*Int. J. Appl. Math. Comput. Sci., 2025, Vol. 35, No. 4, 617–630*
DOI: 10.61822/amcs-2025-0044

# DEEP LEARNING EPILEPTIC SEIZURE DETECTION BASED ON THE MATCHING PURSUIT ALGORITHM AND ITS TIME–FREQUENCY GRAPHICAL REPRESENTATION

MATEUSZ M. KUNIK [a], ARTUR GRAMACKI [a,*]

[a]Institute of Control and Computation Engineering
University of Zielona Góra
ul. Szafrana 2, 65-516 Zielona Góra, Poland
e-mail: {M.Kunik,A.Gramacki}@issi.uz.zgora.pl

Electroencephalography (EEG) is the primary diagnostic and an important prognostic clinical tool for epilepsy. The detection of epileptic activity is usually performed by a human expert and is based on finding specific patterns in the multi-channel electroencephalogram. However, the manual inspection of EEG signals is a time-consuming procedure for neurologists. Therefore, various attempts are made to automate it using both conventional and deep learning techniques. In this article, (i) we investigate the possibility of using time-frequency maps of energy derived from the matching pursuit algorithm for accurate detection of epileptic seizures (to the best of our knowledge, such an approach has not been analyzed so far, making this a pilot study); (ii) we show how to build an effective deep convolutional neural network with the so-called (2+1)D convolution technique; (iii) using carefully selected 79 neonatal EEG recordings, we develop a complete framework for seizure detection employing a deep learning approach, (iv) we share a ready to use R and Python codes which allow reproducing all the results presented in the paper.

**Keywords:** EEG signals, epileptic seizure detection, matching pursuit algorithm, time-frequency representation, deep learning.

## 1. Introduction

Epilepsy is a neurological disorder which can be detected by analyzing the signals produced by brain neurons. The monitoring of these brain signals is commonly done using electroencephalography (EEG), which is the primary diagnostic and an important prognostic clinical tool. These signals are very complex, non-linear, noisy, and non-stationary, and the volume of registered data is very high. Furthermore, there are many different types of epilepsy—general and focal epileptic seizures are summarized in the work of Shoeibi *et al*. (2022). Hence, detecting epileptic seizures is a very difficult and challenging task.

Various researchers have developed a number of approaches to seizure detection using both machine learning (ML) and deep learning (DL) methods. It is not possible to cite all (or even most) of the works on this topic. Therefore, we only provide a few recent works of review type (Bai *et al*., 2025; Xu *et al*., 2024; Zou *et al*., 2024; Miltiadous *et al*., 2022; Shoeibi *et al*., 2022).

In the work of Bai *et al.* (2025), 41 studies up to April 2025 were included in the meta-analysis, focusing on different DL and traditional ML methods. The paper reviews the performance of ML models in seizure detection and analyzes factors such as the model type (DL vs. traditional ML), data preprocessing methods, and dataset types.

In the work of Xu *et al*. (2024), commonly used datasets (nine items) for seizure detection are summarized. The authors then present selected works (77 items) from 2018 to 2023, grouping them by technique, i.e., convolutional neural networks (CNNs, 1D and 2D), recurrent neural networks (RNNs), autoencoders (AEs), graph neural networks (GNNs), CNN-RNNs, based on attention or transformers, and integrating DL with traditional ML methods.

The work of Zou *et al*. (2024) is a systematic review including 28 original studies, with 15 on ML

*Corresponding author

and 13 on DL. All these models were based on electroencephalography data of children. It is worth emphasizing that this list includes 11 articles in which the authors employed the same dataset that we are using in this article (see Section 2). The results from individual works are summarized using forest and funnel plots, separately for the DL and ML methods.

In the work of Miltiadous *et al*. (2022), 190 studies were included from the years 2017–2021, and they are grouped based on what published dataset they were using. The most popular datasets appear to be Bonn and CHB-MIT, which were employed in most of the 190 studies (see Table 7 and Fig. 8 therein). Also, 64 papers were identified in which the authors used more than one dataset in their research. The Bonn and CHB-MIT datasets are also most frequently employed here. The mentioned paper of Miltiadous *et al.* also presents performance results obtained by the authors of the analyzed works.

In the work of Shoeibi *et al*. (2022), the authors collected a vast amount of articles on deep learning techniques for epileptic seizure detection and prediction from the years 2016–2020. They thoroughly analyzed approximately 200 articles, examining various DL methods, epileptic seizure datasets used, types of preprocessing, deep learning architectures and algorithms, and neuroimaging modalities. The performance results obtained by individual authors according to criteria such as AUC, sensitivity, specificity, and F1-score are also presented.

In this article, we propose to use the matching pursuit algorithm (Mallat and Shang, 1993), and in particular the *time-frequency maps of energy distribution* derived from that algorithm. On the other hand, deep learning methods, specifically convolutional neural networks (CNNs), have been widely employed in medical image analysis, such as in pneumonia detection using chest X-rays (Akbar *et al*., 2024) and in breast cancer histopathology classification (Kaczmarek *et al*., 2025). While these methods are well-established in image-based medical problems, the use of the matching pursuit algorithm for seizure detection in EEG signals through the analysis of time-frequency maps has not been analyzed so far, making this a pilot study. In this article, we try to assess the practical usefulness of this approach.

Time-frequency maps have been applied by various authors to analyze EEG signals. The work of Tzallas *et al*. (2009) is an influential, often-cited contribution to EEG seizure detection. The authors compare a short-time Fourier transform (STFT) and 12 different time-frequency distributions (TFDs) to obtain the power spectrum density. This helped popularize a clear, modular pipeline (T-F transform $\rightarrow$ T-F feature design $\rightarrow$ classifier) that many further works reused, extended or compared against. To transform EEG signals into T-F domain, authors very often use techniques such as the STFT, a continuous or discrete wavelet transform (CWT/DWT), Wigner–Ville distribution (WVD), the Stockwell transform and various hybrid approaches (Şengür *et al*., 2016; Pan *et al*., 2022; Dong *et al*., 2024; Bajaj and Pachori, 2013; Truong *et al*., 2018; Türk and Özerdem, 2019; Rashed-Al-Mahfuz *et al*., 2021; Shen *et al*., 2024; Raab *et al*., 2023; Siddartha *et al*., 2024; Tapani *et al*., 2019).

We also note that this article does not address seizure prediction and forecasting (often, incorrectly, these two are used interchangeably). They are entirely different in nature than EEG-based seizure detection. There are studies in which the authors analyze this issue in detail. We encourage those interested to first consult the review paper by Carmo *et al*. (2024).

## 2. Cohort, dataset, initial preprocessing

The study was conducted on a carefully selected dataset of 79 neonatal EEG recordings. The neonates were admitted to the neonatal intensive care unit (NICU) at Helsinki University Hospital between 2010 and 2014. The cohort is described in detail by Stevenson *et al*. (2019); see the source text (in many sources this dataset is called the *Helsinki database*). In addition, appropriate ethical approval was included. All experiments were performed in accordance with relevant guidelines and regulations.

The neonatal EEG dataset consists of (a) 79 raw EDF files, (b) three annotation files in the CSV and Matlab MAT formats. The EDF files contain EEG referential signals recorded with 19 electrodes positioned as per the international 10–20 standard (Fp1, Fp2, F3, F4, F7, F8, Fz, C3, C4, Cz, P3, P4, Pz, T3, T4, T5, T6, O1, O2). The sampling frequency was set to 256 Hz and the signals were recorded in microvolts. The complete dataset is available at `https://zenodo.org/record/4940267`. A very detailed presentation of the dataset used can be found in the paper by Gramacki and Gramacki (2022), hence we do not repeat this information here.

The raw signals are not employed directly. Instead, the so-called bipolar montage was generated known by the slang *double banana* (Fp2-F4, F4-C4, C4-P4, P4-O2, Fp1-F3, F3-C3, C3-P3, P3-O1, Fp2-F8, F8-T4, T4-T6, T6-O2, Fp1-F7, F7-T3, T3-T5, T5-O1, Fz-Cz, Cz-Pz), (see, e.g., Niedermeyer and da Silva, 2005; Klem *et al*., 1999). This bipolar EEG montage was used by three independent experts to annotate the presence of seizures.

Before further processing, the data was preprocessed, i.e., it was filtered with IIR Butterworth filters: a 2-pole high-pass filter with a cut-off frequency of 1 Hz, an 8-pole low-pass filter with a cut-off frequency of 30 Hz and a 2-pole band-stop filter with a center at 50 Hz. The choice of such frequencies is the authors' decision, but many publications indicate similar values (see, e.g., Widmann *et al*., 2015).

Let us note here that in many cases there is a discrepancy in the annotations of the three individual experts mentioned (described as expert A, B and C). For example, for infant number 41, experts A and C indicated significantly more seizures than expert B (see Table 6 by Gramacki and Gramacki (2022)). The lengths of individual seizures also quite often vary between experts. Such a variety of end results (no consensus among experts) is rather quite natural in the field of EEG signal analysis (Stevenson *et al*., 2015).

## 3. Matching pursuit and its time-frequency representation

Given a signal $f \in \mathbb{R}^n$ and a (possibly overcomplete) large redundant dictionary $D = \{g_\gamma\}_{\gamma\in\Gamma}$ of normalized atoms $\|g_\gamma\| = 1$, the matching pursuit algorithm (MP) finds a sparse signal representation,

$$f \approx \sum_{n=0}^{N} a_n g_{\gamma_n}, \tag{1}$$

where $a_n \in \mathbb{R}$ are coefficients, $g_{\gamma_n} \in D$ are atoms selected from the dictionary and $N$ is the desired number of iterations (or a stopping threshold). In most practical cases, $N \ll size(D)$. Also, $g_\gamma$ is the dictionary atom indexed by $\gamma$ and $\Gamma$ is the set of all indices in the directory.

In an ideal situation, the linear expansion (1) should contain all the atoms $g_{\gamma_n}$ that represent relevant structures of the signal $f$. In real signals, such a situation is rarely possible and a kind of approximation is needed. This task can be performed in a very elegant way by using the MP algorithm, which was first proposed by Mallat and Shang (1993) in the context of signal processing.

Each atom $g_\gamma$ is typically a time-frequency shifted and scaled version of a prototype function, such as the Gabor function (often called a Gaussian-windowed sinusoid). The dictionary is constructed to span a wide range of time and frequency characteristics. A real-valued Gabor function has the following form:

$$g_\gamma(t) = K(\gamma)e^{-\pi\left(\frac{t-\mu}{\sigma}\right)^2}\cos(\omega(t-\mu)+\phi), \tag{2}$$

where $\gamma = (\mu, \omega, \sigma, \phi)$ constitute a four-dimmensional space and $K(\gamma)$ is such that $||g_\gamma|| = 1$. It is easy to see that Gabor functions are constructed by multiplying Gaussian envelopes with cosine oscillations of different frequencies $\omega$ and phases offset $\phi$. By multiplying these two functions, we can obtain a wide variety of shapes depending on their parameters. A few examples of Gabor functions are presented in Fig. 1.

In the MP algorithm, the decomposition process is iterative. At each step, the algorithm selects an atom $g_{\gamma_n}$ from the dictionary $D$ that best matches the current residual signal $R$. Formally, starting with the signal $f$ at

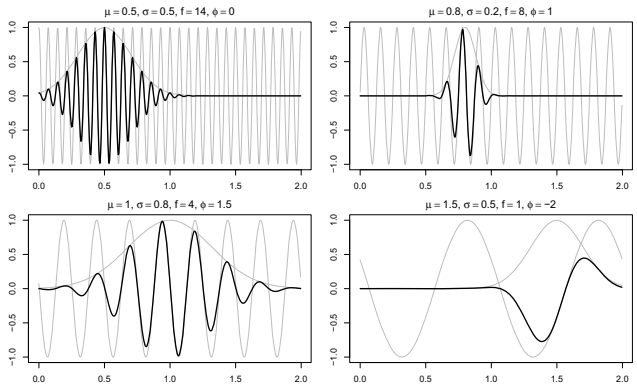



Fig. 1. Four examples of Gabor functions (bold lines). Gaussian envelopes and cosine waves are drawn with thin gray lines.

the iteration $n = 0$, the initial residual and initial function approximation are

$$\begin{aligned} R^0 &= f, \\ f^0 &= 0. \end{aligned} \tag{3}$$

For each iteration $n = \{0, 1, \ldots, N\}$, we find $g_{\gamma_n} \in D$ such that the following inner product $\langle\cdot,\cdot\rangle$ is maximized:

$$g_{\gamma_n} = \arg\max_{\gamma\in\Gamma} |\langle R^n, g_\gamma\rangle|. \tag{4}$$

The coefficients $a_n$ in (1) are

$$a_n = \langle R^n, g_{\gamma_n}\rangle, \tag{5}$$

and the updated function approximation is denoted by

$$f^{n+1} = f^n + a_n g_{\gamma_n}. \tag{6}$$

Similarly, the updated residual is defined as

$$R^{n+1} = R^n - a_n g_{\gamma_n}. \tag{7}$$

After $N$ iterations, the signal $f$ is approximated as

$$f \approx \sum_{n=0}^{N} \langle R^n, g_{\gamma_n}\rangle g_{\gamma_n} \tag{8}$$

or, equivalently,

$$f = \sum_{n=0}^{N} \langle R^n g_{\gamma_n}\rangle g_{\gamma_n} + R^{N+1}. \tag{9}$$

The procedure stops when $\|R^{n+1}\|$ is below a threshold or after a fixed number of iterations.

It should be mentioned here that finding an optimal approximation (1) is an NP-hard problem. A suboptimal solution can be found by means of an iterative procedure, such as the MP algorithm. Also, a key property of MP

is energy conservation. The total energy of the signal is preserved in the MP decomposition,

$$||f||^2 = \sum_{n=0}^{N} |\langle R^n, g_{\gamma_n} \rangle|^2 + ||R^{N+1}||^2. \tag{10}$$

Each selected atom has a well-defined time and frequency, so we can build time-frequency maps of energy (TFMEs) using the MP decomposition. Each Gabor atom contributes an energy blob (ellipses, actually a 2D Gaussian) centered at its time $t$ and frequency $f$, and whose spread depends on scale $\sigma$ and the Heisenberg uncertainty principle. The principle states that the shorter the structure is in time, the wider its frequency content. In signal processing, this means that

$$\Delta t \Delta f \geq \frac{1}{4\pi}, \tag{11}$$

where $\Delta t$ is standard deviation (width) in time domain and $\Delta f$ is standard deviation in frequency domain. Gabor signals minimize this uncertainty product, which means their product reaches the minimum bound

$$\Delta t \Delta f = \frac{1}{4\pi}. \tag{12}$$

This is a manifestation of the time-frequency uncertainty principle—one cannot be arbitrarily sharp in both time and frequency at once.

A Dirac impulse can be regarded as an extremely short structure, so it will be represented as a vertical line. On the other hand, the infinite sine wave has the best localization in frequency domain but the worst in time domain, so it will be represented as a horizontal line. Hence, a narrow blob in time represents atoms with good time resolution and poor frequency resolution. Similarly, a narrow blob in frequency represents good frequency resolution and poor time resolution.

By adding all the $N$ atoms' blobs, we obtain TFME representation like that depicted in Fig. 2(a), where a sample signal was decomposed into 50 atoms. Each of the 50 atoms is visible as a blob with a different color saturation as well as different width (on the horizontal time axis) and length (on the vertical frequency axis).

Figure 3 shows, from top down, the sample signal in time domain (a), its reconstruction based on 50 atoms (b). It can be observed that the reconstruction is almost perfect: the first five atoms (those with the highest energies) (c), the signal reconstructed using only the 20 atoms with the highest energies (d). It can be seen that the reconstruction is not very accurate, but the general character of the input signal is preserved. By comparing Fig. 2(a) with Fig. 3, it is easy to locate in the former the five atoms with the highest energy as the brightest blobs. Details of the implementation of the MP algorithm are beyond the scope of this article. The basic problem here is the correct selection of $N$ atoms from a large (potentially infinite) dictionary $D$. A highly optimized multi-threaded version with GPU support of MP can be downloaded from the GitHub repository (Różański, 2025). More information on the MP algorithm can also be found in the works of Różański (2024), Kuś *et al*. (2013) and Durka (2007).

## 4. Input data

The input to the neural network is time-frequency maps, analogous to those shown in Fig. 2(a). For performance reasons, the resolution of these maps was limited to 64 × 64 pixels, as in Fig. 2(b).

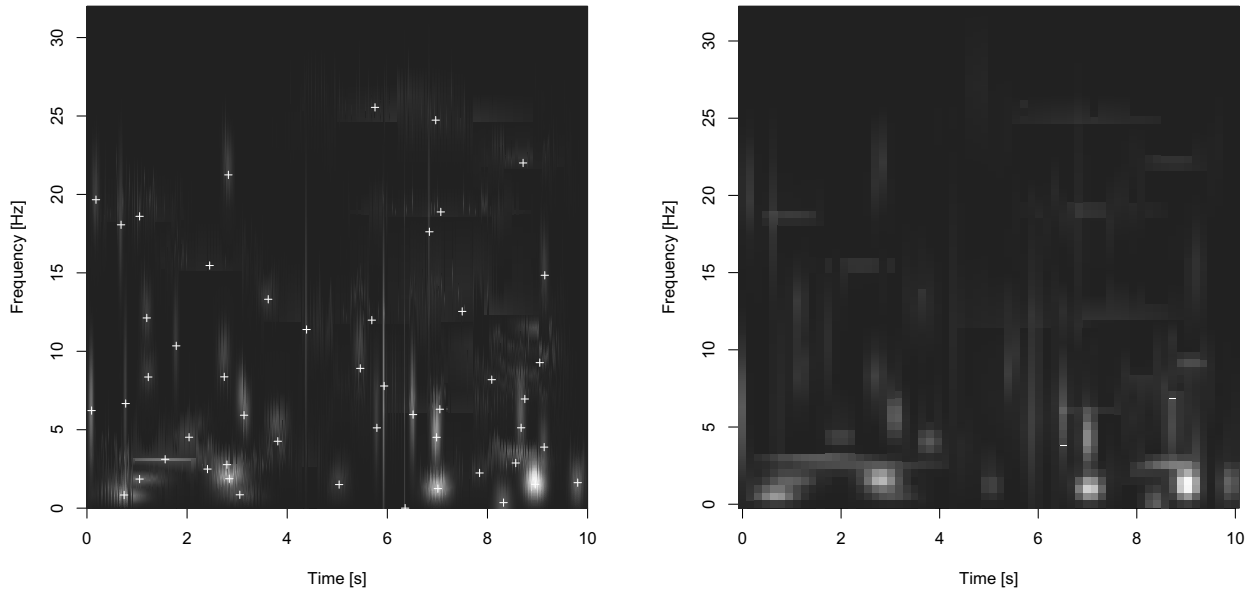
Of course, in the lower resolution figures, various details are lost, but as experiments show, even such small TFMF maps allow very good results to be obtained. Experiments conducted with larger maps did not yield significant improvement in the results, while computation times and the size of the neural network increased significantly.

Since the EEG data used consists of 18 channels, one data sample covers 18 maps, which gives us a tensor of size 18 × 64 × 64. Positive and negative samples (e.g., seizure and nonseizure fragments from EEG recordings) are generated according to exactly the same principle, which was presented in detail by Gramacki and Gramacki (2022) (so-called *sliding window design*; see the *Data preprocessing* section therein). Using the results presented in Tables 3, 4 and 5 (included in the work of Gramacki and Gramacki (2022)), the following values were chosen: *Window size* = 10 and *Number of contiguous chunks* = 20. For these parameter values, the total number of chunks based on annotations provided by experts A, B and C is presented in Table 1.

It was decided that the input signals would be decomposed into no more than 50 atoms. This was motivated by the fact that such a number of atoms makes the signal's total energy $\int_0^T |x(t)|^2\, dt$ after MP decomposition $E_{decomp}$ only slightly smaller than the original signal's total energy $E_{orig}$. The average ratio $E_{decomp}/E_{orig}$ for all sets A, B and C was about 96% with a standard deviation of about 4.5%. This means that only a very small amount of energy was not explained by the atoms.

### 4.1. Two modes of input data preparation.

The size of the training data (see Table 1) is not very small, but in practice it may be insufficient to achieve satisfactory results. Therefore, two types of experiments were performed: (a) data from individual experts formed three independent (expert-secific) datasets, (b) data from all three experts were merged into one bigger dataset. Each approach has its advantages.




(a) (b)

Fig. 2. MP decomposition of a sample signal. Every colored blob represents one atom localized precisely in time and frequency. The saturation of individual blocks is proportional to the energy of individual signal fragments. Small white crosses indicate centers of the atoms and are not present in the data that we feed into the neural network. The data is stored as a matrix which is normalized to the range [0-1] (a). MP decomposition of a sample signal from the left hand side, but after limiting the resolution to the size of 64×64 pixels. The data is stored as a matrix which is normalized to the range [0-1] (b). (Note: The visible colors are for visualization only and have no significance for the analysis.)

.

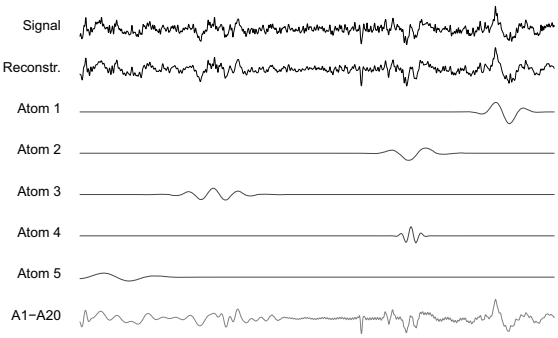



Fig. 3. Looking from top to bottom: the analyzed sample signal and its reconstruction based on 50 atoms (a), the first five atoms with the highest energies (b), the signal reconstructed using only the 20 atoms with the highest energies (c). The signal's total energy, defined as $\int_0^T |x(t)|^2\,dt$, of the original signal is $121,668$, while the total energy of the reconstructed signal is $109,731$. This means that for this particular signal as much as $90.2\%$ of the signal's energy was explained by the MP atoms.

The expert-specific datasets allow the evaluation of inter-rater variability and the model's ability to adapt to individual annotation styles used by the individual experts A, B and C. On the other hand, analyzing Tables 6 and 7 from the work of Gramacki and Gramacki (2022), there is of course some kind of agreement between the experts' annotations, but many differ. As mentioned above, such a lack of consensus among experts evaluating EEG signals is not unusual. The merged dataset, on the other hand, benefits from increased sample diversity and size, which can enhance generalization and model robustness.

**4.2. Cross-validation strategy.** To ensure the reliability of the results, we applied 5-fold cross-validation. The training set was divided into five equal subsets, four of which were used for training and one for validation in each fold. A separate test set containing unseen data was used exclusively for the final performance evaluation. Both the model's final weights and the best-performing weights on the validation set were evaluated on this test set. The size of the test set was established as $10\%$ of the total data. Given the $k$-fold cross-validation using $k = 5$, in each iteration the training set covers about $72\%$ of the data and the validation set contains about $18\%$ of the data.

**4.3. EEG seizure detection pipeline.** Below are the basic steps we take to prepare the data for training the neural network.

1. Read all EEG files (with the *edf* extension), generate the so-called bipolar montage (known by the

slang *double banana*), execute filtering (high-pass, low-pass, band-stop) and prepare binary files (with the *bin* extension) with positive and negative samples (separately for data from experts A, B and C). Use the *sliding window design* as described in detail by Gramacki and Gramacki (2022). This step is implemented in the *edf_to_bin.R* script; see Section 8.

2. Perform MP time-frequency decomposition on each binary file using the very fast implementation available for download (see Różański, 2025). The results are saved in the SQLite format (with the *db* extension). This step is implemented in the *bin_to_db.R* script; see Section 8.

3. Generate TFME maps (*RData* files) based on the contents of the SQLite files. Each positive or negative sample contains data from 18 EEG channels, so, for example, for data from expert A, $6139 \times 18 = 110502$ files are generated (see Table 1). The range in the $f$-axis was arbitrarily set from 0 to 32 Hz, although the sampling frequency of 256 Hz was used for recording EEG signals. EEG gamma waves, i.e., oscillations with frequencies above 30 Hz (most often 30–100 Hz), are not traditionally considered a primary indicator of epileptic seizures, but they may be important in some clinical and research contexts. The range in the $t$-axis was was arbitrarily set from 0 to 10 seconds (see Fig. 2). This gives a resolution of 0.15625 seconds per pixel on the $t$-axis and 2 Hz per pixel on the $f$-axis. This step is implemented in the *db_to_RData.R* script; see Section 8.

4. Prepare tensors of size $n \times 18 \times 64 \times 64$, where $n$ is the total number of chunks based on annotations provided by experts A, B and C (see Table 1). In addition to the data from experts A, B and C, merge data from experts A, B and C into one large dataset ABC (see Section 4.1). Save tensors as *hdf5* files. This step is implemented in the *RData_to_hdf5.py* script; see Section 8.

5. Split the data into train, validation and test sets, according to the remarks given in Section 4.2. This step is implemented in the *split_data.py* script; see Section 8.

6. First, the required parameters for initializing and training the neural network are loaded from a *config.yml* file. Next, the preprocessed data are loaded and split into the training, validation, and test sets, with the corresponding data loaders created for each subset. The neural network model, optimizer, and loss function are then initialized. Training is performed using the *train_and_validate_model* function, which iteratively updates the model weights based on the training data and evaluates performance on the validation set. Finally, the training artifacts are saved, including the complete training history and the model weights corresponding to the best achieved performance, enabling reproducibility and further analysis. The entire deep learning pipeline is ready to be executed from the *pipeline_example.ipynb* file, providing a convenient entry point for end-to-end experimentation.

Table 1. Total number of chunks based on annotations provided by experts A, B and C. Negative samples are generated in such a way that the positive and negative ones are approximately equal in number.

| Expert(s) | Positive samples | Negative samples | Sum |
|---|---|---|---|
| A | 3059 | 3080 | 6139 |
| B | 3478 | 3498 | 6976 |
| C | 3444 | 3454 | 6898 |
| ABC | 9981 | 10032 | 20013 |

## 5. Experiments

Given the specific structure of our data, the architecture of the proposed neural network was designed to accommodate the dimensionality of the input. Each data sample is represented as a tensor of size $18 \times 64 \times 64$, where 18 corresponds to the number of EEG channels, and $64 \times 64$ denotes a time-frequency representation generated via the MP algorithm. It is crucial to emphasize that the data do not exhibit a standard spatiotemporal structure as in video data, where consecutive frames correspond to different time points. In this case, the "channel" dimension represents independent spatial signals rather than a temporal axis.

Under these conditions, it is natural to consider architectures capable of learning from 3D input data. However, standard 3D convolutions are often computationally expensive and challenging to train due to the large number of learnable parameters. To address this issue, we adopted the (2+1)D convolutional approach, which decomposes the full 3D convolution into two simpler operations, thus improving computational efficiency and model performance.

**5.1. Convolution (2+1)D.** The (2+1)D convolution technique was introduced as an alternative to standard 3D convolutions, notably by Tran *et al*. (2018). It decomposes a conventional 3D kernel of size $t \times d \times d$ into a spatial 2D convolution of size $1 \times d \times d$ followed by a 1D convolution of size $t \times 1 \times 1$. In our setup, the 2D convolution

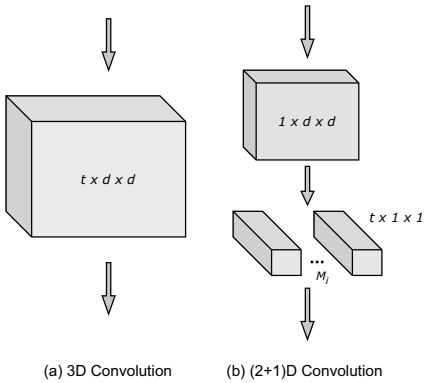



Fig. 4. Convolutions 3D and (2+1)D: a full 3D convolution filter of size $t \times d \times d$, where $t$ denotes the temporal extent and $d$ is the spatial height and width (a), decomposition of 3D convolution into a spatial 2D convolution followed by a temporal 1D convolution, where $M_i$ is the number of 2D filters so that the number of parameters in the (2+1)D block matches that of the full 3D convolutional block (b).

captures spatial features within a single time-frequency map, while the 1D convolution models the relationships between channels, which in this context can be interpreted as a quasi-temporal dimension.

Figure 4 (Xi *et al.*, 2024) illustrates this decomposition. A critical aspect of this design is the determination of the number of intermediate filters $M_i$, which corresponds to the number of filters in the 2D convolutional layer. As proposed by Tran *et al.* (2018), this value is chosen so that the total number of parameters in the (2+1)D block approximates that of the original 3D convolutional block. Therefore, $M_i$ is computed using the following formulae:

$$M_i = \left\lfloor \frac{td^2 C_{in} C_{out}}{d^2 C_{in} + tC_{out}} \right\rfloor, \tag{13}$$

where $t$ is the size of the kernel along the channel dimension, $d$ is the spatial kernel size, $C_{in}$ is the number of input channels, and $C_{out}$ is the number of output channels.

The use of (2+1)D convolutions offers several key advantages. First, it significantly reduces the number of parameters, thereby lowering the risk of overfitting and improving training efficiency. Second, by explicitly decoupling spatial and inter-channel interactions, the network gains greater representational capacity and interpretability. Third, as shown in prior research (Tran *et al.*, 2018), such decompositions lead to faster convergence and better generalization, which is especially valuable in medical applications where labeled data are limited. These benefits are further supported by our empirical comparison presented in Table 2, which demonstrates that the (2+1)D architecture

Table 2. Comparison between a residual (2+1)D convolutional network and a standard 3D convolutional architecture with a similar number of trainable parameters, determined by $M_i$.

| Architecture | Total mult-adds | Memory | Total size |
|---|---|---|---|
| R(2+1)D | 3.35 M | 4.00 MB | 6.26 MB |
| 3D | 159.00 M | 7.95 MB | 10.21 MB |

entails significantly fewer computations and consumes less memory than the standard 3D convolutional network, while maintaining a similar number of trainable parameters.

In our implementation, the (2+1)D block first reshapes the input tensor to apply 2D convolutions independently to each 64 $\times$ 64 time-frequency map. The resulting feature maps are then reorganized to allow the 1D convolution to operate along the channel axis. This ensures that the output tensor maintains a structure compatible with further layers in the network.

**5.2. Deep learning CNN architecture.** The basis of our network is a residual architecture utilizing (2+1)D convolutions. This combination, known as R(2+1)D, was originally introduced by Tran *et al.* (2018). Our architecture builds upon R(2+1)D and has been tailored to address our specific task. The network begins with a (2+1)D convolutional layer with 16 filters and a kernel of size $3\times7\times7$, followed by batch normalization and a ReLU activation function. This is succeeded by three residual blocks with increasing numbers of filters (16, 32, and 64, respectively), each composed of two residual submodules: a residual convolutional block and a residual identity block. After processing through the residual stages, the final feature map is reduced using a global average pooling layer, and the resulting vector is passed to a simple classification block. The complete model architecture is illustrated in Fig. 6.

As introduced by He *et al.* (2016), the residual mechanism enables more effective gradient propagation in deep networks, thereby alleviating the vanishing gradient problem. Structurally, each residual block adds its input to the output of a nonlinear transformation, formally expressed as

$$y = \mathcal{F}(x, W_i) + x, \tag{14}$$

where $\mathcal{F}$ denotes the composition of convolution, normalization and activation operations, and $x$ is the original input.

Figure 5 provides a schematic view of a basic residual block with identity mapping. This visual representation emphasizes how the direct addition of input $x$ to the transformed features $\mathcal{F}(x)$ facilitates training in deeper networks.

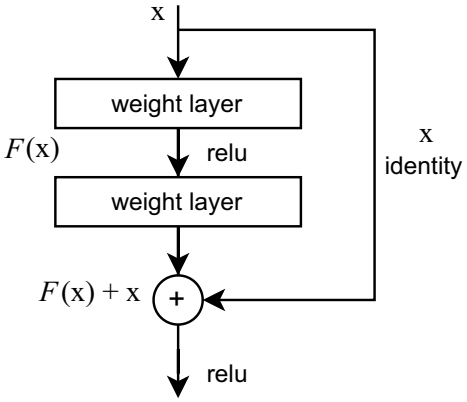



Fig. 5. Residual connection with identity mapping as introduced by He *et al*. (2016). The input $x$ is added to the output of the residual function $\mathcal{F}(x)$, facilitating gradient flow and enabling the training of deeper networks.

Each residual block in our architecture consists of two submodules. The first one is the residual convolutional block, responsible for adjusting the dimensions of the input (e.g., spatial downsampling and increasing the number of filters). It includes two (2+1)D convolutional operations with a $3\times3\times3$ kernel, the first of which is applied with a $2\times2\times2$ stride to perform downsampling and reduce the spatial and temporal resolution, and the second with a standard $1\times1\times1$ stride. It also contains a parallel shortcut path using a (2+1)D convolution with a $1\times1\times1$ kernel and a matching stride.

The second submodule is the residual identity block, which maintains the dimensions of the input tensor and primarily serves to deepen the representation. It also consists of two (2+1)D convolutions with a $3\times3\times3$ kernel and a $1\times1\times1$ stride. The skip connection here is an identity mapping, meaning the input is added directly to the transformed output.

In both types of residual blocks, each convolutional operation is separated by batch normalization and a ReLU activation, and the block concludes with dropout for regularization.

After passing through the three residual blocks, the output tensor is spatially aggregated via a 3D global average pooling operation, resulting in a compact feature vector. This vector is fed into the classification block, which is intentionally kept simple. It consists of two fully connected layers. The first one reduces the dimensionality to 10 neurons, followed by a ReLU activation and dropout regularization. The final one outputs logits corresponding to the two target classes: seizure and non-seizure. Despite its simplicity, this classifier has proven effective given the structured nature of the input data.

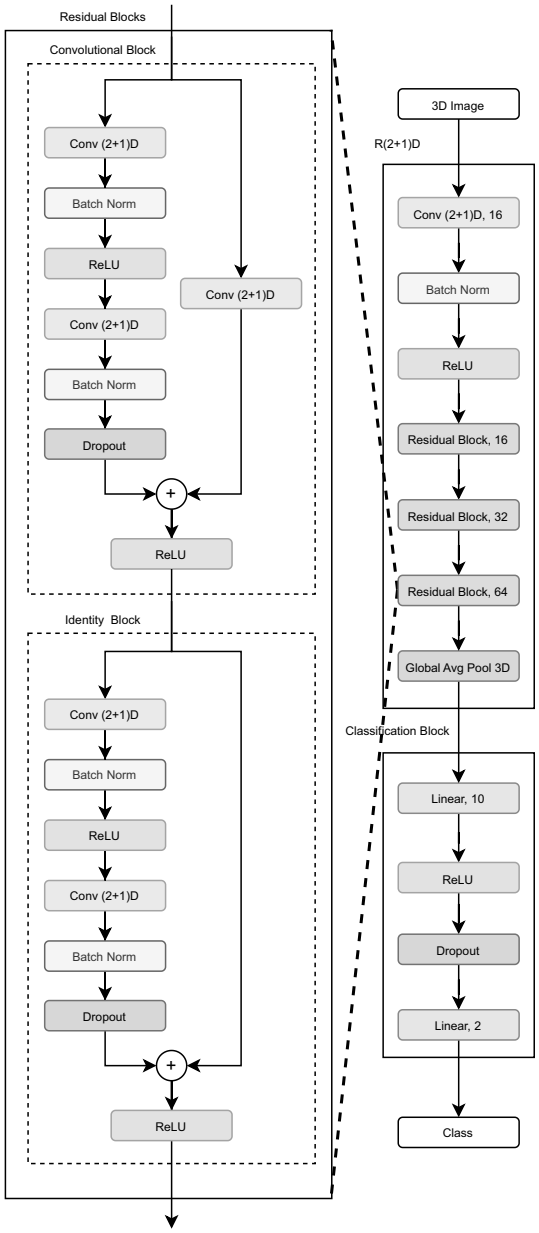



Fig. 6. Architecture of the R(2+1)D model used in our study. The network employs residual blocks with (2+1)D convolutions, which decompose 3D convolutions into separate spatial (2D) and temporal (1D) operations, reducing the number of parameters and enhancing training efficiency. The final classification block consists of two fully connected layers.

**5.3. Data augmentation.** Data augmentation was applied exclusively to the training set to improve generalization and prevent overfitting. Two stochastic transformation methods were used. The first one, RandomChannelShuffle, is a novel augmentation inspired by the channel shuffle operation originally introduced by Zhang *et al*. (2018) in ShuffleNet, which enables inter-group information exchange in grouped convolutions. In our case, the input is a 3D tensor of 18

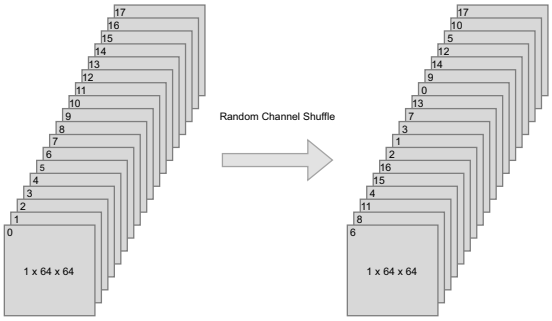



Fig. 7. Illustration of the RandomChannelShuffle operation applied to an 18-channel time-frequency map from distinct EEG sensors. Maps are randomly permuted along the channel axis without altering individual feature maps.

channels, each representing a $64 \times 64$ time-frequency map from a distinct EEG sensor. Unlike traditional image data, these channels have no inherent sequential order, allowing their permutation without semantic distortion. Randomly shuffling these channels introduces structured variability, helping the model avoid overfitting to specific sensor arrangements and encouraging it to learn order-invariant features.

Moreover, similar inter-channel augmentation strategies have been explored in other domains, such as meta-learning (Yao *et al*., 2021) and signal classification (Wei *et al*., 2023), where channel permutations have been shown to improve generalization under limited data conditions.

The second transformation, RandomInvert, inverts pixel intensities according to the formula

$$x' = 1 - x. \tag{15}$$

This augmentation was introduced due to the skewed color distribution of the time-frequency images, where most pixels are black, and only salient features appear in bright colors. Inverting the image increases diversity in visual patterns and adds robustness to the model.

Each augmentation was applied probabilistically during training, meaning only a subset of samples were augmented in each epoch. This maintained a balance between diversity and data consistency.

**5.4. Model training.** During training, the binary cross-entropy loss function was used as the primary optimization objective. This function is well-suited for binary classification tasks such as seizure detection. In addition to the loss, several evaluation metrics were monitored, including accuracy, precision, recall, and F1-score. Since the dataset was well-balanced, accuracy served as a reliable primary metric. However, F1-score was also considered critical for evaluating the model's performance, especially in detecting infrequent seizure events.

We trained the model using the Adam optimizer (Kingma and Ba, 2014), which provides adaptive learning rates and generally exhibits faster and more stable convergence. For comparison, we also evaluated the stochastic gradient descent (SGD), but Adam consistently demonstrated superior performance in this context.

Both the final and best-validation models were evaluated on the test set using all predefined metrics, ensuring comprehensive assessment of the model's seizure detection capabilities.

To ensure optimal model performance and prevent overfitting, a range of hyperparameters was systematically explored during the training phase. The evaluated settings included the batch size, dropout rate, learning rate, weight decay, and the data augmentation techniques mentioned earlier. Additionally, training duration and stopping criteria were carefully tuned using early stopping with checkpointing to retain the best-performing model on the validation set. An overview of the hyperparameter ranges considered during the experiments is presented in Table 3. All configurations were selected based on empirical validation performance and guided by prior literature on deep learning for biomedical signal analysis.

**5.5. Results.** The initial stage of our experimental pipeline focused on optimizing the architecture of the R(2+1)D neural network to achieve high classification performance while minimizing model complexity. This involved systematic tuning of the number of residual blocks, the number of filters within these blocks, and the size of the final classification block. The final architecture, as described earlier and illustrated in Fig. 6, proved to be a favorable trade-off between accuracy and computational efficiency. The adoption of (2+1)D convolutions within residual blocks significantly contributed to reducing overfitting and accelerating convergence, as further supported by the architectural comparison in Table 2.

Subsequently, we conducted a thorough evaluation of several key training hyperparameters. Our goal was to identify ranges of values that consistently led to robust and stable learning, thereby narrowing the hyperparameter search space for subsequent experiments. The evaluated parameters included the optimizer type, batch size, and the probabilities of applying two data augmentation techniques described earlier. Adam outperformed the SGD in terms of convergence speed and final validation accuracy, and was thus selected as the default optimizer in later experiments. Additionally, small batch sizes (e.g., 16, 32) often led to training instability, rapid overfitting, and oscillations in the validation metrics. Increasing the batch size to 64 or 128 improved training stability. Similarly, applying RandomChannelShuffle with probabilities greater than 0.5 and RandomInvert with

Table 3. Overview of hyperparameters and their tested ranges during model development and training. Values were selected empirically based on validation performance. The columns IndExpA,B, IndExpC, and MergExps show the best-performing hyperparameters for the model trained on data labeled individually by expert A, expert B, expert C, and the merged dataset of all three experts.

| Setting type | Hyperparameter | Value range | IndExpA,B | IndExpC | MergExps |
|---|---|---|---|---|---|
| Data loading | Batch size | {16, 32, 128, 256} | 128 | 128 | 256 |
| | Number of workers | {0, 1, 2} | 1 | 1 | 2 |
| Regularization | Dropout | [0, 0.5] | 0.15 | 0.1 | 0.1 |
| | RandomChannelShuffle | [0, 1] | 1 | 1 | 1 |
| | RandomInvert | [0, 1] | 0.1 | 0.1 | 0.2 |
| Optimization | Optimizer | {SGD, Adam} | Adam | Adam | Adam |
| | Learning rate | (0, 0.01] | 0.0001 | 0.0001 | 0.0001 |
| | Weight decay | (0, 0.01] | 0.00001 | 0.00001 | 0.00001 |
| Training control | Number of epochs | [30, 200] | 100 | 100 | 100 |
| | Early stopping patience | 10 | 10 | 10 | 10 |
| | Early stopping delta | 1 | 1 | 1 | 1 |

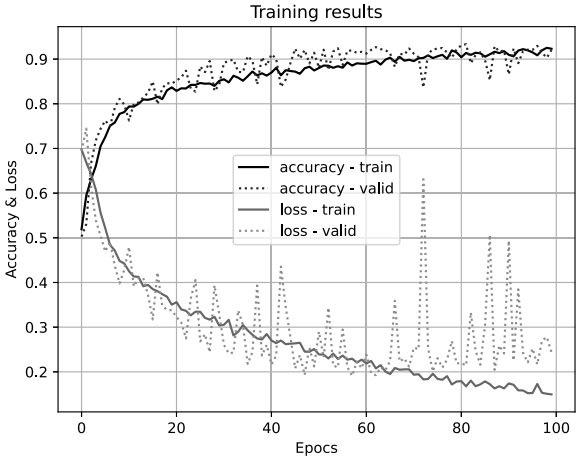



Fig. 8. Training and validation curves (accuracy and loss) for the model trained on data labeled by expert C. The accuracy increases steadily during training, while the loss generally decreases. The observed fluctuations in the validation loss curve are likely due to the limited size of the dataset, which makes the model more sensitive to the specific composition of validation folds. Despite this, the overall training dynamics indicate relatively stable convergence.

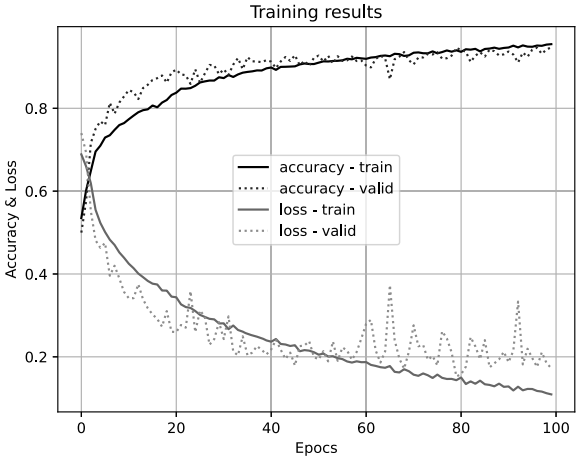



Fig. 9. Training and validation curves (accuracy and loss) for the model trained on the merged dataset (experts A, B and C). Compared to expert-specific models, the merged model demonstrates higher accuracy, faster convergence, and significantly reduced fluctuations in the validation loss. This stability highlights the benefits of using a more diverse and comprehensive dataset.

probabilities above 0.1 reduced overfitting and improved the model's ability to generalize, as reflected in both training and validation performance. Building on the previous optimization steps, we trained and evaluated four separate models: three using data labeled by individual experts and one employing a merged dataset aggregating annotations from all three experts. This experimental setup allowed us to assess both the variability stemming from inter-rater differences and the benefits of increased data diversity when merging annotation sources.

As shown in Table 4, model performance varied notably across datasets. The model trained on expert C's annotations yielded the most favorable results among the expert-specific models, demonstrating both high performance metrics and stable training dynamics, as confirmed by the corresponding learning curves shown in Fig. 8. This suggests that the consistency and internal coherence of expert C's labeling style may have contributed positively to the model's ability to learn relevant patterns.

Table 4. Performance metrics obtained for models trained on datasets labeled by individual experts and on the merged multi-expert dataset. For each model, results for all five folds of the cross-validation procedure are reported, along with the mean and standard deviation across the folds. Among the individual experts, expert C's annotations yielded the highest scores, while the best overall performance was achieved by the model trained on the combined dataset, confirming the benefit of integrating multi-expert annotations in seizure detection tasks.

| | Dataset: IndExpA | | | | | Dataset: IndExpB | | | |
|---|---|---|---|---|---|---|---|---|---|
| Fold no. | Loss | Accuracy | Precision | Recall | Fold no. | Loss | Accuracy | Precision | Recall |
| 1 | 0.2678 | 0.9039 | 0.9734 | 0.8794 | 1 | 0.2787 | 0.8881 | 0.9161 | 0.9162 |
| 2 | 0.2714 | 0.9235 | 0.9603 | 0.9048 | 2 | 0.2948 | 0.8838 | 0.9591 | 0.8699 |
| 3 | 0.2939 | 0.8974 | 0.9690 | 0.8635 | 3 | 0.2735 | 0.8881 | 0.8873 | 0.8873 |
| 4 | 0.2762 | 0.9023 | 0.9474 | 0.8921 | 4 | 0.2873 | 0.8895 | 0.9351 | 0.8960 |
| 5 | 0.2473 | 0.9039 | 0.9295 | 0.9048 | 5 | 0.2435 | 0.9067 | 0.9377 | 0.9046 |
| Mean | 0.2713 | 0.9062 | 0.9559 | 0.8889 | Mean | 0.2756 | 0.8912 | 0.9271 | 0.8948 |
| Std | 0.0168 | 0.0100 | 0.0178 | 0.0177 | Std | 0.0197 | 0.0089 | 0.0270 | 0.0175 |
| | Dataset: IndExpC | | | | | Dataset: MergExps | | | |
| Fold no. | Loss | Accuracy | Precision | Recall | Fold no. | Loss | Accuracy | Precision | Recall |
| 1 | 0.2565 | 0.9029 | 0.9307 | 0.9377 | 1 | 0.1485 | 0.9430 | 0.9576 | 0.9346 |
| 2 | 0.1922 | 0.9261 | 0.9576 | 0.9348 | 2 | 0.1794 | 0.9410 | 0.9453 | 0.9448 |
| 3 | 0.2220 | 0.9333 | 0.9541 | 0.9490 | 3 | 0.1952 | 0.9450 | 0.9617 | 0.9243 |
| 4 | 0.2711 | 0.8957 | 0.9072 | 0.9263 | 4 | 0.1672 | 0.9405 | 0.9820 | 0.9305 |
| 5 | 0.2147 | 0.9217 | 0.9489 | 0.8952 | 5 | 0.1648 | 0.9450 | 0.9454 | 0.9448 |
| Mean | 0.2313 | 0.9159 | 0.9397 | 0.9286 | Mean | **0.1710** | **0.9429** | **0.9584** | **0.9358** |
| Std | 0.0321 | 0.0159 | 0.0209 | 0.0209 | Std | **0.0174** | **0.0021** | **0.0151** | **0.0090** |

Most notably, the model trained on the merged dataset achieved the best overall performance across all metrics, with significantly lower variance across folds. These results confirm the intuition that combining annotation sources can enhance model generalization by exposing it to a broader range of labeling patterns and signal characteristics. As illustrated in Fig. 9, the training process for the merged model also exhibited higher stability, faster convergence, and fewer fluctuations in validation loss compared to expert-specific models. Such findings are particularly valuable in medical applications like seizure detection from EEG, where expert disagreement is common and single-rater annotations may not fully capture the underlying variability of the data.

## 6. Hardware and software configuration

The experiments were carried out on a system running Windows 11, equipped with an NVIDIA RTX 6000 Ada Generation (48 GB VRAM), an AMD Ryzen 9 7950X 16-core processor, and 64 GB of RAM. Python 3.11 was used as the primary programming language, with PyTorch 2.6.0+cu118 serving as the core deep learning framework.

## 7. Conclusions

In this article, we dealt with the issue of epileptic seizure detection based on EEG signals. This topic has been discussed in a large number of publications—several review articles on this topic are mentioned in Section 1. The authors of those works used a variety of approaches, from classical machine learning techniques to various types of deep learning ones. To the best of our knowledge, no previous attempts have been made to use the matching pursuit algorithm for this task, and in particular the time-frequency maps of energy generated on its basis. Our results show that this is absolutely possible and the obtained outcomes are at least satisfactory.

When preparing the data in the form of time-frequency maps, it was arbitrarily assumed that they would be in the shape of squares with a size of $64\times64$ pixels. As for the first assumption, the shape of the maps is not of major importance—it is only crucial that all maps have the same shape. The map size of $64\times64$ is set solely to ensure an acceptable network training time. Of course, a larger map size allows the presentation of more details. However, the results obtained during our experimens were so good that it was decided not to try larger map sizes. The last arbitrary assumption was to generate a maximum of 50 atoms in the program by Różański (2025). In the future (and for other types of data) one can experiment with other values.

The results of our deep learning experiments additionally highlight the importance of the annotation strategy in supervised medical classification tasks. We observed that, while models trained on data from individual experts yielded satisfactory performance, the inclusion of annotations from all three experts

significantly improved both accuracy and training stability. This suggests that models benefit from the variability introduced by multiple labeling perspectives, which can enhance generalization, especially in domains with inherent subjectivity, such as EEG interpretation.

These findings support the view that not only the representation of input data (e.g., time-frequency maps) but also the structure and origin of labels play a crucial role in the overall effectiveness of machine learning-based diagnostic systems.

The Helsinki database, on which our work relies, has also been used in many other works. The performance of our method can be compared with previous studies focused on this database (Tapani *et al*., 2019; O'Shea *et al*., 2020; Caliskan and Rencuzogullari, 2021; Tanveer *et al*., 2021; Abbas *et al*., 2021; Raeisi *et al*., 2022; Diykh *et al*., 2022; Zhou *et al*., 2024; Nelson *et al*., 2024). In the works cited above, the overall performance was evaluated by the AUC value. Therefore, we cannot directly compare our outcomes. Most of the results achieve the AUC value higher than 90%. In our work, we summarize the obtained results using metrics typical of deep learning: loss, accuracy, as well as calculating precision and recall metrics. Accuracy values exceed 90%, with the best results being around 95%. These are roughly equivalent to those in the above cited works, including the systematic review and meta-analysis reports presented in Section 1.

A literature search on the use of time-frequency maps (see, for example, the references listed in Section 1) indicates very frequent use of spectrograms as time-frequency maps. Therefore, experiments with this technique were also conducted and the results are stored in the GitHub electronic supplement (see Section 8). A suitable spectrogram-based dataset is created by executing the *bin_to_RData_STFT.R* scripts. Comparing the results, the following conclusions can be drawn: the results for datasets prepared separately for each of the three experts are slightly better for the MP-based method. In turn, for the merged dataset (experts A, B, C), slightly better results were observed for the spectrogram-based method. In other words, for smaller sets, MP yields better results than spectrograms. This is undoubtedly an argument in favor of MP-based datasets. Medical datasets are often not as large as we would like. Therefore, a method that performs better on a smaller dataset (in our case, MP vs. spectrogram) will be preferred.

Finally, it should also be noted that the method proposed in this paper does not have to be limited to EEG data. In principle, any time-series data can be used.

## 8. Data and code availability

All data analyzed during this study, as well as the R and Python source codes which allow reproducing all the results presented in the paper, can be downloaded from the GitHub electronic supplement available at `https://github.com/artur-gramacki/esd-mp`. Begin by running the *__run_it_first__.R* script. To avoid unexpected problems, we suggest first reviewing the four *important notes* opening the script.

At the beginning of the *__run_it_first__.R* script, the required directory structure is created (or recreated). Using all the input edf files from the `https://zenodo.org/record/4940267` repository will result in a very long generation of the final datasets. Therefore, for testing purposes, we suggest using only two selected edf files (set the appropriate values for the *s.IDs* and *ns.IDs* variables in the *edf_to_bin.R* script). For convenience, in the *data* directory in the repository there are links to download the already generated complete hdf5 files.

In addition to the data preprocessing scripts, the repository also contains a complete deep learning pipeline in the *deep_learning* folder, which allows end-to-end training and evaluation of neural network models. The pipeline can be executed directly from the *pipeline_example.ipynb* notebook, which sequentially loads the preprocessed datasets, initializes the model, trains it on the training set, evaluates it on the validation and test sets, and saves all relevant artifacts including model weights and training history. This setup enables users to reproduce all reported results efficiently, or to experiment with modified parameters and datasets.

## References

Abbas, A.K., Azemi, G., Ravanshadi, S. and Omidvarnia, A. (2021). An EEG-based methodology for the estimation of functional brain connectivity networks: Application to the analysis of newborn EEG seizure, *Biomedical Signal Processing and Control* **63**: 102229, DOI: 10.1016/j.bspc.2020.102229.

Akbar, W., Soomro, A., Hussain, A., Hussain, T., Ali, F., Haq, M.I.U., Attar, R.W., Alhomoud, A., Alzubi, A.A. and Alsagri, R. (2024). Pneumonia detection: A comprehensive study of diverse neural network architectures using chest x-rays, *International Journal of Applied Mathematics and Computer Science* **34**(4): 679–699, DOI: 10.61822/amcs-2024-0045.

Bai, L., Litscher, G. and Li, X. (2025). Epileptic seizure detection using machine learning: A systematic review and meta-analysis, *Brain Sciences* **15**(6), DOI: 10.3390/brainsci15060634.

Bajaj, V. and Pachori, R.B. (2013). Automatic classification of sleep stages based on the time-frequency image of EEG signals, *Computer Methods and Programs in Biomedicine* **112**(3): 320–328, DOI: 10.1016/j.cmpb.2013.07.006.

Caliskan, A. and Rencuzogullari, S. (2021). Transfer learning to detect neonatal seizure from electroencephalography signals, *Neural Computing and Applications* **33**: 12087–12101, DOI: 10.1007/s00521-021-05878-y.

Carmo, A.S., Abreu, M., Baptista, M.F., de Oliveira Carvalho, M., Peralta, A.R., Fred, A., Bentes, C. and da Silva, H.P. (2024). Automated algorithms for seizure forecast: A systematic review and meta-analysis, *Journal of Neurology* **271**: 6573–6587, DOI: 10.1007/s00415-024-12655-z.

Diykh, M., Miften, F.S., Abdulla, S., Deo, R.C., Siuly, S., Green, J.H. and Oudahb, A.Y. (2022). Texture analysis based graph approach for automatic detection of neonatal seizure from multi-channel EEG signals, *Measurement* **190**: 110731, DOI: 10.1016/j.measurement.2022.110731.

Dong, X., He, L., Li, H., Liu, Z., Shang, W. and Zhou, W. (2024). Deep learning based automatic seizure prediction with EEG time-frequency representation, *Biomedical Signal Processing and Control* **95**: 106447, DOI: 10.1016/j.bspc.2024.106447.

Durka, P.J. (2007). *Matching Pursuit and Unification in EEG Analysis*, Artech House Engineering in Medicine and Biology, Boston/London.

Gramacki, A. and Gramacki, J. (2022). A deep learning framework for epileptic seizure detection based on neonatal EEG signals, *Scientific Reports* **12**: 1–21, Article no. 13010, DOI: 10.1038/s41598-022-15830-2.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA*, pp. 770–778, DOI: 10.48550/arXiv.1512.03385.

Kaczmarek, M., Kowal, M. and Korbicz, J. (2025). Exploring data preparation strategies: A comparative analysis of vision transformer and ConvNeXT architectures in breast cancer histopathology classification, *International Journal of Applied Mathematics and Computer Science* **35**(2): 329–339, DOI: 10.61822/amcs-2025-0023.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization, *arXiv* 1412.6980, DOI: 10.48550/arXiv.1412.6980.

Klem, G.H., Lüders, H.O., Jasper, H.H. and Elger, C. (1999). The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology, *Electroencephalography and Clinical Neurophysiology* **52**: 3–6, DOI: 10.1016/0022-510x(84)90023-6, PubMed ID: 10590970.

Kuś, R., Różański, P.T. and Durka, P.J. (2013). Multivariate matching pursuit in optimal Gabor dictionaries: Theory and software with interface for EEG/MEG via Svarog, *BioMedical Engineering OnLine* **12**: 1–28, Article no. 94, DOI: 10.1186/1475-925X-12-94.

Mallat, S. and Shang, Z. (1993). Matching pursuits with time-frequency dictionaries, *IEEE Transactions on Signal Processing* **41**: 3397–3415, DOI: 10.1109/78.258082.

Miltiadous, A., Tzimourta, K., Giannakeas, N., Tsipouras, M., Glavas, E., Kalafatakis, K. and Tzallas, A. (2022). Machine learning algorithms for epilepsy detection based on published EEG databases: A systematic review, *IEEE Access* **11**: 564–594, DOI: 10.1109/ACCESS.2022.3232563.

Nelson, M., Rajendran, S., Osamah Ibrahim, K. and Hamam, H. (2024). Deep-learning-based intelligent neonatal seizure identification using spatial and spectral GNN optimized with the Aquila algorithm, *AIMS Mathematics* **9**(7): 19645–19669, DOI: 10.3934/math.2024958.

Niedermeyer, E. and da Silva, F.L. (2005). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*, Lippincott Williams & Wilkins, Philadelphia.

O'Shea, A., Lightbody, G., Boylan, G. and Temko, A. (2020). Neonatal seizure detection from raw multi-channel EEG using a fully convolutional architecture, *Neural Networks* **123**: 12–25, DOI: 10.1016/j.neunet.2019.11.023.

Pan, Y., Zhou, Xiaoyu nad Dong, F., Wu, J., Xu, Y. and Zheng, S. (2022). Epileptic seizure detection with hybrid time-frequency EEG input: A deep learning approach, *Computational and Mathematical Methods in Medicine* **2022**, Article ID: 8724536, DOI: 10.1155/2022/8724536.

Raab, D., Theissler, A. and Spiliopoulou, M. (2023). XAI4EEG: Spectral and spatio-temporal explanation of deep learningbased seizure detection in EEG time series, *Neural Computing and Applications* **35**: 10051–10068, DOI: 10.1007/s00521-022-07809-x.

Raeisi, K., Khazaei, M., Croce, P., Tamburro, G., Comani, S. and Zappasodi, F. (2022). A graph convolutional neural network for the automated detection of seizures in the neonatal EEG, *Computer Methods and Programs in Biomedicine* **222**: 106950, DOI: 10.1016/j.cmpb.2022.106950.

Rashed-Al-Mahfuz, M., Moni, M.A., Uddin, S., Alyami, S.A., Summers, M.A. and Eapen, V. (2021). A deep convolutional neural network method to detect seizures and characteristic frequencies using epileptic electroencephalogram (EEG) data, *IEEE Journal of Translational Engineering in Health and Medicine* **9**: 1–12, DOI: 10.1109/JTEHM.2021.3050925.

Różański, P.T. (2024). EMPI: GPU-accelerated matching pursuit with continuous dictionaries, *ACM Transactions on Mathematical Software* **50**: 1–17, DOI: 10.1145/3674832.

Różański, P.T. (2025). *Enhanced Matching Pursuit Implementation (empi)*, `https://github.com/develancer/empi`.

Şengür, A., Guo, Y. and Akbulut, Y. (2016). Time-frequency texture descriptors of EEG signals for efficient detection of epileptic seizure, *Brain Informatics* **3**: 101–108, DOI: 10.1007/s40708-015-0029-8.

Shen, M., Yang, F., Wen, P. and Li, Y. (2024). A real-time epilepsy seizure detection approach based on EEG using short-time Fourier transform and Google-Net convolutional neural network, *Heliyon* **10**(11), DOI: 10.1016/j.heliyon.2024.e31827.

Shoeibi, A., Moridian, P., Khodatars, M., Ghassemi, N., Jafari, M., Alizadehsani, R., Kong, Y., Gorriz, J.M., Ramírez, J., Khosravi, A., Nahavandi, S. and Acharya, U.R. (2022). An overview of deep learning techniques for epileptic seizures detection and prediction based on neuroimaging modalities: Methods, challenges, and future

works, *Computers in Biology and Medicine* **149**: 106053, DOI: 10.1016/compbiomed.2022.106053.

Siddartha, K.M., Yuvaraj, R., Thomas, J. and Jac Fredo, A.R. (2024). Comparative study of time-frequency spectrogram techniques for the classification of seizure types using EEG signals and deep learning models, *Expert Systems with Applications*, DOI: 10.2139/ssrn.4883195 (preprint).

Stevenson, N.J., Clancy, R.R., Vanhatalo, S., Rosén, I., Rennie, J.M. and Boylan, G.B. (2015). Interobserver agreement for neonatal seizure detection using multichannel EEG, *Annals of Clinical and Translational Neurology* **2**(11): 1002–1011, DOI: 10.1002/acn3.249.

Stevenson, N.J., Tapani, K., Lauronen, L. and Vanhatalo, S. (2019). A dataset of neonatal EEG recordings with seizure annotations, *Scientific Data* **6**, DOI: 10.1038/sdata.2019.39.

Tanveer, M.A., Khan, M.J., Sajid, H. and Naseer, N. (2021). Convolutional neural networks ensemble model for neonatal seizure detection, *Journal of Neuroscience Methods* **358**: 109197, DOI: 10.1016/j.jneumeth.2021.109197.

Tapani, K.T., Vanhatalo, S. and Stevenson, N.J. (2019). Time-varying EEG correlations improve automated neonatal seizure detection, *International Journal of Neural Systems* **29**(4), DOI: 10.1142/S0129065718500302.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y. and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA*, pp. 6450–6459, DOI: 10.48550/arXiv.1711.11248.

Truong, N.D., Nguyen, A.D., Kuhlmann, L., Bonyadi, M.R., Yang, J., Ippolito, S. and Kavehei, O. (2018). Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram, *Neural Networks* **105**: 104–111, DOI: 10.1016/j.neunet.2018.04.018.

Türk, O. and Özerdem, M.S. (2019). Epilepsy detection by using scalogram based convolutional neural network from EEG signals, *Brain Sciences* **9**(5), DOI: 10.3390/brainsci9050115.

Tzallas, A.T., Tsipouras, M.G. and Fotiadis, D.I. (2009). Epileptic seizure detection in EEGs using time-frequency analysis, *IEEE Transactions on Information Technology in Biomedicine* **13**(5): 703–710, DOI: 10.1109/TITB.2009.2017939.

Wei, S., Sun, Z., Wang, Z., Liao, F., Li, Z. and Mi, H. (2023). An efficient data augmentation method for automatic modulation recognition from low-data imbalanced-class regime, *Applied Sciences* **13**(5): 3177, DOI: 10.3390/app13053177.

Widmann, A., Schröger, E. and Maess, B. (2015). Digital filter design for electrophysiological data—A practical approach, *Journal of Neuroscience Methods* **250**: 34–46, DOI: 10.1016/j.jneumeth.2014.08.002.

Xi, H., Ren, K., Lu, P., Li, Y. and Hu, C. (2024). SSGait: Enhancing gait recognition via semi-supervised self-supervised learning, *Applied Intelligence* **54**(7): 5639–5657, DOI: 10.1007/s10489-024-05385-2.

Xu, J., Yan, K., Deng, Z., Yang, Y., Liu, J.-X., Wang, J. and Yuan, S. (2024). EEG-based epileptic seizure detection using deep learning techniques: A survey, *Neurocomputing* **610**: 128644, DOI: 10.1016/j.neucom.2024.128644.

Yao, H., Huang, L.-K., Zhang, L., Wei, Y., Tian, L., Zou, J., Huang, J. and Li, Z. (2021). Improving generalization in meta-learning via task augmentation, *International Conference on Machine Learning (ICML 2021)*, pp. 11887–11897, DOI: 10.48550/arXiv.2007.13040, (virtual event).

Zhang, X., Zhou, X., Lin, M. and Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA*, pp. 6848–6856, DOI: 10.48550/arXiv.1707.01083.

Zhou, W., Zheng, W., Feng, Y. and Li, X. (2024). LMA-EEGNet: A Lightweight multi-attention network for neonatal seizure detection using EEG signals, *Electronics* **13**(12), DOI: 10.3390/electronics13122354.

Zou, Z., Chen, B., Xiao, D., Tang, F. and Li, X. (2024). Accuracy of machine learning in detecting pediatric epileptic seizures: Systematic review and meta-analysis, *Journal of Medical Internet Research* **26**: e55986, DOI: 10.2196/55986.

**Mateusz M. Kunik** holds BSc and MSc degrees in mathematics, specializing in mathematical modeling, from the University of Zielona Góra, Poland. He is pursuing his PhD in technical informatics and telecommunications and concurrently serves as an assistant lecturer at the Institute of Control and Computation Engineering of the University of Zielona Góra. His research interests include applications of mathematics and computer science, with a particular focus on deep neural networks with the use in medicine and strength sports. ORCID: 0009-0009-1050-816X.

**Artur Gramacki** received his PhD degree in electrical engineering from the Technical University of Zielona Góra, Poland, in 2000 and his DSc degree in computer science from the Częstochowa University of Technology in 2018. Currently, he is an associate professor at the Institute of Control and Computation Engineering, University of Zielona Góra. In his research and teaching he mainly deals with issues such as relational and NoSQL databases, geographic information systems, data analysis, data mining, machine learning and deep learning, and R Project. ORCID: 0000-0002-1610-9743.

Received: 4 July 2025
Revised: 30 September 2025
Accepted: 1 October 2025