



# Deep Feature Learning and Visualization for EEG Recording Using Autoencoders

Yue Yao<sup>(✉)</sup>, Jo Plested, and Tom Gedeon

Research School of Computer Science,  
The Australian National University, Canberra, Australia  
u6014942@anu.edu.au

**Abstract.** In this era of deep learning and big data, the transformation of biomedical big data into recognizable patterns is an important research focus and a great challenge in bioinformatics. An important form of biomedical data is electroencephalography (EEG) signals, which are generally strongly affected by noise and there exists notable individual, environmental and device differences. In this paper, we focus on learning discriminative features from short time EEG signals. Inspired by traditional image compression techniques to learn a robust representation of an image, we introduce and compare two strategies for learning features from EEG using two specifically designed autoencoders. Channel-wise autoencoders focus on features in each channel, while Image-wise autoencoders instead learn features from the whole trial. Our results on a UCI EEG dataset show that using both Channel-wise and Image-wise autoencoders achieve good performance for a classification problem with state of art accuracy in both within-subject and cross-subject tests. A further experiment using shared weights shows that the shared weights technique only slightly influenced learning but it reduced training time significantly.

**Keywords:** Deep learning · Neural network · Autoencoders  
Convolutional neural network · EEG · Brain-Computer Interface

## 1 Introduction

As an important part of Brain-Computer Interfaces (BCIs), EEG has found a variety of interesting and useful applications for users and has become increasingly important in various areas. Especially for the medical field, diagnosis of epilepsy for example, EEG has shown success [1, 2]. Gathered from the scalp, the EEG is a signal containing information about the electrical activity of the brain. Electrodes placed on the scalp are used to detect electrical information from the brain under the scalp, bone and other tissues. Since it is an overall measurement of human brain electrical activity, it may contain a wealth of information. This is the reason why EEG can be applied to diverse areas like personal recognition, disease identification [1], sleep stage classification [3], visual image generation using brain waves [4], and so on.

On the other hand, using EEG signals faces many difficulties. First, being full of information also here means full of noise and interference, making it very hard to extract reliable features. Further, depending on the collection device, EEG will have a

different format, hence it becomes difficult to construct standard algorithms to extract features from EEG. Third, EEG signals have large individual differences, making it hard for cross-subject tests [5] to achieve high accuracy. These three difficulties make EEG feature engineering still a work in progress.

For feature learning tasks in bioinformatics, a wide range of traditional machine learning algorithms have been applied and achieved success. In some areas, such as for bio-signals like EEG, many well-known algorithms have been applied like support vector machine, random forests, Bayesian networks, and hidden Markov models [6]. The good performance of conventional machine learning algorithms relies heavily on features extracted [7]. Traditional learned features are not always as good as we want since they are not always robust and not designed to counter noise. For this reason, we need algorithms that can learn features from big data automatically.

Deep learning, a neural network based technique, is a rising subfield of machine learning. It has achieved great success in computer vision (CV), natural language processing (NLP) and many other areas in recent years. But unlike CV and NLP which have many successful algorithms and datasets using deep learning, bio-information areas have no widely accepted learning algorithms or even a well-known and popular dataset like imagenet [8] in CV. Human brain waves have commonalities and differences, and it is these commonalities and differences which are exactly the properties that we want. We believe that only when all these properties are better understood can we make it possible to design a robust and recognized deep learning method in this area – even deep learning approaches need some understanding of the structure of the data to extract features well. So that is the reason why both learning and visualization are introduced in this paper.

To address these difficulties, deep learning approaches are utilized in this paper to achieve both learning and visualization. Two autoencoder-based techniques are used for feature learning and dimensionality reduction for short time EEG signals. They are referred to as Channel-wise autoencoders and Image-wise autoencoders. Channel-wise autoencoders are inspired by one-dimensional convolutions. For EEG data, the number of channels is often significantly less than the timescale length, forming an unbalanced matrix input. In such settings, for applying convolutional neural networks based techniques, it is usual to perform one-dimensional convolution for feature extraction [6, 9]. Thus, as a first step in our work, we design a group of channel-wise autoencoders which only focus on features from a single channel using simple fully connected layers. The Image-wise autoencoders are designed based on Fast Fourier Transform (FFT) and CNN. Using FFT, we can obtain the three EEG frequency bands, then we use these frequency bands to achieve an RGB-color visualization (an image) [10]. Then, a CNN based autoencoder is designed to extract features from these color images.

## 2 Related Work

Convolutional neural networks (CNNs) are feature extraction networks proposed by Lecun [11], based on the structure of the mammalian visual cortex – thus providing structural information about the data via the network topology. The difference between convolution neural networks and the traditional neural networks is the convolution

layer. We consider the convolution layers as feature extractors. Then, the fully connected layer serves as a ‘classifier’ trying to find decision boundaries between each class. From another point of view, the role of the fully connected layer is similar to the kernel method, warping the high-level feature space to make each class approximately linearly separable.

Much CNN based research has been applied to EEG. Depending on the type of the kernel, CNN based work can be divided into normal CNN as well as frequency-based CNN. Normal CNN takes the raw EEG as the input while frequency-based CNN extracts frequency features from raw EEG. Examples of normal CNN approaches include Deep4net [12] and EEGNet [5]. The SyncNet [13] is the latest example of a frequency-based CNN for EEG. An interesting commonality is that one-dimensional convolutions are often applied among convolution procedures [6, 9].

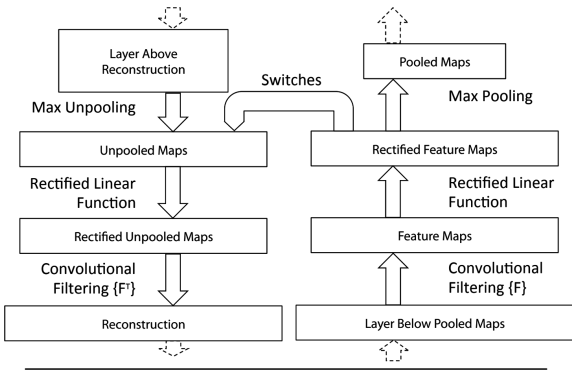
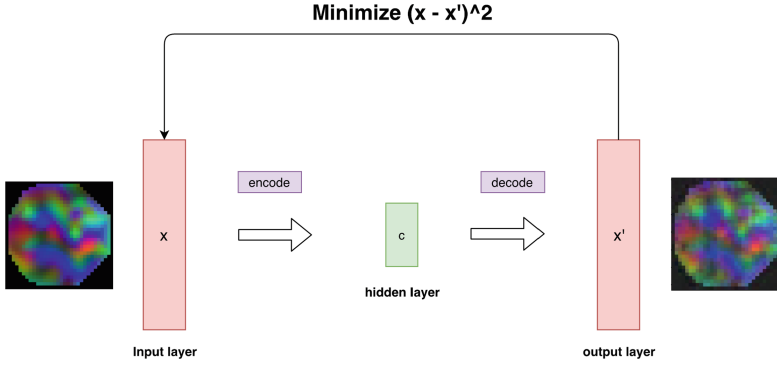


Fig. 1. The structure of DNN [14]

**Deconvolution neural network (DNN)** was first used for visualization in CV by Zeiler and Fergus [14]. A high-level feature map with many high dimensional features is difficult to interpret intuitively. A DNN projects the response value of the specified convolution layer into the input pixel space by reversing the CNN, thus revealing the contribution made by each pixel of the input image to the response value, thus creating a more comprehensible feature map visualization. These operations are shown in Fig. 1. The right side is the process of forward propagation, while the left side is the process of mapping the response value back to the input pixel space. By using a DNN, an autoencoder with CNN as encoder and DNN as decoder can be easily implemented.

**Autoencoder** is a sort of compression algorithm, or dimension reduction algorithm, which has similar properties to Principal Components Analysis (PCA). But compared with PCA, the autoencoder has no linear constraints. The autoencoder structure has been widely used for image compression, for example [15], which inspired us to try an autoencoder based learning algorithm. From Fig. 2, an autoencoder can be divided into two parts, an encoder and a decoder. The number of nodes in the hidden layer is



**Fig. 2.** Structure of autoencoder

generally less than the nodes in the input layer and the output layer. That is, the original input is compressed to a smaller feature vector. In Eq. 1 below,  $\phi$  and  $\psi$  stand for encoder and decoder, respectively, and  $L$  means squared loss. The objective of the autoencoder is to minimize the difference between the input and the generated output. A CNN based autoencoder [16] uses convolution operations as the encoder and deconvolution operations for the decoder, making it better for operating on image data.

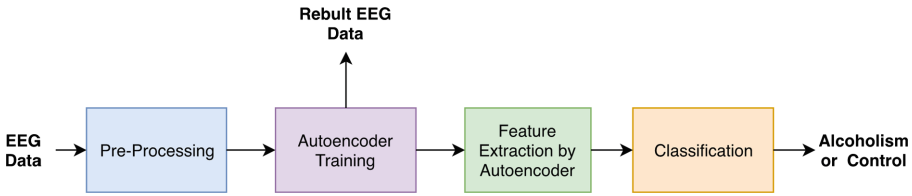
$$\phi, \psi = \operatorname{argmin}_{\phi, \psi} L(X, (\phi \circ \psi)X). \quad (1)$$

A major purpose of our work is to solve the difficulties we identified and advance the state of art in feature abstraction of signal analysis for short time EEG biosignals. Autoencoders are a mature method to extract robust features. Prior to our work, a number of autoencoder related methods have been applied to EEG signals. Stober [17] use convolutional autoencoders with custom constraints to learn features and improve generalization across subjects and trials. It achieved commendable results but it uses CNN directly on the time domain features from EEG signals but not frequency domain features like our methods. But Stober's work inspired us that it could be a general conclusion that the autoencoder based structure can increase the cross-subject accuracy, forming our basic inspiration to try autoencoder based structures.

The most similar work to our model is by Tabar and Halici [18]. They used EEG motor imagery signals and a combined CNN and fully connected stacked autoencoders (SAE) to find discriminative features. They used Short-time Fourier transform (STFT) to build an EEG motor imagery (MI) which is unlike our 3-D electrode location mapping as in our work (described in Sect. 3). Also, their autoencoder design is quite different from ours since they used a CNN followed by an 8-layer SAE. Nevertheless, they have demonstrated that autoencoders can help to learn robust features from EEG signals.

### 3 Methodology

The block diagram shows the general procedure for Channel-wise Autoencoders and Image-wise Autoencoders, as depicted in Fig. 3. We first pre-process the raw EEG data into a useable form. Then feature extraction and dimensionality reduction are done by using autoencoders. Finally, fully connected (FC) layers are utilized to do classification and evaluation. In our procedure, we extract features prior to applying the classification. To achieve this, two kinds of autoencoders are used to enhance features. That is, Channel-wise autoencoders and Image-wise autoencoders.



**Fig. 3.** Structure of general procedure for learning discriminative features based on autoencoders

#### 3.1 Dataset

The dataset we use is from UCI, the EEG dataset from Neurodynamics Laboratory at the State University of New York. It has a total of 122 subjects with 77 diagnosed with alcoholism and 45 control subjects [13, 19]. Each subject has 120 separate trials. If a subject is labeled with alcoholism, all 120 trails belonging to that subject will be labeled as alcoholism. The stimuli they use are several pictures selected from the Snodgrass and Vanderwart picture set. It is a sort time EEG where one trial of EEG signal is of one second length and is sampled at 256 Hz with 64 electrodes. Models are first evaluated using data within subjects, which is randomly split as 7:1:2 for training, validation and testing for one person [5]. We further test them using data across subjects [5], using the same setting as Li [13]. The classification task is to recover whether the subject has been diagnosed with alcoholism or is a control subject. Also, we note that this is not a balanced dataset. It is a two-task classification but alcoholism trials account for more than 70% of the data.

The usual challenges of handling EEG make it more difficult to apply deep learning methods compared with computer vision data or natural language processing data. The UCI EEG dataset is not an exception.

First, a label is usually applied to one trial. But as one trial contains 64 channels and 256 time series data, making it become a  $64 \times 256$  large matrix. In other words, a single EEG trial has  $64 \times 256$  attributes, difficult for a neural network to find meaningful features if treated as 16,384 independent inputs.

Second, EEG is a kind of time-series data but it lacks recognizable patterns in single time slices (1/sampling rate) compared with natural language processing, since each word in NLP often has a specific meaning.

Third, as previous work has shown, if we consider an EEG signal as a picture and directly use a convolution neural network for raw EEG data, there is always a serious problem to determine the size of the kernels to use at each stage [6, 18]. That is, because the original features could be distributed with different time differences in a single trial depending on the scenario (different classification task for example). Furthermore, due to the huge personal differences in EEG data, the cross-subject test result is often far from satisfactory. To address these difficulties, we used two kinds of autoencoders as described in the following.

3.2 Channel-Wise Autoencoders

The key idea of applying a channel-wise autoencoder is to separate the feature extraction procedure into two parts. The channel-wise autoencoder only focuses on features in one channel while the final fully connected layer will combine features across channels to make a final prediction. This is very much like using 1-D convolutions to get channel-wise features followed by a fully connected layer to make a prediction [9].

As shown in Fig. 4. An EEG trial with the  $64 \times 256$  dimensions will be separated into 64  $1 \times 256$  signals, then each signal will be the input for one autoencoder only trained on that channel. These 64 autoencoders are just 2 layers of a fully connected neural network with 16 hidden units in the middle. The input of autoencoders will be normalized to  $[-1, 1]$  and we use a tanh activation function for the output layer to match the output to  $[-1, 1]$  as well. The shared weight technique derived from image compression [15] is also used for signal compression, which takes the transpose of encoder weights for the decoder weights.

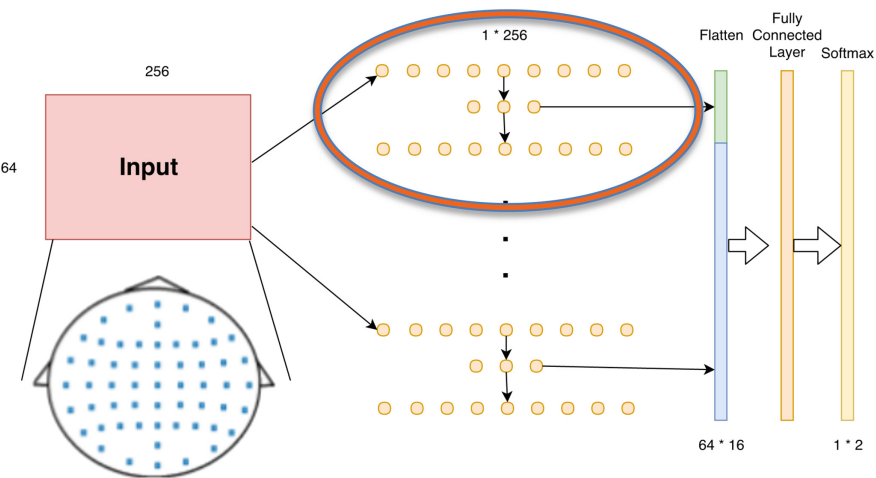


Fig. 4. Structure of channel-wise autoencoders

3.3 Image-Wise Autoencoders

The image-wise autoencoders take the images as input while using a CNN to extract features. The whole procedure is shown in Fig. 5, below is some further explanation.

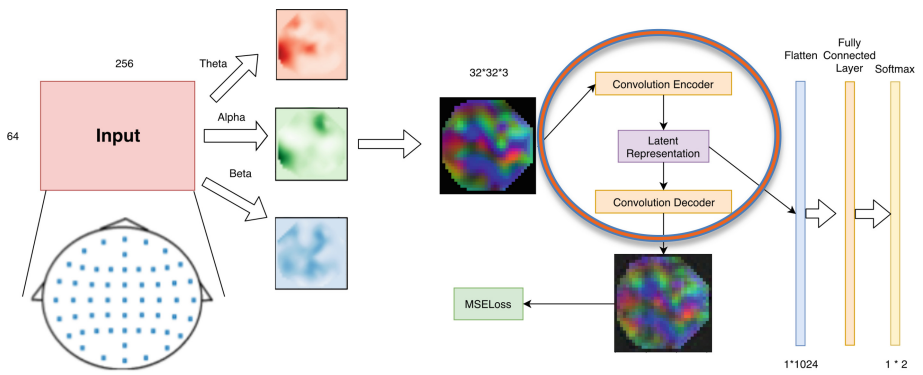


Fig. 5. Structure of image-wise autoencoder

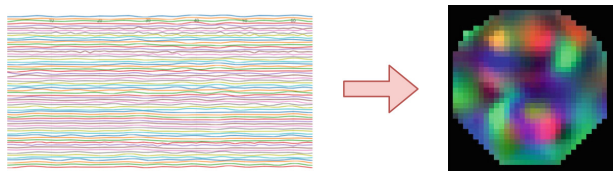


Fig. 6. EEG signal to image example

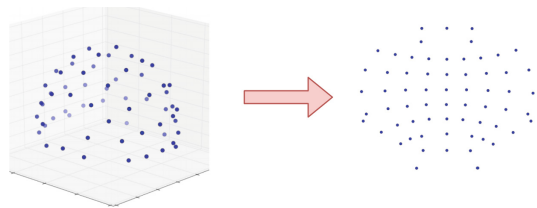


Fig. 7. Transform 3-D coordinate to 2-D coordinate [10]

A. EEG to Image

The method is derived from Bashivan’s work [10]. As shown in Fig. 6, it is a method that combines the time-series information and spatial channel locations information over the scalp in a trial of EEG signals. An FFT is performed on the time series to estimate the

power spectrum of the signal for each trial ( $64 \times 256$ ). Then, three frequency bands of theta (4–7 Hz), alpha (8–13 Hz), and beta (13–30 Hz) are extracted, and the sum of squared absolute values in these frequency bands are used, forming a  $64 \times 3$  map. To form an RGB EEG image, the theta frequency will be the red channel, alpha the green channel and beta the blue channel. For each frequency band ( $64 \times 1$ ), shown in Fig. 7, Azimuthal Equidistant Projection (AEP) also known as Polar Projection is used to map the three-dimensional 64 channel position into two-dimensional positions on a flat surface. That is, all EEG electrodes positions are mapped into a consistent 2-D space because the original EEG electrodes are distributed over the scalp in a three-dimensional fashion. In this way, each  $64 \times 1$  frequency band can be mapped to a  $32 \times 32$  mesh, forming  $32 \times 32 \times 3$  data. The CloughTocher scheme is used for estimating the values in-between the electrodes over the  $32 \times 32$  mesh. Thus, a trial of  $64 \times 256$  EEG signals is transformed to  $32 \times 32 \times 3$  color pictures.

B. Autoencoder design

The design of this CNN based autoencoder is inspired by the CNN for CIFAR-10 [20]. The CIFAR-10 dataset consists of 60,000  $32 \times 32$  color images in 10 classes, with 6,000 images per class, with the same input dimension as our generated EEG pictures. Our encoder and decoder are described in Table 1. Our shared weight CNN Autoencoder is in the same structure as the normal CNN autoencoder but the weight of the three deconvolution layers is fixed and derived from the encoder’s convolution layer. The Rectified Linear Unit (ReLU) is used for activation layers to speed up the training process while dropout is performed after every activation layer to make the model more robust, since it forces all the layers before the dropout to extract redundant representations. Adam optimizer is used with  $1e-4$  learning rate and the batch size is set to 64. Xavier normal initialization is used for convolution kernels.

**Table 1.** The detailed encoder and decoder structure

Encoder	Decoder
Input $32 \times 32 \times 3$ Color Image	Input $16 \times 8 \times 8$ Matrix
$3 \times 3$ conv, $2 \times 2$ max-pooling	$3 \times 3$ deconv, $2 \times 2$ max-un-pooling
ReLU, 0.25 dropout	ReLU, 0.25 dropout
$3 \times 3$ conv, $2 \times 2$ max-pooling	$3 \times 3$ deconv, $2 \times 2$ max-un-pooling
ReLU, 0.25 dropout	ReLU, 0.25 dropout
$3 \times 3$ conv, ReLU	$3 \times 3$ deconv

3.4 Classification Task

The features extracted from channel-wise autoencoders and image-wise autoencoders will be flattened into a long vector, composed of 16 hidden unit representations  $\times 64$  autoencoders in the channel-wise case or  $16 \times 8 \times 8$  matrix in the image-wise case. Then we use a feedforward network with three hidden layers. During training of these



three fully connected layers using  $4e-5$  learning rate, the encoder of both the channel-wise and image-wise autoencoders will also be fine-tuned by the classification loss using a much smaller learning rate ( $1e-7$ ).

## 4 Results and Discussions

Our experiment was to compare the classification accuracy using normal channel-wise autoencoders, shared weight channel-wise autoencoders, normal image-wise autoencoders and shared weight image-wise autoencoders. The code was written in python and pytorch. All experiments were done on an i5-7500 CPU, Nvidia GTX1050Ti, 8g RAM and Windows environment. Below is the classification result for the different classification task.

**Table 2.** Comparison between two image-wise autoencoders

Method	Within accuracy	Final loss	Training time (100 epoch)
Shared weight Image-wise Autoencoders	0.897	0.00026	<b>132.99 s</b>
Normal Image-wise Autoencoders	<b>0.917</b>	<b>0.00019</b>	150.68 s

**Table 3.** Comparison between image-wise autoencoders and common CNN

Method	Within accuracy	Cross accuracy
Normal Image-wise Autoencoders	<b>0.917</b>	<b>0.756</b>
Image-wise CNN	0.915	0.712

**Table 4.** Classification accuracy – within-subject tests

Method	Accuracy
Normal Channel-wise Autoencoders	0.864
Shared weight Channel-wise Autoencoders	0.858
Normal Image-wise Autoencoders	<b>0.917</b>
Shared weight Image-wise Autoencoders	0.897
EEGNet (Lawhern et al. 2016)	0.878
SyncNet (Li et al. 2017)	<b>0.923</b>
DE (Zheng and Lu 2015)	0.821
PSD (Zheng and Lu 2015)	0.816
rEED (O'Reilly et al. 2012)	0.702

**Table 5.** Classification accuracy – cross-subject tests

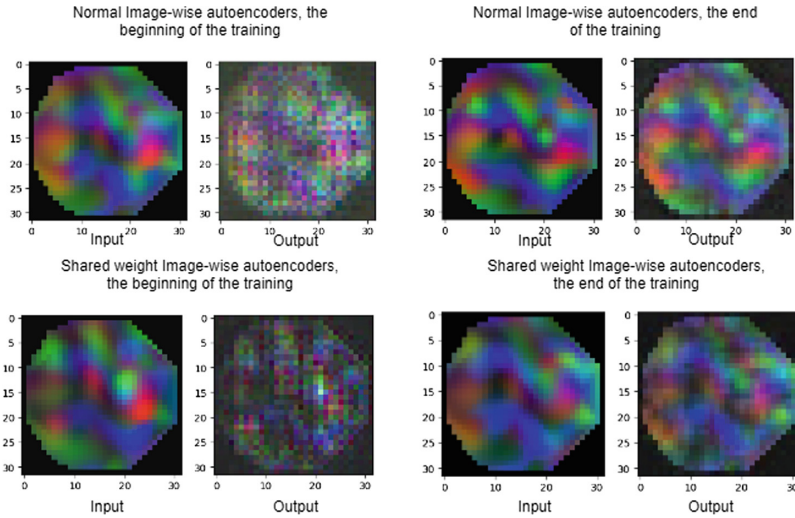
Method	Accuracy
Normal Channel-wise Autoencoders	0.731
Shared weight Channel-wise Autoencoders	0.713
Normal Image-wise Autoencoders	<b>0.756</b>
Shared weight Image-wise Autoencoders	0.740
EEGNet (Lawhern et al. 2016)	0.672
SyncNet (Li et al. 2017)	<b>0.723</b>
DE (Zheng and Lu 2015)	0.622
PSD (Zheng and Lu 2015)	0.605
rEED (O'Reilly et al. 2012)	0.614

The accuracy of prediction on the UCI EEG dataset, from a variety of methods, is given in Tables 4 and 5. The accuracy of other methods is listed from Li’s paper [13].

We first check cross-subject result, which is the test format we are more likely to meet in real life for classifying disease, and the part we are focusing on to improve through autoencoders. As shown in Table 5, all of our autoencoders except the shared weight channel-wise autoencoders achieve state-of-art cross-subject test accuracy. We believe this is because autoencoders can encourage feature extraction without overfitting, and will prevent the model from performing badly on new data. In other words, this prevents our model from learning the disease condition by merely remembering the personal identity, and instead makes our model focus on the common features for alcoholism. This could explain why autoencoders based methods perform best in the cross-subject test. Further evidence is shown in Table 3 in order to show the performance of using autoencoder, we construct an Image-wise CNN which has the same structure of the encoder of Image-wise autoencoder with a three-layer FC as the classifier. The result shows that though it can achieve similar within subject accuracy as Image-wise autoencoder but it performs badly in the cross-subject test. That is, an autoencoder structure helps to improve the ability to extract robust features.

From the result above in the within-subject test, we can see that the accuracy of our autoencoder based method is better than most of the past methods except the SyncNet [13] published last year.

Apart from these, there are also some general conclusions. Image-wise autoencoders perform better than channel-wise autoencoders while the normal autoencoders perform slightly better than shared weight autoencoders. From Table 2, we can see the normal image-wise autoencoder has better within subject accuracy and lower final test loss than the shared weight image-wise autoencoders. Also from Fig. 8, the picture generated by normal Image-wise autoencoder is slightly clearer and more similar to the original image. On the other hand, shared weight image-wise autoencoders have lower training time. This is an advantage of the shared weight technique because it cuts half of the parameters. From these results above, we can see that the image-wise autoencoders find the best discriminative features among different methods. We believe this is because frequency-based feature learning methods can obtain more discriminative information – both our image-wise autoencoders and the SyncNet approach are frequency based and they achieved the best performance.



**Fig. 8.** Image-wise autoencoders' performance

## 5 Conclusion

Feature extraction for EEG data is very challenging because EEG signals contain a lot of noise, use different collection standards and huge personal differences exist. This paper introduced two kinds of autoencoders: Image-wise autoencoders and Channel-wise autoencoders. These two types of autoencoders were tested for feature extraction, and both achieve state of the art accuracy in cross-subject tests and comparable accuracy in the (less important in classifying disease) within-subject tests. The experiment results demonstrate that the autoencoder based feature learning is discriminative and robust for new data. Also, we found that a shared weight technique can noticeably reduce the training time with only a small discriminative information loss.

## 6 Limitation and Future Work

Many further experiments should be done. First of all, the UCI dataset also contains other labels that can be classified. We should further test our extracted features with those labels to ensure our extracted features are discriminative for multiple task classifications. Furthermore, since we are using the EEG2image technique, our method should be a general framework without further fine-tuning for other datasets as long as the 3-D electrode location information is provided. Then, other popular datasets like DEAP should be tested in the same setting as the UCI dataset we used. Then we can turn our attention to more frequency based methods since both Image-wise autoencoders and SyncNet are frequency based. Finally, we may try LSTM based work – there also exist many RNN feature extractors for EEG data. If we have realized all these feature extraction methods, we would do more visualization procedures. Unlike the

computer vision area, the features of EEG signals are not obvious, so visualization will be a good choice for understanding EEG features. Our ultimate goal is to get a deeper understanding of EEG features and make it possible to design stronger feature extractors and classifiers.

Currently, there is a very limited work for applying deep learning work for bio-signals. Since many successful examples exist in CV and NLP areas, it will be very worthwhile to try them in bioinformatics areas.

## References

1. Truong, N.D., Nguyen, A.D., Kuhlmann, L., Bonyadi, M.R., Yang, J., Kavehei, O.: A Generalised Seizure Prediction with Convolutional Neural Networks for Intracranial and Scalp Electroencephalogram Data Analysis (2017). arXiv preprint: [arXiv:1707.01976](https://arxiv.org/abs/1707.01976)
2. Thodoroff, P., Pineau, J., Lim, A.: Learning robust features using deep learning for automatic seizure detection. In: Machine Learning for Healthcare Conference, pp. 178–190 (2016)
3. Ebrahimi, F., Mikaeili, M., Estrada, E., Nazeran, H.: Automatic sleep stage classification based on EEG signals by using neural networks and wavelet packet coefficients. In: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2008, pp. 1151–1154. IEEE (2008)
4. Palazzo, S., Spampinato, C., Kavasidis, I., Giordano, D., Shah, M.: Generative adversarial networks conditioned by brain signals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3410–3418 (2017)
5. Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P., Lance, B.J.: Eegnet: A compact convolutional network for eeg-based brain-computer interfaces (2016). arXiv preprint: [arXiv:1611.08024](https://arxiv.org/abs/1611.08024)
6. Min, S., Lee, B., Yoon, S.: Deep learning in bioinformatics. *Brief. Bioinform.* **18**, 851–869 (2017)
7. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: *Deep Learning*. MIT Press, Cambridge (2016)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
9. Hajinoroozi, M., Mao, Z., Jung, T.-P., Lin, C.-T., Huang, Y.: EEG-based prediction of driver's cognitive performance by deep convolutional neural network. *Signal Process. Image Commun.* **47**, 549–555 (2016)
10. Bashivan, P., Rish, I., Yeasin, M., Codella, N.: Learning representations from EEG with deep recurrent-convolutional neural networks (2015). arXiv preprint: [arXiv:1511.06448](https://arxiv.org/abs/1511.06448)
11. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. In: *The Handbook of Brain Theory and Neural Networks*, p. 3361 (1995)
12. Schirmer, R.T., et al.: Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum. Brain Mapp.* **38**, 5391–5420 (2017)
13. Li, Y., Dzirasa, K., Carin, L., Carlson, D.E.: Targeting EEG/LFP synchrony with neural nets. In: *Advances in Neural Information Processing Systems*, pp. 4623–4633 (2017)
14. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014, Part I. LNCS*, vol. 8689, pp. 818–833. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53)

15. Gedeon, T., Catalan, J., Jin, J.: Image compression using shared weights and bidirectional networks. In: Proceedings 2nd International ICSC Symposium on Soft Computing (SOCO 1997), pp. 374–381 (1997)
16. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) ICANN 2011, Part I. LNCS, vol. 6791, pp. 52–59. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21735-7\\_7](https://doi.org/10.1007/978-3-642-21735-7_7)
17. Stober, S., Sternin, A., Owen, A.M., Grahn, J.A.: Deep feature learning for EEG recordings (2015). arXiv preprint: [arXiv:1511.04306](https://arxiv.org/abs/1511.04306)
18. Tabar, Y.R., Halici, U.: A novel deep learning approach for classification of EEG motor imagery signals. *J. Neural Eng.* **14**, 016003 (2016)
19. Sykacek, P., Roberts, S.J.: Adaptive classification by variational Kalman filtering. In: Advances in Neural Information Processing Systems, pp. 753–760 (2003)
20. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)