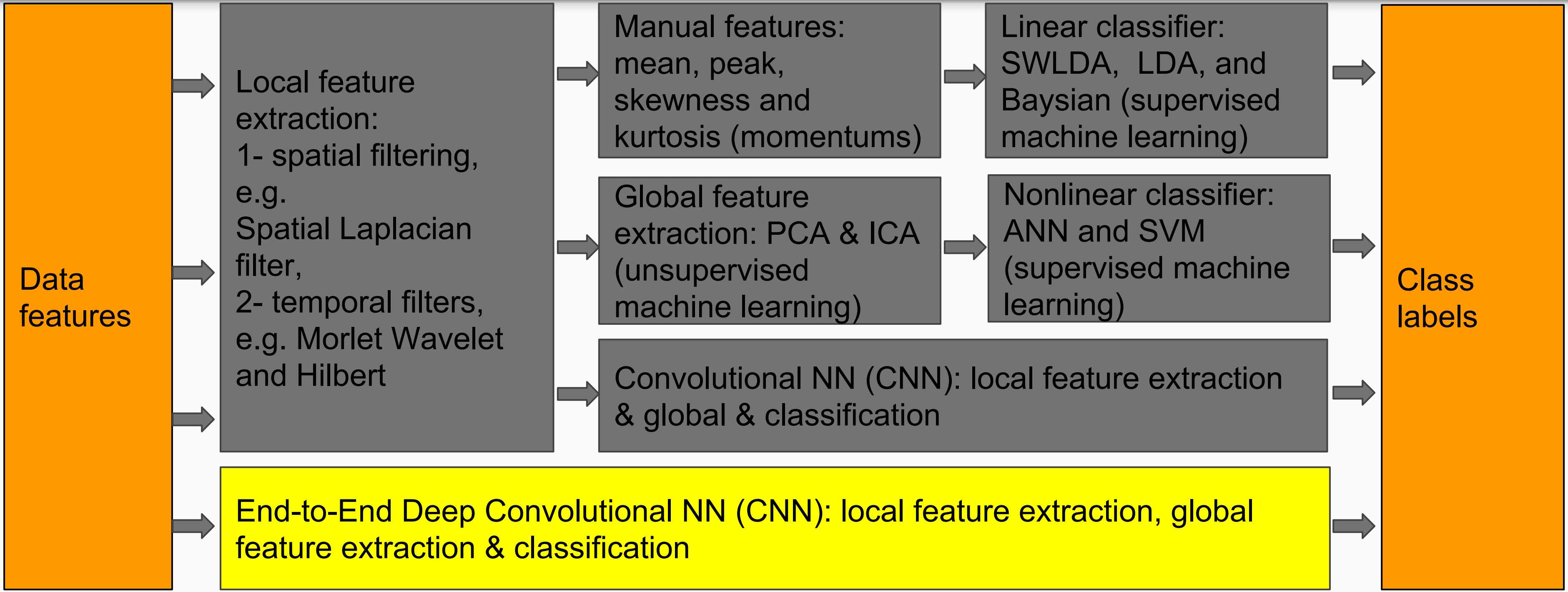
# An end-to-end CNN in BCI with EEG for target/non-target classification task

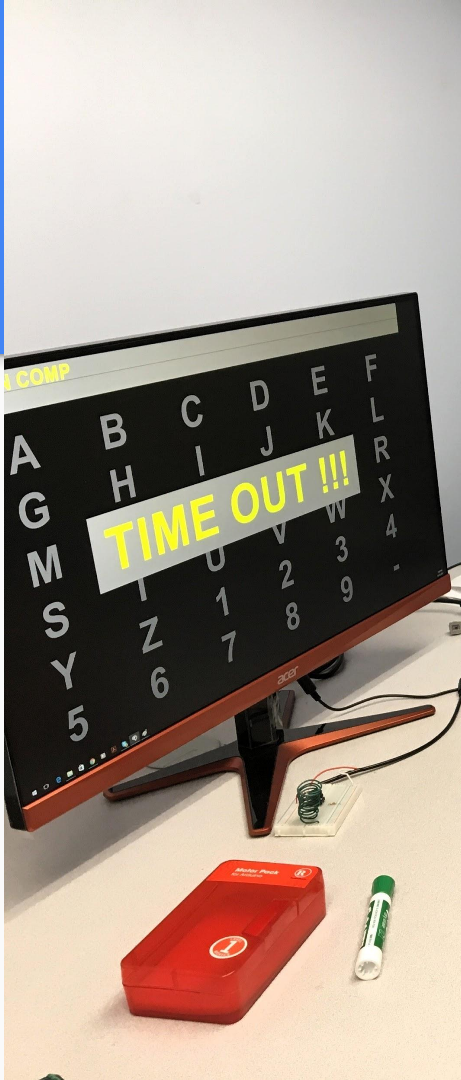
Convolutional vs conventional approaches

# My approach vs Conventional BCI approaches



# Data collection and experimental setup

1. Target vs non-target detection for all subjects
2. How all subjects Faces vs Flips perform in classification
3. Five subjects data
4. Labels were 83% non-target vs 33% target
5. Labels were very unbalanced
6. Walk through github and for data shape and type
7. Data were RAW, no filtering only epochs/ trials
8. Input data features is normalized and labels are one-hot encoded.



# End-to-end CNN

1. Local feature extraction: convolutional filters
2. Global feature extraction: fully connected layer or MLP
3. Classifier: Softmax
4. Loss function: cross entropy
5. Learning approach: backprop

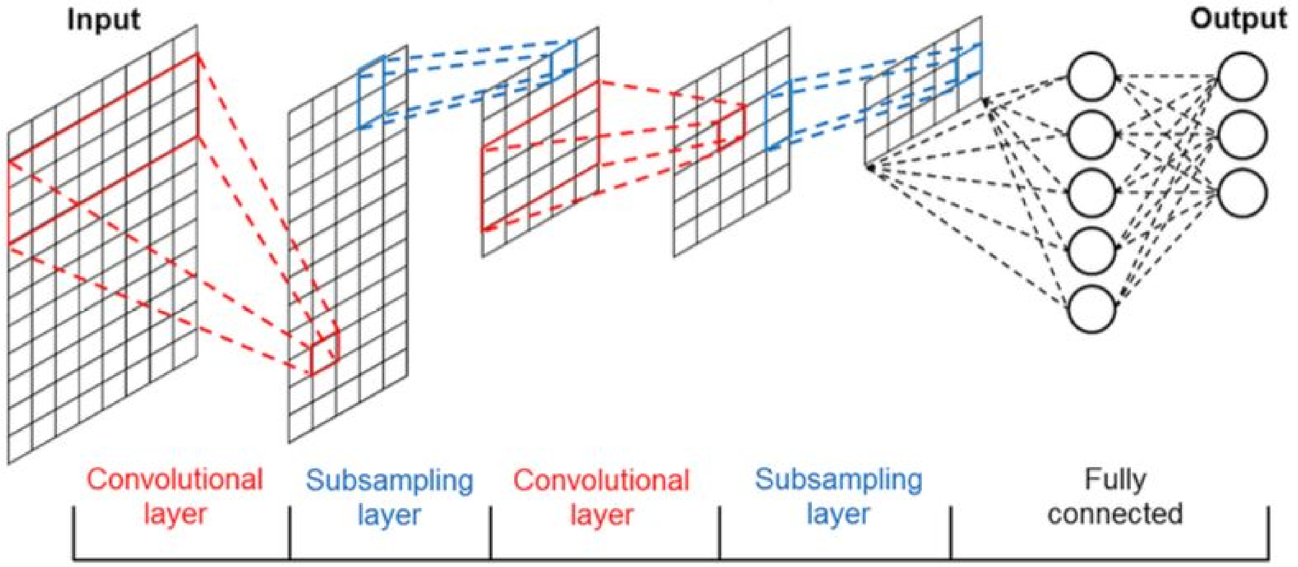
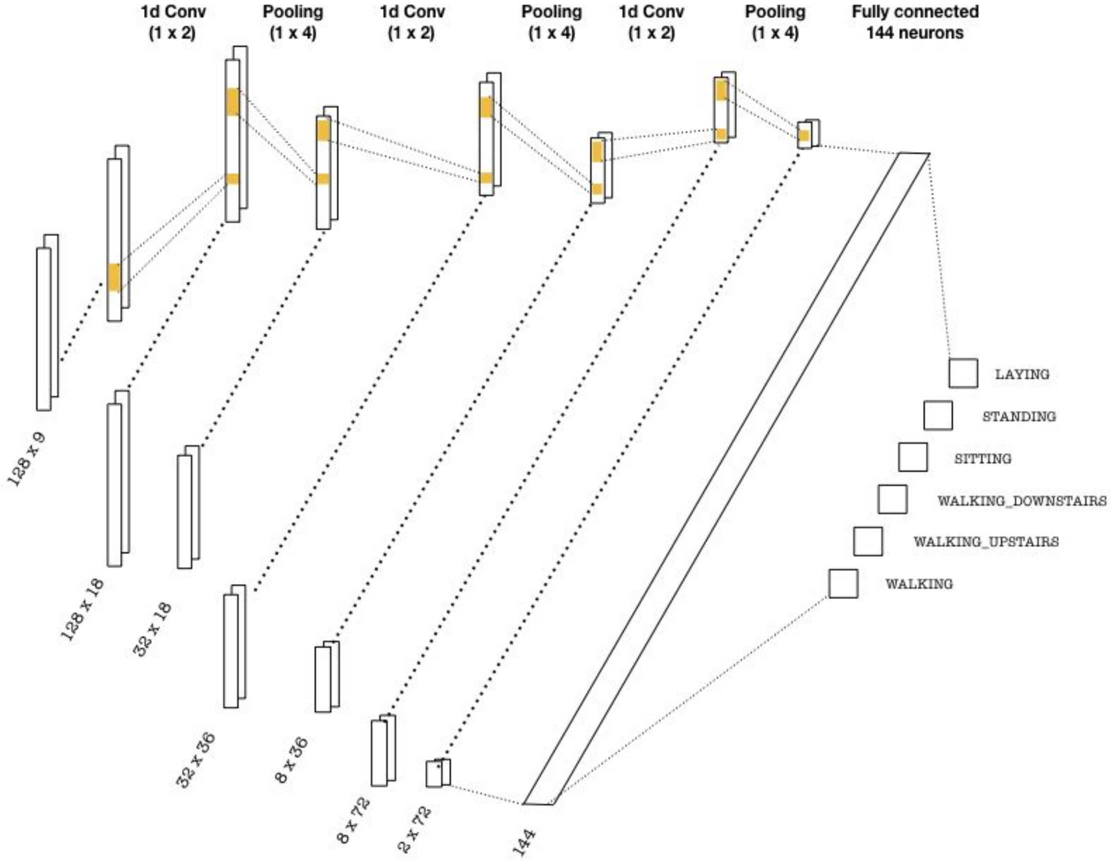
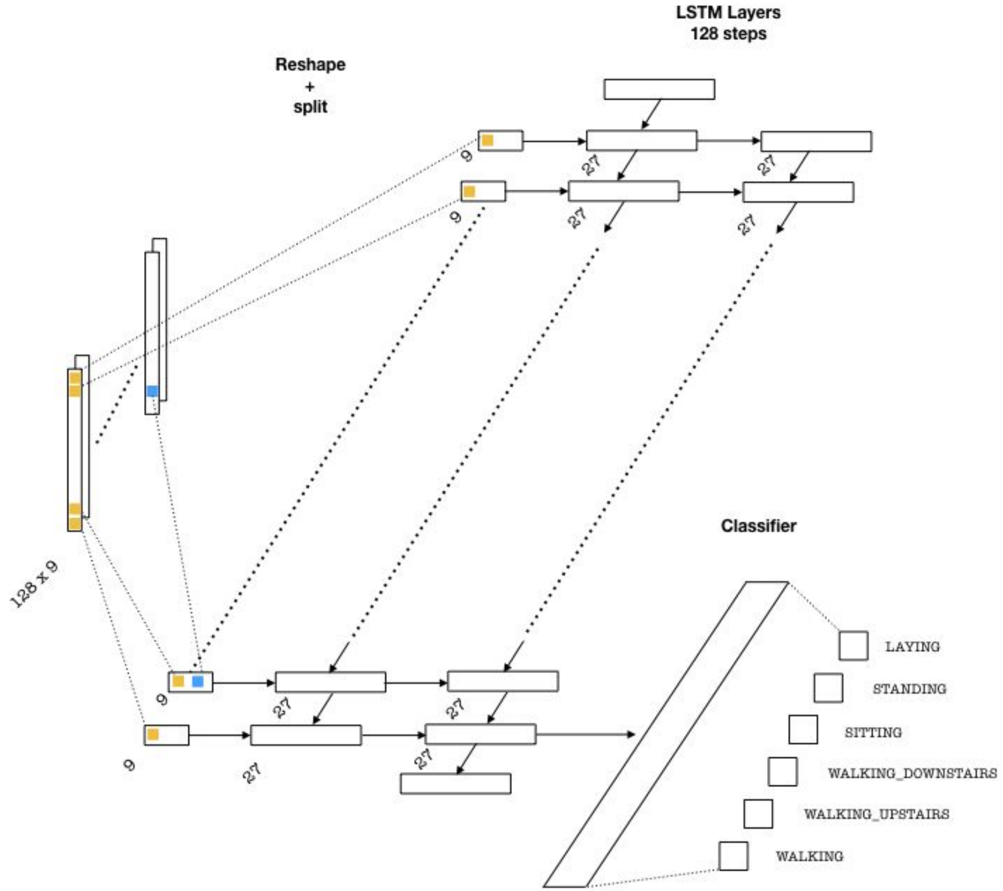



**Fig. 2** The common structure of convolutional neural network.

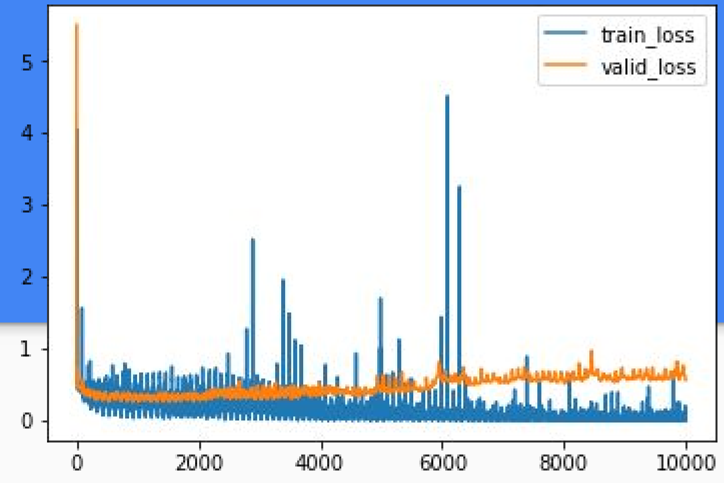

1d Conv (1 x 2)
Pooling (1 x 4)
1d Conv (1 x 2)
Pooling (1 x 4)
1d Conv (1 x 2)
Pooling (1 x 4)
Fully connected 144 neurons
128 x 9
128 x 18
32 x 18
32 x 36
8 x 36
8 x 72
2 x 72
144
LAYING
STANDING
SITTING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS
WALKING

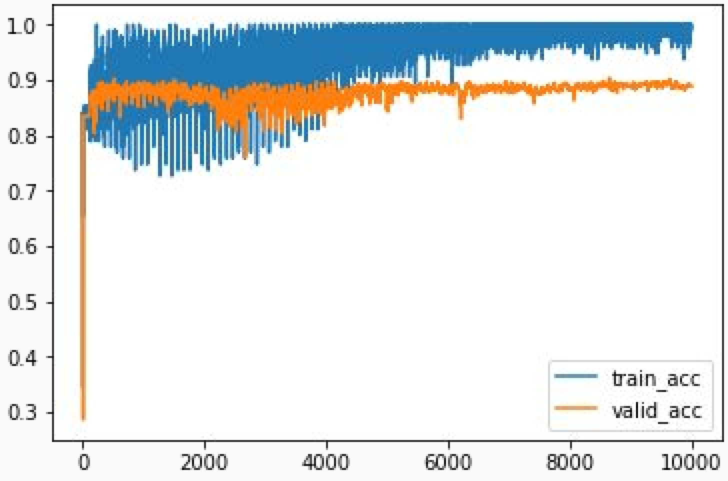


LSTM Layers
128 steps
Reshape
+
split
9
27
27
9
27
27
128 x 9
Classifier
LAYING
STANDING
SITTING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS
WALKING
9
27
27
9
27
27

# Results on all subject Faces

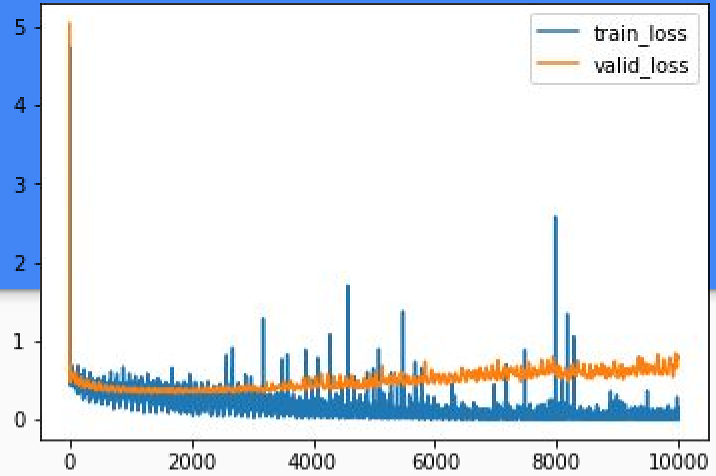
1. Epoch: 100/100 Train loss: 0.003697 Valid loss: 0.468953
2. Epoch: 100/100 Train acc: 1.000000 Valid acc: 0.877434
3. Epoch: 100/100 Test loss: 0.698612 Test acc: 0.880451
4. Overfitting is happening

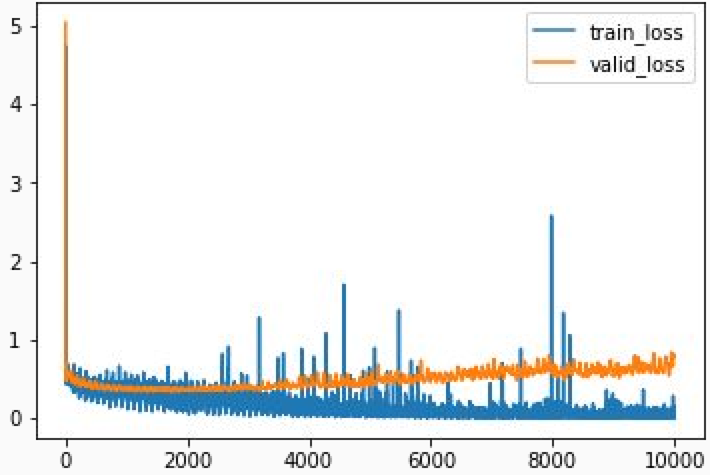


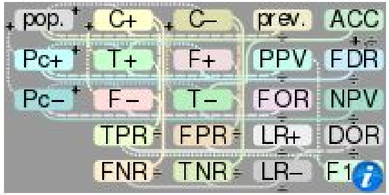
# Results on all subjects Flipps

1. Epoch: 100/100 Train loss: 0.024262 Valid loss: 0.488262
2. Epoch: 100/100 Train acc: 0.989130 Valid acc: 0.861963
3. Epoch: 100/100 Test loss: 0.921476 Test acc: 0.869922
4. Overfitting is also happening



# Confusion matrix

<table>
<tr><td colspan="2"></td><td colspan="2">True condition</td><td colspan="2"></td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td><td>Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$</td><td colspan="2">Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$</td></tr>
<tr><td rowspan="2">Predicted condition</td><td>Predicted condition positive</td><td>True positive</td><td>False positive,<br>Type I error</td><td>Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$</td><td colspan="2">False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$</td></tr>
<tr><td>Predicted condition negative</td><td>False negative,<br>Type II error</td><td>True negative</td><td>False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$</td><td colspan="2">Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$</td></tr>
<tr><td colspan="2" rowspan="2">Click thumbnail for interactive chart:<br></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td><td>Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$</td><td rowspan="2">Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$</td><td rowspan="2">$F_1$ score $= \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$</td></tr>
<tr><td>False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td><td>Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$</td></tr>
</table>

# Final point

Convolutional vs conventional:
CNNs are computational heavy
They need a lot of data.
End-to-end possible
Overfitting problem
Much easier than conventional

“Try to make things simple enough but not simpler than that”

- Probably from Einstein
Provable by Heisenberg uncertainty principle