. ISBN 9781510810587

[37] Liu, Y., Zhou, Y., Wen, S., Tang, C. "A strategy on selecting performance metrics for classifier evaluation", International Journal of Mobile Computing and Multimedia Communications, 6(4), pp. 20–35, 2014.
https://doi.org/10.4018/IJMCMC.2014100102

[38] Ng, W., Minasny, B., de Sousa Mendes, W., Demattê, J. A. M. "The influence of training sample size on the accuracy of deep learning models for the prediction of soil properties with near-infrared spectroscopy data", SOIL, 6(2), pp. 565–578, 2020.
https://doi.org/10.5194/soil-6-565-2020