Data in brief 25 (2019) 104101

Contents lists available at ScienceDirect

Data in brief

journal homepage: www.elsevier.com/locate/dib

Data Article

The ICLabel dataset of electroencephalographic (EEG) independent component (IC) features

Luca Pion-Tonachini [a, b, *], Ken Kreutz-Delgado [b, c], Scott Makeig [a]

[a] Swartz Center for Computational Neuroscience, University of California San Diego, 9500 Gilman Drive, La Jolla CA 92093, USA
[b] Department of Electrical and Computer Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla CA 92093, USA
[c] Pattern Recognition Laboratory, University of California San Diego, 9500 Gilman Drive, La Jolla CA 92093, USA

ARTICLE INFO

Article history:
Received 13 May 2019
Accepted 27 May 2019
Available online 8 June 2019

Keywords:
EEG
ICA
Classification
Crowdsourcing

ABSTRACT

The ICLabel dataset is comprised of training and test sets of a set of spatiotemporal features of electroencephalographic (EEG) independent components (IC). The **ICLabel training set** feature sets were computed for over 200,000 EEG ICs from more than 6,000 existing EEG recordings. More than 8,000 of these ICs have accompanying crowdsourced IC labels across seven IC categories: **Brain, Muscle, Eye, Heart, Line Nosie, Channel Noise,** and **Other**. The feature-sets included in the ICLabel dataset are scalp topography images, channel-based scalp topography measures, power spectral densities (PSD) measures (median, variance and kurtosis) and autocorrelation functions, equivalent current dipole (ECD) model fits for single and bilaterally symmetric dipole models, plus features used in several published IC classifier approaches. The **ICLabel test set** is comprised of 130 ICs from 10 datasets not included in the training set. Each of the test set ICs has an associated IC label estimated based on labels provided by six ICA-EEG experts. Files necessary for adding to and amending the dataset are also included, plus a python class containing useful methods for interacting with the dataset, and IC classifications produced by several existing IC classifiers. These data are linked to the article, "ICLabel: An automated electroencephalographic independent

DOI of original article: https://doi.org/10.1016/j.neuroimage.2019.05.026.
* Corresponding author. Swartz Center for Computational Neuroscience, University of California San Diego, 9500 Gilman Drive, La Jolla CA 92093, USA.
E-mail address: lpionton@ucsd.edu (L. Pion-Tonachini).

https://doi.org/10.1016/j.dib.2019.104101

component classifier, dataset, and website" [1]. An active tutorial and crowdsourcing website is available: i*clabel.ucsd.edu/tutorial/overview*.

Specifications Table

Subject area	*Neuroscience*
More specific subject area	*EEG, Independent Component Analysis, Metadata*
Type of data	*Feature-sets computed from independent components of EEG data acquired in many different task paradigms.*
How data was acquired	*Computed from available EEG datasets*
Data format	*Anonymized, Processed, Partially-Normalized*
Experimental factors	*Over 200,000 ICs from more than 6,000 datasets*
Experimental features	*Various*
Data source location	*San Diego, CA, USA*
Data accessibility	*Data is available through G-Node*
Related research article	*Pion-Tonachini, L., Kreutz-Delgado, K., Makeig, S., "ICLabel: An automated electroencephalographic independent component classifier, dataset, and website." Submitted to NeuroImage.*

Value of the data

- This dataset contains extensive summary statistics for over 200,000 independent components (ICs) of high-density EEG datasets, a subset of which are labeled.
- The data can be used to develop and evaluate EEG independent component classifiers.
- The EEG recordings included in this dataset encompass many experimental paradigms, recording environments, preprocessing recipes, and blind source separation algorithms.
- The data could be used in combination with other similar datasets.
- Meta-analysis can be performed on this dataset to learn common properties of EEG independent components including EEG effective brain sources.

1. Data

The ICLabel dataset is comprised of files containing sets of EEG IC features from a wide variety of found, anonymized EEG recordings, plus files containing IC labels for a subset of those components and a *sqlite* database of the class label submissions used to estimate the IC labels. The files used to extract the IC features included in the ICLabel dataset are included in the folder *features/*. Feature extraction is performed using the MATLAB function *ICL_feature_extraction_full.m*. The files needed to combine the crowd labels from the *sqlite* database into useable label estimates are included in the folder *labels*/and use the python scripts *CLLDA_for_ICLabel.py* and *CLLDA_for_ICLabel_test.py* for the training set and test set, respectively. The dataset is accompanied by a python class containing methods to load the IC features, to match ICs with their labels, and to preprocess the IC features, plus methods for visualizing some of the IC features. The python class for interacting with the ICLabel dataset is included in the folder *dataset/*. Files containing the actual ICLabel dataset features and labels are in the folders *dataset/features/*and *dataset/labels*, respectively. The data can be found at *https://web.gin.g-node.org/doi/ICLabel-Dataset* https://doi.org/10.12751/g-node.e3ddb5. These data are linked to the article, "ICLabel: An automated electroencephalographic independent component classifier, dataset, and website" [1].

2. Experimental design, materials, and methods

The ICLabel dataset is a compilation of extracted features from found, anonymized EEG datasets in the EEGLAB [2] data format (as **.set* files) that have each been decomposed using independent component analysis (ICA) and have attached channel location information. Features were extracted

from each EEG dataset using MATLAB function *ICL_feature_extractor.m* that returns a matrix of features (number of ICs by number of features). The features extracted are illustrated here:

- **Scalp topography images**, interpolated and extrapolated representations of the spatial pattern by which the IC process projects to the scalp, are calculated using the function *topoplotFast.m*, a modified version of *topoplot.m* from EEGLAB. They are stored as vectors but can be converted back to a 32x32 pixel greyscale image using the method *pad_topo* in the *ICLabelDataset* class in *icldata.py*. An example scalp topography is shown in the top-left of Fig. 1. Training set scalp topographies are stored in file *features_0D1D2D.mat*.
- **Channel-based scalp topography measures** are comprised of channel names and locations along with IC loadings onto each channel. These measures are equivalent to the necessary input for *topoplot.m* mentioned above.
- **Power spectral densities (PSD)** features are calculated by applying the fast Fourier transform to 50%-overlapping 1-s windows and taking the median across windows. These windows are then combined into an estimate of the PSD by taking the median across trials. Measures of PSD stability are also included; these were calculated by computing the variance and kurtosis across windows of each frequency bin. All three measures were calculated from 1 to 100 Hz at 1 Hz intervals using the included file *eeg_rpsd.m*. A sample PSD estimate is shown in the bottom right of Fig. 1. Training set PSD features are stored in file *features_PSD_med_var_kurt.mat*.
- **Autocorrelation functions** are computed up to a time-lag of 1 s and are normalized such that the 0-lag value equals 1 before being up-or-downsampled to 100 Hz. The 0-lag value is not included as it is always identically 1. Autocorrelation functions can be calculated using file *eeg_autocorr.m*, although two other versions (*eeg_autocorr_fft.m* and *eeg_autocorr_welch.m*) are also included to maintain efficient computation on recordings with varying properties. Training set autocorrelation features are stored in file *features_Autocorr.mat*.

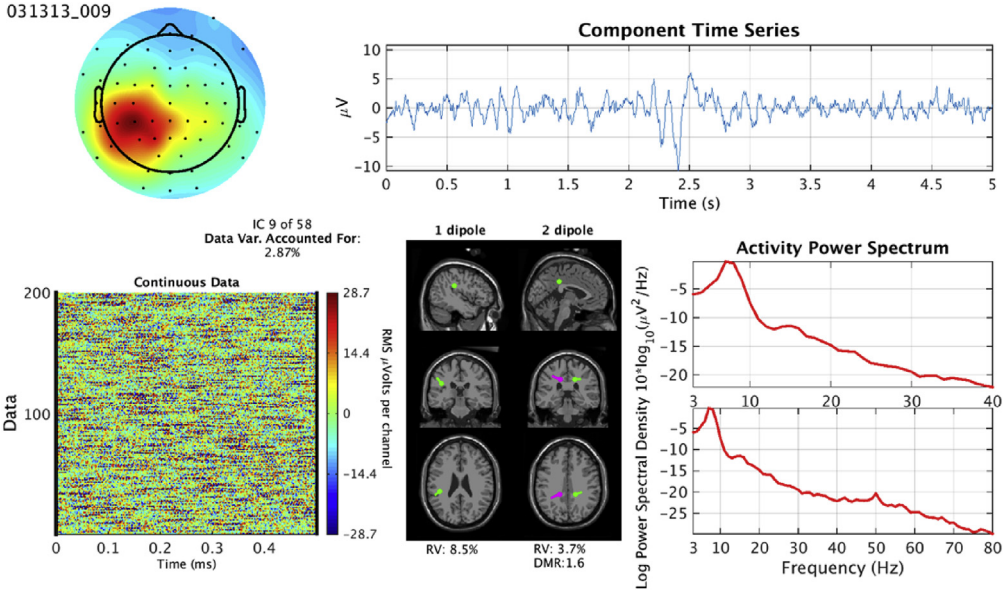

Fig. 1. Graphical summary of an EEG independent component (IC). This is representative of what was shown to volunteer IC labelers who visited iclabel.ucsd.edu. The circle to the top-left is a scalp topography. The time series to the top-right shows IC activity, as does the plot to the bottom-left. The bottom-center illustration shows the single-dipole and bilaterally-symmetric-dipole model fits. The bottom-right illustrates the IC power spectral density (PSD) with two different frequency scales. RV stands for "residual variance", or how well the dipole fit models the data. DMR stands for "dipole moment ratio" which is the ratio of the bilaterally-symmetric stronger to weaker dipole moment norms.

- **Equivalent current dipole (ECD) model fits** were calculated using the *dipfit* plug-in for EEGLAB. Each IC was modeled twice: using a single equivalent dipole model and using a bilaterally-symmetric dual-dipole model (with orientations of the two dipoles free to differ). Each model contains a three-element dipole position per dipole, a three-element dipole moment per dipole, and a scalar value for the residual variance of the IC scalp projection after subtracting the learned model. An example of the single- and dual-dipole fits are illustrated on the bottom-center of Fig. 1. Training set ECD features are stored in file *features_0D1D2D.mat.*
- **Handcrafted features** used in several published IC classifiers (ADJUST [3], FASTER [4], SASICA [5]) were computed using code extracted from the SASICA plug-in for EEGLAB. Additional descriptive features were also included. Most of these features can be calculated using the function *myeeg_SASICA.m* and are summarized in Table 1 (with more details in Ref. [5] The measures not computed by *myeeg_SASICA.m* are simple properties of the dataset. Although "signal to noise ratio" is included in the files, the ICLabelDataset python class removes that feature when loading the dataset, as it is unusable for most datasets. Training set handcrafted features are stored in file *features_0D1D2D.mat*.

All test set features are stored in features_testset_full.mat.

Redundant IC labels were gathered from dozens of volunteer EEG researchers for a subset of ICs ($n > 8{,}000$) in the ICLabel training set using the ICLabel website (*iclabel.ucsd.edu/tutorial/overview*). The redundant labels were compiled into a single, unique probabilistic label per IC using crowd labeling latent Dirichlet allocation (CL-LDA) [6] using file *CLLDA_for_ICLabel.py*. Two training-set label options were computed (1) using all submitted labels from labelers who submitted at least ten IC label suggestions and (2) using only the labels submitted by the expert who contributed the most labels to the database. These estimated labels are stored in *ICLabels_expert.pkl* and *ICLabels_onlyluca.pkl*, respectively. Similarly, six expert IC labelers submitted labels for each of the 130 ICs in the ICLabel test set. These labels were also compiled into one unique probabilistic label per IC using CL-LDA with the file *CLLDA_for_ICLabel_test.py*, the results of which are stored in *ICLabels_test.pkl*. The raw label suggestions collected from the ICLabel website are stored in the *sqlite* database *anonymized_database.sqlite*.

In addition to the training and test labels for the ICs in the *ICLabel dataset*, the ICLabelDataset python class in *icldata.py* provides several methods useful for managing and processing the feature-sets and labels comprising the ICLabel dataset. Some of them are listed here:

Table 1
"Handcrafted" IC features available in the ICLabel dataset.

Feature	Origin	Description
Autocorrelation	SASICA	Autocorrelation with a lag of 20 ms
Focal scalp topography	SASICA	Interpolated scalp map showing IC projection polarity and relative strength across the scalp using EEGLAB *topoplot* conventions.
Signal to noise ratio	SASICA	Trial-based measure of evoked potentials (present in file *features_0D1D2D.mat* but ignored by *ICLabelDataset* data loading methods)
Signal variance	SASICA	Sample variance of the IC process activity
Temporal kurtosis	ADJUST	Sample kurtosis of the IC process activity
Spatial eye difference (SED)	ADJUST	Measure of anterior horizontal scalp projection distribution
Spatial average difference (SAD)	ADJUST	Difference between absolute projections to anterior and posterior scalp regions
Differential variance	ADJUST	Difference between squared projections to anterior and posterior scalp regions
Maximum epoch variance (MEV)	ADJUST	Ratio of maximum and mean trial variance
Median gradient value	FASTER	Median of first derivative of IC activity
Kurtosis of spatial map	FASTER	Spatial kurtosis of IC scalp projections
Hurst exponent	FASTER	Measure of time series "memory"
Channel count	–	Number of EEG electrode channels
IC count	–	Number of ICs in the decomposition
Scalp topography radius	–	Radius of the scalp topography image (using EEGLAB *topoplot* conventions)
Epoched dataset	–	Whether the IC activity is continuous or a series of trials
Sample rate	–	Sampling rate of the IC time series
Data points		Total number of sample points in the recording

- ***load_data***: Loads only the requested feature-sets and keeps only the ICs which have all the requested feature-set available. All IC labels are then matched with the appropriate IC features and organized into two groups: labeled and unlabeled ICs. Finally, ICs with non-numeric values (*Inf* and *NaN*) are removed.
- ***load_semi_supervised***: Internally calls *load_data* prior to separating and preprocessing all the individual feature-sets. The processing applied is by no means definitive as there are many other reasonable normalizations which may be applied to the feature sets in addition to those used in the *ICLabelDataset* class.
- ***load_channel_features***: Loads all available channel-based scalp topography measures.
- ***load_test_data***: Similar to *load_semi_supervised* but loads the ICLabel test set ICs and labels.
- ***load_classifications***: Loads classification from several published IC classifiers for a given number of IC categories (two, three, or five). Classifiers included are MARA [7,8], ADJUST [3], FASTER [4], IC_MARC [9], and EyeCatch [10]. MARA and FASTER are only included in the two-class case (brain and non-brain), ADJUST is also capable of the three-class case (adding the eye category), and IC_MARC is further capable of the five-class case (adding muscle and heart categories). EyeCatch is always included as it classifies ICs as eye and non-eye.

Example code for loading the ICLabel dataset in python:

```
# import ICL dataset class
from icldata import ICLabelDataset
# initialize the class: this is where many of the settings governing loading the
dataset can be specified
icl = ICLabelDataset()
# load the ICLabel training set
icl_train_data = icl.load_semi_supervised()
# load the ICLabel test set
icl_test_data = icl.load_test_data()
# load classifications from previous classifiers with 2 categories (brain and
non-brain)
previous_classifications = icl.load_classifications(2)
```

Acknowledgments

This work was supported in part by a gift from The Swartz Foundation (Old Field NY) and by grants from the National Science Foundation [grant number GRFP DGE-1144086] and the National Institutes of Health [grant number 2R01-NS047293-14A1]. The expert-labeled test set was annotated with help from James Desjardins, Agatha Lenartowicz, Thea Radüntz, Lawrence Ward and Elizabeth Blundon, and Matthew Wisniewski. Their contributions are greatly appreciated.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Pion-Tonachini, K. Kreutz-Delgado, S. Makeig, ICLabel: an automated electroencephalographic independent component classifier, dataset, and website, Neuroimage (2019), https://doi.org/10.1016/j.neuroimage.2019.05.026.
[2] A. Delorme, S. Makeig, EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, J. Neurosci. Methods 134 (2004) 9–21.
[3] A. Mognon, J. Jovicich, L. Bruzzone, M. Buiatti, ADJUST: an automatic EEG artifact detector based on the joint use of spatial and temporal features, Psychophysiology 48 (2) (2011) 229–240.
[4] H. Nolan, R. Whelan, R.B. Reilly, FASTER: fully automated statistical thresholding for EEG artifact rejection, J. Neurosci. Methods 192 (1) (2010) 152–162.
[5] M. Chaumon, D.V. Bishop, N.A. Busch, A practical guide to the selection of independent components of the electroencephalogram for artifact correction, J. Neurosci. Methods 250 (2015) 47–63.

[6] L. Pion-Tonachini, S. Makeig, K. Kreutz-Delgado, Crowd labeling latent Dirichlet allocation, Knowl. Inf. Syst. 53 (3) (2017) 749–765.
[7] I. Winkler, S. Haufe, M. Tangermann, Automatic classification of artifactual ICA-components for artifact removal in EEG signals, Behav. Brain Funct. 7 (1) (2011) 30.
[8] I. Winkler, S. Debener, K.R. Müller, M. Tangermann, On the influence of high-pass filtering on ICA-based artifact reduction in EEG-ERP, in: Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE, IEEE, 2015, August, pp. 4101–4105.
[9] L. Frølich, T.S. Andersen, M. Mørup, Classification of independent components of EEG into multiple artifact classes, Psychophysiology 52 (1) (2015) 32–45.
[10] N. Bigdely-Shamlo, K. Kreutz-Delgado, C. Kothe, S. Makeig, EyeCatch: data-mining over half a million EEG independent components to construct a fully-automated eye-component detector, in: Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE, IEEE, 2013, July, pp. 5845–5848.