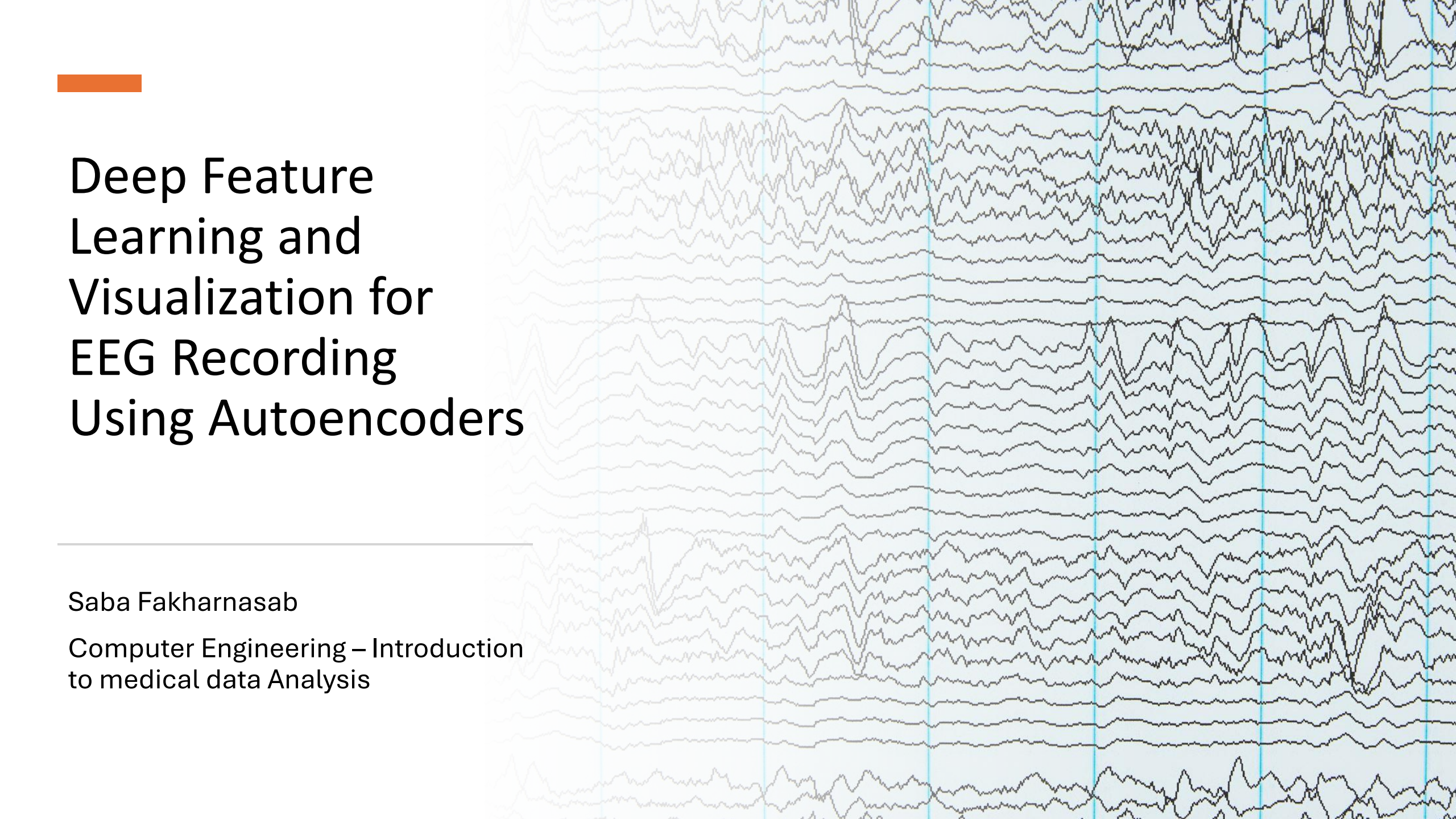
Deep Feature Learning and Visualization for EEG Recording Using Autoencoders

Saba Fakharnasab

Computer Engineering – Introduction to medical data Analysis



What is EEG?

- 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.

Challenges Analyzing EEG

- Using EEG signals faces many difficulties

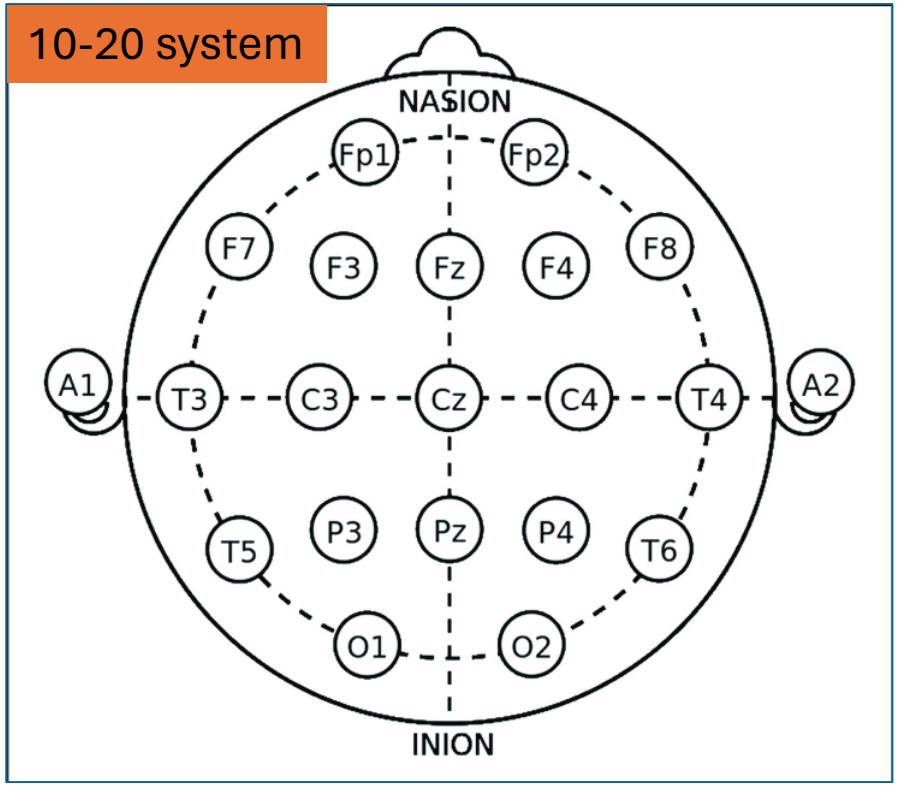
1) being full of information also here means *full of noise* and *interference* making it very hard *to extract reliable feature*

2) Depending on the collection device, EEG will have a different format, hence it becomes difficult to construct standard algorithms to extract features from EEG.

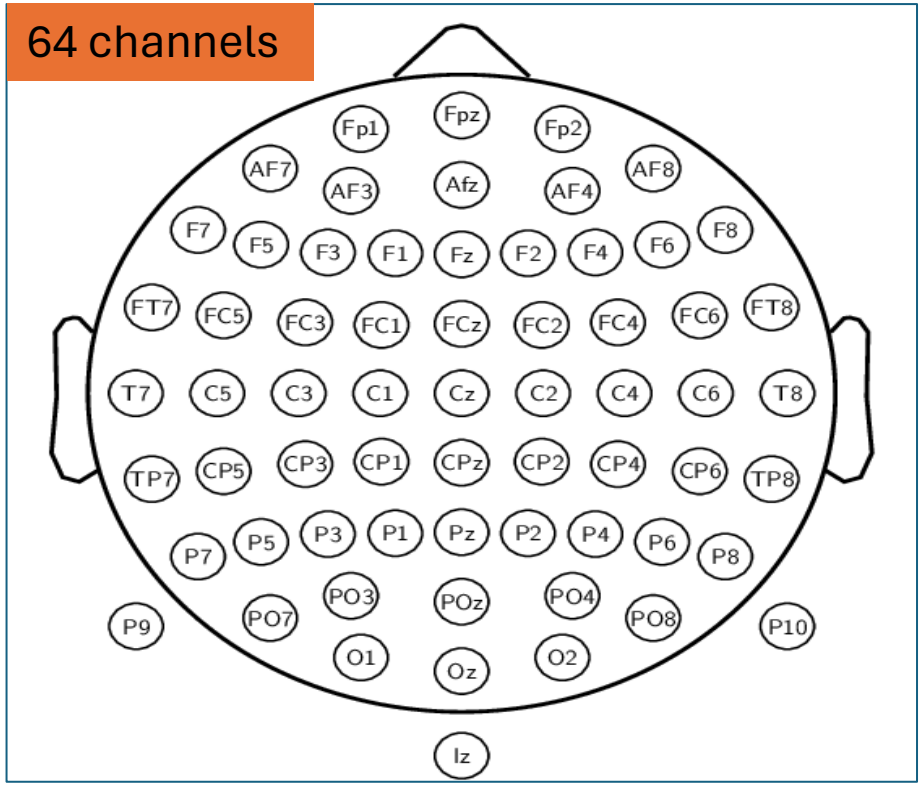
Example: different electrode numbers (10-20 system, 64 electrode system etc.)

Challenges Analyzing EEG

10-20 system



64 channels



Challenges Analyzing EEG

3) EEG signals have large *individual differences*, making it hard for *cross-subject* tests to achieve high accuracy.

Examples are *age*, *gender* and *educational background* differences.

- **These three difficulties make EEG feature engineering still a work in progress.**

Deep learning vs traditional machine learning

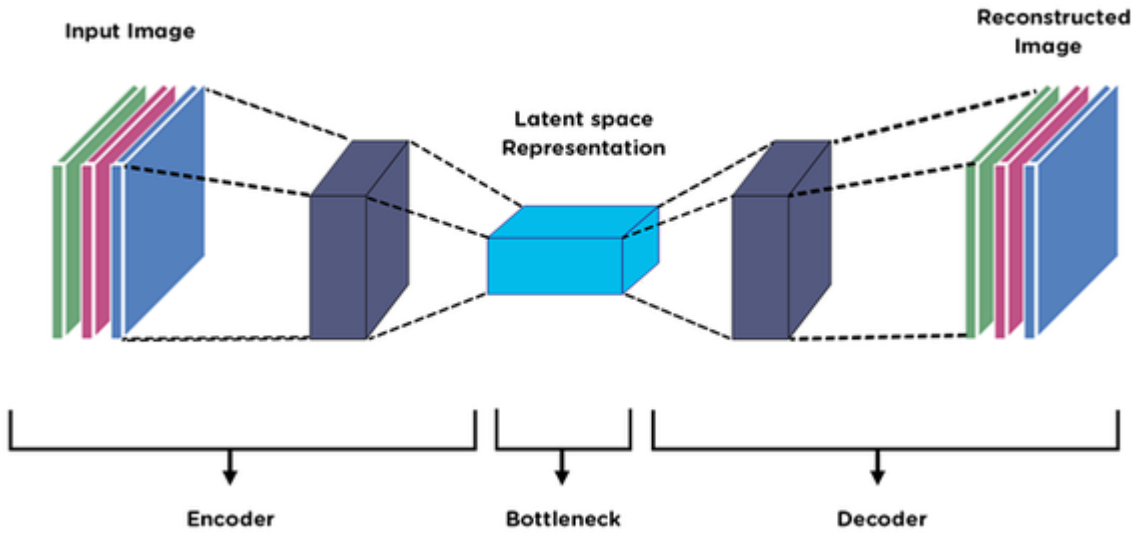
- The good performance of conventional machine learning algorithms relies heavily on *features extracted*
- Extracted features are not always good and informative.
- Models are not robust and not designed to encounter noise.
- Deep learning and neural network-based techniques can provide automatic feature learning.

Article’s deep learning modality

- To address these difficulties, deep learning approaches are utilized in this paper to achieve both *feature learning* and *visualization*
- Two autoencoder based models are used for feature learning and dimensionality reduction:
  - Channel-wise autoencoders
  - Image-wise autoencoders

Typical Autoencoders

- Autoencoder is a sort of compression algorithm, or dimension reduction algorithm.
- Has similar properties to Principal Components Analysis (PCA). But compared with PCA, the autoencoder has no linear constraints.



Dataset

- UCI, the EEG dataset
- It has a total of 122 subjects
  - 77 diagnosed with alcoholism
  - 45 control subjects
  - Unbalanced Dataset
- 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.

Dataset

- Short time EEG = One trial is one second
- Sampling rate = 256 Hz
- Each trial has 256 data points
- 64 electrode system
- Stimuli : pictures

Dataset

- is a two-task classification but alcoholism trials account for more than 70% of the data
- Models are first evaluated using data *within subjects (alcoholism only or control only)*, where 120 trial data is randomly split as 7:1:2 for training, validation and testing for one person
- We further test them using data across subjects

Channel-wise autoencoders

- The key idea: separate the feature extraction procedure into two parts.

1) Each channel has its own autoencoder
2) Finally, a fully connected layer combines flattened features across channels to make the final prediction.

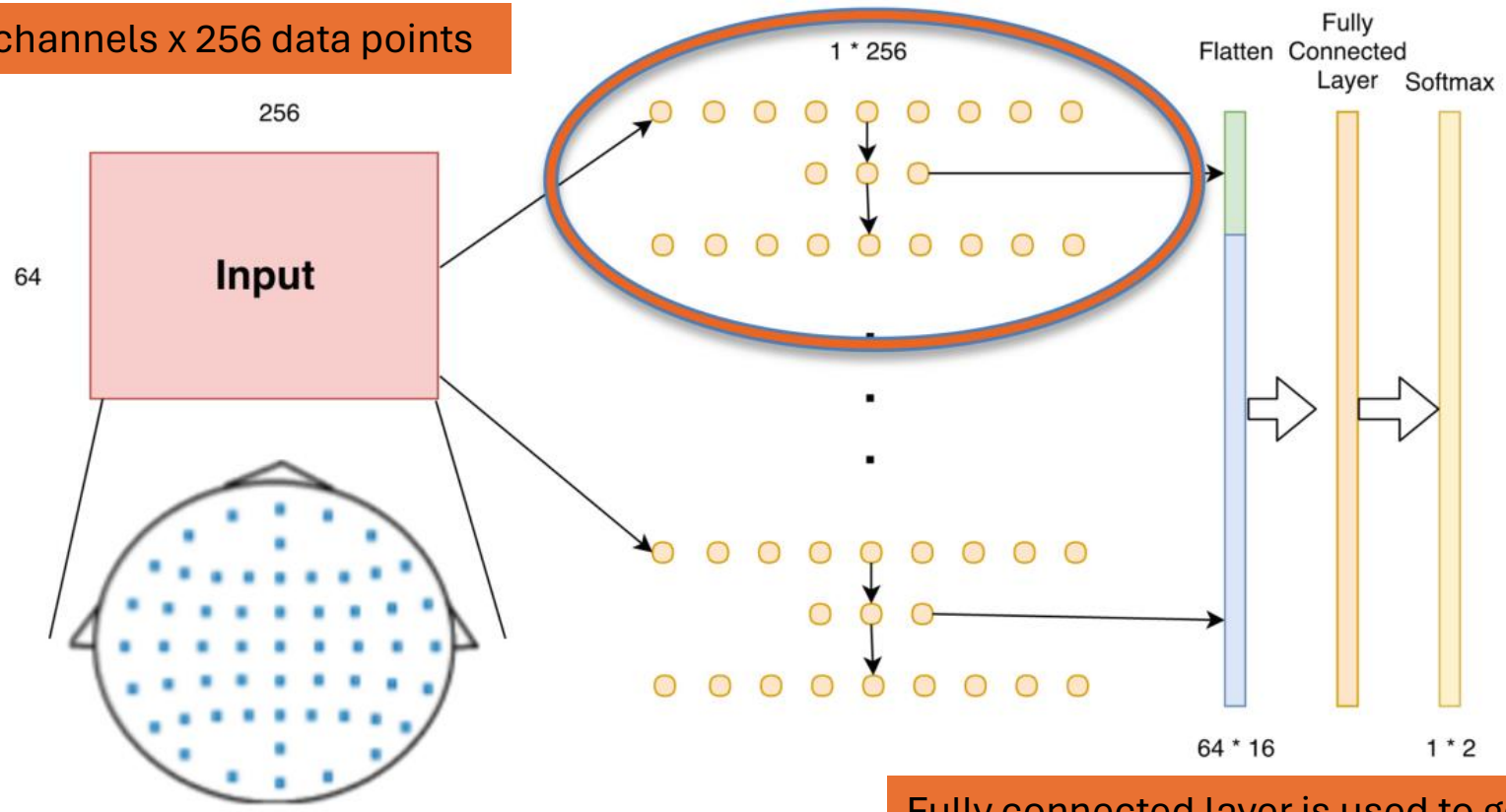
Channel-wise autoencoders - Architecture

- Each autoencoder has *2 fully connected* layers, each having *16 hidden units*.
  - Dimension of data reduced from 256 to 16 in latent space
- The input of autoencoders will be normalized to [−1, 1]
- we use a *tanh activation* function for the output layer to match the output to [−1, 1] as well.
- The *shared weight* technique: decoder will use transpose of encoder weight vectors
  - Using the shared weight technique helps the autoencoder better understand how to compress the input data and then reconstruct it accurately, resulting in improved performance and reduced training time.

Channel-wise autoencoders

Each channel is passed through its own autoencoder

Input dimension = 64 channels x 256 data points



Fully connected layer is used to generalize learned patterns

Fig. 4. Structure of channel-wise autoencoders

Image-wise autoencoders

- The key idea: CNN based autoencoder takes images as input while using convolution to extract features.

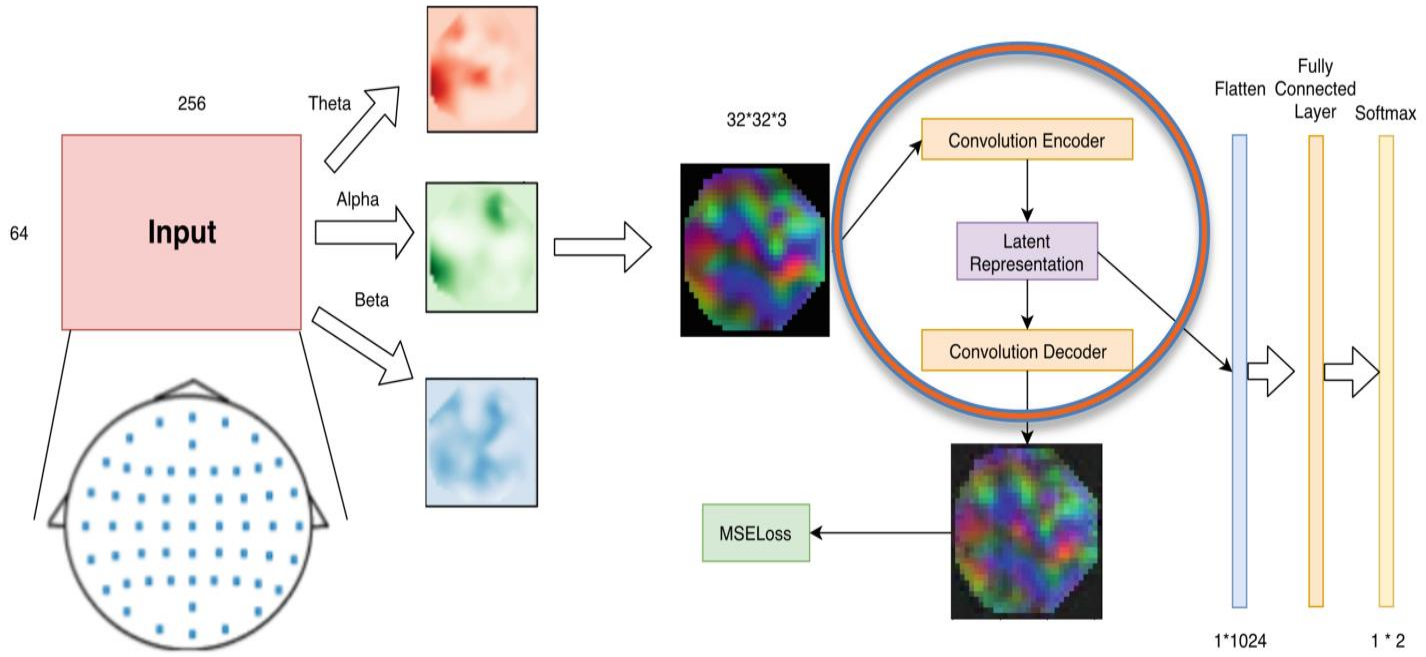


Fig. 5. Structure of image-wise autoencoder

EEG to image

- A method that combines the *time-series information* and *spatial channel locations* information over the scalp in a trial of EEG signals.

EEG to image

- FFT is performed on the time series to estimate the power spectrum of the signal for each trial (64 x 256).
- Three frequency bands information are extracted:
  - theta (4–7 Hz)
  - alpha (8–13 Hz)
  - beta (13–30 Hz)
- The sum of squared absolute values in these frequency bands are used, forming a 64 x 3 map

EEG to image

- To form an RGB EEG image:
  - theta = red channel
  - alpha = green channel
  - beta = blue channel

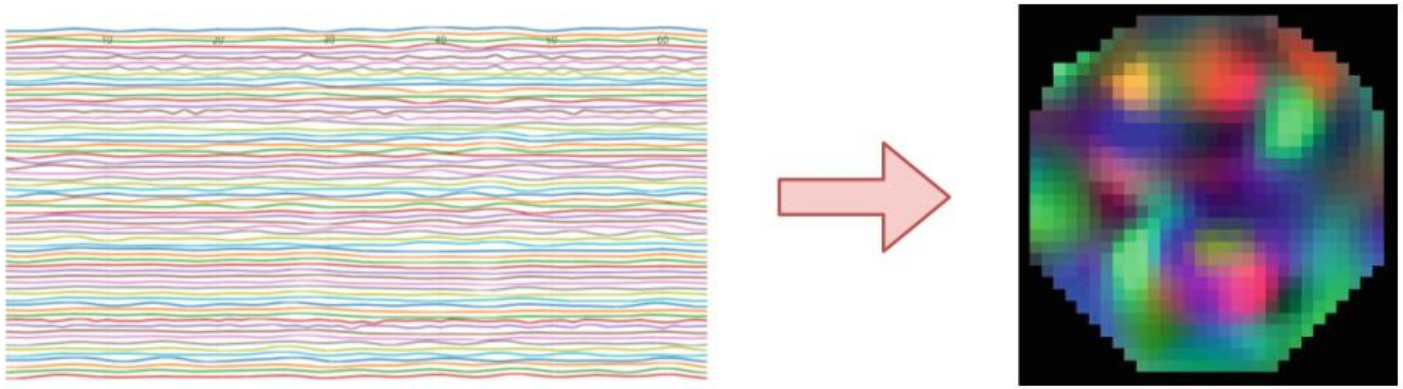


Fig. 6. EEG signal to image example

EEG to image

- For each frequency band (64 x1), shown in Fig. 7, *topology preserving* Azimuthal Equidistant Projection (AEP) also known as Polar Projection is used to map 3D channel position into 2D space.
- “Topology preserving" = preservation of the neighborhood relation
- Location of the electrodes also preserved in relation to the center of the head (reference point)

EEG to image

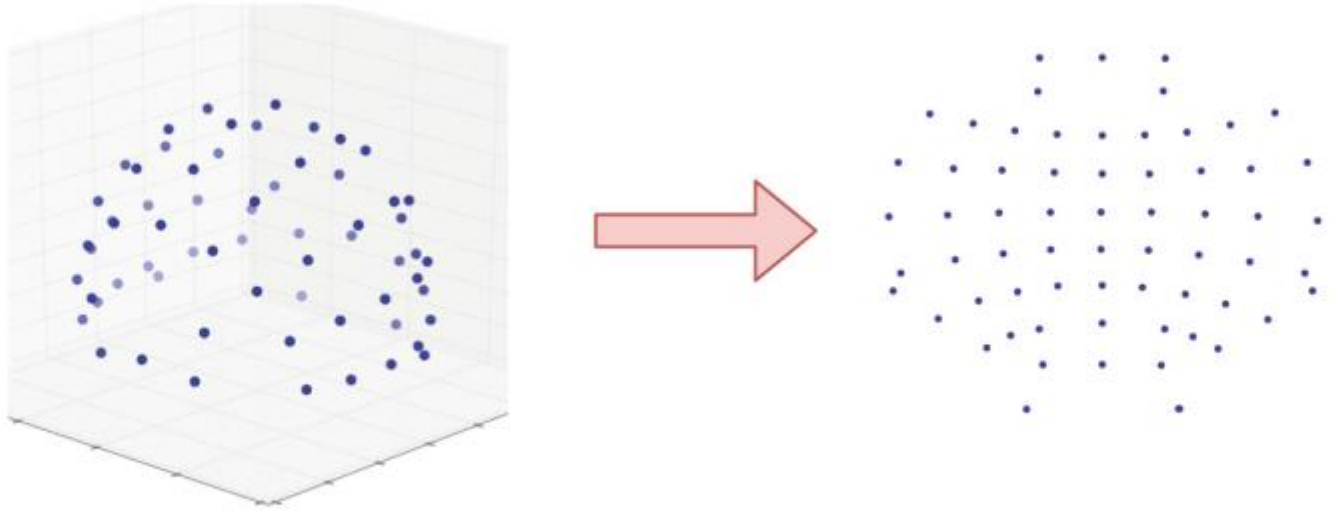


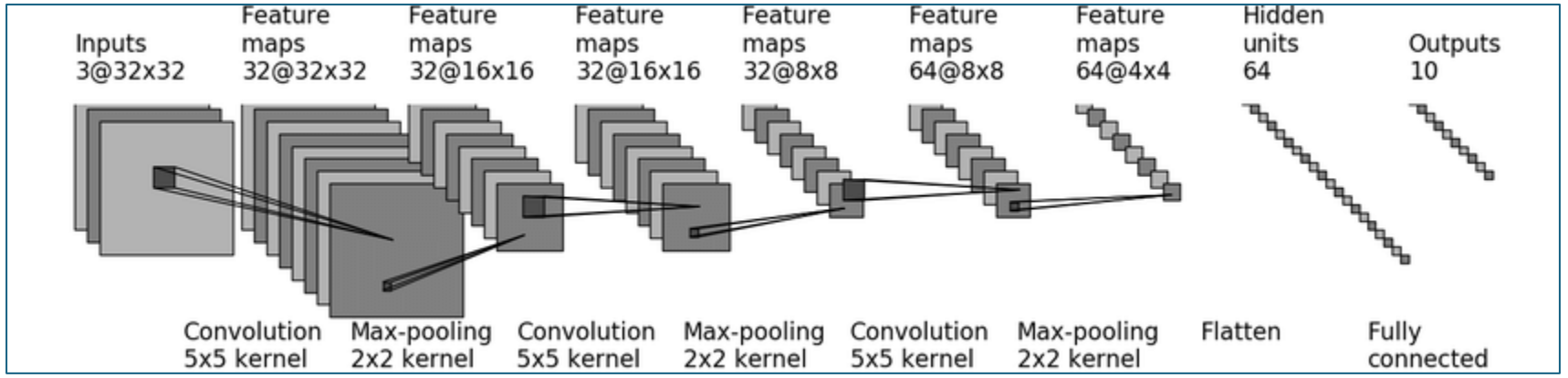
Fig. 7. Transform 3-D coordinate to 2-D coordinate [10]

EEG to image

- Each 64 x 1 frequency band can be mapped to a 32 x 32 mesh, forming 32 x 32 x 3 data.
- One trial of 64 x 256 EEG signals is transformed to color pictures.
  - RGB picture (3 channels)
  - Size = 32 x 32

Autoencoder design

- The design of this CNN-based autoencoder is inspired by the CNN for CIFAR-10.
- The input dimension of our generated EEG pictures is the same as the CIFAR-10 dataset.



Original CIFAR-10 architecture – input dimensions are the same, output is 10 classes

Encoder-Decoder Structure

Table 1. The detailed encoder and decoder structure

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

Classification

- Features from Image-wise and channel-wise autoencoders are combined, flattened into a long vector.
  - 16 x 64 channels wise
  - 16 x 8 x 8 image wise
- Feedforward network with three hidden layers
- Learning rate for training 4e-5
- Fine tuning *Encoders* of both channel wise and image wise autoencoder
  - Small learning rate of 1e-7

Classification

- Features from Image-wise and channel-wise autoencoders are combined, flattened into a long vector.
  - 16 x 64 channels wise
  - 16 x 8 x 8 image wise
- Feedforward network with three hidden layers
- Learning rate for training 4e-5
- Fine tuning *Encoders* of both channel wise and image wise autoencoder
  - Small learning rate of 1e-7

Results: Image-wise autoencoders

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	**132.99 s**
Normal Image-wise Autoencoders	**0.917**	**0.00019**	150.68 s

Results: Image-wise autoencoders vs common CNN

- CNN has the same structure of the encoder of the (image wise autoencoder)
- Three fully connected layers used as classifier

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

Method	Within accuracy	Cross accuracy
Normal Image-wise Autoencoders	**0.917**	**0.756**
Image-wise CNN	0.915	0.712

Results: Image-wise autoencoders vs common CNN

- Image-wise autoencoder
  - 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.

Results: Autoencoders vs famous deep learning structures – within subject

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	**0.917**
Shared weight Image-wise Autoencoders	0.897
EEGNet (Lawhern et al. 2016)	0.878
SyncNet (Li et al. 2017)	**0.923**
DE (Zheng and Lu 2015)	0.821
PSD (Zheng and Lu 2015)	0.816
rEED (O'Reilly et al. 2012)	0.702

Results: Autoencoders vs famous deep learning structures – cross subject

- The accuracy of the proposed autoencoder based method is better than most of the past methods except the SyncNet.

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	**0.756**
Shared weight Image-wise Autoencoders	0.740
EEGNet (Lawhern et al. 2016)	0.672
SyncNet (Li et al. 2017)	**0.723**
DE (Zheng and Lu 2015)	0.622
PSD (Zheng and Lu 2015)	0.605
rEED (O'Reilly et al. 2012)	0.614

Results

- All autoencoders proposed in this paper except the shared weight channel-wise autoencoders achieve state-of-art cross-subject test accuracy.
- Reason: Authors believe that their shared weight autoencoder using the shared weights method focuses on common features in alcoholism rather than personal identity features
- This method prevents overfitting, and performs better on new, unseen data.

General conclusions

- Image-wise autoencoders perform better than channel-wise autoencoders
- Normal autoencoders perform slightly better than shared weight autoencoders.
- Normal image-wise autoencoder has better within subject accuracy and lower final test loss than the shared weight image-wise autoencoders.

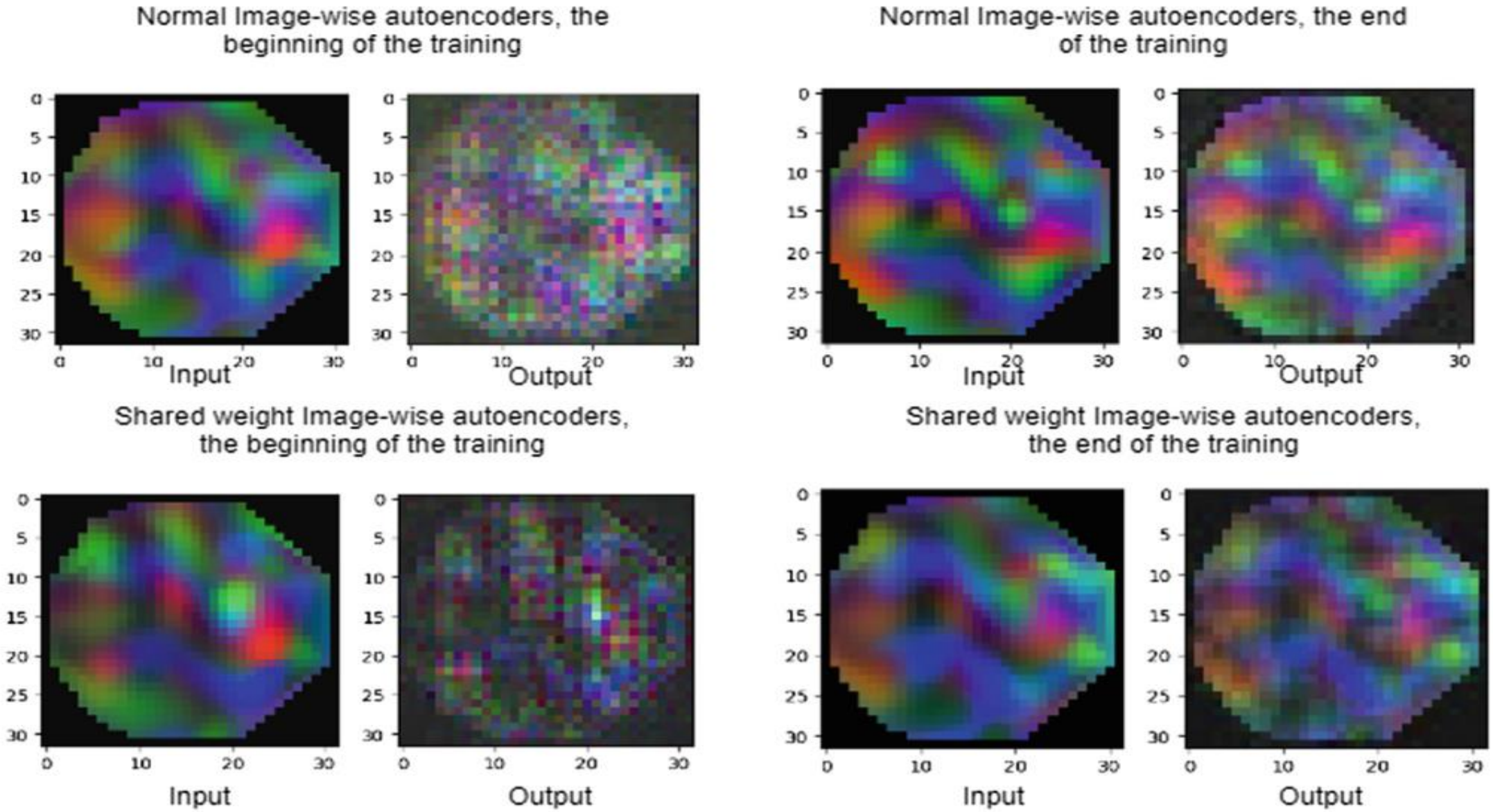


Fig. 8. Image-wise autoencoders' performance

General conclusions

- The image-wise autoencoders find the best discriminative features among different methods.
- Authors believe this is because frequency-based feature learning methods can obtain more discriminative information
- Both proposed image-wise autoencoders and the SyncNet approach are frequency based and they achieved the best performance.

Limitation and further work

- Verify the method's applicability to other datasets without further fine-tuning, as long as 3-D electrode location information is available.
- Investigate LSTM-based approaches and other RNN feature extractors for EEG data.