Eyedrone

TEAM

Albert

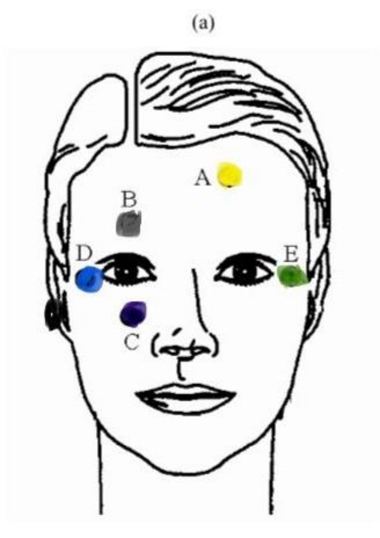
Antonio

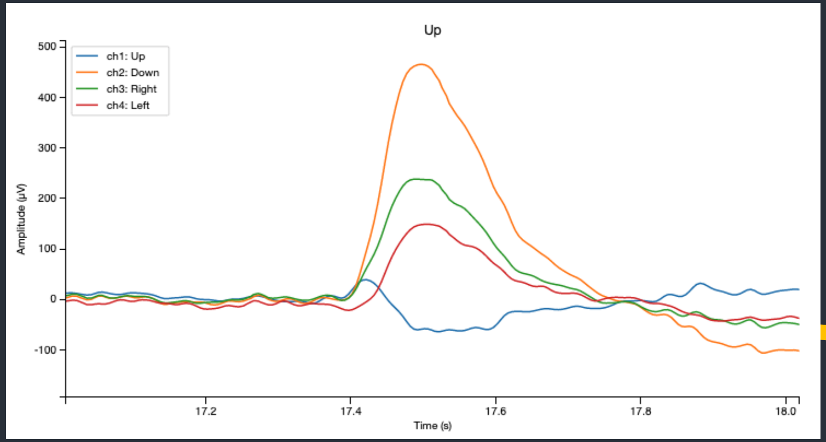
Eva

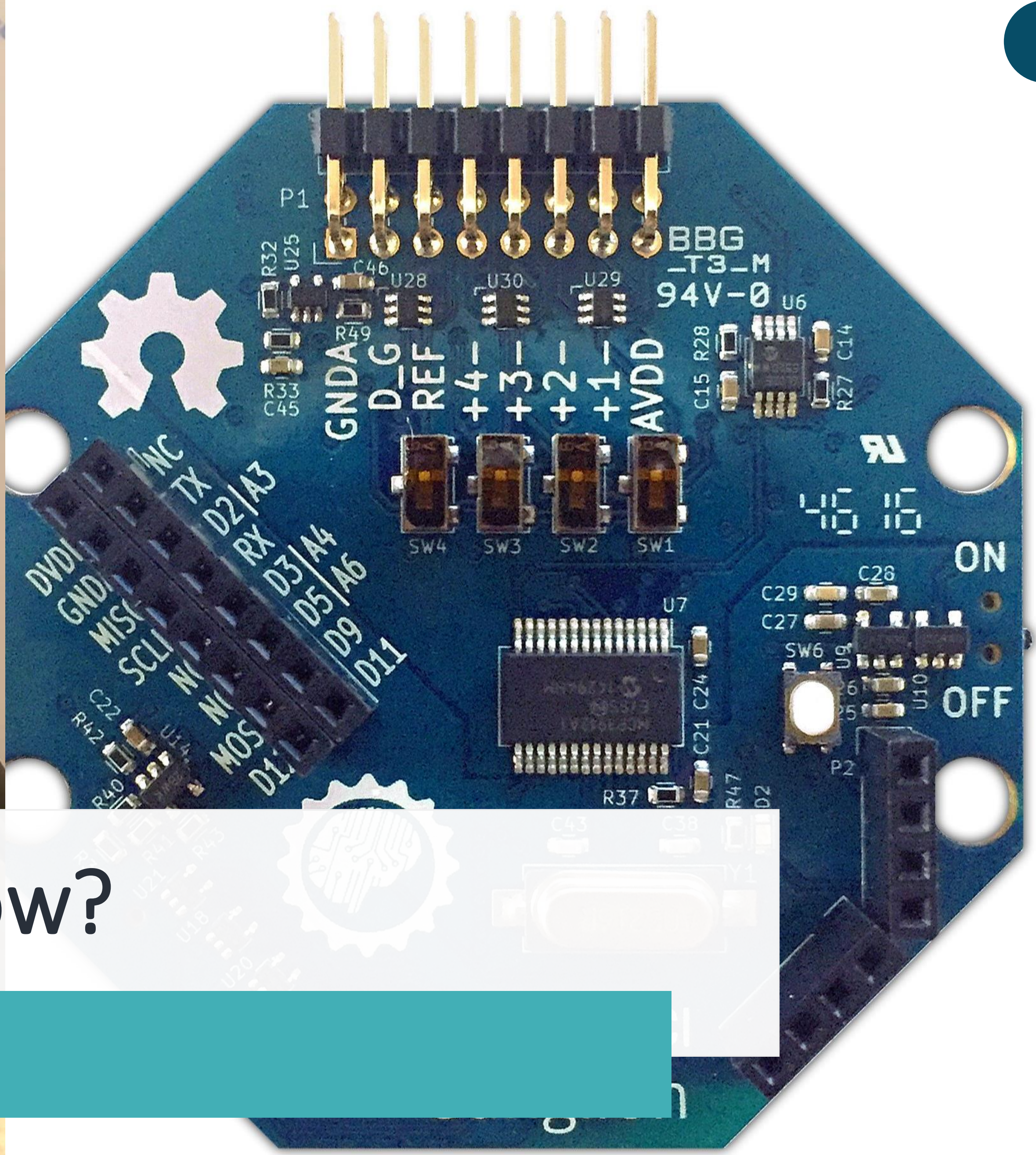
Asma

Ali

Move a **drone** using **your eyes**

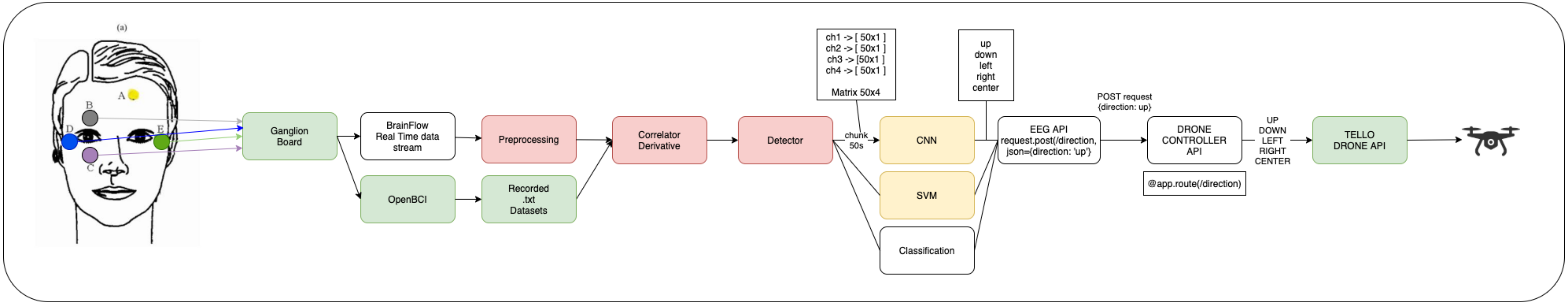





P1
BBG
_T3_M
94V-0
GNDA
D_G
REF
+4-
+3-
+2-
+1-
AVDD
SW4
SW3
SW2
SW1
NC
TX
D2|A3
RX
D3|A4
D5|A6
D9
D11
ON
OFF

How?

The end-to-end system



1. Data acquisition, preprocessing and detection

- Get raw data from sensors

- Process in real time
- When a change is detected, send a chunk of data as input to the neural network

2. Classification using AI

- Get movement of the eye as output

3. Drone controller

- Send movement to Drone
- Drone executes the movement

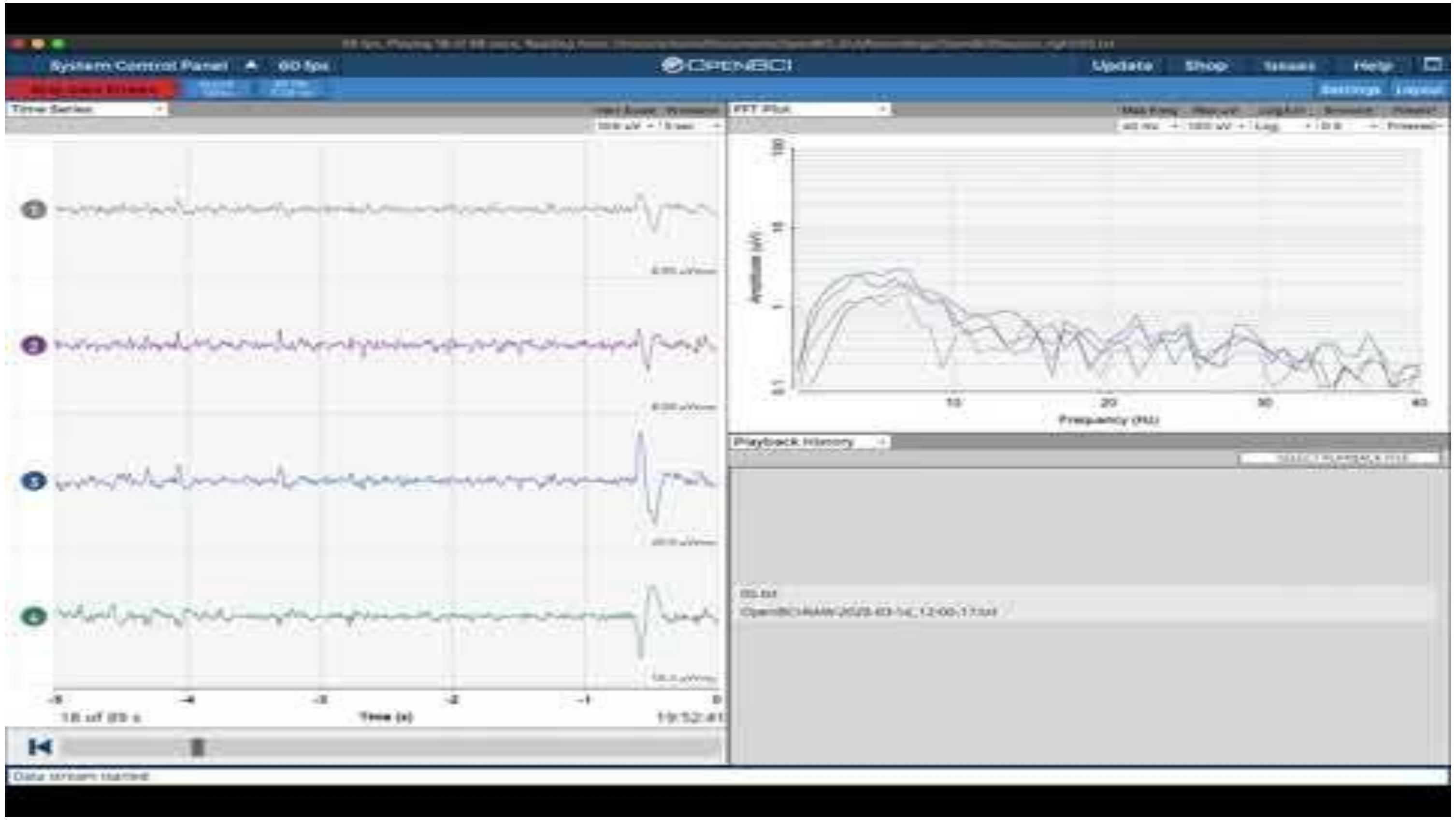
Microservices approach:

parts are interchangeable. Example: instead of using eye movements, we could use brain "thoughts", just by changing the first part

Ganglion Board + Sensors + Open BCI GUI

Recording datasets

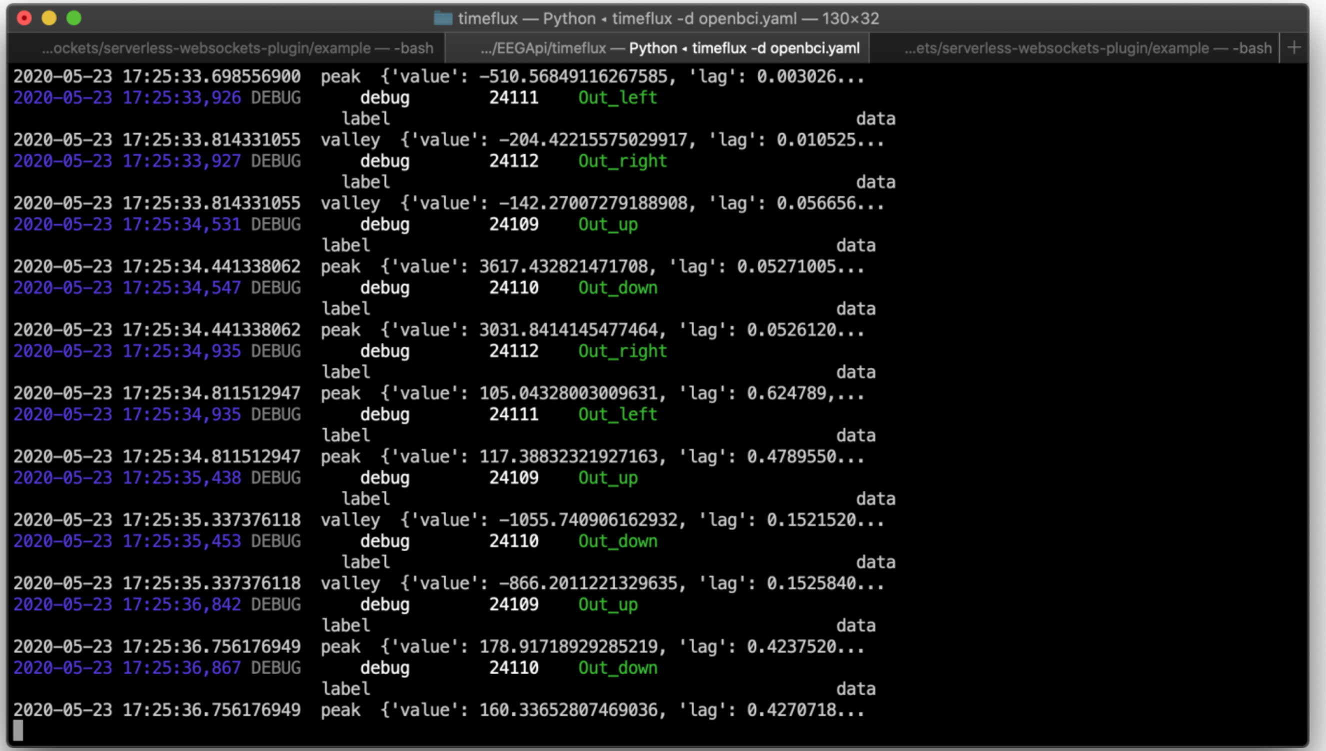
- Usage of Ganglion Board (4-channel EEG) + golden cup sensors.
- Recordings made with OpenBCI GUI

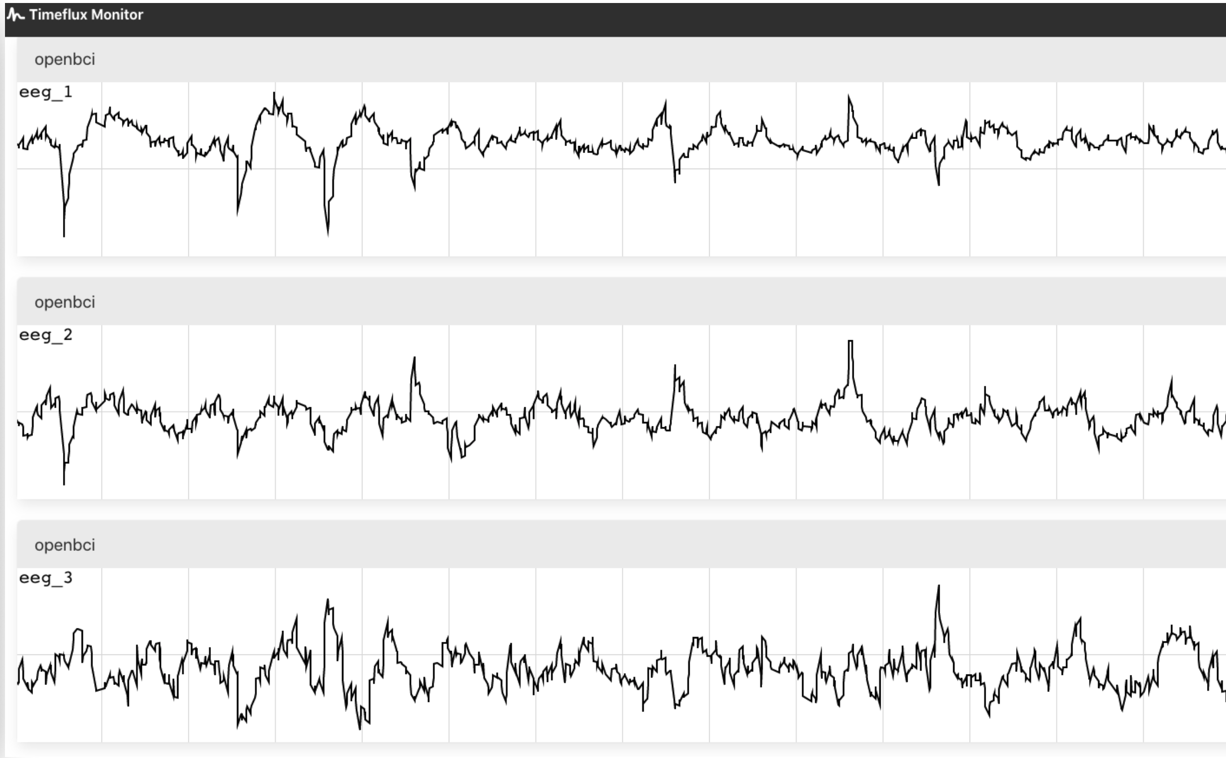


Recording example right eye movement

Ganglion Board + Brainflow + Timeflux

Raw data acquisition directly from board for real –time detection



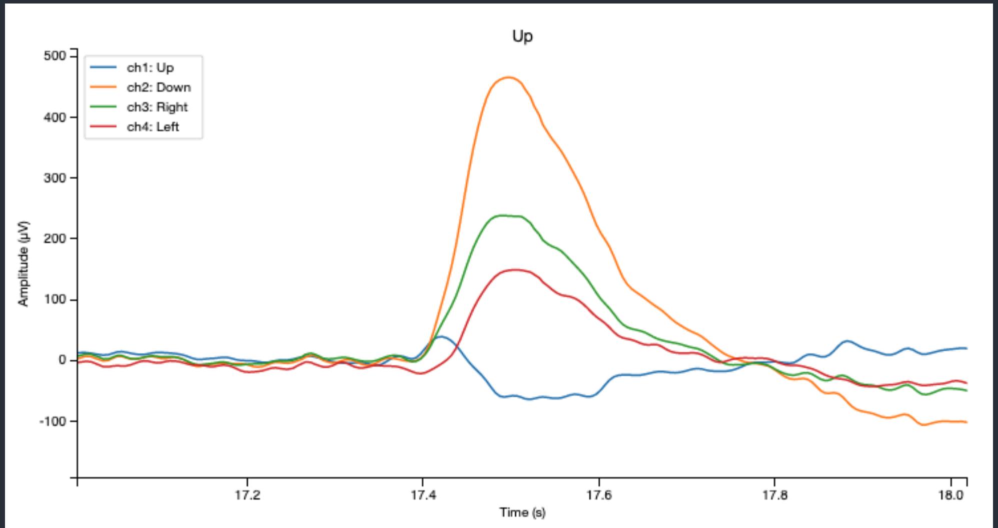


Unfortunately, there are some parts missing:

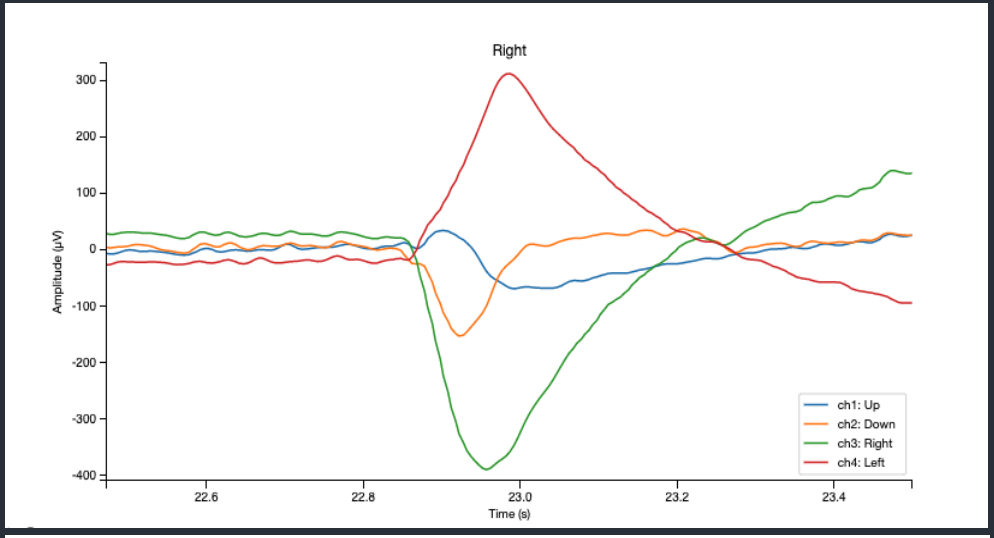
What is missing? Real Time processing.

- Get chunks of data in real time when an event happens (not only the event) and pass this as an input to the Neural Net. Making this work in real-time was not an easy work.
- Pass the output of the neural net to the Drone API in real time.
- Investigate REST vs Websockets to connect with the drone.

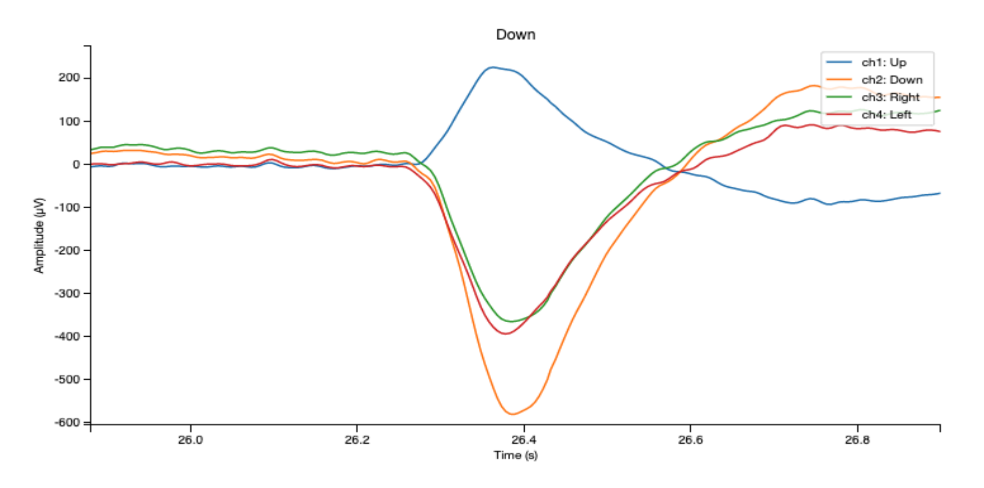
Dataset



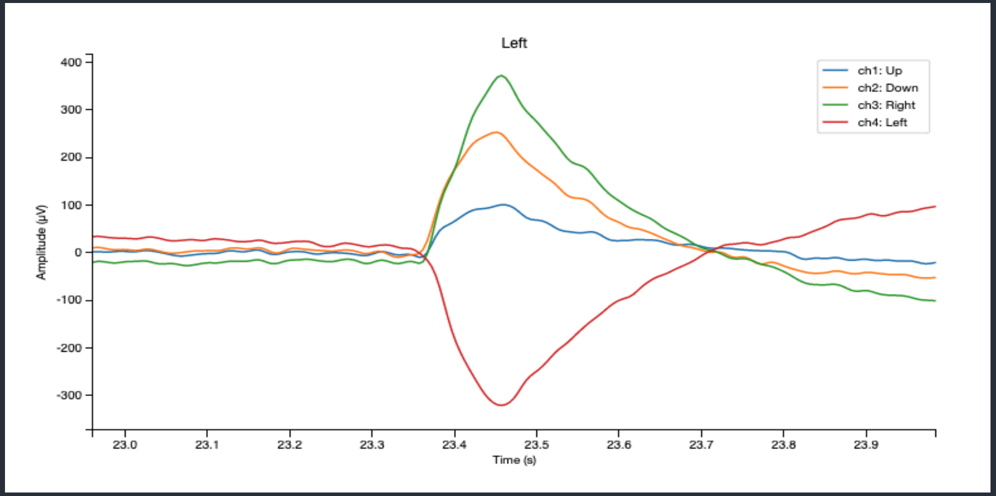
Up
ch1: Up
ch2: Down
ch3: Right
ch4: Left
Amplitude (μV)
500
400
300
200
100
0
-100
17.2
17.4
17.6
17.8
18.0
Time (s)



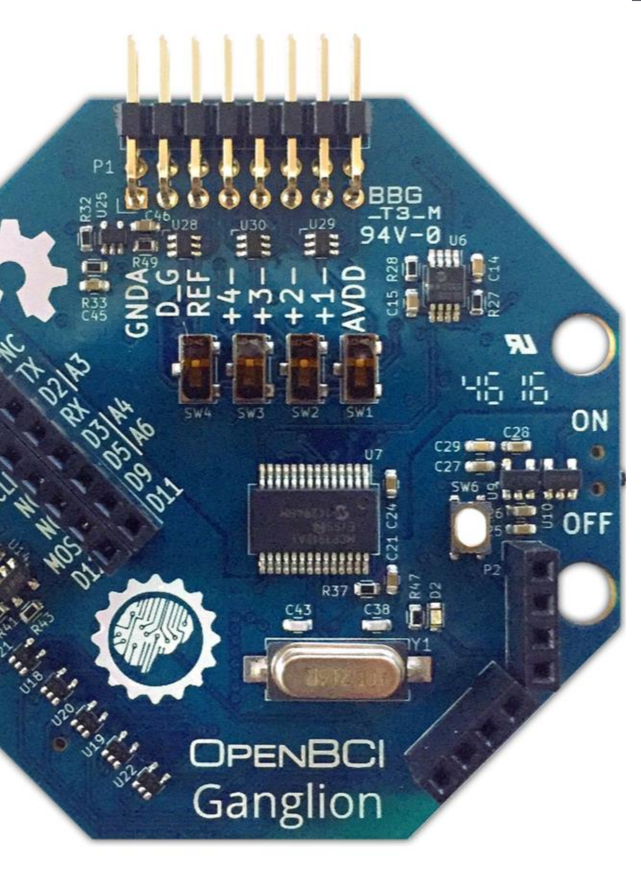
Right
ch1: Up
ch2: Down
ch3: Right
ch4: Left
Amplitude (μV)
300
200
100
0
-100
-200
-300
-400
22.6
22.8
23.0
23.2
23.4
Time (s)

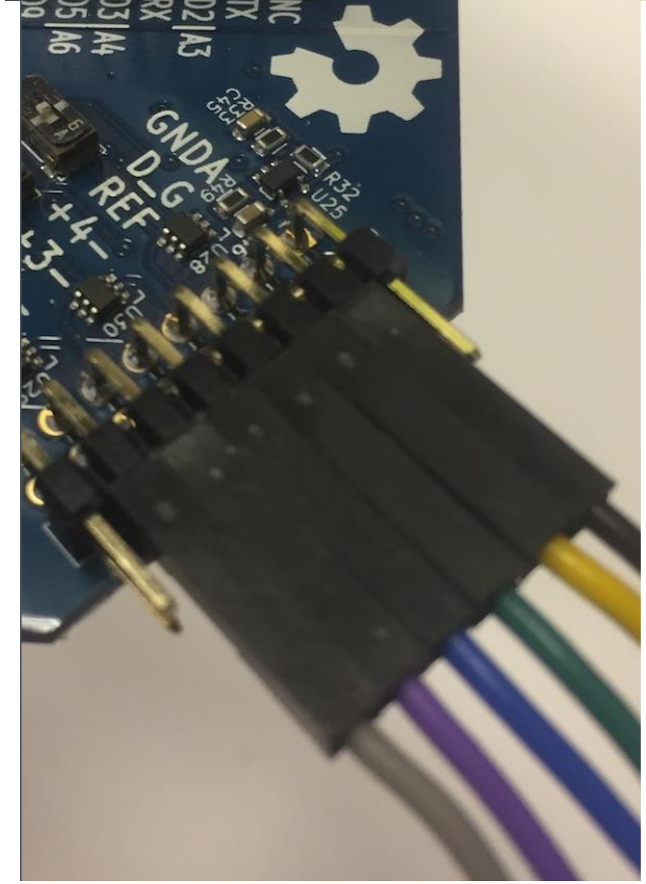


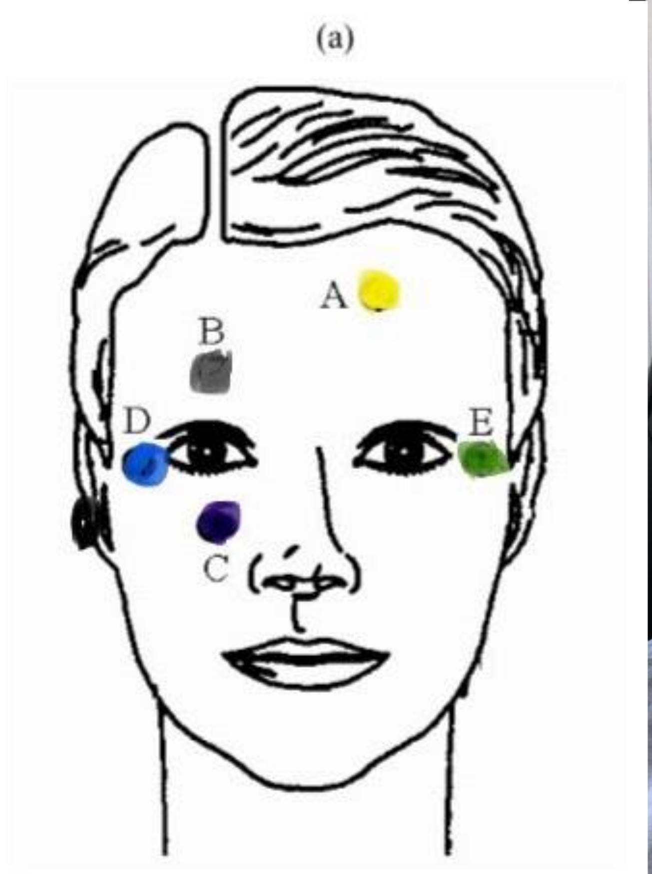
Down
ch1: Up
ch2: Down
ch3: Right
ch4: Left
Amplitude (μV)
200
100
0
-100
-200
-300
-400
-500
-600
26.0
26.2
26.4
26.6
26.8
Time (s)



Left
ch1: Up
ch2: Down
ch3: Right
ch4: Left
Amplitude (μV)
400
300
200
100
0
-100
-200
-300
23.0
23.1
23.2
23.3
23.4
23.5
23.6
23.7
23.8
23.9
Time (s)







Dataset creation

Using OpenBCI Ganglion Board (4-channels)

<u>Tests done:</u>

25x Center --> Up --> Center movement
25x Center --> Down --> Center movements
25x Center --> Left --> Center movements
50x Center --> Right --> Center movements
1x Long Blinking Test

Main problems...

We wanted to:
-Have multiple sources of data
-From each have multiple recordings
-Do data augmentation

Lack of data

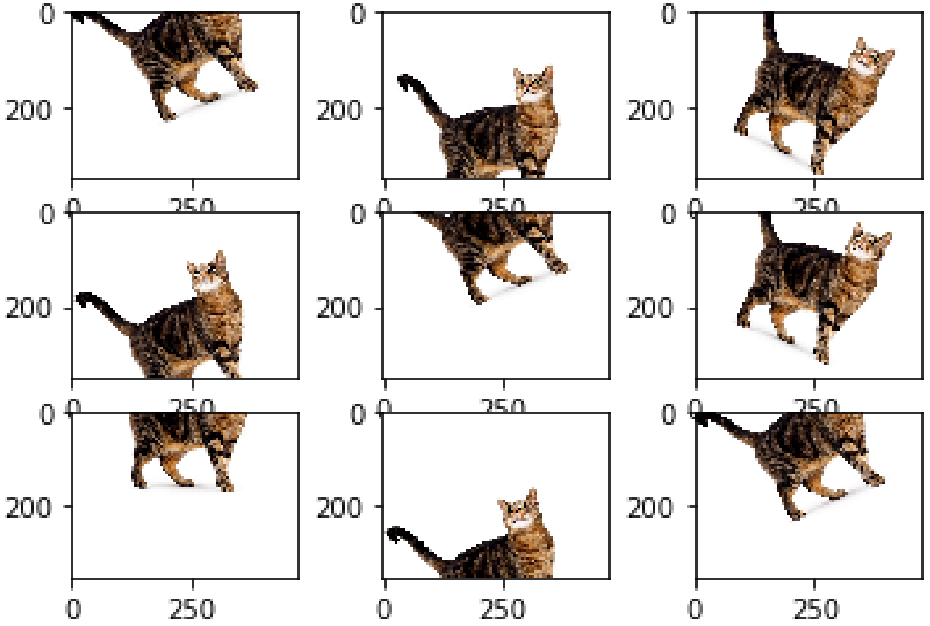
Learning

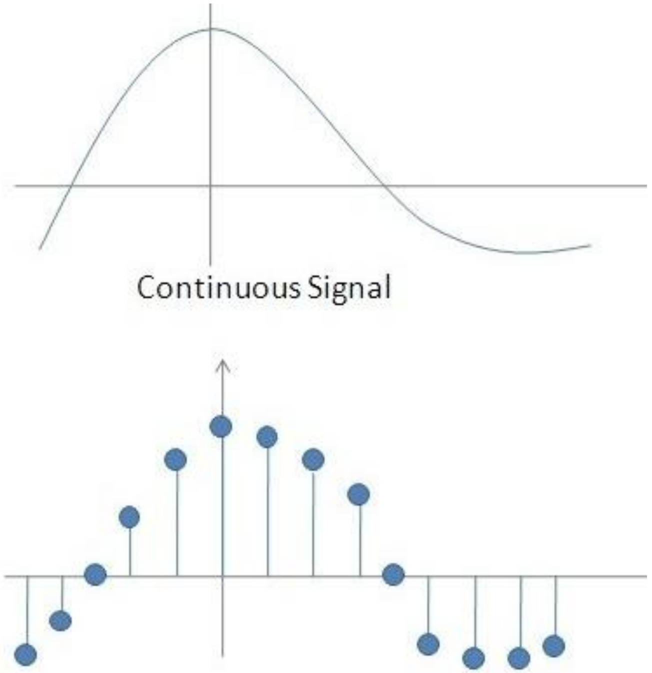
01 Data Augmentation

02 Machine Learning

03 Deep Learning

Data Augmentation

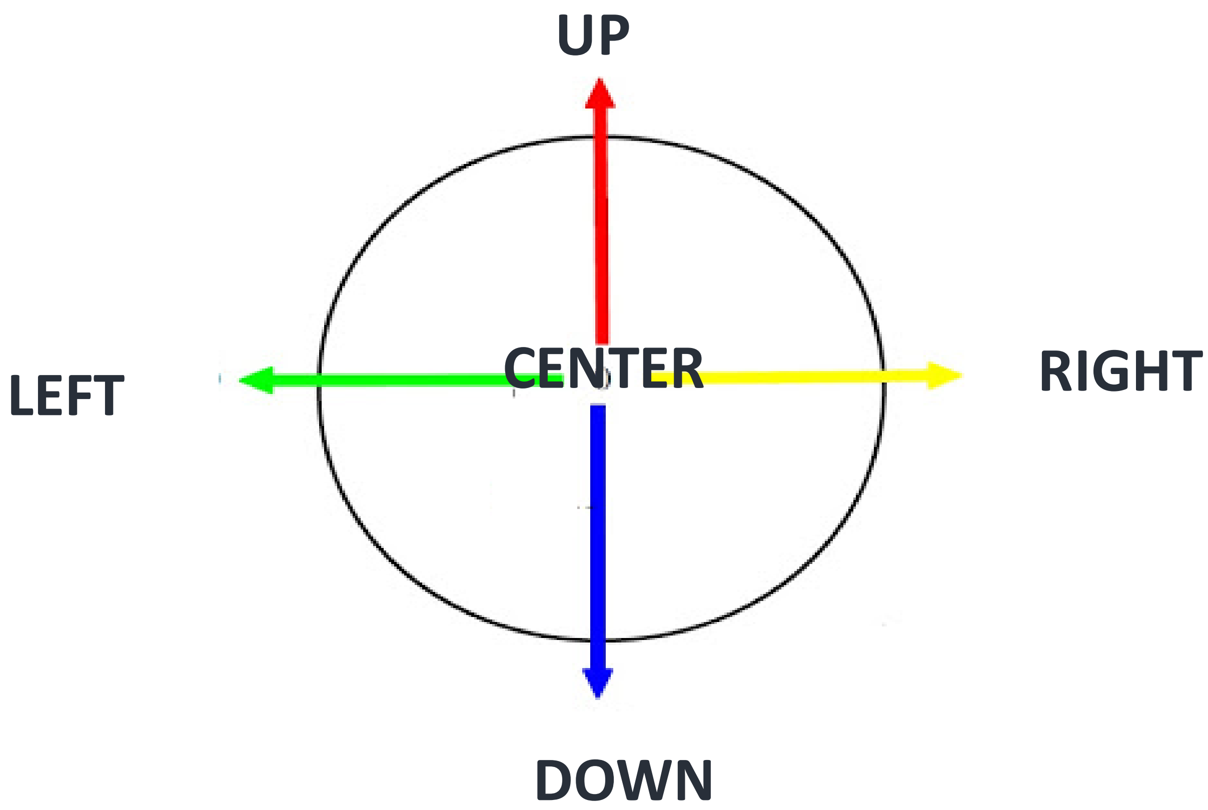


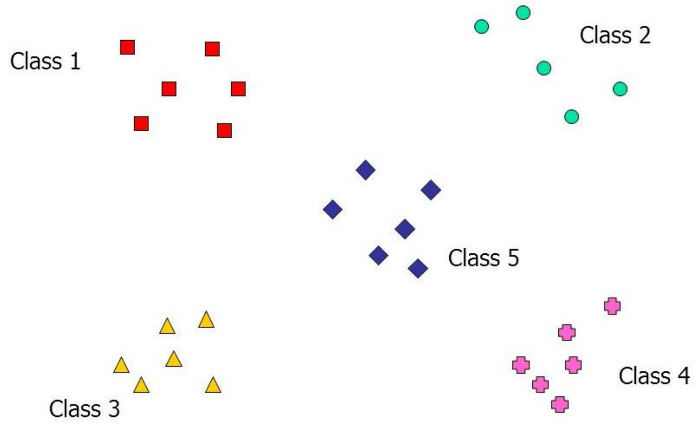


Concept:
- Choose randomly P% of time points;
- Stretch them with[Min-Max]s

Helps:
- Less biased model;
- Generated Noise;

Machine Learning: SVM





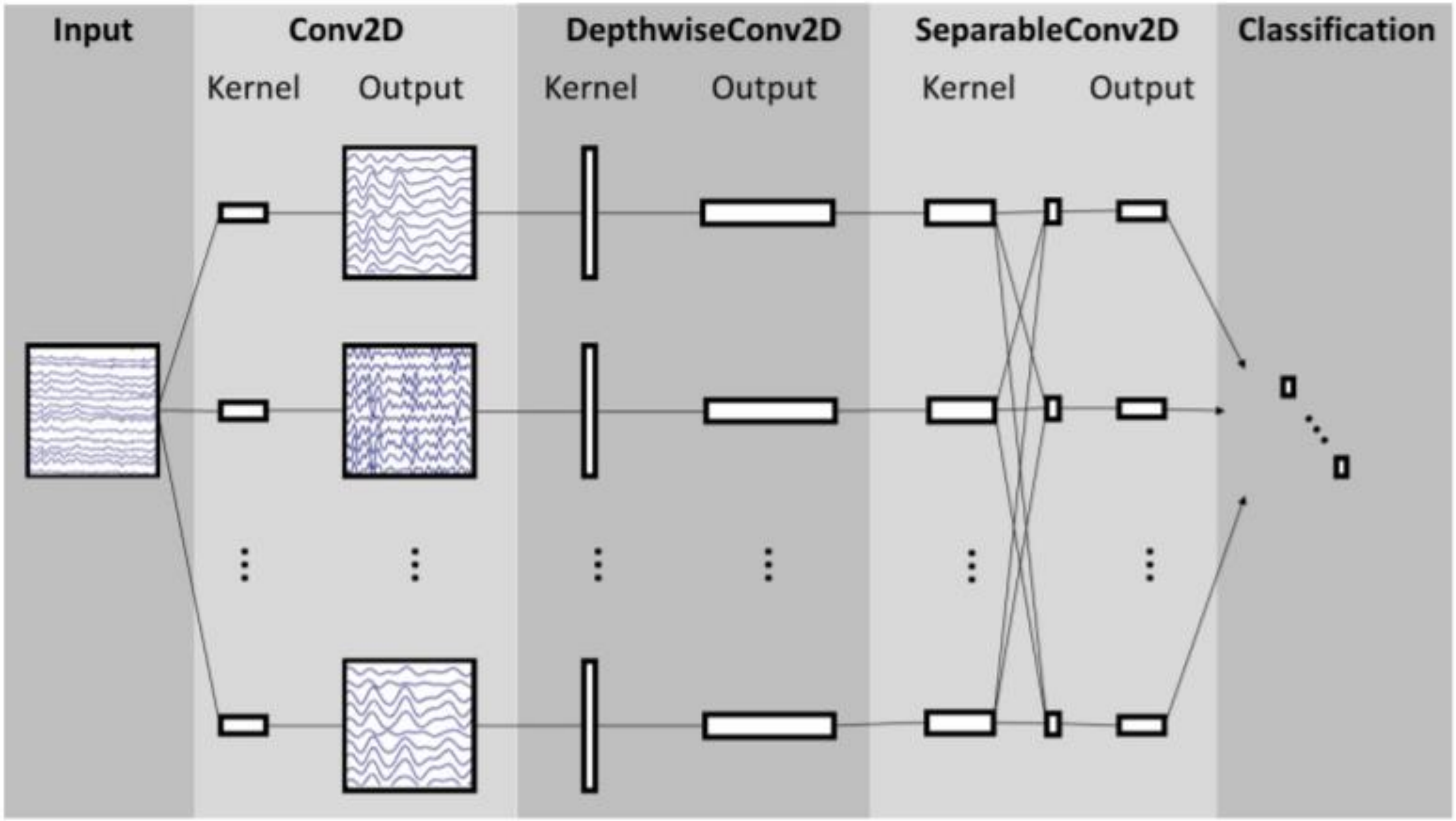
Feature Vector:
- Mean, Max , Min , Var;
- 5 classes

Result:
- **43% accuracy**

Deep Learning: EEGNet

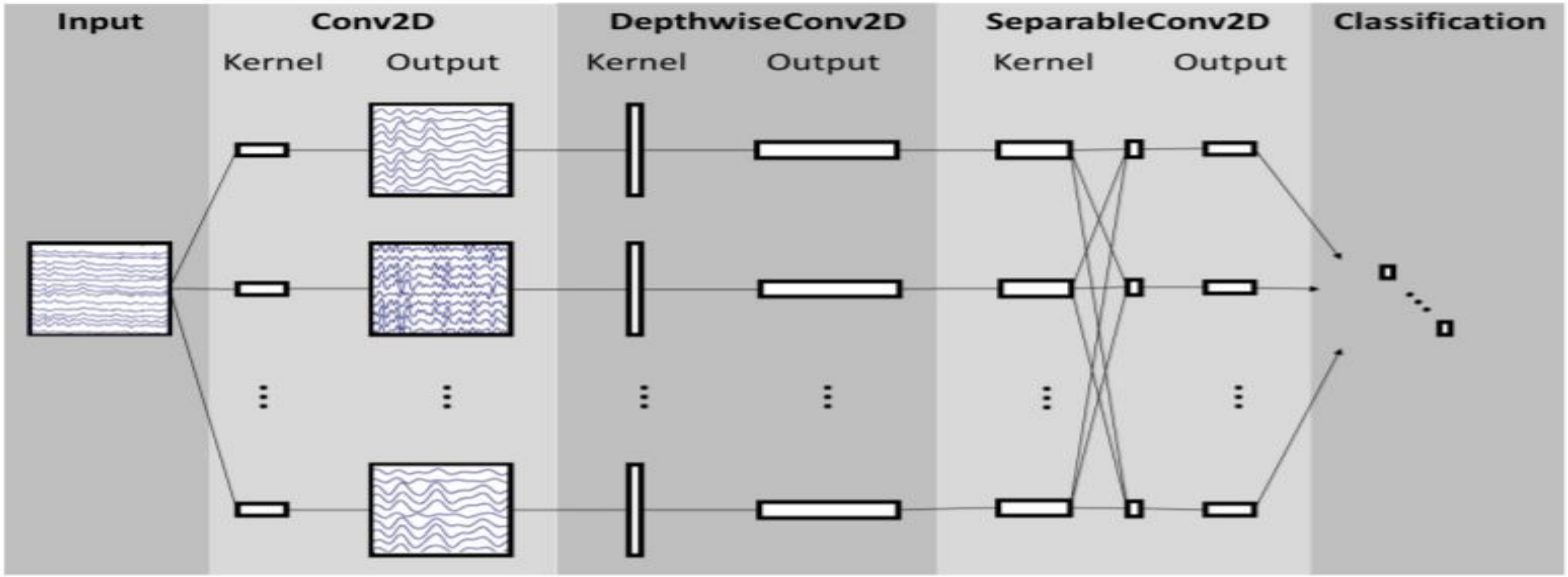
EEGNet: A Compact Convolutional Neural Network for EEG-based Brain-Computer Interfaces

Vernon J. Lawhern1, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon,Chou P. Hung, and Brent J. Lance



- ***Fully supervised***
- ***Few data***
- ***Compact architecture***
- ***Engineering***

Deep Learning: EEGNet



Training and Engineering:

- Number of channels 4
- Frequency 100Hz

Result:

- **50,1% accuracy**

THE DRONE

- Communications
- DroneApp connected with the Drone API
- Real time show
- Flask app for simple navigation controller
- Router port forwarding to be able to work remotely

Progress

Dataset creation: Collect EEG signal with 4 golden cup sensors

Done

01

02

Signal sending: in real-time from a Board to PC and/or Cloud (Amazon Web Services)

03

Signal/data pre processing: Data cleaning, noise

Done

04

Classification:

Data augmentaion **Done**

SVM **Done**

Deep learning **On-going**

API

05

Drone