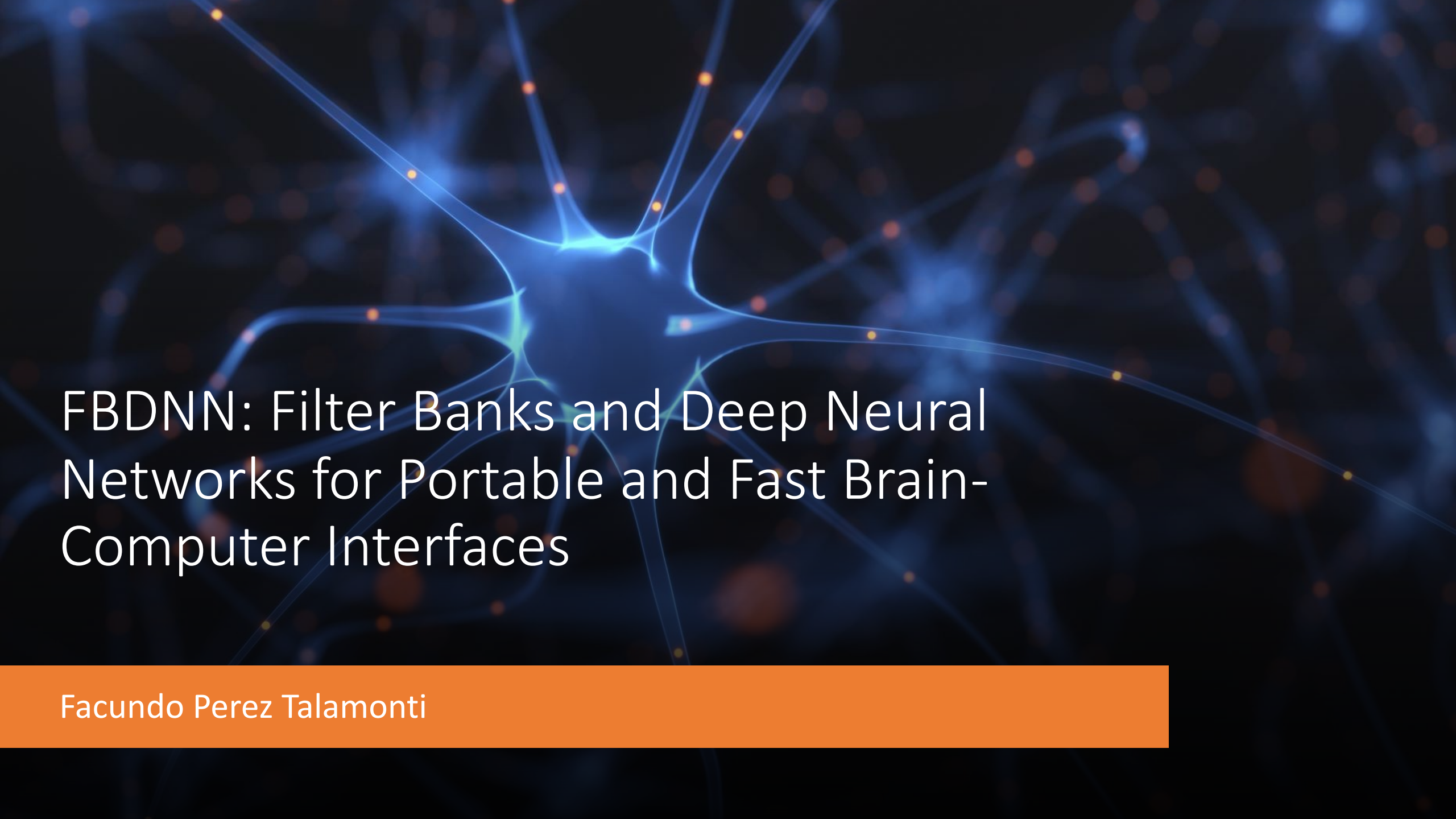
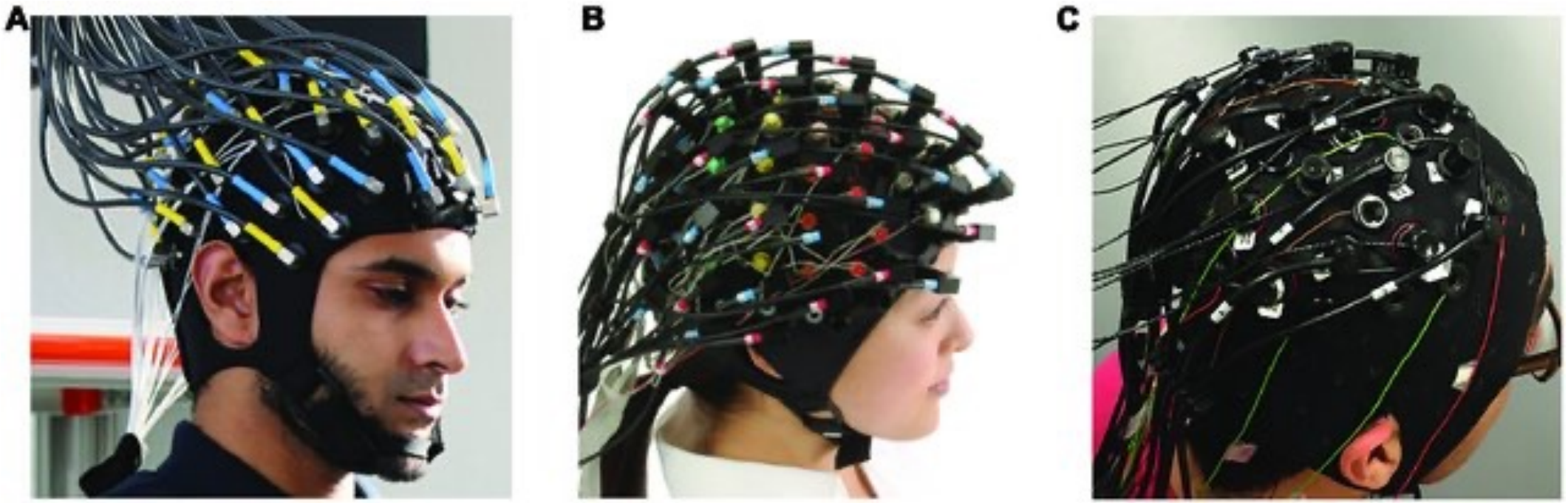


FBDNN: Filter Banks and Deep Neural Networks for Portable and Fast Brain-Computer Interfaces

Facundo Perez Talamonti

Portable?
User Independent?

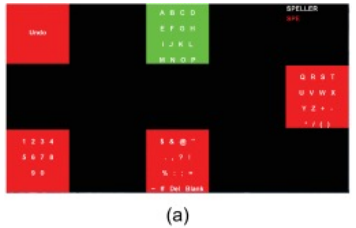
"… is cost-effective and portable "



Source:
Ahn, Sangtae & Jun, Sung. (2017). Multi-Modal Integration of EEG-fNIRS for Brain-Computer Interfaces – Current Limitations and Future Directions. Frontiers in Human Neuroscience.

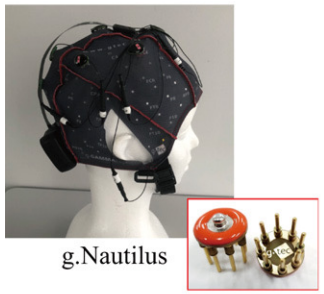
Applications

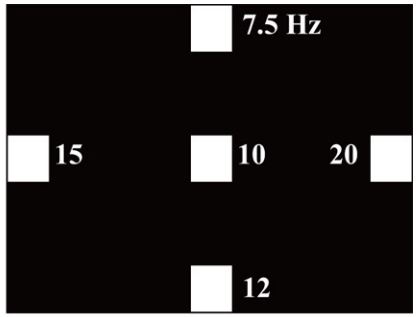
SSVEP speller





[1]





Multi-flicker monitor screen

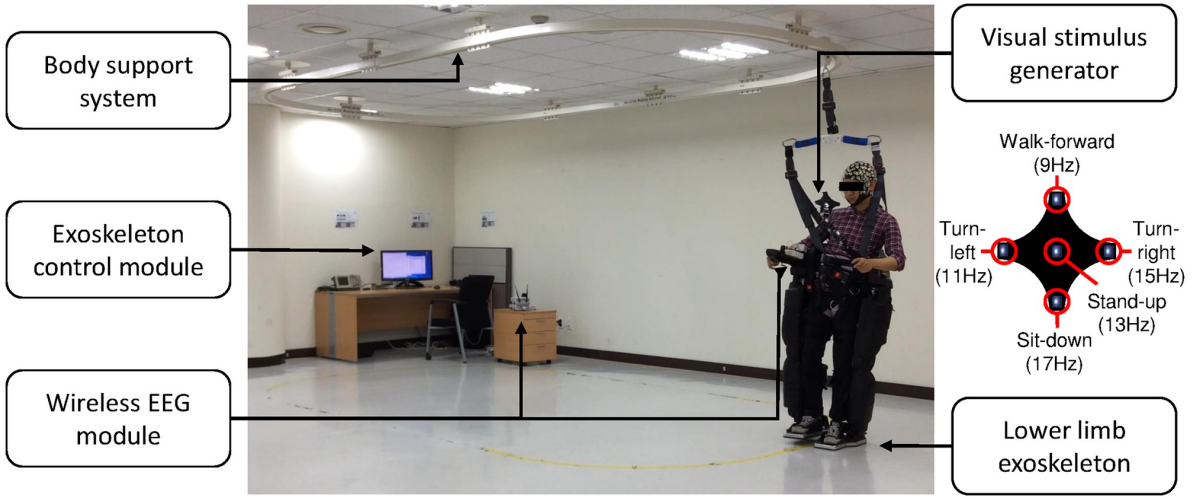


Parrot AR. Drone2.0

[5]

Quadcopter

Exoskeleton



[6]


Previous work

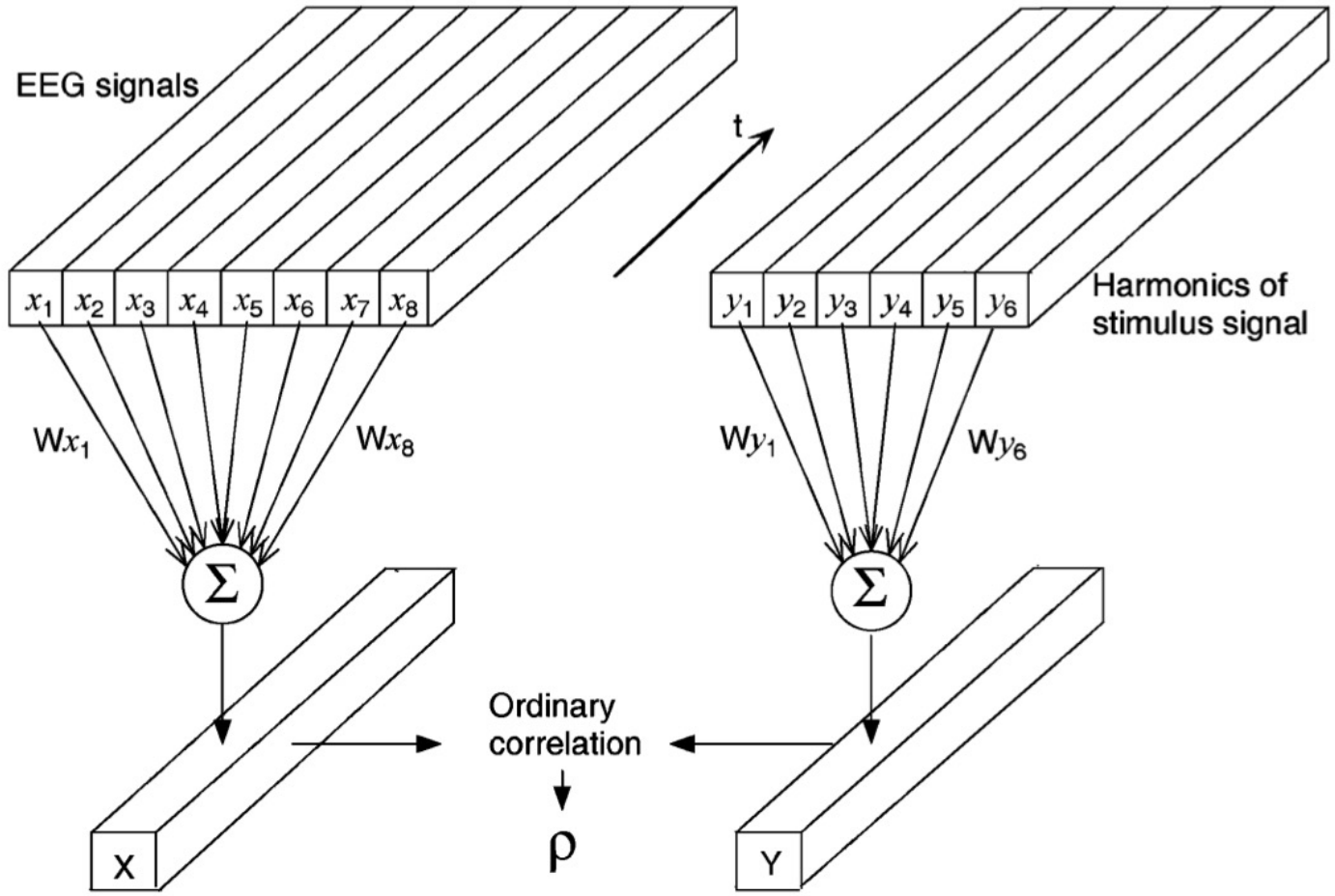
CCA

Pros:

- Low computational cost
- Robust

Cons:

- Relatively weak



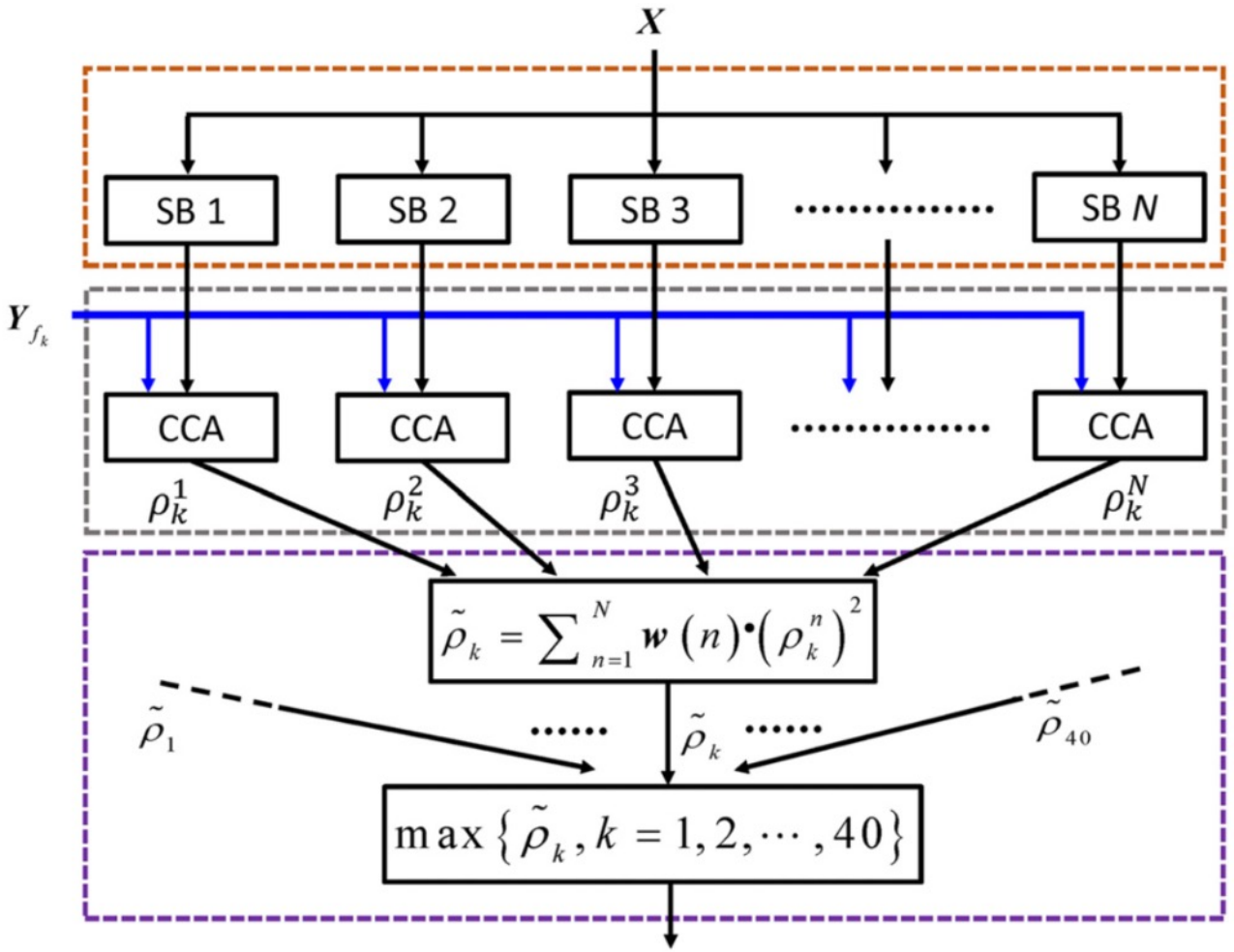
[2]

FBCCA

Allows for better use of

- Fundamental frequency
- Harmonics

Better accuracy!



[3]

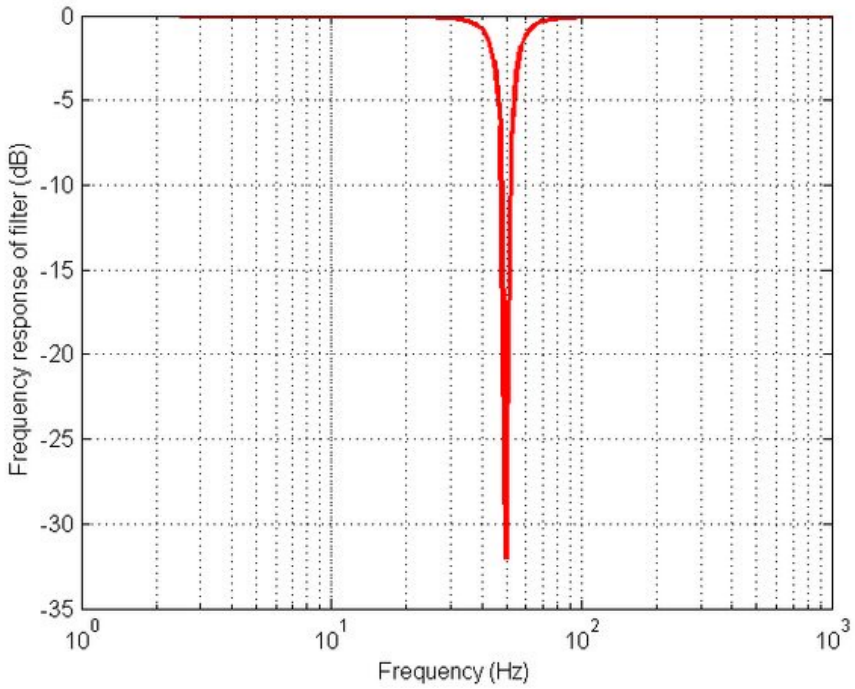
This paper’s contributions



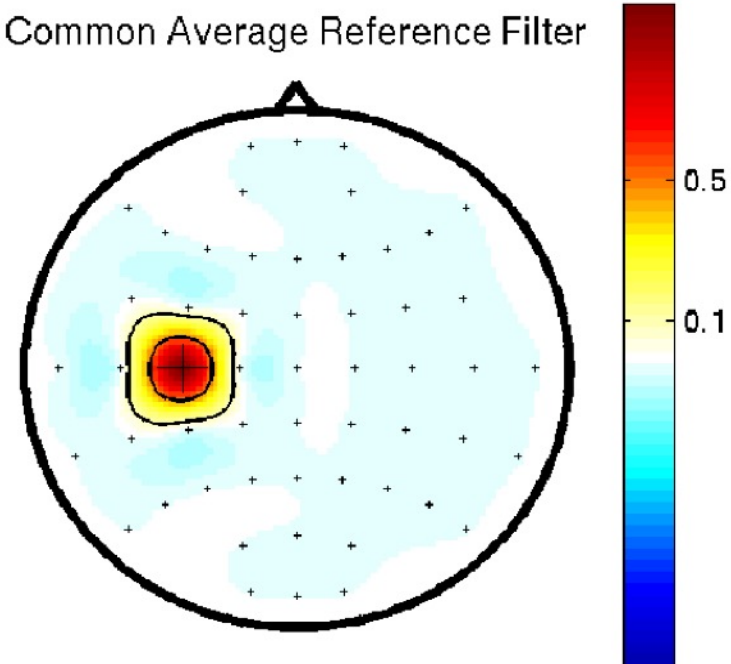
Preprocessing

Signal preprocessing is shared among all models!

Notch filter at 50Hz



[7]



[Lecture 8]

Filter Bank

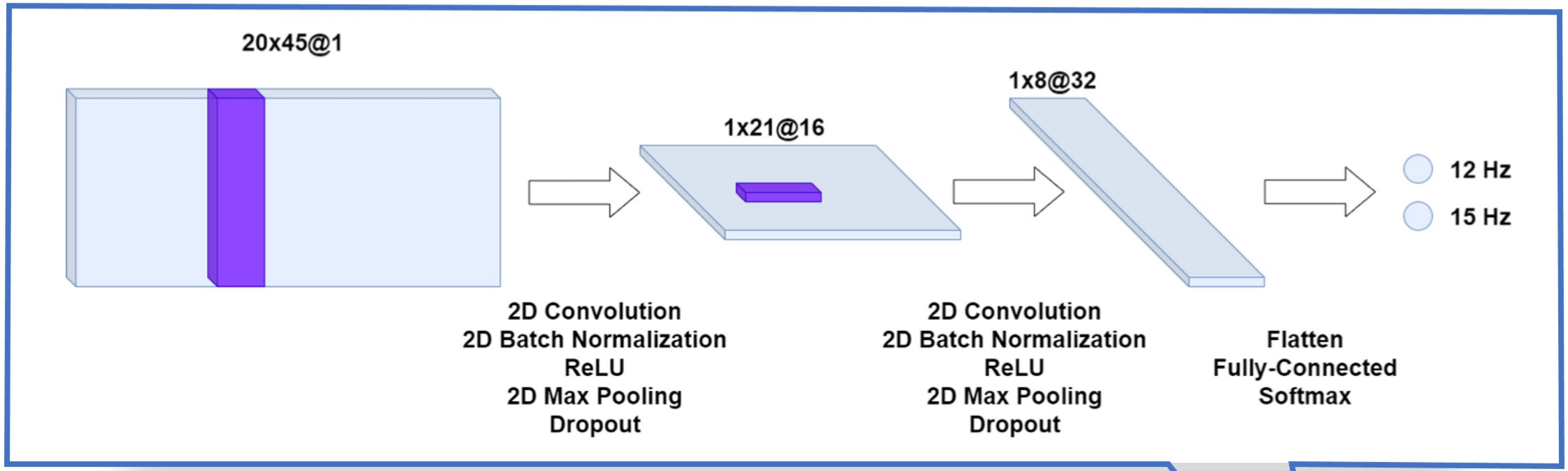
FB 1:

- Trivial case

FB 2:

- Makes better use of harmonics
- Allows the network to learn the details

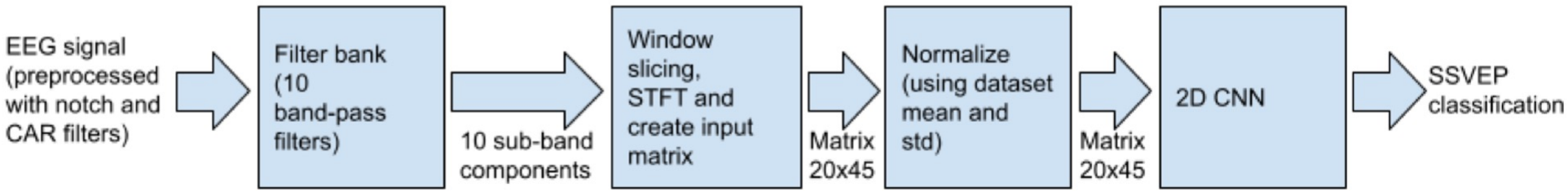
Filter	FB 1	FB 2
1	8Hz to 16Hz	6Hz to 90Hz
2	16Hz to 24Hz	14Hz to 90Hz
3	24Hz to 32Hz	22Hz to 90Hz
4	32Hz to 40Hz	30Hz to 90Hz
5	40Hz to 48Hz	38Hz to 90Hz
6	48Hz to 56Hz	46Hz to 90Hz
7	56Hz to 64Hz	54Hz to 90Hz
8	64Hz to 72Hz	62Hz to 90Hz
9	72Hz to 80Hz	70Hz to 90Hz
10	80Hz to 88Hz	78Hz to 90Hz



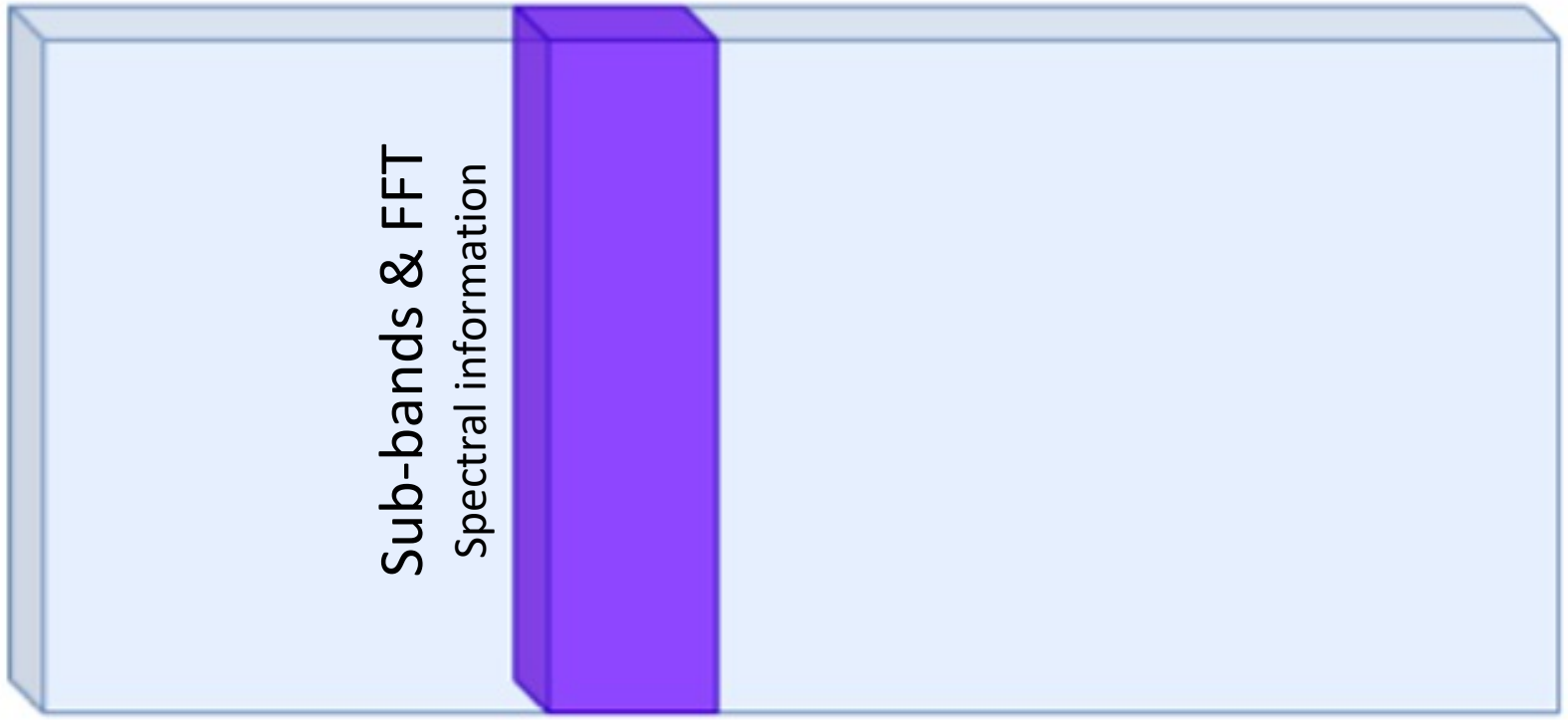
[4]

FBCNN – 2D

1. Windowing [0.0s – 0.5s]
2. Short-Time Fourier Transform [0Hz – 90Hz] (2Hz)
3. Stack Re & Im on **Sub-bands**

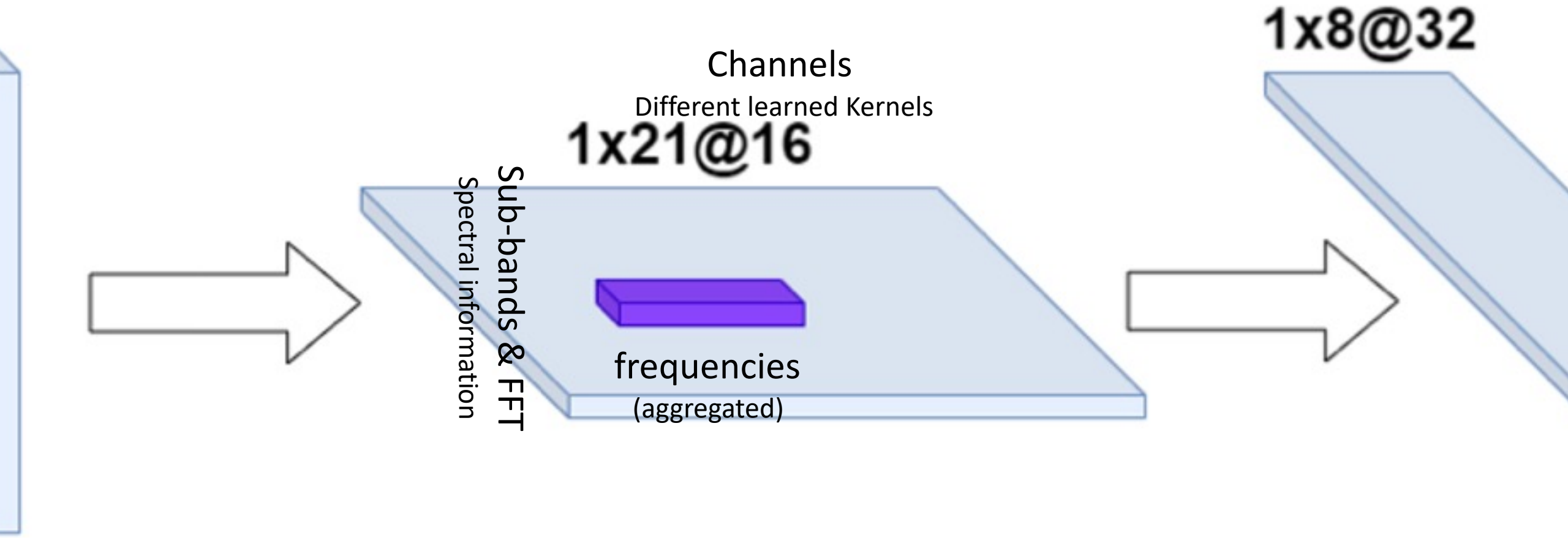


20x45@1



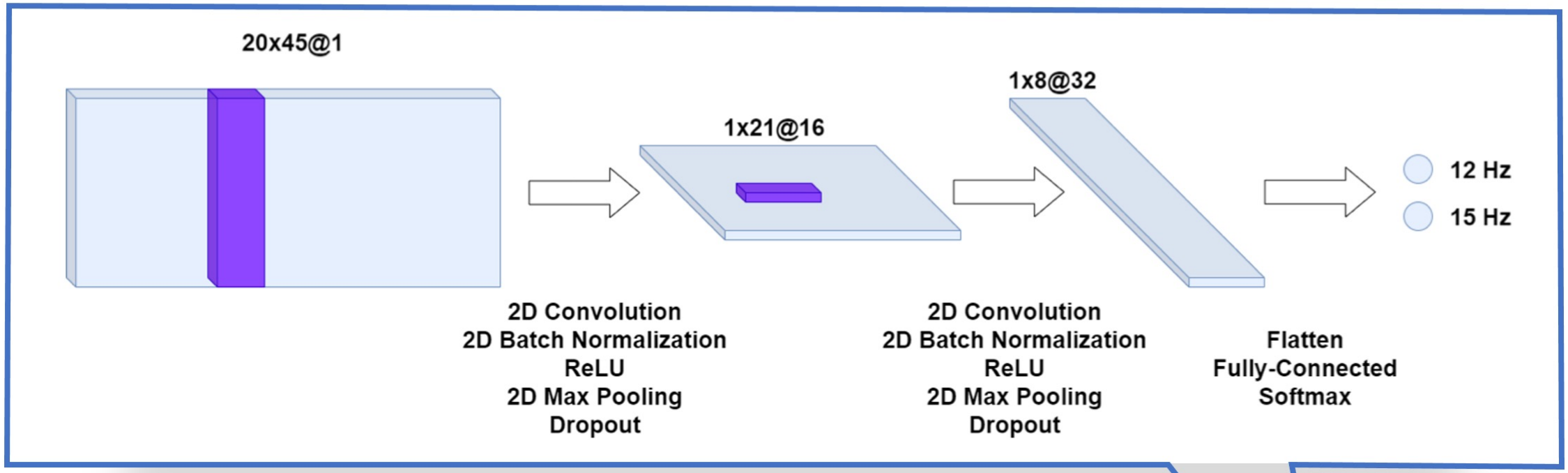
frequencies

2D Convol
2D Batch Norm
ReLU



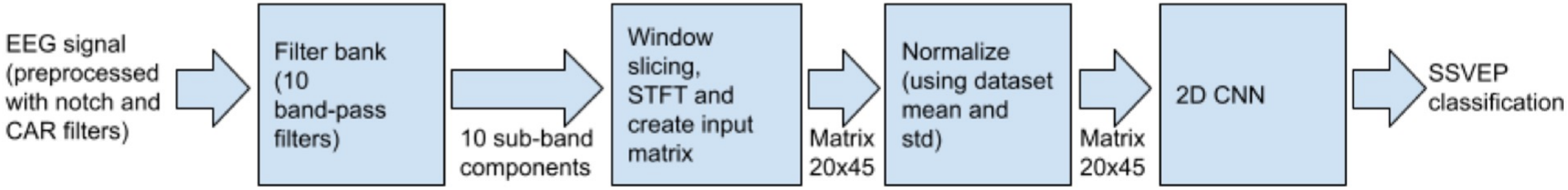
2D Convolution
Batch Normalization

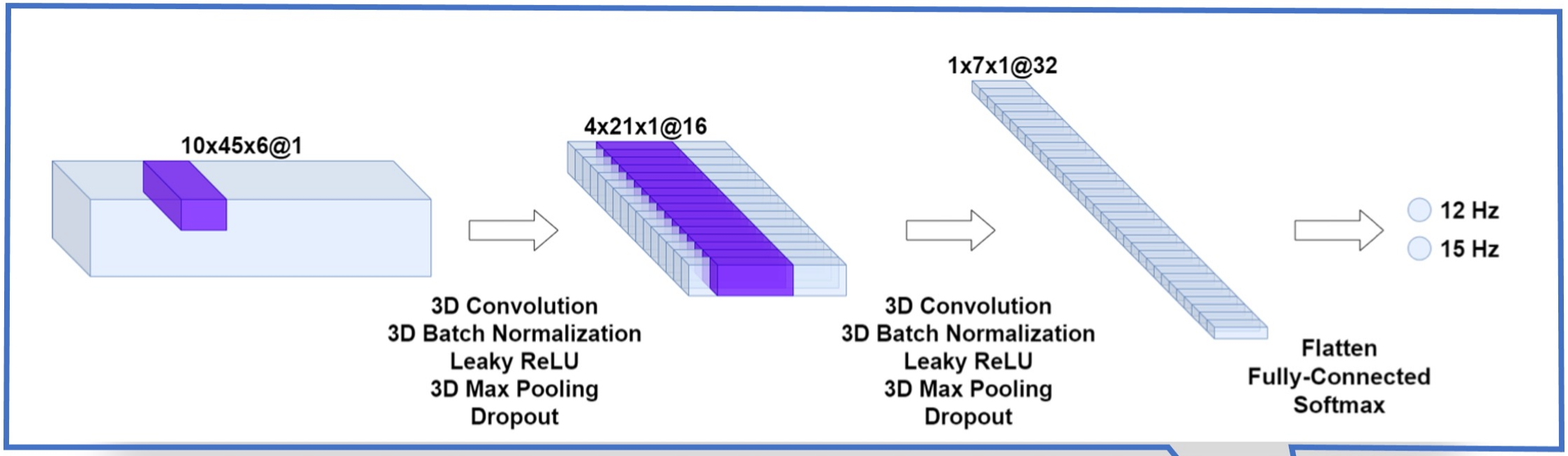
2D Convolution
2D Batch Normalizatio


20x45@1
1x21@16
1x8@32
12 Hz
15 Hz
2D Convolution
2D Batch Normalization
ReLU
2D Max Pooling
Dropout
2D Convolution
2D Batch Normalization
ReLU
2D Max Pooling
Dropout
Flatten
Fully-Connected
Softmax

[4]

FBCNN – 2D

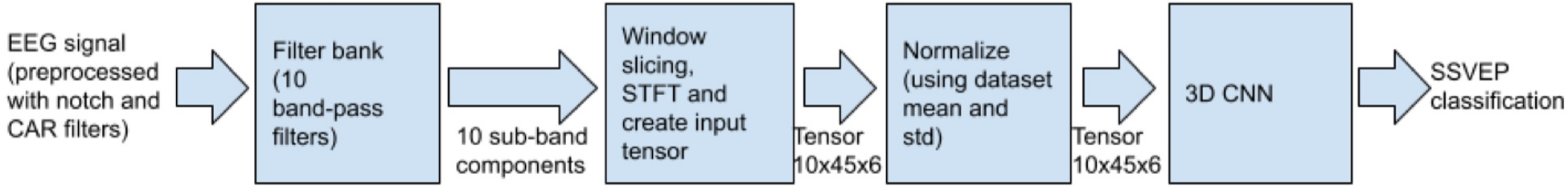

EEG signal (preprocessed with notch and CAR filters)
Filter bank (10 band-pass filters)
10 sub-band components
Window slicing, STFT and create input matrix
Matrix 20x45
Normalize (using dataset mean and std)
Matrix 20x45
2D CNN
SSVEP classification

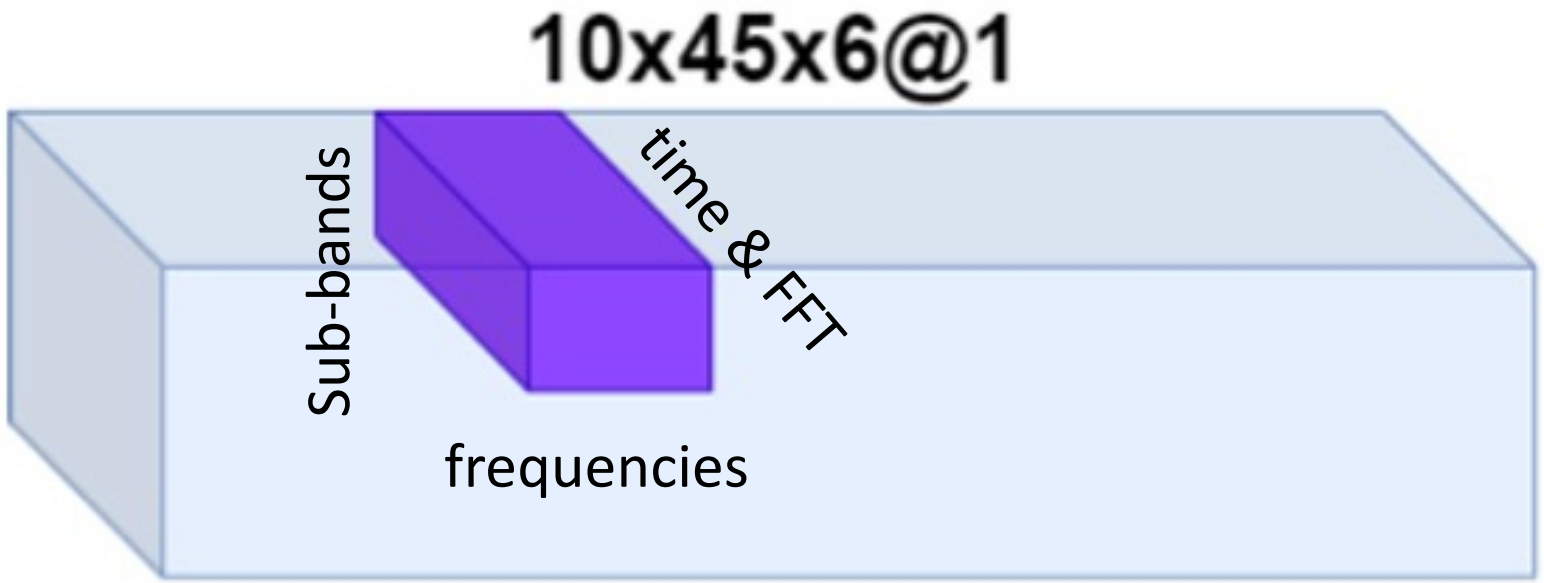


[4]

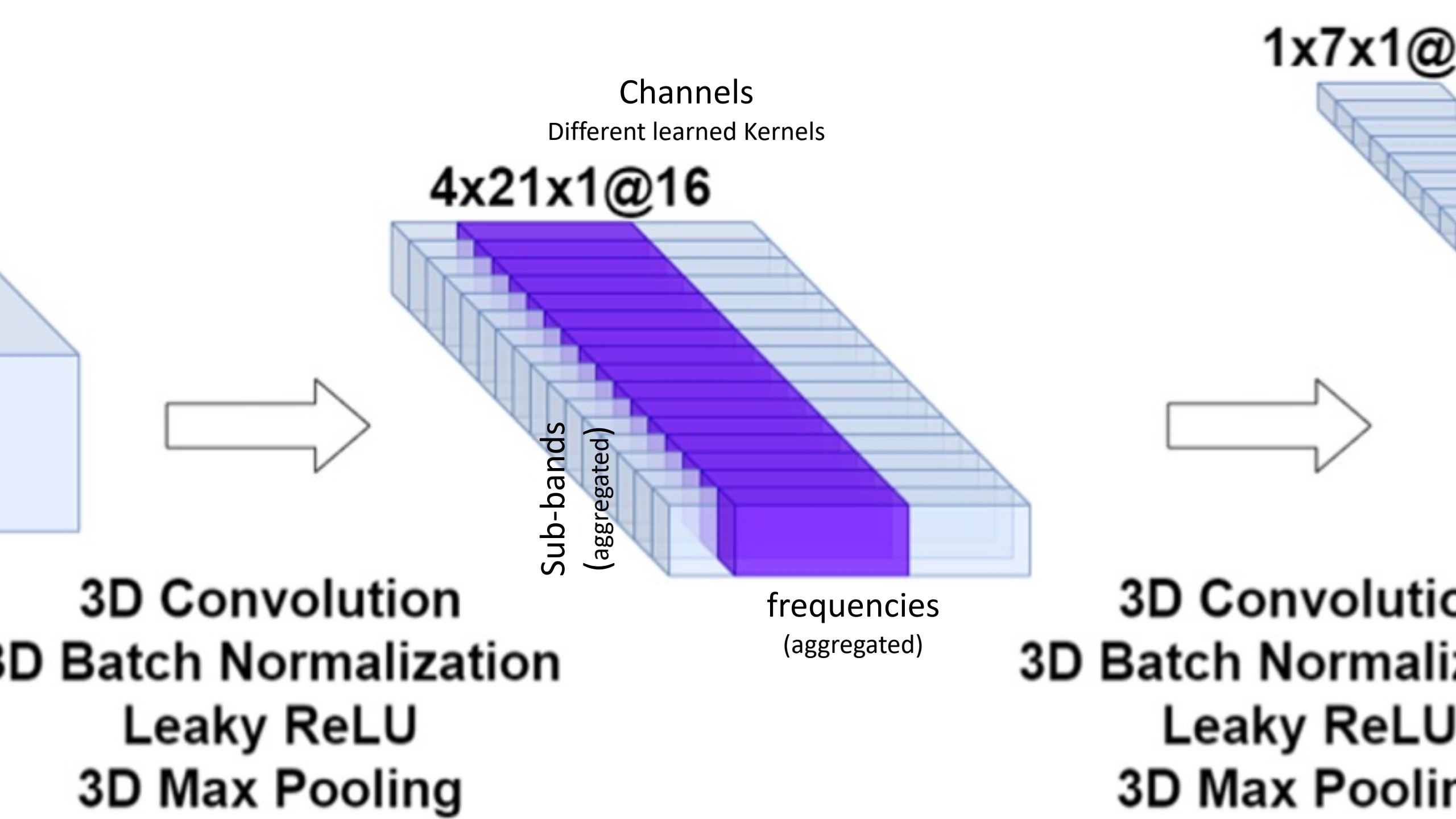
FBCNN – 3D

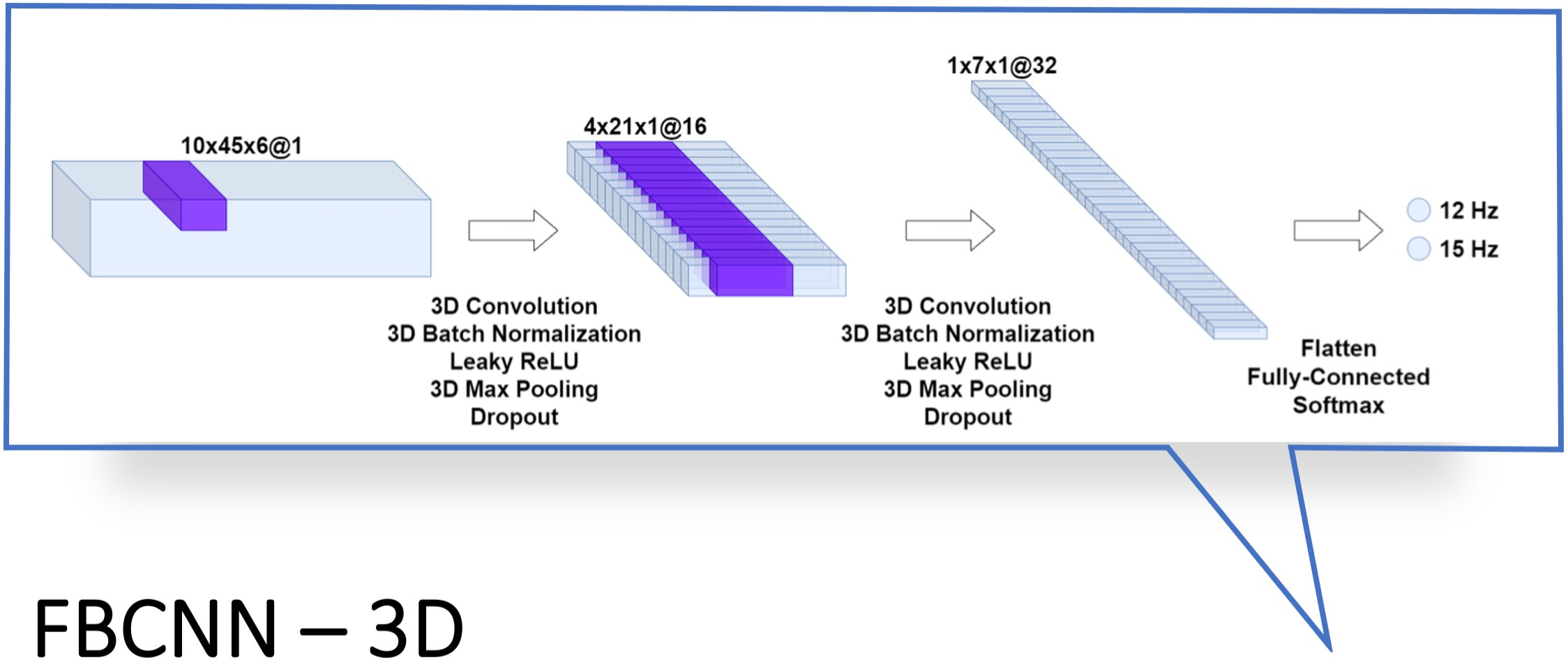
1. Windowing [0.0s – 0.5s] (0.167s)
2. Short-Time Fourier Transform [0Hz – 90Hz] (2Hz)
3. Stack Re & Im on **time axis**





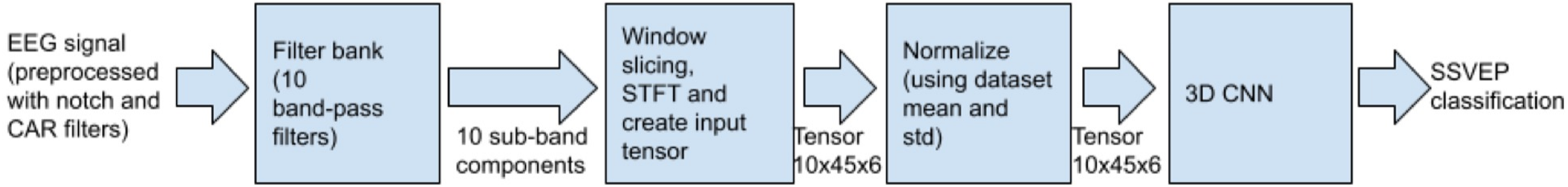
3D Convolu
3D Batch Norm
Leaky Re

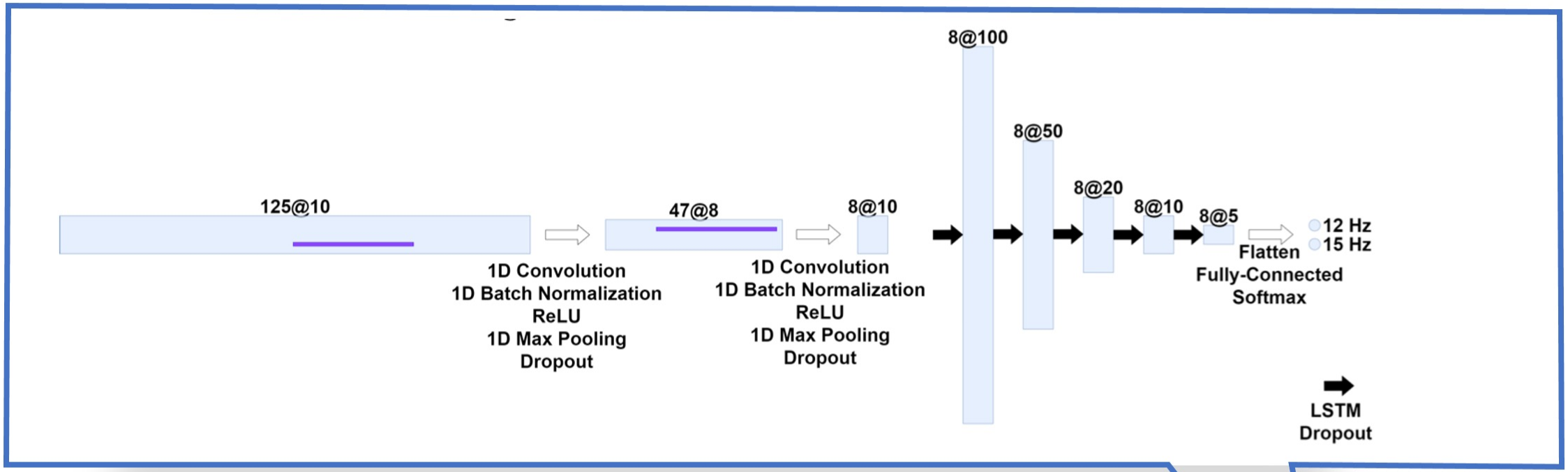

Channels
Different learned Kernels
4x21x1@16
1x7x1@
Sub-bands
(aggregated)
frequencies
(aggregated)
3D Convolution
3D Batch Normalization
Leaky ReLU
3D Max Pooling
3D Convolutio
3D Batch Normali
Leaky ReLU
3D Max Poolin


10x45x6@1
4x21x1@16
1x7x1@32
3D Convolution
3D Batch Normalization
Leaky ReLU
3D Max Pooling
Dropout
3D Convolution
3D Batch Normalization
Leaky ReLU
3D Max Pooling
Dropout
Flatten
Fully-Connected
Softmax
12 Hz
15 Hz

[4]

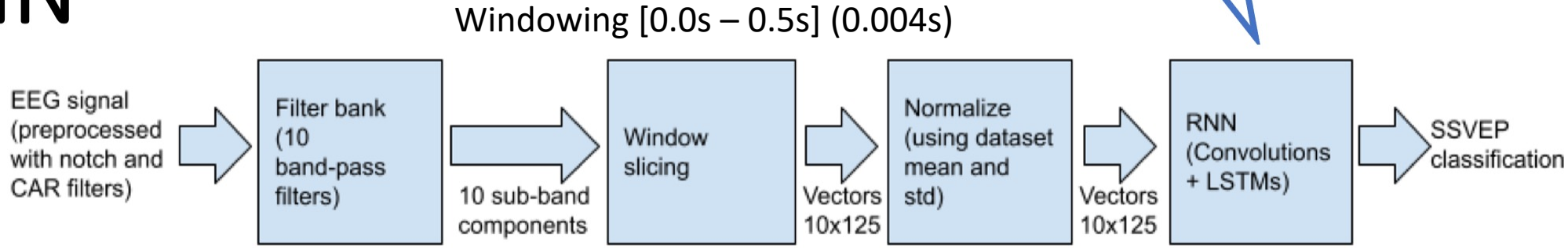
FBCNN – 3D

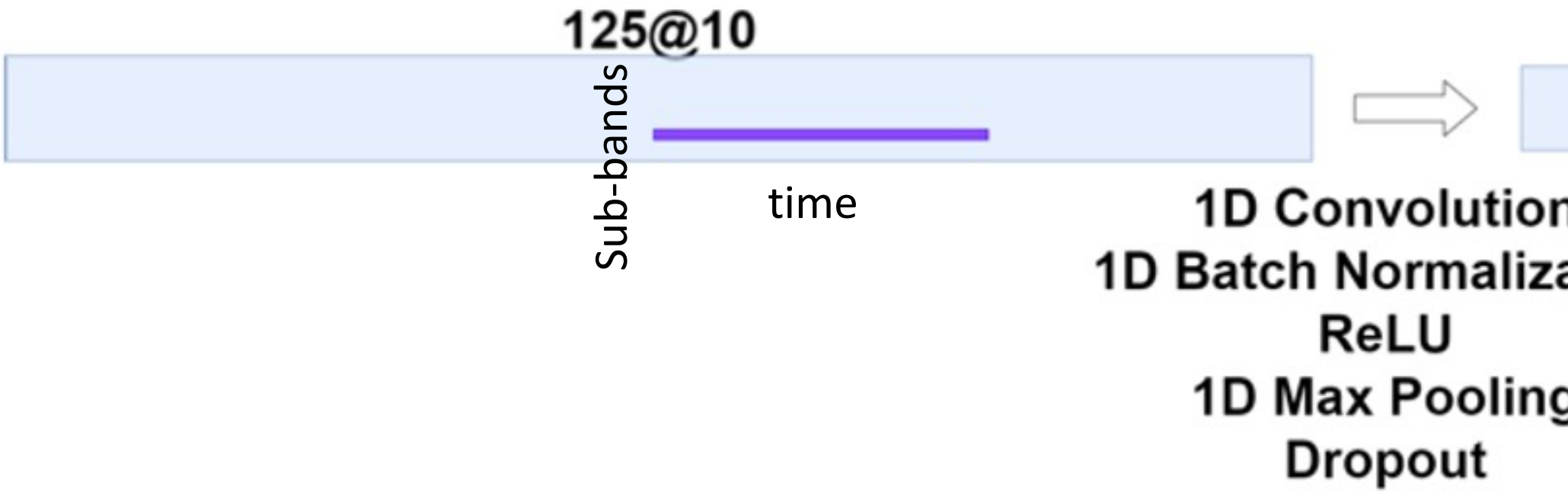

EEG signal (preprocessed with notch and CAR filters)
Filter bank (10 band-pass filters)
10 sub-band components
Window slicing, STFT and create input tensor
Tensor 10x45x6
Normalize (using dataset mean and std)
Tensor 10x45x6
3D CNN
SSVEP classification

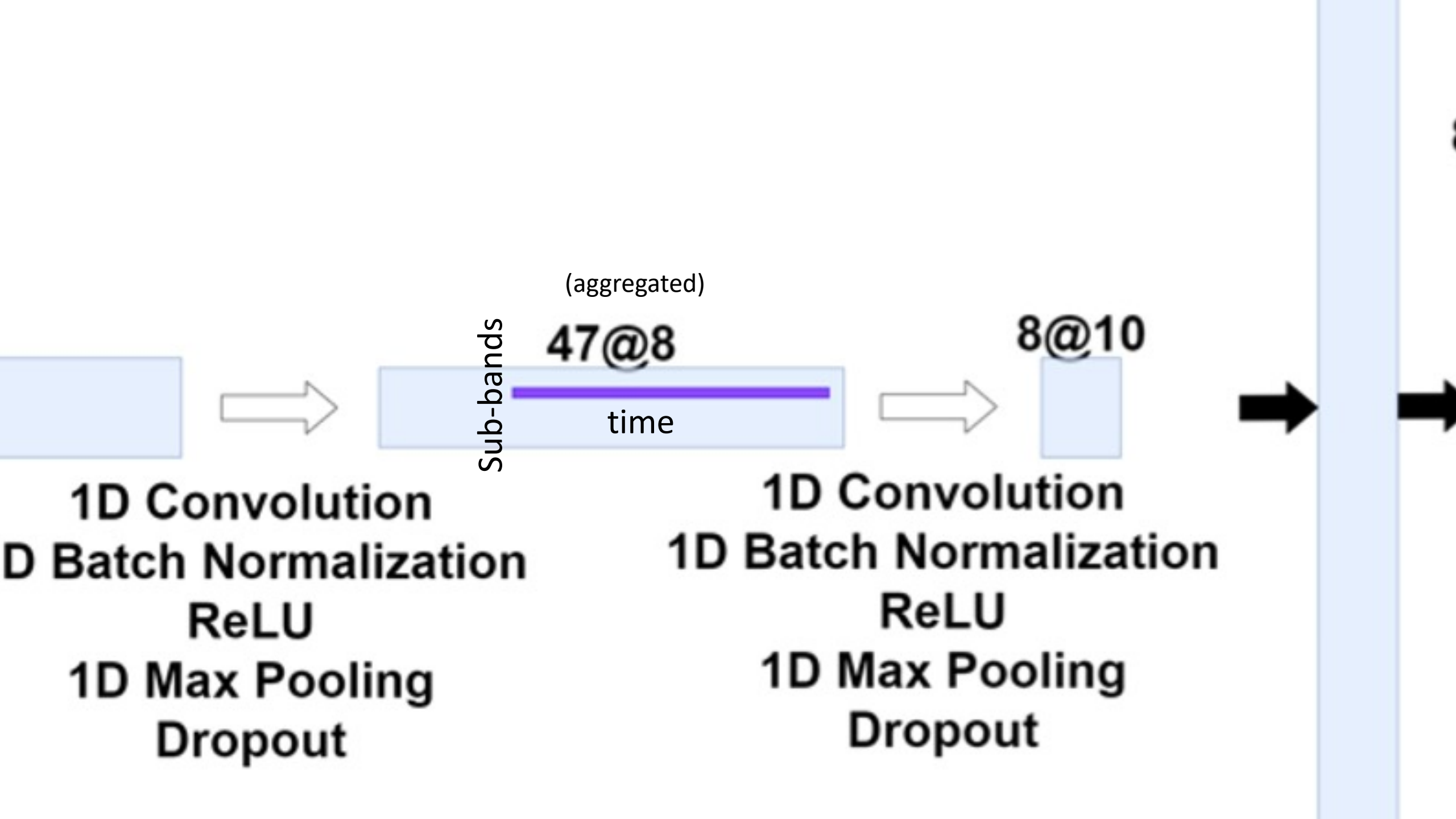

125@10
1D Convolution
1D Batch Normalization
ReLU
1D Max Pooling
Dropout
47@8
1D Convolution
1D Batch Normalization
ReLU
1D Max Pooling
Dropout
8@10
8@100
8@50
8@20
8@10
8@5
Flatten
Fully-Connected
Softmax
12 Hz
15 Hz
LSTM
Dropout

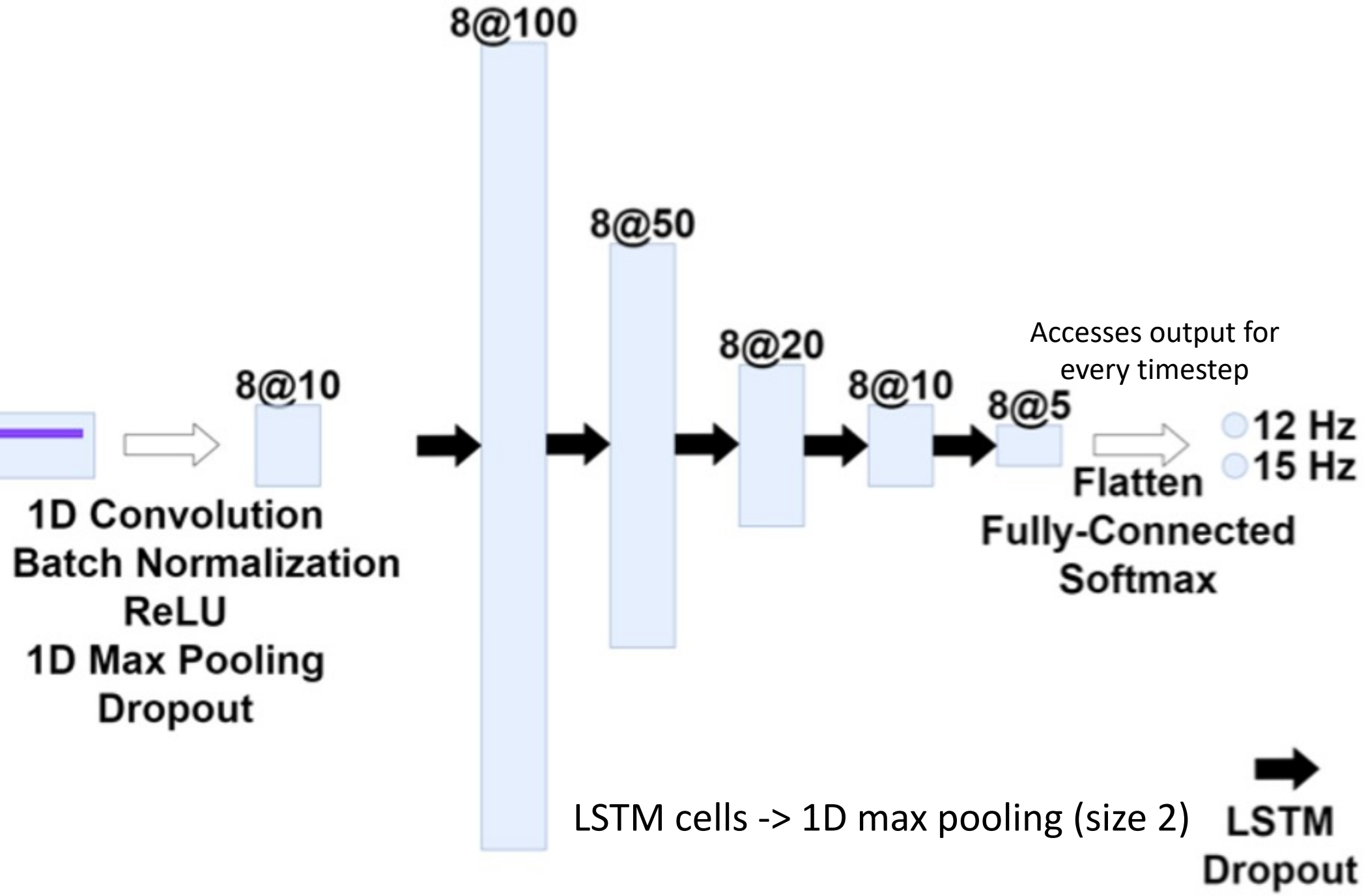
[4]

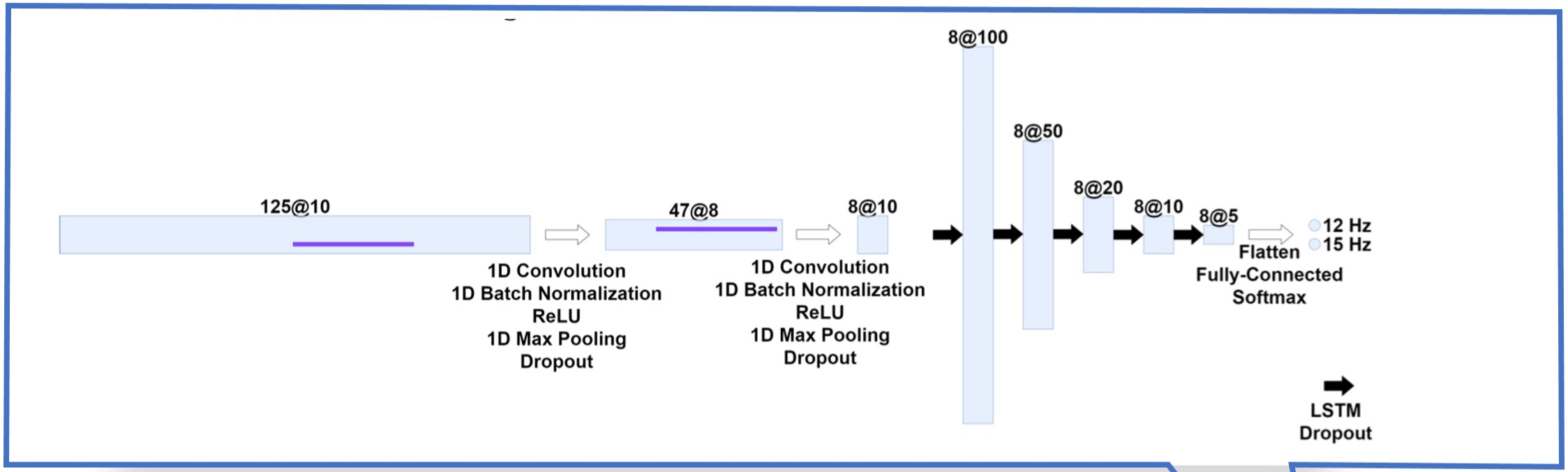
FBRNN


Windowing [0.0s – 0.5s] (0.004s)
EEG signal (preprocessed with notch and CAR filters)
Filter bank (10 band-pass filters)
10 sub-band components
Window slicing
Vectors 10x125
Normalize (using dataset mean and std)
Vectors 10x125
RNN (Convolutions + LSTMs)
SSVEP classification


125@10
Sub-bands
time
1D Convolution
1D Batch Normaliza
ReLU
1D Max Pooling
Dropout

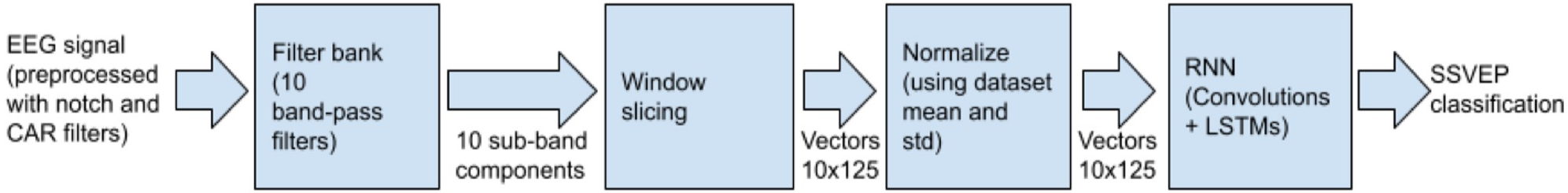

(aggregated)
47@8
Sub-bands
time
8@10
1D Convolution
D Batch Normalization
ReLU
1D Max Pooling
Dropout
1D Convolution
1D Batch Normalization
ReLU
1D Max Pooling
Dropout


8@100
8@50
8@20
8@10
8@10
8@5
Accesses output for every timestep
12 Hz
15 Hz
Flatten
Fully-Connected
Softmax
1D Convolution
Batch Normalization
ReLU
1D Max Pooling
Dropout
LSTM cells -> 1D max pooling (size 2)
LSTM
Dropout

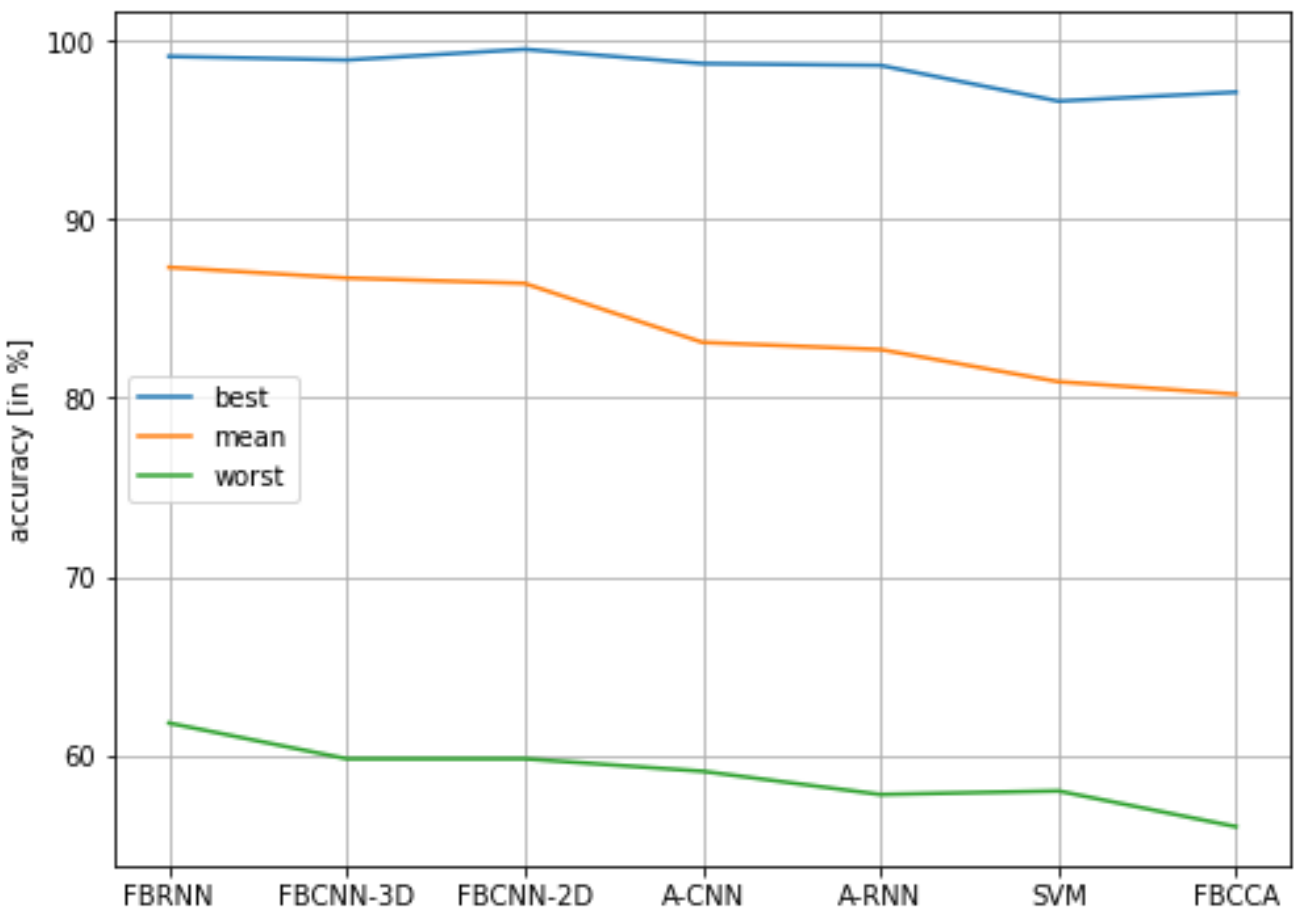

125@10
1D Convolution
1D Batch Normalization
ReLU
1D Max Pooling
Dropout
47@8
1D Convolution
1D Batch Normalization
ReLU
1D Max Pooling
Dropout
8@10
8@100
8@50
8@20
8@10
8@5
Flatten
Fully-Connected
Softmax
12 Hz
15 Hz
LSTM
Dropout

[4]

FBRNN


EEG signal (preprocessed with notch and CAR filters)
Filter bank (10 band-pass filters)
10 sub-band components
Window slicing
Vectors 10x125
Normalize (using dataset mean and std)
Vectors 10x125
RNN (Convolutions + LSTMs)
SSVEP classification

How do they compare?



Subjects	FBRNN	FBCNN-3D	FBCNN-2D	A-CNN	A-RNN	SVM	FBCCA
1	83.7	82.8	81.9	77.7	78.1	80.8	86.8
2	87.9	87.7	87	83.7	79.5	82.6	87.9
3	97.6	97.3	97.5	95.3	94.9	93.7	94.7
4	78.4	77	78.1	75.4	74.8	69.6	70.3
5	97.6	96.2	96.2	93.8	93.5	89.7	96.6
6	96	95.5	96.7	93.5	93.8	90.6	87
7	91.8	93.1	92.4	88.8	88	86.6	88.4
8	96.7	96.6	96.4	93.7	95.3	91.3	88.8
9	86.8	81.9	81.2	82.8	78.6	79.2	80.4
10	92.8	92	94.4	90.4	89.9	86.4	90.8
11	70.5	67.9	66.3	65.6	65.8	62.1	56.2
12	78.3	77.7	78.8	63.8	73.4	66.7	62.9
13	61.8	59.8	59.8	59.1	57.8	58	56
14	76.3	77.9	76.4	70.5	70.8	67.6	65.9
15	92.2	94.6	94.6	90	88.9	88.4	84.1
16	87	85.5	80.4	79	75.7	72.3	79
17	80.3	78.3	79.9	76.6	78.1	70.5	73.6
18	83.5	78.6	79.2	78.6	76.8	77.2	77.4
19	80.8	82.1	78.4	75.9	73	75.4	70.3
20	93.7	92.6	94.4	92.2	92	86.8	90.2
21	78.6	79.5	77.2	75.5	75	75.2	73.7
22	98.7	97.6	96.6	95.8	95.7	94.2	83.5
23	77	75.5	74.8	71.6	69.4	65.2	71.6
24	97.1	96.9	97.3	95.3	92.2	91.5	91.8
25	92.4	94.2	93.7	84.6	86.2	81.5	76.8
26	93.1	94.7	94.2	91.7	90.2	89.1	77.2
27	97.1	95.5	96	93.3	89.9	89.5	95.1
28	96.2	95.5	95.5	94.4	92.4	90.8	96.4
29	73.4	72.6	71.2	54.3	60	58.3	53.3
30	97.3	96.9	96.4	94.4	95.1	93.1	93.5
31	94.4	94.6	96.2	93.5	93.8	90.8	86.1
32	99.1	98.9	99.5	98.7	98.6	96.6	97.1
33	65.4	68.5	66.8	69.6	67.4	72.3	71.4
34	86.2	84.2	84.6	78.8	78.6	77.7	74.3
35	95.7	93.7	93.1	90.8	92.9	89.1	77.4
Mean Accuracy	87.3	86.7	86.4	83.1	82.7	80.9	80.2
Mean F1-Score	0.877	0.872	0.867	0.831	0.829	0.807	0.786

[4]

What did the paper accomplish?

Challenges:

- One electrode
- Small data length

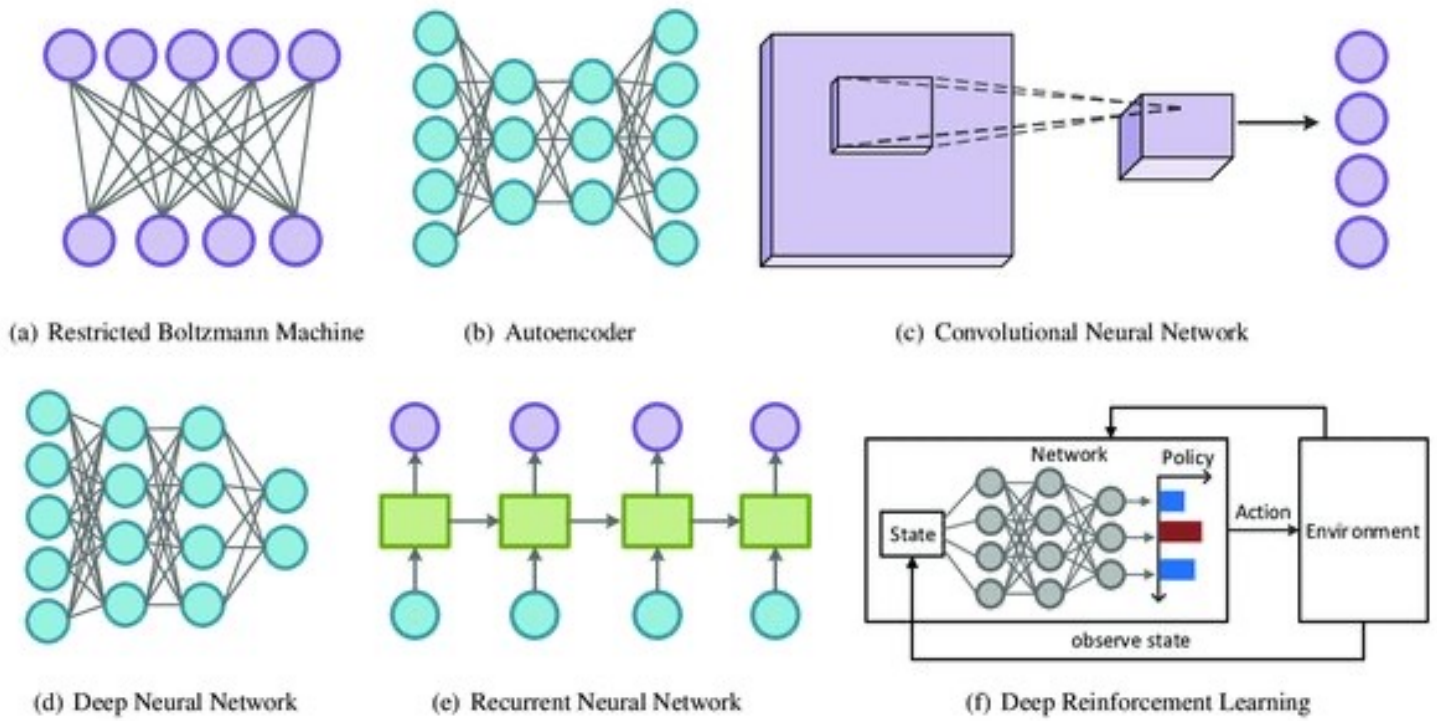
They both usually reduce performance drastically!

Results:

- Very good results considering this complicated scenario

Potential for portable and online application!

Main takeaway



- Choice of model matters!
- The choice isn't always trivial
- Clever preprocessing of the data can improve model performance
- BUT: model still needs freedom to learn the data

Source:
Wang, Fangxin & Zhang, Miao & Wang, Xiangxiang & Ma, Xiaoqiang & Liu, Jiangchuan. (2020). Deep Learning for Edge Computing Applications: A State-of-the-Art Survey. IEEE Access. PP. 1-1.

References

[1]: T. Nguyen and W. Chung, "A Single-Channel SSVEP-Based BCI Speller Using Deep Learning," in IEEE Access, vol. 7, pp. 1752-1763, 2019

[2]: Z. Lin, C. Zhang, W. Wu and X. Gao, "Frequency Recognition Based on Canonical Correlation Analysis for SSVEP-Based BCIs," in *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2610-2614, Dec. 2006

[3]: Chen, Xiaogang & Wang, Yijun & Gao, Shangkai & Jung, Tzyy-Ping & Gao, Xiaorong. (2015). Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain–computer interface. Journal of neural engineering.

[4]: Bassi, Pedro R. A. S. and Romis de Faissol Attux. “FBCNN: A Deep Neural Network Architecture for Portable and Fast Brain-Computer Interfaces.” ArXiv abs/2109.02165 (2021)

[5]: Kazumi Ishizuka, Nobuaki Kobayashi, and Ken Saito, “High Accuracy and Short Delay 1ch-SSVEP Quadcopter-BMI Using Deep Learning,” J. Robot. Mechatron., Vol.32, No.4, pp. 738-744, 2020.

[6]: Kwak NS, Mu¨ller KR, Lee SW. A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. PLOS ONE 2017;12(2):1–20

[7]: Li, Li & Song, Zhenyu & Liang, Lei & Wang, Yiding. (2010). Design a Digital Notch Filter Based on Matlab Program for Photoacoustic Gas Sensing Systems. Electrical and Control Engineering, International Conference on. 671-674.