Application of Convolutional Neural Networks to Four-Class Motor Imagery Classification Problem

Tomas Uktveris, Vacius Jusas

Kaunas University of Technology, Software Engineering Department,
Studentu st. 50, Kaunas, Lithuania,
e-mail: tomas.uktveris@ktu.lt, vacius.jusas@ktu.lt

Abstract. In this paper the use of a novel feature extraction method oriented to convolutional neural networks (CNN) is discussed in order to solve four-class motor imagery classification problem. Analysis of viable CNN architectures and their influence on the obtained accuracy for the given task is argued. Furthermore, selection of optimal feature map image dimension, filter sizes and other CNN parameters used for network training is investigated. Methods for generating 2D feature maps from 1D feature vectors are presented for commonly used feature types. Initial results show that CNN can achieve high classification accuracy of 68% for the four-class motor imagery problem with less complex feature extraction techniques. It is shown that optimal accuracy highly depends on feature map dimensions, filter sizes, epoch count and other tunable factors, therefore various fine-tuning techniques must be employed. Experiments show that simple FFT energy map generation techniques are enough to reach the state of the art classification accuracy for common CNN feature map sizes. This work also confirms that CNNs are able to learn a descriptive set of information needed for optimal electroencephalogram (EEG) signal classification.

Keywords: convolutional neural network; motor imagery; feature map; image classification, FFT energy map.

1. Introduction

Motor imagery classification is one of many widespread machine-learning problems of brain-computer interface (BCI) systems. With the need for human mind controlled applications the recording of electroencephalograms (EEG) has emerged as an optimal solution for non-interventional brain activity analysis. The ability to fully understand this brain induced electrical signal would greatly simplify the life for people with disabilities or break the barrier for natural interaction in entertainment industry.

This work focuses on four-class motor imagery problem where the recorded EEG signal is classified into four different classes that correspond to four different human subject imagined motoric actions (left hand, right hand, feet and tongue movement). Even if a simpler two-class (binary) problem achieves good classification performance, the four-class still struggles to reach the same results and requires more scientific investigation.

A relatively new and perspective approach to EEG data classification was found in deep learning branch of machine learning. Convolutional neural network (CNN) is a novel animal visual cortex inspired method for image based classification that has not been widely used with EEG, let alone motor-imagery task. With the abilities to generalize/pool and self-learn the needed features in non-linear ways it can benefit EEG classification. Since EEG motor imagery task lacks accurate solutions the CNN could be the new perspective way to look deeper into the same problem. Regarding its novelty and success in other fields it was chosen as the main tool for four-class EEG motor imagery problem analysis in this paper.

By using CNN for classification subtle fine tuning is required to receive best results. This involves selecting a proper neural network architecture, feature method and feature map size. These nuances and their effect on classification performance are further analyzed and discussed in this paper.

Furthermore, feature extraction and feature map (image) generation methods for classification are of great significance. In simplest cases, the EEG signal and feature vector can be treated as one-dimensional signal. In order to move to two-dimensional image classification, two dimensional features or feature transformation methods are required. Possible techniques for such a task are presented and discussed in this work.

2. Related work

In recent years, an increasing number of papers that use CNN for EEG classification task have been published. Multiple approaches have been proposed for solving motor imagery and other related problems. A short review of the common techniques is presented in the remainder of this section.

CNN was successfully used by Mirowski et al. [1] to predict epileptic seizures from EEG. The authors have proposed to use four types of bivariate statistical properties of the EEG signal as features for classification. They argue that commonly used univariate features (computed on each EEG channel separately) lack the required channel relationship information. Cross-correlation, non-linear interdependence, Lyapunov exponent and wavelet synchrony feature information was packed into 2D images for classification. Prediction accuracy of 70% was achieved. Another work in the field of EEG analysis was dedicated to solving the SSVEP (Steady State Visually Evoked Potential) signal classification problem by Cecotti and Gräser [2], where a subject is introduced to visual stimulation at a specific frequency. A four layer CNN network topology with a Fourier transform filter in second layer was tested. Selected architecture proved to achieve up to 97% classification accuracy. It was noted that the switch from time domain to frequency domain gave a positive effect on the classification performance. However, introduced reliability rejection criteria for each class made the final solution less robust, produced a lot of sample rejections and gave average generalization. Different application of CNN to the SSVEP is described in a paper by Bevilacqua et al. [3]. The authors used a four layer network architecture with a hidden L2 Fast Fourier Transform (FFT) layer for frequency extraction. Due to the nature of the problem the signal analysis was done in frequency domain. Channels Pz, PO3, PO4, Oz (of 10-20 electrode system) were used to record EEG samples at 256Hz within 2 second windows. Images of 4x512 elements were composed of filtered EEG data and used as input for the CNN classifier. Network was trained for 1000 epochs. Mean accuracy of 88% was obtained by this method.

CNN capability of detecting P300 events from EEG was showcased by Cecotti and Gräser [4] with accuracy of 95%. The signal analysis was conducted separately in time and space domains. Images of 64x64 in size created from 64 channels of downsampled EEG data were used for classification. Seven different CNN models were verified. Additionally, the work employed a strategy to use vector based CNN kernels instead of matrix kernels in order to prevent mixing features related to space and time domains. A technique based on trained network first layer weight analysis was used to extract 8 most relevant electrodes for each subject.

Recently, CNN has been used by Manor et al. [5] to solve RSVP (Rapid Serial Visual Presentation) task (where a subject has to detect a target image within five possible categories). The authors introduced, a spatio-temporal regularization penalty for EEG classification to reduce network overfitting. Accuracy of 75% was reached with CNN architecture of three layers having 64x1 convolutional, two pooling and two fully connected filters. Images of 64x64 (64 channels by 64 time samples) were used as input for the network. Advantages of using neural network models against manually designed feature extraction algorithms were presented along with criticism for the manual method for unclear and endless possibilities of combining different methods in an efficient way.

Various techniques directly related to the current motor imagery problem have been proposed over the years in literature. Qin and He in [6] describe an analysis of a two-class motor imagery problem. Authors proposed a technique to analyze the EEG in frequency domain. A time-frequency distribution (TFD) images were constructed based on complex Morlet wavelet decomposition for electrode pairs. The TFDs were subtracted from symmetrical channels to form weight matrices that were used to compute in weighted energy for classification. A Laplacian filter was used for signal preprocessing. Average classification rate of 78% was achieved for this method. Another approach based on energy entropy preprocessing and Fisher class separability criteria was proposed in [7] by Xiao et al. Authors analyzed a two-class motor imagery problem in time-frequency domain. Similar TFD distributions (spectrograms) were constructed from EEG short-term Fourier transform (STFT) data. Three different classification methods were compared. Classification accuracy for the two class problem was 85%. A more complicated approach for 3-class motor imagery analysis was done by Zhou et al. in [8]. The study proposed a new method to extract the MRICs (movement related independent components) and utilized ICA (Independent Component Analysis) spatial distribution patterns for such a task. Different ICA filter designs were tested. ICA filter design was confirmed to be subject invariant. Classification accuracy of 62% was received.

A more recent study by Bai et al. [9] on 4-class motor imagery proposed a novel Wavelet-CSP (Common Spatial Patterns) with ICA-filter method. The EEG artifacts were removed using negative entropy-based ICA. Mean accuracy of 76% was achieved using SVM (Support Vector Machine) classifier.

One of the latest works in the field of CNN and 4-class motor imagery is the paper by Yang et al. [10]. The authors proposed a frequency complementary feature map selection (FCMS) method. ACSP (Augmented CSP) feature filtering was used in their work. Two other feature selection methods - random map selection (RMS) and selection of all feature maps (SFM) were analyzed. FCMS was the best performing method due to its ability to limit the ACSP feature redundancy in different frequency bands. The CNN used 5 layer architecture with 5x5 filters (kernels). The work also demonstrated that CNNs are capable of learning discriminant, deep structure features for EEG classification without relying on the handcrafted features. Average classification accuracy achieved was 69%.

3. Methods of analysis

3.1. Convolutional Neural Networks (CNN)

Convolutional neural networks are biologically-inspired variants of MLPs (multi-layer perceptrons). They have been successfully used for character recognition in the past by LeCun et al. [11] and currently have gained interest from researchers due to performance capabilities. CNNs consist of one or more convolutional layers, with weights of the layer shared across the input. Multiple of such layers form a non-linear "filter" chain. The convolution is designed to handle 2D data, as opposed to other neural networks that operate on 1D vectors. This ability makes the extracted features easier to view and interpret.

3.1.1. Feed-forward neural network

A typical neural network function as presented by Vedaldi and Lenc [12] is defined as:

$$f(x) = f_n(\dots f_2(f_1(\boldsymbol{x};\boldsymbol{w}_1);\boldsymbol{w}_2)\dots,\boldsymbol{w}_n) \tag{1}$$

$$f\colon \mathbb{R}^{M\times N\times K} \rightarrow \mathbb{R}^{M'\times N'\times K'} \tag{2}$$

where $\boldsymbol{x}=(x_1,\dots,x_k)$ is the network layer input (a M×N size image with K channels), $\boldsymbol{w}=(w_1,\dots,w_n)$ is the vector of learned parameters.

3.1.2. Convolution

A 3-dimensional convolution operation with *k'* filter count can be expressed as:

$$y_{i'j'k'} = \sum_{ijk} \boldsymbol{w}_{ijkk'}\boldsymbol{x}_{i+i',j+j',k} \tag{3}$$

where *y* is the output of the convolution.

3.1.3. Pooling

CNN concept of pooling is a form of non-linear down-sampling. Pooling partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum or average value. This way it is possible to reduce the feature size (and computation) as required and provide translation invariance. Pooling function is given by:

$$y_{ijk} = max\{y_{i'j'k}: i \le i' < i+p, j \le j' < j+p\} \tag{4}$$

where *y* is the output, *p* is padding.

3.1.4. Non-linear gating

Typical CNN non-linear filters use linear functions with a non-linear gating function, applied identically to each component of a feature map. The simplest such function is the Rectified Linear Unit (ReLU). Such filter can be written as:

$$y_{ijk} = max\{0, x_{ijk}\}. \tag{5}$$

3.1.5. Normalization

Another important CNN building block is channel-wise normalization. This operator normalizes the vector over feature channels at each spatial location in the input map $\boldsymbol{x}$. The form of the normalization operator is:

$$y_{ijk'} = \frac{x_{ijk}}{\left(\kappa + \alpha \sum_{k\in G(k')} x_{ijk}^2\right)^{\beta}}, \tag{6}$$

where y is the output; *κ*, *α*, *β* are normalization parameters, $G(k)=[k-\lfloor\rho 2\rfloor,k+\lceil\rho 2\rceil]\cap\{1,2,\dots,K\}$ is a group of *ρ* consecutive feature channels in the input map.

3.1.6. Softmax

The operation computes the softmax operator across feature channels and, in a convolutional manner, at all spatial locations. It is a combination of an activation function (exponential) and a normalization operator:

$$y_{ijk} = \frac{e^{x_{ijk}}}{\sum_{t=1}^{D} e^{x_{ijt}}}. \tag{7}$$

3.2. Common Spatial Patterns (CSP)

CSP is a widely adopted signal pre-processing method that decomposes the raw EEG into subcomponents (spatial patterns) having maximum differences in variance as shown by Naeem et al. [13]. Wang et al. in [14] concluded that this technique allows better feature separation in feature space and thus more accurate signal classification. Also, the property of CSP to decrease feature dimensionality is very suitable for EEG data complexity reduction. It has been shown by Uktveris and Jusas in [15] and other works that this method gives a substantial EEG signal classification performance increase, thus is a highly recommended filtering method.

The filter is a spatial coefficient matrix W:

$$S = W^T E$$

where *S* is the filtered signal matrix, *E* is original EEG signal. Columns of *W* denote spatial filters, while inverse, i.e. W^{-1}, are spatial patterns of EEG signal. The criterion of CSP for a two C_1, C_2 class problem is given by:

$$maximize: tr(W^T \textstyle\sum_1 W) \tag{8}$$

$$subject\ to: W^T(\textstyle\sum_1 + \sum_2)W = I \tag{9}$$

where $\sum_1$ and $\sum_2$ are the class covariance matrices.

Solution can be acquired by solving a generalized eigenvalue problem. Since CSP was designed for a binary problem, multiclass solutions are combined of multiple spatial filters.

Due to the broad and positive acknowledgement of CSP, the method was used in the current work to filter EEG data before commencing feature extraction.

3.3. Feature extraction methods

A multitude of EEG feature extraction methods have been studied by Uktveris and Jusas in [15] and other literature. Their output usually is a one dimensional feature vector that can be used for classification. The ability to adapt the algorithms for two-dimensional CNN has not been thoroughly analyzed. It is also important to know if the adapted methods can give similar or better results when applied in 2D for CNN. Thus, a review of the most common feature extraction techniques and their implementations for CNN is presented in this work. A short description of the EEG feature methods that were tested and analyzed in this paper is given next.

3.3.1. Mean channel energy (MCE)

The energy of each *i*-th EEG channel is computed as the mean of squared time domain samples (10). The result is then transformed using a *Box-Cox* [16] transformation (i.e. logarithm) in order to make the features more normally distributed, and finally, the resulting values are combined into a feature vector:

$$y_i = \log\left(\frac{1}{N}\sum_{k=1}^{N} x_i[k]^2\right), i = \overline{1,n}\,. \tag{10}$$

3.3.2. Channel variance (CV)

Variance for each *i*-th EEG channel is the second moment of the signal computed about its mean $\bar{x}$. The result is normalized using *Box-Cox* for final feature vector:

$$y_i = \log\left(\frac{1}{N}\sum_{k=1}^{N} (x_i[k] - \bar{x}_i)^2\right), i = \overline{1,n}\,. \tag{11}$$

An example of a feature map generated using this technique is given in Figure 1.

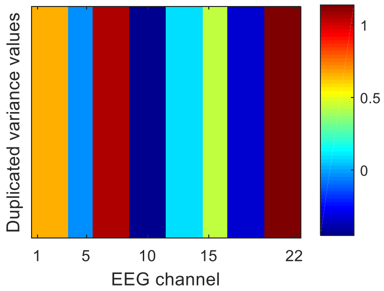

Figure 1. Feature map generated with CV method

3.3.3. Mean window energy (MWE)

This technique computes (12) the mean signal energy of *N* windows of size *W=s/N* for each *i*-th EEG channel (where *s* is EEG channel sample count). The resulting coefficients are *Box-Cox* transformed (12) to form final map:

$$H_{i,j} = \log\left[\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{W}\sum_{k=0}^{W} x_i[k]^2\right)\right], N = \overline{1,p}\,. \tag{12}$$

The maximum window count in experiments was selected as *p=n (*where *n* is EEG channel count*)* in order to form rectangle feature maps.

3.3.4. Principal Component Analysis (PCA)

PCA is a filtering technique that decomposes input signal into main components by using orthogonal transformations (13). Wang et al. showed in [17] that it also can be used to suppress artifacts and noise in EEG signal. The decomposition (13) is carried out multiple times – initially to determine the principal components, secondly – to suppress noisy components at decomposition levels 1-3:

$$\hat{\boldsymbol{x}}_i = |PCA(\boldsymbol{x}_i)|, \qquad i = \overline{1,n}\,. \tag{13}$$

The final feature vector consists of filtered EEG mean energy elements (14) that were normalized via *Box-Cox*:

$$y_i = \log\left(\frac{1}{N}\sum_{k=1}^{N} \hat{x}_i[k]^2\right), i = \overline{1,n}\,. \tag{14}$$

Multi-resolution of 5 levels with Daubechies' least-asymmetric wavelet (4 vanishing moments) was used for the decomposition in this work.

3.3.5. Mean band power (BP)

Algorithm calculates the power of three major frequency bands: 8-14Hz, 19-24Hz and 24-30Hz (corresponding to *Mu*, *Alfa* and *Beta* brain waves) by first band-pass filtering the signal using the 4-th order Butterworth finite impulse response (FIR) filter.

$$\bar{p}[n] = ln\left(\frac{1}{w}\sum_{k=0}^{w} p[n-k]^2\right) \tag{15}$$

The resulting signal is then squared to obtain the power, and a *w*-sized smoothing window operation is performed to filter the signal as shown in (15). The mean power values (16):

$$y_i = \frac{1}{N}\sum_{k=1}^{N} \bar{p}[k], i = \overline{1,n} \tag{16}$$

of each computed band are then used as feature vector components.

3.3.6. Channel FFT energy (CFFT)

As analyzed by Cecotti and Gräser in [2], this method employs the Fast Fourier Transform (FFT) for computing *i*-th EEG channel signal energy estimation in the frequency domain. The FFT result is squared and the sum of all elements is computed:

$$y_i = \log\left(\sum_{k=1}^{N} FFT(\boldsymbol{x}_i)^2\right), i = \overline{1,n} \tag{17}$$

Final feature vector components are formed after *Box-Cox* transformation is applied.

3.3.7. Channel Discrete Cosine Transform (DCT)

Signal energy concentration can be estimated via DCT as shown by Birvinskas et al. in [18]. The sum of squared DCT coefficients of each *i*-th EEG channel forms the feature vector components of this method (18). Features are normalized using *Box-Cox* transform:

$$y_i = \log\left(\sum_{k=1}^{N} DCT(\boldsymbol{x_i})^2\right), i = \overline{1,n}. \tag{18}$$

3.3.8. Time Domain Parameters (TDP)

Time domain parameters compute time-varying energy of the first k derivatives of the i-th EEG channel. Obtained derivative values (19) are smoothed using exponential moving average and a logarithm is taken as given by (20). The resulting signal mean is used in feature vector generation.

$$p_j(t) = \frac{d^j x(t)}{dt^j}, j = 0,1,\dots,k \tag{19}$$

$$y_i = \frac{1}{N}\sum_{k=1}^{N} \ln\left(u \cdot p_j[n] - (1-u) \cdot p_j[n-1]\right), j = \overline{1,m} \tag{20}$$

here u is the moving average parameter, $u \in [0; 1]$.

3.3.9. Teager-Kaiser Energy Operator (TKEO)

TKEO is a more accurate signal energy calculation method that allows to detect high frequency and low amplitude components. Approximation for discrete i-th EEG channel signals is given by (21).

$$\Psi[x[n]] = x^2[n] - x[n-1]x[n+1] \tag{21}$$

$$y_i = \log\left(\frac{1}{N}\sum_{k=1}^{N} \Psi[k]\right), i = \overline{1,n} \tag{22}$$

The components of the final feature vector are computed by using (22).

3.3.10. FFT energy map (FFTEM)

This method generates a 2D feature map from EEG by using FFT. Each i-th EEG channel signal is transformed into frequency domain and forms a single row in the feature map as shown in (23). Full signal window was used to gain a global energy view as opposed to the work by Hu et al. [19], which used short-term FFT windows.

$$\boldsymbol{H}_i = |FFT(\boldsymbol{x_i})|, \qquad i = \overline{1,n} \tag{23}$$

The computed map $\boldsymbol{H}$ is scaled to required feature map size for CNN classification. Figure 2 shows an example result map generated with this method.

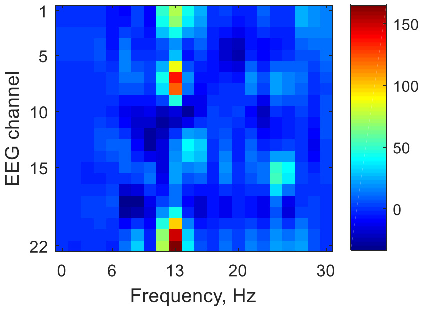

Figure 2. FFT energy map example

3.3.11. Complex Morlet Wavelet Transform (CWT)

CWT is a time-frequency analysis method used by Le Van Quyen et al. in [20] for obtaining wavelet coefficient maps W (24) at specific frequencies, that were analyzed more by Qin and He in [6]:

$$W_x(\tau, f) = \int_{-\infty}^{+\infty} x(u)\,\Psi_{\tau,f}(u)du. \tag{24}$$

All EEG channel signals combined as one *x(t)* signal are convolved with a number of different frequency Morlet Wavelets (25), where $\sigma = n/2\pi f$ and n is the number of wavelet cycles.

$$\Psi_{\tau,f}(u) = \sqrt{f}e^{i2\pi f(u-\tau)}e^{-\frac{(u-\tau)^2}{2\sigma^2}} \tag{25}$$

Finally, the $W_x(\tau, f)$ is decomposed back to initial EEG dimensions and the mean energy coefficients of each channel form a single row (26) in the feature map:

$$\boldsymbol{H}_i = \frac{1}{N}\sum_{k=1}^{N} \left|W_x(\tau, f_i)\right|, \qquad i = \overline{1,n}. \tag{26}$$

In this work, 22 different frequencies were used from [0; 30] Hz range band along with wavelet cycles from range [0.5; 5]. An example output of this method is given in Figure 3.

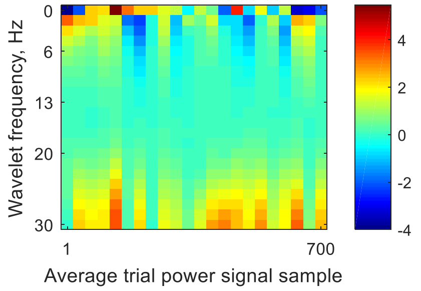

Figure 3. Feature map generated with CWT

3.3.12. Raw signal features (RAW)

RAW is a baseline method that uses the initially pre-processed EEG signal as values for the feature map. Each i-th EEG signal channel directly maps to feature map **H** i-th row as shown in (27):

$$\boldsymbol{H}_i = \boldsymbol{x_i}, \qquad i = \overline{1,n}. \tag{27}$$

If necessary, the resulting feature map is scaled to the required image size for CNN training.

3.3.13. Signal energy map (SEM)

This method is using raw EEG signal energy values for feature map generation. The *Box-Cox* normalized energy of each i-th EEG signal channel is computed and the resulting vector is directly mapped to feature map **H** i-th row as shown in (28):

$$\boldsymbol{H}_i = \log \boldsymbol{x}_i^2, \qquad i = \overline{1,n}. \tag{28}$$

If necessary, the resulting feature map is scaled to the required image size for CNN training. An example map generated with this method is given in Figure 4.

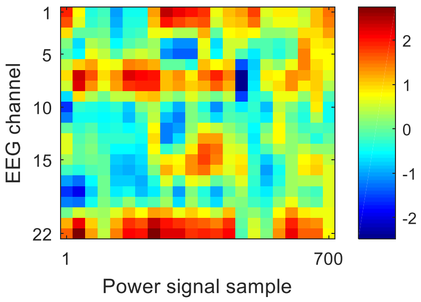

Figure 4. Feature map generated with SEM

3.4. CNN architecture selection

Choosing the correct network architecture for the problem gives a greater probability of getting better classification results. CNN supports serially connected layers. Due to the large number of different layer types it is not trivial to find an optimal chain that closely matches the given problem.

Tests for 11 different CNN architectures were completed. Starting from simplest and ending with more complex ones. The architecture configurations in a simplified notation are given in Table 1. The used notation is explained in Table 2.

Table 1. Different evaluated CNN architectures

#	CNN configuration	Notes
1	IC(4)RPFSO	4 filters
2	IC(4)RP(4)FSO	stride 4
3	IC(8)RPFSO	8 filters
4	ICRPFSO	
5	IC(32)RPFSO	32 filters
6	IC(64)RPFSO	64 filters
7	ICRPCRPFSO	
8	ICRFSO	
9	ICFSO	
10	IC(7x1)RC(1x7)RPFSO	Non-rect filters
11	IC(1x7)RPC(7x1)RPFSO	Non-rect filters

Table 2. CNN layer symbolic notation

Notation	Description (default parameters)
I	input layer of size (44x44x1)
C	convolutional layer (7x7, 16 filters)
R	ReLU layer
P	max pooling layer (2x2, stride 2)
F	fully connected layer (4 classes)
S	softmax layer
O	classification (output) layer

Evaluation results are shown in Figure 5. It can be noted that testing accuracy is around ~65% between most of the configurations. However, training accuracy displays a more dynamic profile from 50% to 80%. In this case, the CNN configuration with the least amount of computational-processing resources (i.e. simplest) should be selected as optimal – 1, 2, 4 or 10.

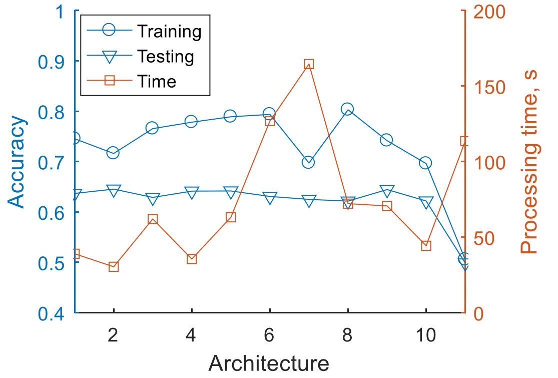

Figure 5. CNN architecture evaluation

3.5. CNN parameter tuning

CNNs are more complex since they have more hyper-parameters than a standard MLP. However, the usual learning rates and regularization constants still apply. CNN training parameters, initial learning rate, momentum, batch size and the number of epochs, must be tuned for best performance. Since a 4D parameter grid based search is too resource intensive, a parameter range scanning approach was carried out to find optimal values.

The momentum value denotes the contribution for the next gradient value from previous iteration in Stochastic Gradient Descent (SGD) method. Larger parameter values decrease the effectiveness of faster learning as shown in Figure 6. In tests, values above 0.6 push the CNN to overfitting and thus decrease generalization and testing accuracy. Value of zero for momentum is not recommended since that invokes a loss of historical gradient learning information.

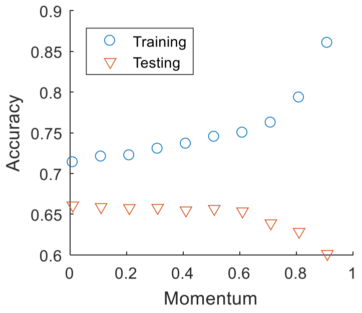

Figure 6. Momentum evaluation

The optimal number of training epochs ensures that the network learns and generalizes the provided features. Excessive epochs deteriorate the testing accuracy since the network is overfitting. Figure 7 shows that the optimal count for training is 400-500 epochs.

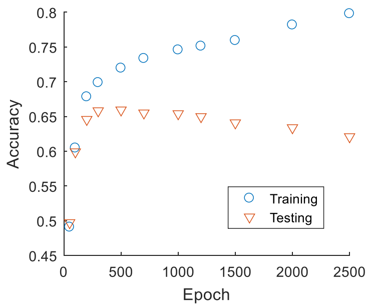

Figure 7. Epoch count evaluation

Batch size is the image count that is used for single epoch training. It has direct effect on the network learning quality as shown in Figure 8. The maximum batch size is the number of total images, e.g. N=288 in experiments. The values lower than N/4 prevent the network from fully maximizing learning efficiency, greater values only increase computational costs at the price of no change in testing accuracy.

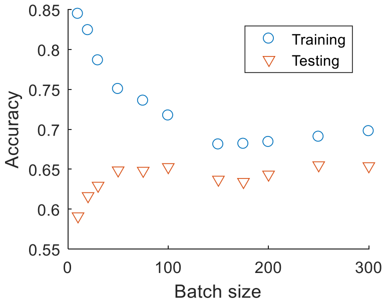

Figure 8. Batch size evaluation

Initial learning rate must be adopted for each problem. Experiments show that the value should be no bigger than 0.1, while the network testing accuracy peak is achieved with values close to 0.01 as shown in Figure 9. Lower values allow to learn fine grained features, while large ones have the tendency to overfit the network.

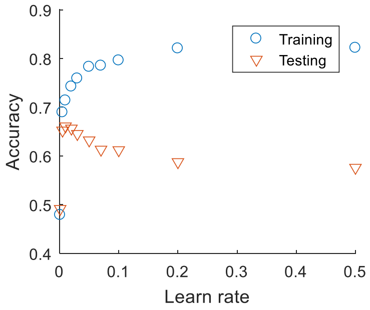

Figure 9. Initial learning rate evaluation

3.6. Feature map generation

3.6.1. Feature duplication

Many feature extraction methods form a single one-dimensional vector of coefficients known as a feature vector. A problem arises since CNNs are designed to process two-dimensional images. Two approaches for image generation form a viable solution.

The first is to interpret the one-dimensional signal as a 2D single row image. However, the negative aspect of this approach is that only a single row CNN filters/kernels will be usable.

The second method, exploiting the CNN translational nature, is to find such a transformation H that allows to convert a 1D signal into 2D:

$$H: \mathbb{R} \rightarrow \mathbb{R}^2. \quad (29)$$

A simple example for such a transformation is to duplicate the feature vector y in both directions to fill the feature map space. Some additional filtering can be applied to new repeated copies. An example of such feature map is given in Figure 10.

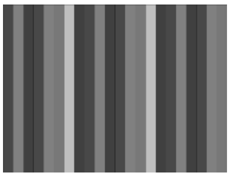

Figure 10. Feature map generated via vector duplication

3.6.2. Feature map scaling

A baseline method and the simplest approach from all feature extraction techniques is to classify the raw EEG signal samples. The raw EEG data form factor of *NxM*, (where *N* is number of channels, *M* is number of samples, *N<<M)* restricts a direct use of it for CNN feature images due to large number of samples. Thus it must be scaled down. Generally, a feature map of *WxH* size (where *W* is width and *H* is height) can be formed by down/up-scaling the raw EEG signal or extracted feature data. The technique of resizing can use bilinear or other type of filtering in order to prevent sharp data transitions, limit noise and smooth out the final feature map. An example of filtering applied to raw EEG feature maps can be seen in Figure 11.

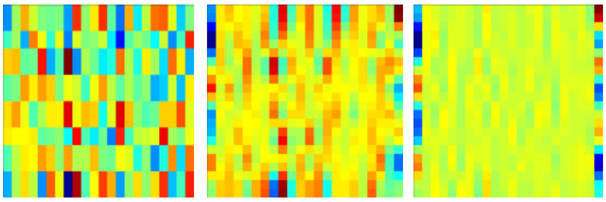

Figure 11. Example of 22x22 raw EEG feature maps (from left - nearest, bilinear, bicubic filtering)

Initial testing results of the three different image filtering techniques for raw EEG signal classification are given in Table 3. Results show ~10% difference in classification accuracy when various filtering techniques are applied. It can be seen that for raw EEG signal analysis nearest filtering method should be used in order to retain original signal details as much as possible. For other feature types the effect could be the opposite.

Table 3. Raw EEG feature map resize filtering accuracy

Filter method	Training	Testing
Nearest	0.47 ± 0.14	0.43 ± 0.11
Bilinear	0.35 ± 0.11	0.33 ± 0.12
Bicubic	0.33 ± 0.10	0.32 ± 0.11

3.7. Feature map and filter dimensions

The problem is to find the right level of granularity in order to create data abstractions at the proper scale, given a particular dataset. Different feature maps and filter sizes were analyzed for the motor imagery problem. Dimensions from 8x8 to 64x64 of feature maps were tested. Test results are given in Figure 12.

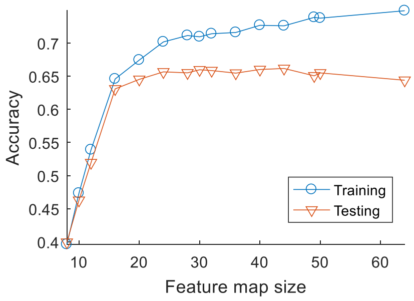

Figure 12. Feature map size evaluation

The plot shows that the optimal feature map size is 24x24 with accuracy of 65%, even though a more accurate solution of 66% exists at size 44x44. Choosing a smaller size feature map ensures faster computation and processing speeds. Also note that the accuracy convergence is reached when the feature map size is at least twice (15x15) the size as the convolution layer filter size (7x7 in the experiment). When the optimal size is reached, the further increase in dimension only introduces extra computational costs.

Convolution layer filter size limits the learning granularity by encompassing fixed size feature map regions. Ten different filter sizes were tested in range [2; 11] for 22x22 feature maps. Test results are displayed in Figure 13.

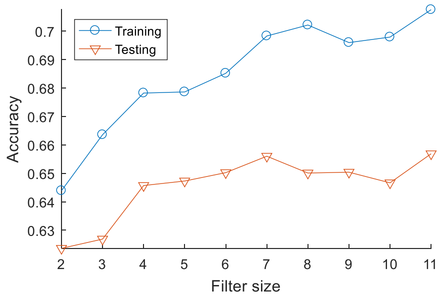

Figure 13. Convolution layer filter size evaluation

The optimal filter size, that gives highest accuracy, is 7x7 and 11x11. Choosing the smaller filter size ensures faster processing speeds. Filters of size 2x2 and 3x3 exhibit too few weights to fully learn the details of the provided data.

4. Experiments

The main purpose of the experimentation activities was to investigate the capabilities of CNN classifier for four-class motor imagery classification problem. Also, to analyze influence of various CNN architectures, feature maps, filter sizes and other parameters to classification accuracy. Experiments were conducted in the analysis step (tuning the CNN network parameters) and also in the main motor imagery classification step (for each subject).

The experiment results were measured and evaluated using normalized accuracy in range [0; 1]. The CNN network parameters were tuned and verified before final classification step. Tests were carried out using ten-fold cross validation. Also, the ability of CNN to learn from feature data was validated visually by inspecting the learned filter/weight images.

Final classification results for each subject are provided in the results section further.

4.1. Dataset

BCI signal Dataset 2a (contributed and described by Brunner et al. [21]) from the BCI IV competition held in 2008 was used for classifier training and testing. The data consists of 22 channels of 250 Hz sample rate recorded EEG signal for 9 healthy test subjects (total 288 motor imagery trials per subject). The EEG signal was bandpass-filtered between 0.5 Hz and 100 Hz and additional 50 Hz notch filter was enabled to suppress line noise. For each subject, two sessions on different days were recorded. During each session and using a cue-paced (synchronous) mode of operation, test subjects were asked to imagine movement of one out of four different motions (left hand, right hand, feet, tongue) for 3 s. Each of the trials (Figure 14) in the dataset started with an audible signal (beep), followed by visual information (cue) on screen to perform one of the mental tasks and a short break after the mental task. Before the experiments additional artifact correction of EEG data was done to discard invalid trials as described in [21] by Brunner et al. The corrected EEG data were bandpass-filtered between 7 Hz and 30 Hz in order to cover *mu* and *beta* rhythm bands.

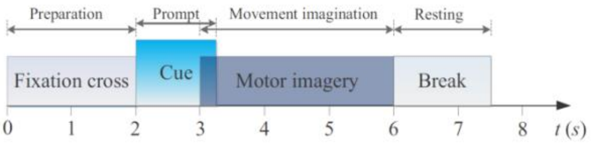

Figure 14. Single trial timing scheme

4.2. Implementation details

Software code for experiments was implemented in MATLAB 2016b/9.1 numerical computation environment. CNN is a new MATLAB functionality (starting from the 2016a/9.0 version), which uses GPU processor for parallel computations. Other alternatives for convolutional neural networks exist such as the open source MatConvNet library created by Vedaldi and Lenc [12], however, the library was left as an option for future CNN evaluations. Parts of the open source BioSig library for biomedical signal processing and imaging were used in EEG signal analysis.

CNN convolution layer initial filter weights in all tests were set to have a Gaussian distribution with a mean of 0 and standard deviation of 0.01. The default for the initial bias was 0.

All MATLAB source code used in this work with results are available at the Github Git repository: https://github.com/tomazas/itc2017.

4.3. Results

Final classification results were obtained after analysis and CNN parameter fine-tuning step. A CNN with initial learning rate of 0.01, momentum of 0.1, batch size of 128, 200 epochs and architecture I(22x22)C(4x4,16)RPFSO was trained and tested for final evaluation on all subjects. Results were verified by using 10-fold cross-validation scheme. The accuracies with their standard deviation values are displayed in Table 4. From the results, it can be seen that the best performing approach (70% in training) and (68% in testing) is the FFT energy map method. The second and third best methods in tests are the Channel variance (68%/61%) and Signal energy map (67%/61%) features. The lowest accuracy of (41%/31%) was achieved by the TDP feature method.

Table 4. Classification results for feature methods

Method	Training	Testing
MCE	0.66 ± 0.19	0.58 ± 0.20
CV	**0.68 ± 0.18**	**0.61 ± 0.22**
MWP	0.66 ± 0.19	0.58 ± 0.20
PCA	0.61 ± 0.16	0.55 ± 0.20
BP	0.52 ± 0.18	0.39 ± 0.11
CFFT	0.66 ± 0.19	0.58 ± 0.20
DCT	0.54 ± 0.17	0.42 ± 0.11
TDP	0.41 ± 0.11	0.31 ± 0.07
TKEO	0.43 ± 0.12	0.34 ± 0.05
FFTEM	**0.70 ± 0.18**	**0.68 ± 0.20**
CWT	0.46 ± 0.10	0.43 ± 0.13
RAW	0.48 ± 0.14	0.37 ± 0.11
SEM	**0.67 ± 0.18**	**0.61 ± 0.20**

5. Conclusions

This work analyzed Convolutional Neural Networks and their application to four-class motor-imagery based problem. After an in-depth CNN analysis and parameter fine-tuning, promising results were achieved for the selected problem. The FFT energy map method demonstrated the best feature determination abilities and achieved 68% mean testing accuracy for all the BCI IV competition 2a dataset subjects. The gained accuracy is slightly better than in new techniques proposed by Tabar and Halici in [22] and similar to more complex state-of-the-art EEG analysis techniques by Yang et. al [10]. The use of simpler feature extraction methods like FFT energy map shows a high CNN method potential for motor imagery EEG analysis.

Further work will continue in order to provide more efficient feature extraction methods favoring processing speed and accuracy.

Acknowledgments

We would like to present our thanks to anonymous reviewers for their helpful suggestions.

References

[1] **P. Mirowski, Y. LeCun, D. Madhavan, R. Kuzniecky.** Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG, *IEEE Workshop Machine Learning Signal Processing*, 2008, https://doi.org/10.1109/MLSP.2008.4685487

[2] **H. Cecotti, A. Gräser,** Convolutional neural network with embedded fourier transform for EEG classification, *Proceedings of the 19th International Conference on PatternRecognition (ICPR '08),* 2008, 1–4, https://doi.org/10.1109/ICPR.2008.4761638

[3] **V. Bevilacqua et al.**, A novel BCI-SSVEP based approach for control of walking in Virtual Environment using a Convolutional Neural Network, *International Joint Conference on Neural Networks (IJCNN)*, 2014, 4121-4128, https://doi.org/10.1109/IJCNN.2014.6889955

[4] **H. Cecotti, A. Gräser,** Convolutional neural networks for P300 detection with application to brain–computer interfaces, I*EEE Trans. Pattern Anal. Mach. Intell.*, 2011, Vol. 33, No. 3, 433-445, https://doi.org/10.1109/TPAMI.2010.125

[5] ***R. Manor, A. B. Geva***, Convolutional Neural Network for Multi-Category Rapid Serial Visual Presentation BCI, *Frontiers Media S.A*, 2015, https://dx.doi.org/10.3389%2Ffncom.2015.00146

[6] **L. Qin, B. He.** A wavelet-based time-frequency analysis approach for classification of motor imagery for brain-computer interface applications, *J. Neural Eng.*, 2005, Vol. 2, 65-72, https://doi.org/10.1088/1741-2560/2/4/001

[7] **D. Xiao, Z. D. Mu, J. F. Hu.** Classification of motor imagery EEG signals based on energy entropy, 2009 *International Symposium on Intelligent Ubiquitous Computing and Education*, 2009, 61-64, https://doi.org/10.1109/IUCE.2009.57

[8] **Zhou, B. , Wu, X. , Zhang, L. , Lv, Z. and Guo, X**. Robust Spatial Filters on Three-Class Motor Imagery EEG Data Using Independent Component Analysis. *Journal of Biosciences and Medicines*, 2014, Vol. 2, 43-49, http://dx.doi.org/10.4236/jbm.2014.22007

[9] **X. Bai, X. Wang, S. Zheng, M. Yu**. The offline feature extraction of four-class motor imagery EEG based on ICA and Wavelet-CSP, *Control Conference (CCC)*, 2014, 7189-7194, https://doi.org/10.1109/ChiCC.2014.6896188

[10] **H. Yang, S. Sakhavi, K. K. Ang, C. Guan,** On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification, *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, 2620-2623, https://doi.org/10.1109/EMBC.2015.7318929

[11] **Y. LeCun, L. Bottou, Y. Bengio, P. Haffner**. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324, https://doi.org/10.1109/5.726791

[12] **A. Vedaldi, K. Lenc**, MatConvNet - Convolutional Neural Networks for MATLAB, *Proc. of the ACM Int. Conf. on Multimedia*, 2015, https://arxiv.org/abs/1412.4564

[13] **M. Naeem, C. Brunner, G. Pfurtscheller.** Dimensionality Reduction and Channel Selection of Motor Imagery Electroencephalographic Data, *Computational Intelligence and Neuroscience*, 2009, http://dx.doi.org/10.1155/2009/537504

[14] **Y. Wang, S. Gao, X. Gao.** Common Spatial Pattern Method for Channel Selection in Motor Imagery Based Brain-Computer Interface, *IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, 5392-5395, https://doi.org/10.1109/IEMBS.2005.1615701

[15] **T. Uktveris, V. Jusas.** Comparison of Feature Extraction Methods for EEG BCI Classification, *Information and Software Technologies: 21st International Conference*, 2015, 81-92, https://doi.org/10.1007/978-3-319-24770-0_8

[16] **G. E. P. Box, D. R. Cox.** An analysis of transformations, Journal of the Royal Statistical Society, Series B, 1964, Vol. 26, No. 2, 211-252, http://www.jstor.org/stable/2984418

[17] **D. Wang, D. Miao, G. Blohm**. Multi-Class Motor Imagery EEG Decoding for Brain-Computer Interfaces. *Frontiers in Neuroscience*, 2012, https://doi.org/10.3389/fnins.2012.00151

[18] **D. Birvinskas, V. Jusas, I. Martišius, R. Damaševičius**. EEG dataset reduction and feature extraction using discrete cosine transform. *UKSim-AMSS EMS 2012: 6th European Modelling Symposium on Mathematical Modeling and Computer Simulation 2012*, 186-191, https://doi.org/10.1109/EMS.2012.88

[19] **J. Hu, Z. Mu, D. Xiao.** Application of Energy Entropy in Motor Imagery EEG Classification. JDCTA, 2009, Vol. 3, 83-90, http://dblp.dagstuhl.de/rec/bib/journals/jdcta/HuXM09

[20] **M. Le Van Quyen, J. Foucher, J. Lachaux, E. Rodriguez, A. Lutz, J. Martinerie, F. J. Varela.** Comparison of Hilbert transform and wavelet methods for the analysis of neuronal synchrony. *Journal of Neuroscience Methods*, 2001, Vol. 111, 83-98, https://doi.org/10.1016/S0165-0270(01)00372-7

[21] **Brunner, C., et al.,** BCI Competition 2008 – Graz data set A, 2008, http://www.bbci.de/competition/iv/desc_2a.pdf

[22] **Y. R. Tabar, U. Halici.** A novel deep learning approach for classification of EEG motor imagery signals, *Journal of Neural Engineering*, 2016, Vol. 14, No. 1, https://doi.org/10.1088/1741-2560/14/1/016003