

Article

Two-Stream Convolutional Neural Networks for Breathing Pattern Classification: Real-Time Monitoring of Respiratory Disease Patients

Jinho Park ¹, Thien Nguyen ¹, Soongho Park ^{1,2}, Brian Hill ¹, Babak Shadgan ^{3,4} and Amir Gandjbakhche ^{1,*}

- ¹ Eunice Kennedy Shriver National Institute of Child Health and Human Development, National Institutes of Health, 49 Convent Dr., Bethesda, MD 20894, USA; jinho.park@nih.gov (J.P.); thien.nguyen4@nih.gov (T.N.); soongho.park@nih.gov (S.P.); brian.hill@nih.gov (B.H.)
- ² National Heart, Lung and Blood Institute, National Institutes of Health, 10 Center Dr., Bethesda, MD 20892, USA
- ³ Implantable Biosensing Laboratory, International Collaboration on Repair Discoveries, Vancouver, BC V5Z 1M9, Canada; babak.shadgan@ubc.ca
- ⁴ Department of Pathology & Laboratory Medicine, University of British Columbia, Vancouver, BC V6T 1Z7, Canada
- * Correspondence: gandjbaa@mail.nih.gov

Abstract: A two-stream convolutional neural network (TCNN) for breathing pattern classification has been devised for the continuous monitoring of patients with infectious respiratory diseases. The TCNN consists of a convolutional neural network (CNN)-based autoencoder and classifier. The encoder of the autoencoder generates deep compressed feature maps, which contain the most important information constituting data. These maps are concatenated with feature maps generated by the classifier to classify breathing patterns. The TCNN, single-stream CNN (SCNN), and state-of-the-art classification models were applied to classify four breathing patterns: normal, slow, rapid, and breath holding. The input data consisted of chest tissue hemodynamic responses measured using a wearable near-infrared spectroscopy device on 14 healthy adult participants. Among the classification models evaluated, random forest had the lowest classification accuracy at 88.49%, while the TCNN achieved the highest classification accuracy at 94.63%. In addition, the proposed TCNN performed 2.6% better in terms of classification accuracy than an SCNN (without an autoencoder). Moreover, the TCNN mitigates the issue of declining learning performance with increasing network depth, as observed in the SCNN model. These results prove the robustness of the TCNN in classifying breathing patterns despite using a significantly smaller number of parameters and computations compared to state-of-the-art classification models.

Keywords: COVID-19; deep learning; convolutional neural network; respiratory disease; NIRS; wearable device



Citation: Park, J.; Nguyen, T.; Park, S.; Hill, B.; Shadgan, B.; Gandjbakhche, A. Two-Stream Convolutional Neural Networks for Breathing Pattern Classification: Real-Time Monitoring of Respiratory Disease Patients. *Bioengineering* **2024**, *11*, 709. <https://doi.org/10.3390/bioengineering11070709>

Academic Editor: Luca Mesin

Received: 29 May 2024

Revised: 26 June 2024

Accepted: 11 July 2024

Published: 12 July 2024



Copyright: © 2024 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

In 2019, the respiratory illness Coronavirus Disease 2019 (COVID-19) caused by SARS-CoV-2 started spreading globally at an unprecedented rate. The virus has spread to over 228 countries and, as of May 2024, has affected approximately 775 million people, resulting in over 7.05 million deaths [1]. Common symptoms of COVID-19 include cough, breathing complications, wheezing, and tachypnea, which often accompany respiratory distress, fever, headache, fatigue, hypoxia, and lethargy [2–6]. Various methods have been used to monitor infected patients, such as CT scans [7–9], X-rays [7,10], thermal cameras [11], depth cameras [4], ultrasound technology [12], and radar [13–17]. CT scans and X-ray images provide high precision and resolutions, but they are expensive and expose the patient to harmful radiation. Ultrasound technology is noninvasive and harmless but has limited sensitivity. Changes in chest displacement can be observed using depth cameras and radar.

However, these methods are constrained by a person's movement and position making it difficult to observe the displacement accurately. To overcome these issues, various wearable medical devices are being developed that can monitor patients in real time [18]. Among these, near-infrared spectroscopy (NIRS) is gaining attention for its simplicity, low cost, and ability to noninvasively monitor tissue hemodynamics [19].

Common symptoms experienced by patients with respiratory infectious diseases include difficulties in breathing and changes in respiration rate, which are either rapid and shallow (tachypnea) or slow and deep [2,4,20,21]. Therefore, the respiration rate is an important parameter to be monitored [20–24], and various sensors and devices have been developed to measure human respiration rates and patterns. In particular, flexible and wearable sensors have garnered significant interest in terms of respiration monitoring [25–27]. Many of these sensors also measure the peripheral arterial blood oxygen saturation (SpO₂), which has been used as a key marker for the monitoring of the progression of respiratory diseases [28]. Using these sensors to monitor SpO₂ can be convenient but may result in inaccurate readings due to factors like body movement, reduced blood circulation, the skin thickness, and the skin color [29]. While the SpO₂ levels are a measure of the systemic arterial blood oxygen saturation, NIRS devices can measure the more localized tissue oxygen saturation (StO₂) within the microvasculature of tissue [30], which represents an advantage of NIRS devices over pulse oximeters [31]. Studies have shown that NIRS devices can detect the effects of hypoxia better than standard pulse oximeters [32]. Therefore, in this study, we use an NIRS device to evaluate the hemodynamic responses and monitor the respiratory patterns in patients with infectious respiratory diseases.

Infectious respiratory diseases are often transmitted through droplets, person-to-person contact, or vectors [3,33–35]. Therefore, diagnosis through direct contact with infected individuals increases the risk of infection for both contacts and healthcare workers [36]. To minimize the occurrence and prevent the spread of highly contagious respiratory diseases from infected individuals to those around them, it is crucial to detect infections early, enforce self-isolation, and ensure appropriate treatment for a swift recovery. To achieve this, non-contact monitoring methods that can check the patient's condition in real time without direct contact are necessary [37–39]. An Internet of Things (IoT)-based monitoring system serves as a prime example. Shafi et al. proposed a system utilizing wearable devices to measure a patient's temperature, heart rate, and oxygen saturation, transmitting and storing these data in an IoT cloud. Medical staff can access the cloud data to monitor the patient's health status without direct contact [40]. While this system effectively monitors patients' conditions without direct contact, clinical decisions must be made by medical staff. However, medical decision-making poses challenges, as healthcare providers face time and physical limitations in diagnosing extensive patient data. Therefore, an intelligent method for the automatic detection of abnormalities from monitored signals is needed.

Deep learning and machine learning algorithms have been applied to various medical fields and clinical decisions, including DNA classification, brain tumor diagnosis, eye diseases, and malaria diagnosis [41,42]. Commonly used algorithms include support vector machines (SVM), extreme gradient boosting (XGBoost), artificial neural networks, and convolutional neural networks (CNN) [43]. For example, Cho et al. utilized a thermal camera to monitor breathing patterns by analyzing temperature fluctuations around the nose regions of individuals. Subsequently, a CNN-based algorithm was employed to automatically categorize psychological stress levels [11]. Shah et al. introduced a technique to detect distinctive patterns associated with COVID-19 from CT scan images utilizing a deep learning algorithm [44]. Hong et al. proposed a method that utilizes mm-wave sensors to measure the displacement of the patient's chest height and extracts a feature vector using CNN algorithms. The extracted feature vector is compared with pre-learned feature vectors stored in a database to classify it according to the most similar feature vector [45]. Kim et al. developed an embroidered textile sensor that measures electrical

impedance changes depending on the intensity of compression and chest movement. The measured electrical impedance changes were classified into respiratory patterns using a 1D-CNN. [46]. Conway et al. used capnography to determine 'breath' versus 'no breath' using a CNN based on changes in the CO₂ concentration in a patient's breathing [47]. These studies have shown encouraging results in classifying individuals suffering from respiratory diseases. However, long-term data collection might cause discomfort to patients, making it unsuitable for continuous monitoring.

In our previous study, we used an NIRS device to measure chest tissue hemodynamic responses and classified simulated breathing patterns using the oxyhemoglobin (O₂Hb) signal [48]. The simulated breathing patterns included rest, rapid/shallow, and loaded breathing. From the O₂Hb signals, features including the amplitude, depth, and interval were extracted, and the breathing patterns were classified using the random forest classification method, achieving accuracy of 87%. However, this method is unsuitable for real-time monitoring due to the need for post-processing and feature extraction. Therefore, in another study, we developed a deep learning algorithm based on a 1D-CNN to classify the three breathing patterns, which resulted in classification accuracy of 91.28% [49]. The 1D-CNN model is composed of a single-stream structure. Hence, it faces challenges in parameter learning and optimization as the number of layers increases. As a result, its performance may not improve or may even deteriorate beyond a certain number of layer parameters [50–52].

To overcome the issues associated with a single-stream CNN (SCNN), this study proposes a two-stream CNN (TCNN) for breathing pattern classification. The TCNN consists of an autoencoder for the generation of deep compressed feature maps from breathing signals and a classifier module for classification. The deep compressed feature maps are concatenated with feature maps generated by the CNN-based classifier module to ultimately assist in classifying breathing patterns. We hypothesize that the combination of deep compressed feature maps generated by the autoencoder and feature maps created by the CNN-based classifier module will enhance the classification performance and overcome the parameter learning and optimization issues observed in an SCNN.

2. Materials and Methods

2.1. NIRS Device

The NIRS device consists of one light source with LEDs emitting light at three wavelengths (730 nm, 800 nm, and 850 nm) and two photodetectors, located at 3 cm and 4 cm from the light source (prototype provided by Hamamatsu Photonics K.K., Shizuoka, Japan) (Figure 1). The light source emits a constant amount of light into the tissue, which undergoes absorption and scattering. The backscattered light is detected by the photodetectors, which is then used to calculate changes in O₂Hb, deoxyhemoglobin (HHb), and total hemoglobin (THb) at each distance. The absolute tissue oxygen saturation (StO₂) is also measured. The NIRS device was connected to a mobile phone (Pixel XL) for power supply and data recording. A mobile application on the phone converted the raw data measured by the NIRS device into hemodynamic responses, which included two sets of O₂Hb, HHb, and THb from the two photodetector separations and one set of StO₂. The data were collected at a sampling rate of 120 Hz, providing high-resolution data.

2.2. Breathing Class Definition and Data Collection

To facilitate the non-contact monitoring and classification of breathing patterns for the early detection and monitoring of respiratory disease patients, we define 4 classes of breathing patterns. Patients with respiratory diseases exhibit breathing rates that are faster or slower than those of healthy individuals [2,4,20,21]. In this study, we defined the following four breathing patterns.

- Class 1—normal breathing: Participants breathed normally through their mouth or nose while in a relaxed position (Figure 2a).

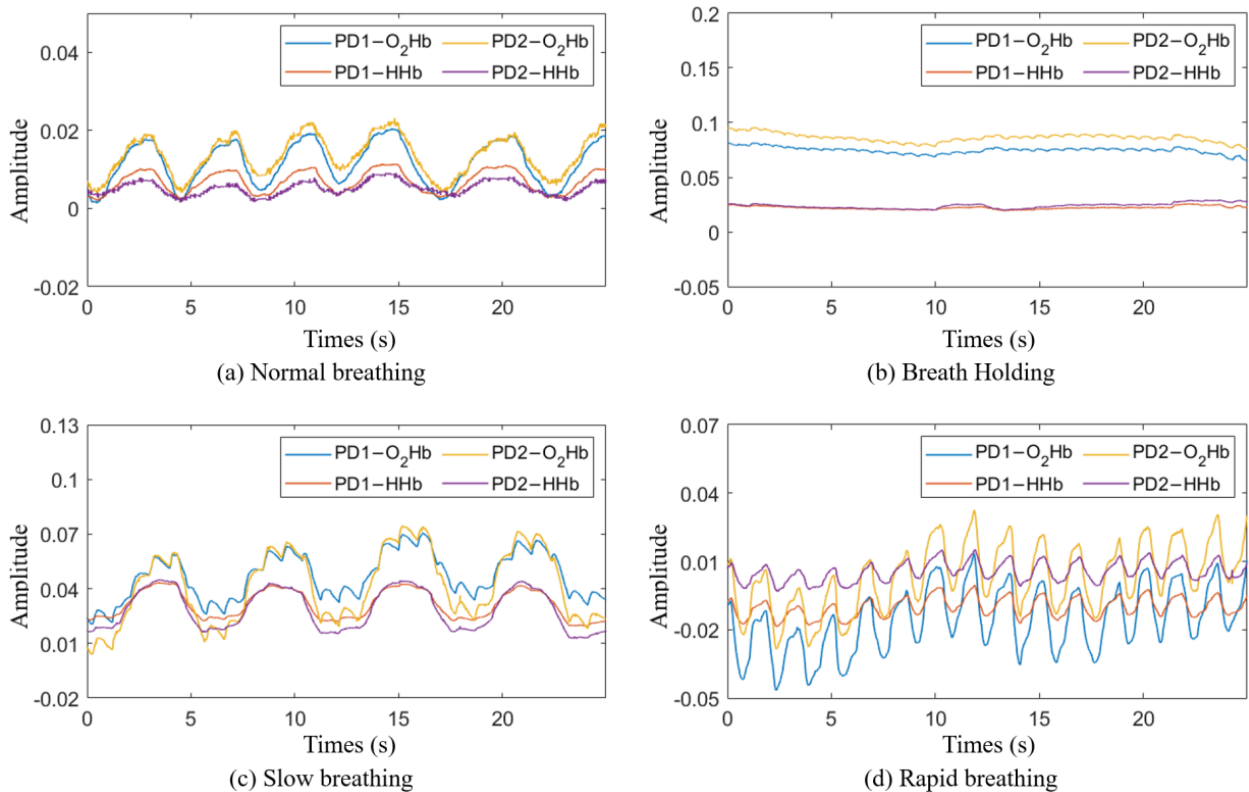
- Class 2—breath holding: Participants exhaled and held their breath for as long as they could (Figure 2b).
- Class 3—slow breathing: Participants breathed at a rate of 10 times per minute for two minutes (Figure 2c).
- Class 4—rapid breathing: Participants breathed at a rate of 30 times per minute for two minutes (Figure 2d).



(a) Data recording system

(b) NIRS device

Figure 1. Data recording system and NIRS device. (a) The signals measured by the NIRS device are converted into O_2Hb , HHb , THb , and StO_2 in real time and recorded on the connected mobile phone. (b) The NIRS device consists of one light source and two photodetectors.



(a) Normal breathing

(b) Breath Holding

(c) Slow breathing

(d) Rapid breathing

Figure 2. Breathing signal in the time domain: (a) normal breathing; (b) breath holding; (c) slow breathing; (d) rapid breathing.

The experiment for data collection was conducted at an outpatient clinic at the National Institutes of Health (NIH) Clinical Center. During data collection, the NIRS device was attached to the participants' chests using medical-grade, double-layer tape. To acquire data on normal breathing, each participant comfortably sat in a chair in a resting position and breathed normally through their mouth or nose for 5 min. The participants then performed a breath holding task where they were asked to exhale and hold their breath for as long as possible, up to two minutes. After resting for two minutes, the task was repeated twice for a total of three repetitions. The participants then rested for five minutes before starting the slow breathing task. During slow breathing, participants took ten breaths per minute for two minutes, rested for two minutes, and then repeated. The slow breathing task ended with a two-minute rest followed by the rapid breathing task, where the participants breathed thirty times per minute for two minutes. This task was also repeated following a two-minute rest period. Figure 2 shows the breathing signals for each class obtained through the NIRS device.

2.3. Participants

Twenty-six healthy adult participants (mean age = 34.7 ± 15.7 years old, 18 females) were enrolled in the study. The study enrollment and procedure followed the protocol of a clinical trial and was approved by the Institutional Review Board of the Eunice Kennedy Shriver National Institute of Child Health and Human Development (NICHD protocol #21CH0028) on 31 August 2021. Data from 12 participants were excluded due to motion artifacts and/or low signal-to-noise ratios in one or more tasks. Data from 14 participants who had good signal-to-noise ratios for all four tasks were retained for further analysis.

2.4. Data Preprocessing

The data recorded by the NIRS device included impulse noise caused by voltage transients during measurement. To remove the impulse noise, the first derivative of the signal was calculated, and the mean and standard deviation (STD) of the first derivative values were computed. Any signals exceeding the 90% confidence interval were considered noise and removed. Figure 3 shows the signal acquired by the NIRS device (a) and the signal after impulse noise removal (b).

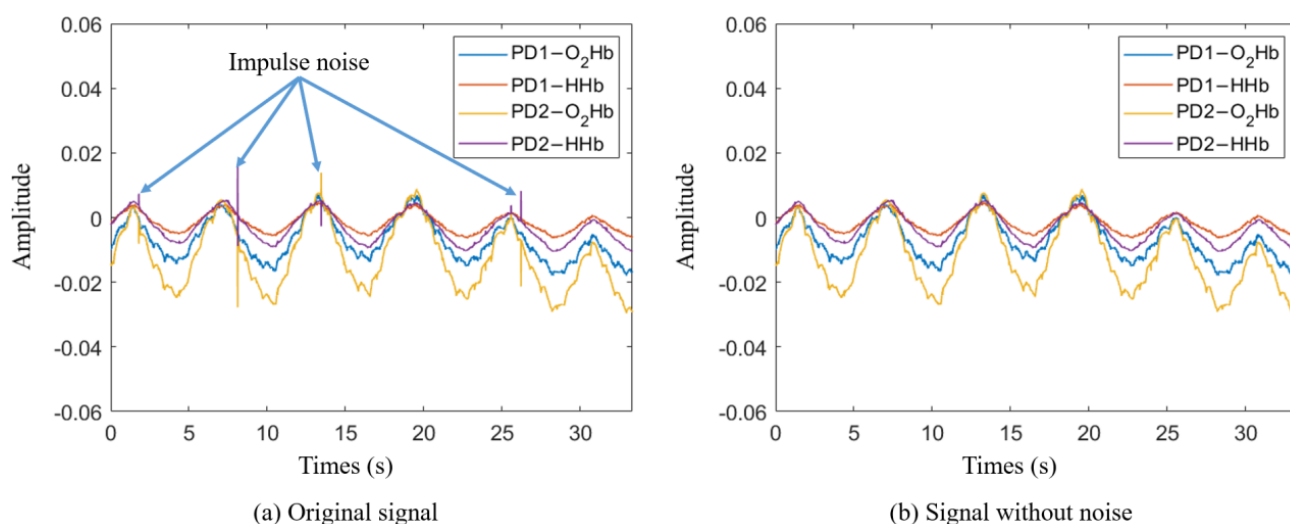


Figure 3. Impulse noise removal: (a) original signal from NIRS device, (b) signal with impulse noise removed.

2.5. Breathing Pattern Classification Model

The proposed TCNN model consists of a CNN-based autoencoder for the generation of deep compressed features and a CNN-based classifier module for the classification of breathing patterns. The autoencoder comprises an encoder that compresses the input

signal and a decoder that reconstructs the high-dimensional features back into the original signal. The high-dimensional features generated from the encoder are trained to detect the most important features that make up the signal. Therefore, we assume that the features generated by the encoder can be helpful in classifying breathing patterns. To test this hypothesis, we designed a breathing pattern classification model, as shown in Figure 4a. The deep compressed features generated by the autoencoder are integrated into the feature model created in the classifier module to classify breathing patterns.

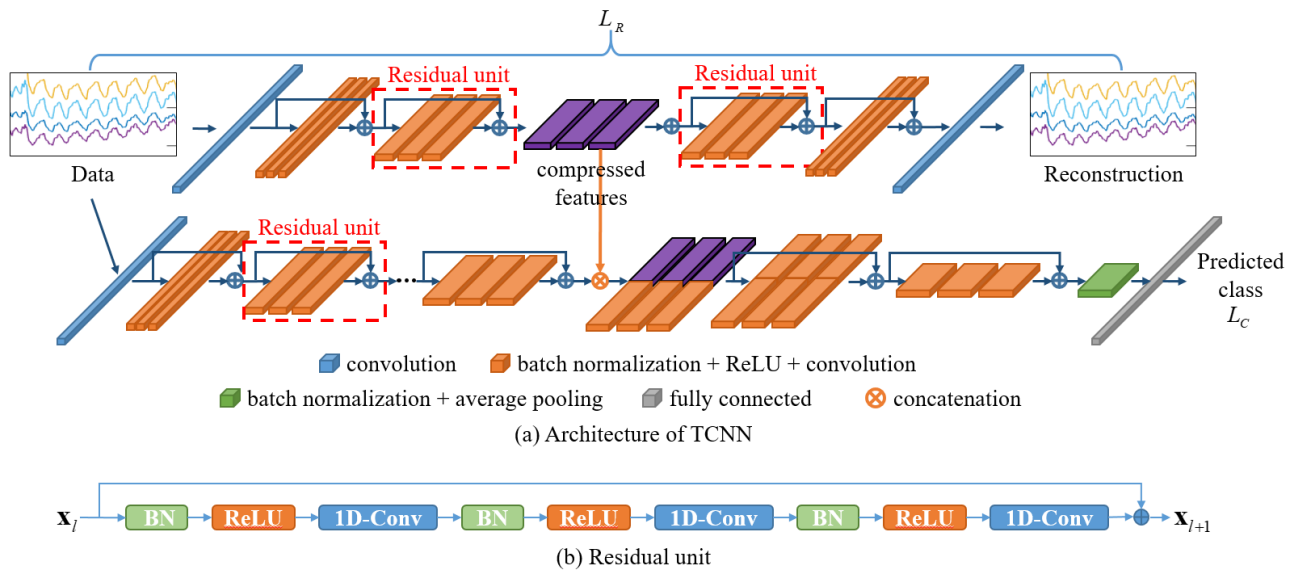


Figure 4. Architecture of TCNN. (a) The autoencoder network is used to generate compressed features from the input data; then, the compressed features are combined with deep features extracted from the classification network. (b) The structure of a residual unit.

The autoencoder and classifier module were designed by modifying the pre-activation residual model (Pre-ResNet), which is an algorithm developed for the classification of 2D images [53]. Typically, the parameters in a deep learning model are optimized by computing the gradient through the differentiation of the error between the predicted values from the model and the actual values and then performing the backpropagation of this gradient [54,55]. However, as the depth of the model increases, the vanishing gradient problem occurs because the transmitted gradient values decrease, making training difficult and leading to overfitting [50–52]. To address these issues, the Pre-ResNet model introduces residual learning by structuring layers in building block units and adding shortcut connections between blocks [53]. Deep learning models composed of residual units mitigate gradient loss during backpropagation by skipping one or more layers. Figure 4b shows a residual unit, which consists of a convolution layer, an activation function (ReLU) [56], and batch normalization (BN) [57].

To apply the residual unit to the 1D signals acquired from the NIRS device, 1D convolutional layers were used instead of 2D convolutional layers, and the size of the convolutional kernels was adjusted to fit the input signal. Detailed information on the network structure and parameters is presented in Tables 1 and 2, with the BN and ReLU functions omitted. Each convolution block within the same group has the same kernel size and number of kernels. The block structure indicates the kernel size and number of kernels included in each residual block. The block number indicates the number of residual units composing each group. The input size and output size represent the sizes of the input and output features at each stage. The feature map generated in Stage 3 of the encoder is combined with the Stage 3 output features of the classifier module and used to train the classification model.

Table 1. Autoencoder architectures.

	Group Name	Input Size Output Size	Block Structure (Kernel Size, Number)	Block Number
Encoder	Stage 0	1×384 1×96	$1 \times 7, 16$	1
	Stage 1	1×96 1×32	$1 \times 7, 16$	1
	Stage 2	1×32 1×32	$1 \times 1, 16$ $1 \times 3, 16$ $1 \times 1, 64$	5
	Stage 3	1×32 1×16	$1 \times 1, 32$ $1 \times 3, 32$ $1 \times 1, 128$	5
Decoder	Stage 0	1×16 1×32	$1 \times 1, 128$ $1 \times 3, 32$ $1 \times 1, 32$	5
	Stage 1	1×32 1×32	$1 \times 1, 64$ $1 \times 3, 16$ $1 \times 1, 16$	5
	Stage 2	1×32 1×96	$1 \times 5, 16$	1
	Stage 3	1×96 1×384	$1 \times 7, 4$	1

Table 2. Classifier module architectures.

	Group Name	Input Size Output Size	Block Structure (Kernel Size, Number)	Block Number
	Stage 0	1×384 1×96	$1 \times 7, 16$	1
	Stage 1	1×96 1×32	$1 \times 7, 16$	1
	Stage 2	1×32 1×32	$1 \times 1, 16$ $1 \times 3, 16$ $1 \times 1, 64$	22
	Stage 3	1×32 1×16	$1 \times 1, 32$ $1 \times 3, 32$ $1 \times 1, 128$	11
	Concatenation	$1 \times 16 \rightarrow 1 \times 32$	-	-
	Stage 4	1×32 1×16	$1 \times 1, 32$ $1 \times 3, 32$ $1 \times 1, 128$	11
	Stage 5	1×16 1×8	$1 \times 1, 64$ $1 \times 3, 64$ $1 \times 1, 256$	22
	Average pooling	1×8 1×256	1×8	1
	Fully connected layer	1×256 Class number	-	1

Stride values of 4 and 3 were used to downsample the feature maps in Stage 0 and Stage 1 of the classifier module and the encoder, respectively. In the remaining stages,

except in Stage 2, a stride value of 2 was used in the first convolutional layer of the first residual unit to downsample the feature maps to half their size, and the dimensions of the feature map were expanded twice. As the depth of the network increased, more complex and comprehensive features were extracted.

In the decoder, transposed convolution was used in the first convolutional layer of each stage's first residual unit to upsample the feature maps to twice their size and halve the dimensionality of the feature maps [58,59]. Stride values of 3 and 4 were used in Stage 2 and Stage 3 to increase the size of the feature maps threefold and fourfold, respectively. As a result, the generated feature maps resembled the input signal more closely as the depth of the network increased.

The total objective function of the TCNN is defined as

$$L_{total} = L_R + \alpha L_C \quad (1)$$

where L_R and L_C represent the reconstruction loss of the autoencoder and the cross-entropy loss of the classifier module, respectively. L_R is defined as follows:

$$L_R = \frac{1}{N} \sum_{i=1}^N (X_i - \hat{X}_i)^2 \quad (2)$$

where X_i and \hat{X}_i represent the input signal and the reconstructed signal, respectively, and N is the number of samples.

2.6. Dataset

Among the acquired signals, the changes in O₂Hb and HHb were used to classify breathing patterns. THb, which is the sum of O₂Hb and HHb, and StO₂, which represents the relationship between O₂Hb and THb, were excluded because they are directly dependent on changes in O₂Hb and HHb. Signals were cropped into intervals of 384 data points, corresponding to approximately 3.2 s of measurement time. As a result, the dataset acquired from the study consisted of a total of 113,456 samples, with 28,364 samples per class. Eleven out of 14 participants were randomly selected to have their data (22,286 samples per class) used for the training of the classification model, while the dataset obtained from the remaining 3 participants (6078 samples per class) was used for model testing.

2.7. Data Augmentation

The performance of deep-learning-based algorithms heavily relies on having an extensive training dataset to prevent overfitting. To enhance the size and quality of our training dataset, we applied data augmentation. The main purpose of data augmentation is to increase the quantity and diversity of the training data, and this technique is widely used in various fields, such as computer vision, image classification, and natural language processing [60–63]. To determine the data augmentation methods, we observed changes in the data during each participant's breathing. The participants exhibited similar respiratory rates while performing respiration corresponding to each class. On the other hand, the amplitude, shape, and direct current (DC) offset of the signals varied among the participants. Hence, a random amplitude scale, horizontal flipping with 50% probability, and DC offset were applied as data augmentation methods to maintain the signal period and increase the diversity of the signal amplitude, shape, and DC values. Figure 5 shows an example of data augmentation.

Each data augmentation method was applied before the signal was input into the model and is defined as follows.

- DC offset: performed by adding a DC value β to the original signal.

$$f_{DCoffset}(I_s) = I_s + \beta \quad (3)$$

where I represents the sample data input to the model during model training, and the subscript s denotes the data source. Let D and d represent the training dataset and the data sample belonging to the training dataset, respectively; then, the range of β is defined as follows.

$$\beta \in \left[\min_{d_s^i \in D_s} \left(\min(d_s^i) \right) - \min(I_s), \max_{d_s^j \in D_s} \left(\max(d_s^j) \right) - \max(I_s) \right] \quad (4)$$

where $i, j \in [1, 2, \dots, N]$ represent the indices of the data samples, and N denotes the total number of samples in the training dataset.

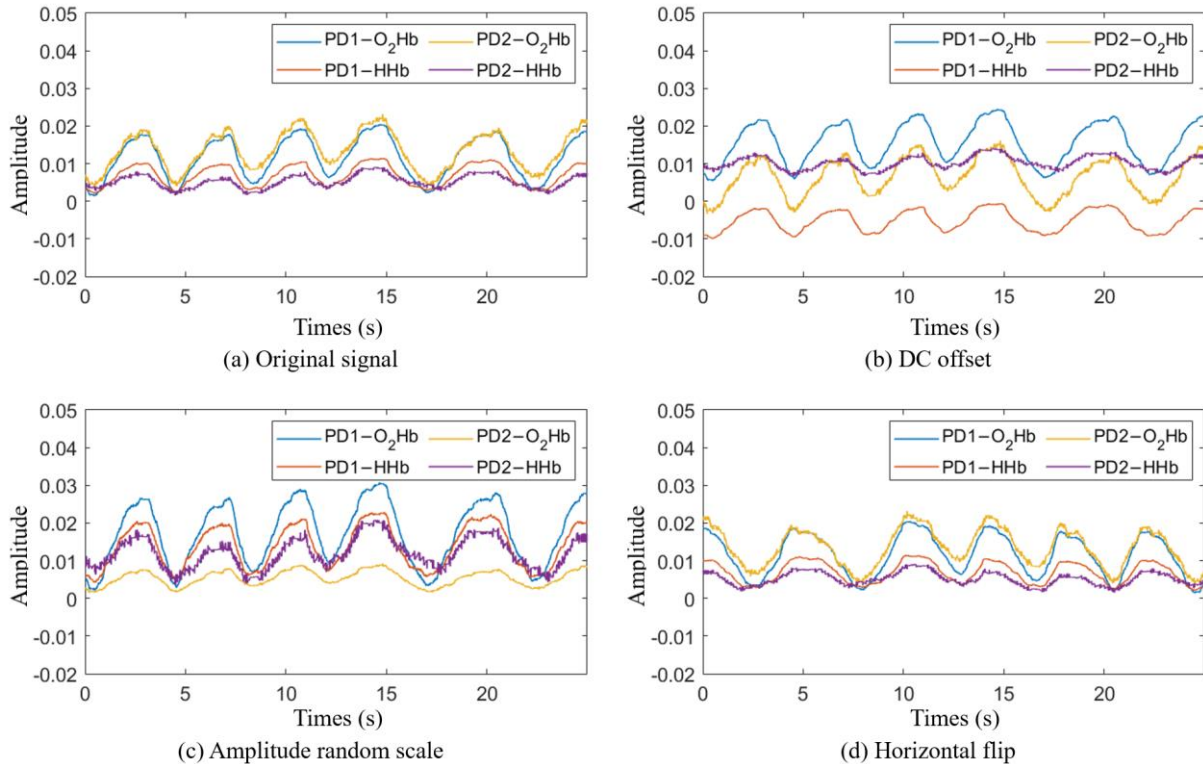


Figure 5. Example of the data augmentation methods. (a) The original signals recorded using the mobile phone. (b) The result of adding DC offset values of 0.004, -0.012 , -0.007 , and 0.005 to the PD1-O₂Hb, PD1-HHb, PD2-O₂Hb, and PD2-HHb signals, respectively. (c) The result of multiplying the PD1-O₂Hb, PD1-HHb, PD2-O₂Hb, and PD2-HHb signals by scale values of 1.5, 2, 0.4, and 2.3, respectively. (d) The result of a horizontal flip occurring with a 50% probability.

- Amplitude random scaling: involves reducing or increasing the amplitude of the data by multiplying the scaling value γ with the original data.

$$f_{Scale}(I_s) = I_s \times \gamma \quad (5)$$

γ is determined by the smallest and largest amplitude values among the training dataset.

$$\gamma \in \left[\min_{d_s^i \in D_s} \left(\max(d_s^i) - \min(d_s^i) \right), \max_{d_s^j \in D_s} \left(\max(d_s^j) - \min(d_s^j) \right) \right] \quad (6)$$

- Horizontal flip $f_{Flip}(I_s)$: performed by flipping the original data horizontally and occurs with a probability of 50%. Horizontal flipping is a simple technique that can effectively improve a model's learning performance.

In this experiment, the augmented data $f_{Aug}(I_s)$ are finally defined as follows:

$$f_{Aug}(I_s) = f_{Flip}\left(f_{DC\ offset}(f_{Scale}(I_s))\right) \quad (7)$$

3. Results

3.1. TCNN with and without Autoencoder

We used a computer equipped with a CPU i5-12600k, a GPU RTX-3090, and 64 GB DDR4 RAM for the experiment. We first pre-trained an autoencoder with the same structure as in Figure 4. The weights of the pre-trained autoencoder were used as the initial values of the autoencoder of the TCNN. The pre-training of the autoencoder and the training of the TCNN were conducted using stochastic gradient descent (SGD) for 140 epochs, with momentum of 0.9 and a batch size of 128. The initial learning rate for the pre-training of the autoencoder was set to 0.1, while the initial learning rate for the training of the TCNN was set to 0.01. The learning rate was divided by 10 every 30 epochs. In this experiment, the hyper-parameter α was set to 10.

Table 3 presents the experimental results to assess the impact of the compressed feature maps generated from the autoencoder on the classification performance. Each model was tested five times. The SCNN (without an autoencoder) is a model derived from the TCNN excluding the autoencoder. The SCNN has no concatenation group, and the output size of Stage 3 is 1×32 . The classification accuracy is defined by the following formula.

$$Accuracy(\%) = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \times 100(\%). \quad (8)$$

Table 3. Classification performance compared with an SCNN (without autoencoder) and the data augmentation effect.

Method	Mean Accuracy (%)	STD	Best Accuracy (%)
TCNN (201 layers)	94.00	0.449	94.63
TCNN (without data augmentation) (201 layers)	88.51	0.337	89.04
SCNN (201 layers)	91.50	0.458	92.04
SCNN (264 layers)	91.13	0.443	91.74

The experimental results showed that the TCNN models had better performance than the SCNN models. In addition, the SCNN with more layers (264 layers) showed slightly worse performance than the SCNN with less layers (201 layers). Furthermore, applying data augmentation techniques during the training process of the TCNN model led to an approximately 5% improvement in the classification accuracy compared to not applying data augmentation.

3.2. Confusion Matrix

Figure 6 shows the normalized confusion matrix for the breathing pattern classification results. The ground truth represents the labels for the data samples, while the predicted labels indicate the results predicted by the model. Therefore, a larger value in the diagonal matrix indicates that the predicted label has been correctly classified. It can be observed that the diagonal matrix is prominent in all three confusion matrices. This indicates that the majority of the data samples have been correctly classified into their respective classes. It is worth noting that in all three confusion matrices, the normal class has the lowest classification accuracy.

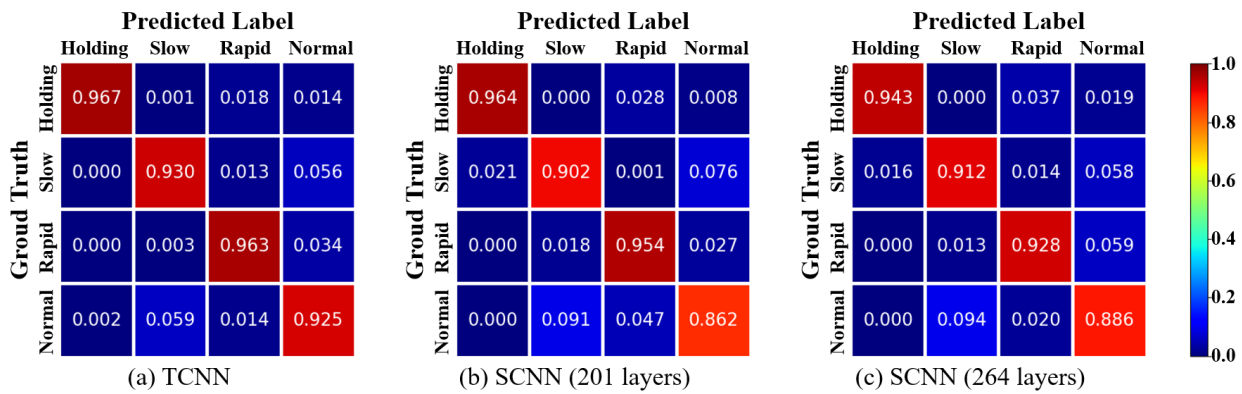


Figure 6. Confusion matrix from predicted results of the TCNN and SCNN (without autoencoder).

3.3. Performance of Well-Known Classification Algorithms

Table 4 shows a performance comparison of several classification methods. To compare the classification performance of the proposed method with that of other state-of-the-art classification algorithms, we used random forest [64], EfficientNetV2-M [65], PyramidNet [66], a modified 1D-PreResNet [49], and a coarse-to-fine CNN (CF-CNN) [67]. The random forest algorithm consists of multiple decision trees and is an ensemble learning method used for the training and classification of data. For our experiment, we employed a model consisting of 100 decision trees. To train the random forest, we utilized the average amplitude, depth, and interval of the NIRS signals [48]. EfficientNetV2-M and PyramidNet are CNN-based algorithms developed for image classification, renowned for their exceptional performance and stability and widely utilized across various domains [68–71]. PyramidNet consists of 272 layers, and the widening step factor value for increasing feature map dimensions is set to 200. CF-CNN consists of a main network for fine classification and several coarse sub-networks with a hierarchical structure for coarse classification. To train a new hierarchical structure for coarse classification, we generated hierarchical group labels using the group label augmentation method [67]. The coarse network of CF-CNN consists of a downsized version of the base network that constitutes the main network. This model is also proposed to overcome the limitations of learning in single-stream networks. For comparison, the 2D convolutional layers of each network were replaced with 1D convolutional layers. For the evaluation of the CF-CNN model, we chose PyramidNet as the base network. The number of layers in each of the Coarse 1, Coarse 2, and main networks was set to 26, 52, and 272, respectively. For Coarse 1 and Coarse 2 classification, the number of group labels was set to 2 and 3, respectively.

Table 4. Classification performance compared with state-of-the-art methods.

Method	Number of Parameters	FLOP (for 1 Sample)	Mean Accuracy (%)	STD	Best Accuracy (%)
Random Forest [64]	-	-	88.10	0.321	88.49
1D Pre-ResNet (200 layers) [49]	1.36 M	14.8 M	90.98	0.428	91.54
EfficientNetV2-M [65]	52 M	61.6 M	93.96	0.459	94.45
PyramidNet [66]	17 M	198 M	89.74	0.412	90.17
CF-CNN [67]	29.7 M	244 M	91.51	0.406	92.01
Proposed Method	1.53 M	16.4 M	94.00	0.449	94.63

Each model was tested five times, and, for each test, the model was trained from scratch. To train the 1D Pre-ResNet, EfficientNetV2-M, PyramidNet, and CF-CNN, the input data were downsampled using the bilinear interpolation method to match the input

size of each model at the input layer. While our proposed method had a larger number of parameters and floating-point operations (FLOP) compared to the 1D Pre-ResNet, it achieved the highest classification performance among the compared methods.

4. Discussion

The aim of our study was to develop a breathing pattern classification method for the real-time monitoring of patients with infectious respiratory diseases such as COVID-19. First, we defined four common breathing patterns observed in patients with respiratory diseases: normal, breath holding, slow, and rapid. Then, we collected data from participants using a wearable NIRS device. Second, we proposed the TCNN, consisting of an autoencoder and a classifier module, to classify breathing patterns. We hypothesized that the deep compressed features generated by the encoder of the autoencoder would contain information about the composition of the signal, which could potentially help to improve the classification performance. To validate this hypothesis, we compared the classification performance of the proposed TCNN with that of an SCNN model. Comparing the breathing pattern classification accuracy between the two SCNN models with 201 layers and 264 layers, we observed that the SCNN with less layers exhibited higher classification accuracy. This demonstrates the challenge of parameter learning becoming more difficult as the number of parameters increases. Comparing the classification performance of the TCNN and SCNN, we observed that the TCNN achieved higher classification performance. This result suggests that the deep compressed features generated by the encoder of the autoencoder contribute to the improvement in classification performance. Furthermore, as training is performed on networks with different objectives, the diversity of the feature maps is enhanced, and the burden on parameter optimization is reduced.

The results of the TCNN (without data augmentation) showed significantly lower classification accuracy compared to the TCNN, primarily due to insufficient training data. Deep learning algorithms can prevent overfitting and enhance the model's generalization when trained with diverse and abundant data. To improve the model's generalization performance, we applied three data augmentation methods: DC offset, amplitude random scale, and horizontal flip. As a result, the classification performance was improved by approximately 5%. This outcome suggests that data augmentation effectively enhanced the model while preserving the essential features necessary for breathing pattern classification.

For all three models in Figure 6, the classification accuracy for normal breathing was the lowest. One plausible explanation is that the most distinguishing feature in breathing pattern classification is the respiratory rate. In some instances, a normal breathing rate can be confused with slow and rapid breathing.

Among all classification models used in this study, the proposed TCNN model exhibited the best classification accuracy. Moreover, the TCNN model used significantly a smaller number of parameters and FLOP, which are directly related to the hardware storage space and computational load. FLOP represents the number of floating-point operations required to execute each classification model. Therefore, a lower FLOP count indicates a shorter classification time. Although the TCNN shows slightly more parameters and FLOP than the 1D-PreResNet, which has the fewest parameters and FLOP, it achieves higher classification accuracy. When compared to the performance of the EfficientNetV2-M, PyramidNet, and CF-CNN models, the proposed method demonstrates the best performance with significantly fewer parameters and FLOP. The classification accuracy when using the random forest algorithm was the lowest among the compared methods. Traditional machine learning methods like random forest perform well when the input features are diverse and contain distinct information. However, creating diverse feature extractors manually is very challenging for humans. On the other hand, deep learning algorithms show better performance than traditional machine learning methods because they automatically generate diverse and meaningful features during the model learning process.

Future studies could expand on our results by collecting signals from individuals and patients with respiratory diseases. We are considering the application of noise removal

techniques during the data preprocessing stage to enhance the classification performance. Additionally, we are exploring the integration of sensors such as temperature sensors, accelerometers, and NIRS devices to obtain diverse and meaningful signals.

5. Conclusions

This study proposed a two-stream convolutional neural network (TCNN) model for the classification of breathing patterns to facilitate the evaluation and monitoring of patients with infectious respiratory diseases. This method utilizes the changes in oxyhemoglobin and deoxyhemoglobin measured using a wearable NIRS device. The proposed TCNN consists of an autoencoder and a classifier module. The deep compressed feature maps generated in the encoder of the autoencoder are combined with the feature maps generated in the classifier module to classify breathing patterns. The compressed features generated by the encoder contain important information that constitutes the signal, unlike the features generated by the classifier. Therefore, compressed features help to improve the breathing pattern classification performance because they provide additional information for the classification of breathing patterns. The proposed method alleviates the issue of decreasing parameter learning capabilities as the depth of a typical single-stream network model increases. For the breathing pattern classification experiment, we defined four common breathing patterns observed in infectious respiratory disease patients: normal breathing, breath holding, slow breathing, and rapid breathing. We then conducted the classification experiment. As a result, we achieved classification accuracy of 94.63%. The proposed method demonstrates its potential for the remote and real-time monitoring of respiratory diseases such as COVID-19, pneumonia, and influenza.

Author Contributions: Conceptualization, J.P., T.N., B.H. and A.G.; Methodology, J.P., S.P., T.N. and A.G.; Software, J.P.; Validation, J.P. and A.G.; Formal Analysis, J.P.; Investigation, J.P. and A.G.; Resources, A.G.; Data Curation, T.N. and J.P.; Writing—Original Draft Preparation, J.P.; Writing—Review and Editing, T.N., S.P., J.P., B.H., B.S. and A.G.; Visualization, J.P.; Supervision, A.G.; Project Administration, T.N. and A.G.; Funding Acquisition, A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Intramural Research Program at the Eunice Kennedy Shriver National Institute of Child Health and Human Development, National Institutes of Health, grant number [Z01-HD000261].

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Institutional Review Board of the Eunice Kennedy Shriver National Institute of Child Health and Human Development on 31 August 2021 (NICHD protocol #21CH0028).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data supporting the reported results available at <https://github.com/dkskzmfps/TCNNforBP>, accessed on 11 July 2024.

Acknowledgments: We acknowledge the Hamamatsu Photonics company for providing their wearable NIRS prototype to conduct the experiments.

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

References

1. WHO Coronavirus (COVID-19) Dashboard. Available online: <https://covid19.who.int/> (accessed on 5 December 2023).
2. Pan, L.; Mu, M.; Yang, P.; Sun, Y.; Wang, R.; Yan, J.; Li, P.; Hu, B.; Wang, J.; Hu, C. Clinical characteristics of COVID-19 patients with digestive symptoms in Hubei, China: A descriptive, cross-sectional, multicenter study. *Am. J. Gastroenterol.* **2020**, *115*, 766–773. [[CrossRef](#)] [[PubMed](#)]
3. Gandhi, L.; Maisnam, D.; Rathore, D.; Chauhan, P.; Bonagiri, A.; Venkataramana, M. Respiratory illness virus infections with special emphasis on COVID-19. *Eur. J. Med. Res.* **2022**, *27*, 236. [[CrossRef](#)] [[PubMed](#)]

4. Wang, Y.; Hu, M.; Li, Q.; Zhang, X.-P.; Zhai, G.; Yao, N. Abnormal respiratory patterns classifier may contribute to large-scale screening of people infected with COVID-19 in an accurate and unobtrusive manner. *arXiv* **2020**, arXiv:2002.05534.
5. Tleyjeh, I.M.; Saddik, B.; Ramakrishnan, R.K.; AlSwaidan, N.; AlAnazi, A.; Alhazmi, D.; Aloufi, A.; AlSumait, F.; Berbari, E.F.; Halwani, R. Long term predictors of breathlessness, exercise intolerance, chronic fatigue and well-being in hospitalized patients with COVID-19: A cohort study with 4 months median follow-up. *J. Infect. Public Health* **2022**, *15*, 21–28. [[CrossRef](#)] [[PubMed](#)]
6. Martelletti, P.; Bentivegna, E.; Spuntarelli, V.; Luciani, M. Long-COVID headache. *SN Compr. Clin. Med.* **2021**, *3*, 1704–1706. [[CrossRef](#)] [[PubMed](#)]
7. Benmalek, E.; Elmhamdi, J.; Jilbab, A. Comparing CT scan and chest X-ray imaging for COVID-19 diagnosis. *Biomed. Eng. Adv.* **2021**, *1*, 100003. [[CrossRef](#)] [[PubMed](#)]
8. Bai, H.X.; Hsieh, B.; Xiong, Z.; Halsey, K.; Choi, J.W.; Tran, T.M.L.; Pan, I.; Shi, L.-B.; Wang, D.-C.; Mei, J. Performance of radiologists in differentiating COVID-19 from non-COVID-19 viral pneumonia at chest CT. *Radiology* **2020**, *296*, E46–E54. [[CrossRef](#)] [[PubMed](#)]
9. Ceniccola, G.D.; Castro, M.G.; Piovacari, S.M.F.; Horie, L.M.; Corrêa, F.G.; Barrere, A.P.N.; Toledo, D.O. Current technologies in body composition assessment: Advantages and disadvantages. *Nutrition* **2019**, *62*, 25–31. [[CrossRef](#)] [[PubMed](#)]
10. 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*, 19549. [[CrossRef](#)]
11. Cho, Y.; Bianchi-Berthouze, N.; Julier, S.J. DeepBreath: Deep learning of breathing patterns for automatic stress recognition using low-cost thermal imaging in unconstrained settings. In Proceedings of the 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), San Antonio, TX, USA, 23–26 October 2017; pp. 456–463.
12. Taylor, W.; Abbasi, Q.H.; Dashtipour, K.; Ansari, S.; Shah, S.A.; Khalid, A.; Imran, M.A. A Review of the State of the Art in Non-Contact Sensing for COVID-19. *Sensors* **2020**, *20*, 5665. [[CrossRef](#)]
13. Rahman, A.; Lubecke, V.M.; Boric-Lubecke, O.; Prins, J.H.; Sakamoto, T. Doppler radar techniques for accurate respiration characterization and subject identification. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2018**, *8*, 350–359. [[CrossRef](#)]
14. Ahmad, A.; Roh, J.C.; Wang, D.; Dubey, A. Vital signs monitoring of multiple people using a FMCW millimeter-wave sensor. In Proceedings of the 2018 IEEE Radar Conference (RadarConf18), Oklahoma City, OK, USA, 23–27 April 2018; pp. 1450–1455.
15. Prat, A.; Blanch, S.; Aguasca, A.; Romeu, J.; Broquetas, A. Collimated beam FMCW radar for vital sign patient monitoring. *IEEE Trans. Antennas Propag.* **2018**, *67*, 5073–5080. [[CrossRef](#)]
16. Podder, P.; Das, S.R.; Mondal, M.R.H.; Bharati, S.; Maliha, A.; Hasan, M.J.; Piltan, F. Lddnet: A deep learning framework for the diagnosis of infectious lung diseases. *Sensors* **2023**, *23*, 480. [[CrossRef](#)] [[PubMed](#)]
17. Purnomo, A.T.; Lin, D.-B.; Adiprabowo, T.; Hendria, W.F. Non-contact monitoring and classification of breathing pattern for the supervision of people infected by COVID-19. *Sensors* **2021**, *21*, 3172. [[CrossRef](#)] [[PubMed](#)]
18. Kim, J.; Campbell, A.S.; de Ávila, B.E.-F.; Wang, J. Wearable biosensors for healthcare monitoring. *Nat. Biotechnol.* **2019**, *37*, 389–406. [[CrossRef](#)] [[PubMed](#)]
19. Nguyen, T.; Khaksari, K.; Khare, S.M.; Park, S.; Anderson, A.A.; Bieda, J.; Jung, E.; Hsu, C.-D.; Romero, R.; Gandjbakhche, A.H. Non-invasive transabdominal measurement of placental oxygenation: A step toward continuous monitoring. *Biomed. Opt. Express* **2021**, *12*, 4119–4130. [[CrossRef](#)] [[PubMed](#)]
20. Massaroni, C.; Lo Presti, D.; Formica, D.; Silvestri, S.; Schena, E. Non-contact monitoring of breathing pattern and respiratory rate via RGB signal measurement. *Sensors* **2019**, *19*, 2758. [[CrossRef](#)] [[PubMed](#)]
21. Tobin, M.J.; Chadha, T.S.; Jenouri, G.; Birch, S.J.; Gazeroglu, H.B.; Sackner, M.A. Breathing patterns: 2. Diseased subjects. *Chest* **1983**, *84*, 286–294. [[CrossRef](#)]
22. AL-Khalidi, F.Q.; Saatchi, R.; Burke, D.; Elphick, H.; Tan, S. Respiration rate monitoring methods: A review. *Pediatr. Pulmonol.* **2011**, *46*, 523–529. [[CrossRef](#)]
23. Nicolò, A.; Massaroni, C.; Schena, E.; Sacchetti, M. The importance of respiratory rate monitoring: From healthcare to sport and exercise. *Sensors* **2020**, *20*, 6396. [[CrossRef](#)]
24. Cretikos, M.A.; Bellomo, R.; Hillman, K.; Chen, J.; Finfer, S.; Flabouris, A. Respiratory rate: The neglected vital sign. *Med. J. Aust.* **2008**, *188*, 657–659. [[CrossRef](#)] [[PubMed](#)]
25. Rohmetra, H.; Raghunath, N.; Narang, P.; Chamola, V.; Guizani, M.; Lakkaniga, N.R. AI-enabled remote monitoring of vital signs for COVID-19: Methods, prospects and challenges. *Computing* **2023**, *105*, 783–809. [[CrossRef](#)]
26. Costanzo, I.; Sen, D.; Rhein, L.; Guler, U. Respiratory monitoring: Current state of the art and future roads. *IEEE Rev. Biomed. Eng.* **2020**, *15*, 103–121. [[CrossRef](#)] [[PubMed](#)]
27. Qi, W.; Aliverti, A. A multimodal wearable system for continuous and real-time breathing pattern monitoring during daily activity. *IEEE J. Biomed. Health Inform.* **2019**, *24*, 2199–2207. [[CrossRef](#)] [[PubMed](#)]
28. Torjensen, I. COVID-19: Patients to use pulse oximetry at home to spot deterioration. *BMJ* **2020**, *71*, m4151. [[CrossRef](#)]
29. Todd, B. Pulse oximetry may be inaccurate in patients with darker skin. *Am. J. Nurs.* **2021**, *121*, 16.
30. Takegawa, R.; Hayashida, K.; Rolston, D.M.; Li, T.; Miyara, S.J.; Ohnishi, M.; Shiozaki, T.; Becker, L.B. Near-infrared spectroscopy assessments of regional cerebral oxygen saturation for the prediction of clinical outcomes in patients with cardiac arrest: A review of clinical impact, evolution, and future directions. *Front. Med.* **2020**, *7*, 587930. [[CrossRef](#)] [[PubMed](#)]
31. Samraj, R.S.; Nicolas, L. Near infrared spectroscopy (NIRS) derived tissue oxygenation in critical illness. *Clin. Investig. Med.* **2015**, *38*, E285–E295. [[CrossRef](#)] [[PubMed](#)]

32. Cheung, A.; Tu, L.; Macnab, A.; Kwon, B.K.; Shadgan, B. Detection of hypoxia by near-infrared spectroscopy and pulse oximetry: A comparative study. *J. Biomed. Opt.* **2022**, *27*, 077001. [[CrossRef](#)]
33. Koonin, E.V.; Krupovic, M.; Agol, V.I. The Baltimore classification of viruses 50 years later: How does it stand in the light of virus evolution? *Microbiol. Mol. Biol. Rev.* **2021**, *85*, e00053-21. [[CrossRef](#)]
34. Monto, A.S.; Gravenstein, S.; Elliott, M.; Colopy, M.; Schweinle, J. Clinical signs and symptoms predicting influenza infection. *Arch. Intern. Med.* **2000**, *160*, 3243–3247. [[CrossRef](#)] [[PubMed](#)]
35. Piret, J.; Boivin, G. Pandemics throughout history. *Front. Microbiol.* **2021**, *11*, 631736. [[CrossRef](#)] [[PubMed](#)]
36. Wang, J.; Zhou, M.; Liu, F. Reasons for healthcare workers becoming infected with novel coronavirus disease 2019 (COVID-19) in China. *J. Hosp. Infect.* **2020**, *105*, 100–101. [[CrossRef](#)] [[PubMed](#)]
37. Lewnard, J.A.; Lo, N.C. Scientific and ethical basis for social-distancing interventions against COVID-19. *Lancet Infect. Dis.* **2020**, *20*, 631–633. [[CrossRef](#)] [[PubMed](#)]
38. Salathé, M.; Althaus, C.L.; Neher, R.; Stringhini, S.; Hodcroft, E.; Fellay, J.; Zwahlen, M.; Senti, G.; Battegay, M.; Wilder-Smith, A. COVID-19 epidemic in Switzerland: On the importance of testing, contact tracing and isolation. *Swiss Med. Wkly.* **2020**, *150*, w20225. [[CrossRef](#)] [[PubMed](#)]
39. Kim, P.S.; Read, S.W.; Fauci, A.S. Therapy for early COVID-19: A critical need. *JAMA* **2020**, *324*, 2149–2150. [[CrossRef](#)] [[PubMed](#)]
40. Shafi, I.; Din, S.; Farooq, S.; Díez, I.d.I.T.; Breñosa, J.; Espinosa, J.C.M.; Ashraf, I. Design and development of patient health tracking, monitoring and big data storage using Internet of Things and real time cloud computing. *PLoS ONE* **2024**, *19*, e0298582. [[CrossRef](#)]
41. Shafi, I.; Ansari, S.; Din, S.; Jeon, G.; Paul, A. Artificial neural networks as clinical decision support systems. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6342. [[CrossRef](#)]
42. Fasihfar, Z.; Rokhsati, H.; Sadeghsalehi, H.; Ghaderzadeh, M.; Gheisari, M. AI-driven malaria diagnosis: Developing a robust model for accurate detection and classification of malaria parasites. *Iran. J. Blood Cancer* **2023**, *15*, 112–124. [[CrossRef](#)]
43. Ghaderzadeh, M.; Asadi, F.; Ramezan Ghorbani, N.; Almasi, S.; Taami, T. Toward artificial intelligence (AI) applications in the determination of COVID-19 infection severity: Considering AI as a disease control strategy in future pandemics. *Iran. J. Blood Cancer* **2023**, *15*, 93–111. [[CrossRef](#)]
44. Shah, V.; Keniya, R.; Shridharani, A.; Punjabi, M.; Shah, J.; Mehendale, N. Diagnosis of COVID-19 using CT scan images and deep learning techniques. *Emerg. Radiol.* **2021**, *28*, 497–505. [[CrossRef](#)] [[PubMed](#)]
45. Hong, J.-W.; Kim, S.-H.; Han, G.-T. Detection of multiple respiration patterns based on 1D SNN from continuous human breathing signals and the range classification method for each respiration pattern. *Sensors* **2023**, *23*, 5275. [[CrossRef](#)] [[PubMed](#)]
46. Kim, J.; Kim, J. Classification of Breathing Signals According to Human Motions by Combining 1D Convolutional Neural Network and Embroidered Textile Sensor. *Sensors* **2023**, *23*, 5736. [[CrossRef](#)] [[PubMed](#)]
47. Conway, A.; Goudarzi Rad, M.; Zhou, W.; Parotto, M.; Jungquist, C. Deep learning classification of capnography waveforms: Secondary analysis of the PRODIGY study. *J. Clin. Monit. Comput.* **2023**, *37*, 1327–1339. [[CrossRef](#)] [[PubMed](#)]
48. Mah, A.J.; Nguyen, T.; Ghazi Zadeh, L.; Shadgan, A.; Khaksari, K.; Nourizadeh, M.; Zaidi, A.; Park, S.; Gandjbakhche, A.H.; Shadgan, B. Optical Monitoring of breathing patterns and tissue oxygenation: A potential application in COVID-19 screening and monitoring. *Sensors* **2022**, *22*, 7274. [[CrossRef](#)] [[PubMed](#)]
49. Park, J.; Mah, A.J.; Nguyen, T.; Park, S.; Ghazi Zadeh, L.; Shadgan, B.; Gandjbakhche, A.H. Modification of a conventional deep learning model to classify simulated breathing patterns: A step toward real-time monitoring of patients with respiratory infectious diseases. *Sensors* **2023**, *23*, 5592. [[CrossRef](#)]
50. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [[CrossRef](#)] [[PubMed](#)]
51. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
52. Basodi, S.; Ji, C.; Zhang, H.; Pan, Y. Gradient amplification: An efficient way to train deep neural networks. *Big Data Min. Anal.* **2020**, *3*, 196–207. [[CrossRef](#)]
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part IV 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
54. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
55. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [[CrossRef](#)]
56. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
57. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
58. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
59. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2528–2535.
60. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [[CrossRef](#)]

61. Mumuni, A.; Mumuni, F. Data augmentation: A comprehensive survey of modern approaches. *Array* **2022**, *16*, 100258. [[CrossRef](#)]
62. Feng, S.Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; Hovy, E. A survey of data augmentation approaches for NLP. *arXiv* **2021**, arXiv:2105.03075.
63. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time series data augmentation for deep learning: A survey. *arXiv* **2020**, arXiv:2002.12478.
64. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; pp. 278–282.
65. Tan, M.; Le, Q. Efficientnetv2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 10096–10106.
66. Han, D.; Kim, J.; Kim, J. Deep pyramidal residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5927–5935.
67. Park, J.; Kim, H.; Paik, J. Cf-cnn: Coarse-to-fine convolutional neural network. *Appl. Sci.* **2021**, *11*, 3722. [[CrossRef](#)]
68. Long, X.; Deng, K.; Wang, G.; Zhang, Y.; Dang, Q.; Gao, Y.; Shen, H.; Ren, J.; Han, S.; Ding, E. PP-YOLO: An effective and efficient implementation of object detector. *arXiv* **2020**, arXiv:2007.12099.
69. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
70. Kim, S.; Rim, B.; Choi, S.; Lee, A.; Min, S.; Hong, M. Deep learning in multi-class lung diseases' classification on chest X-ray images. *Diagnostics* **2022**, *12*, 915. [[CrossRef](#)]
71. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.J.; Pla, F. Deep pyramidal residual networks for spectral–spatial hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 740–754. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.