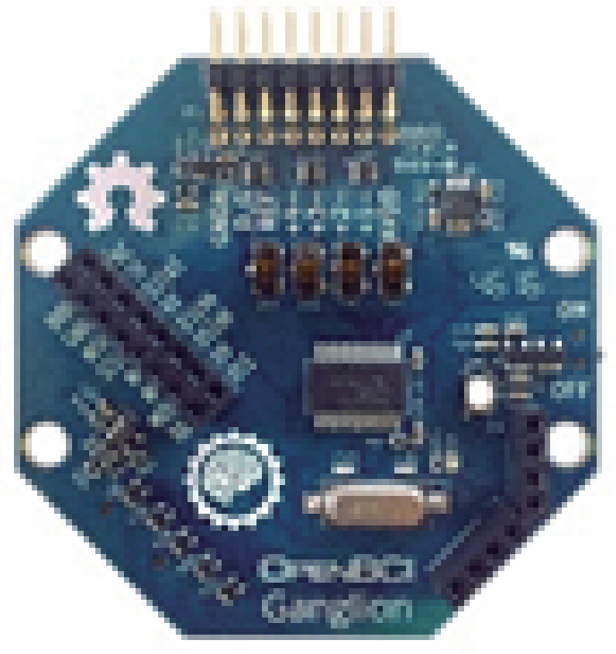
Enhancement to MindDesktop
Improvement to a General Purpose Brain Computer Interface

Students: Tang Wang Hin Kenneth, Lau Chun Lam Frederick   Supervisor: Prof. Anthony Tam

This project proposes methods for improvement of MindDesktop, a general purpose Brain Computer Interface (BCI) aiming of controlling the computer with Microsoft Windows as the running OS. The project composed of two major parts: Graphical User Interface (GUI) and the mental command training with Deep Neural Network. There is also a Connector Component to seperate the GUI module from the mental command predictions to enhance modularity.

Information

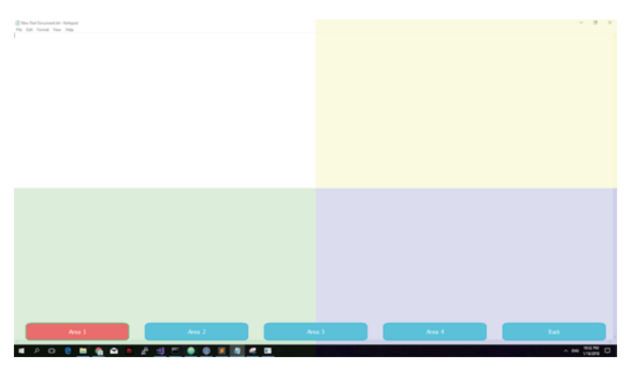
- Programming Language: Python
- BCI Device: OpenBCI Ganglion (4 channels)
- Platform: Windows 10
- Goal:
  After training, user can use 2 Mental Commands, “Next” and “Confirm”, to enable keyboard input and mouse clicking on computers that run Windows 10.

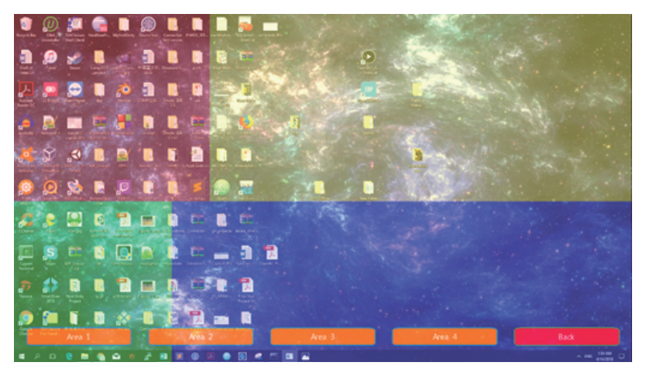


GUI

Improvements:

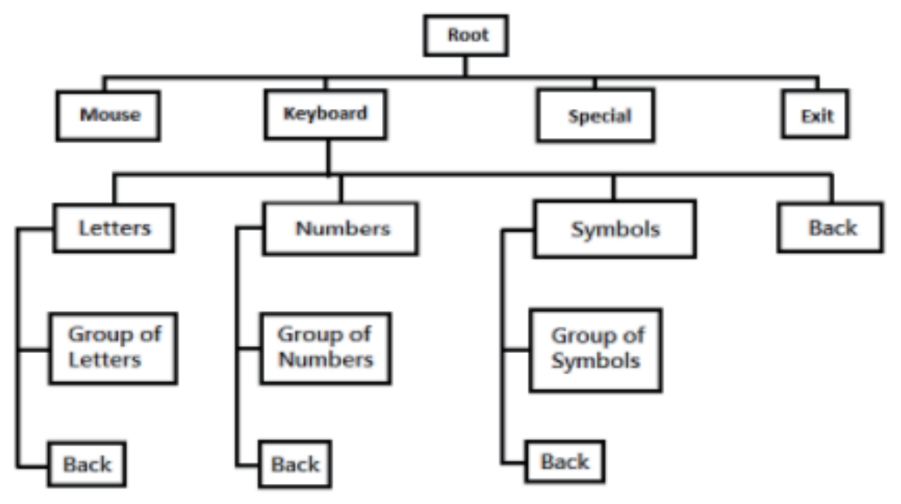
- Three (Next, Confirm, Back) ➤ Two (Next, Confirm) commands for control
- Reduce steps for screen division to get to the target area on the screen for emulating mouse clicks, by dividing the selected area into unequal areas through the division algorithm base on the number of elements. In MindDesktop, it simply divides the selected areas into four equal areas





Features:

- Self-developed GUI framework with various controllable styles similar to css
  Basic Components: Panel, Button, Text
- Consists of a hierarchy tree that will go back to

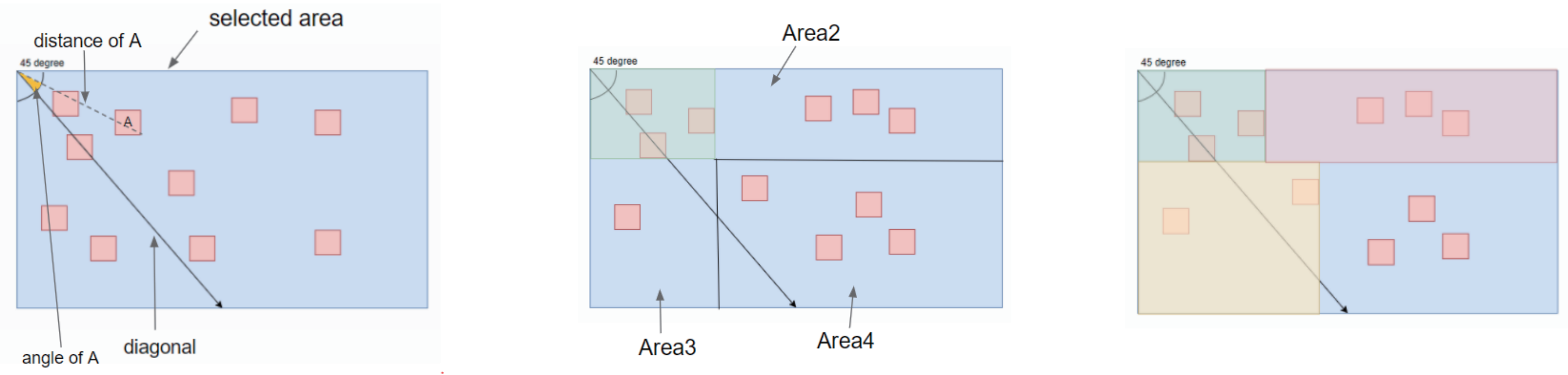


Mental Command Training

- Keras + Tensorflow for rapid Neural Network development
- Simple Sequential Model
- Four identical models for four channels since each channel has distinct signals
- FFT features of EEG signals as training data
- Multiclass classification: Neutral, Next, Confirm
- Unique training profiles for each user (containing the weights of trained model)

Selection Division Algorithm

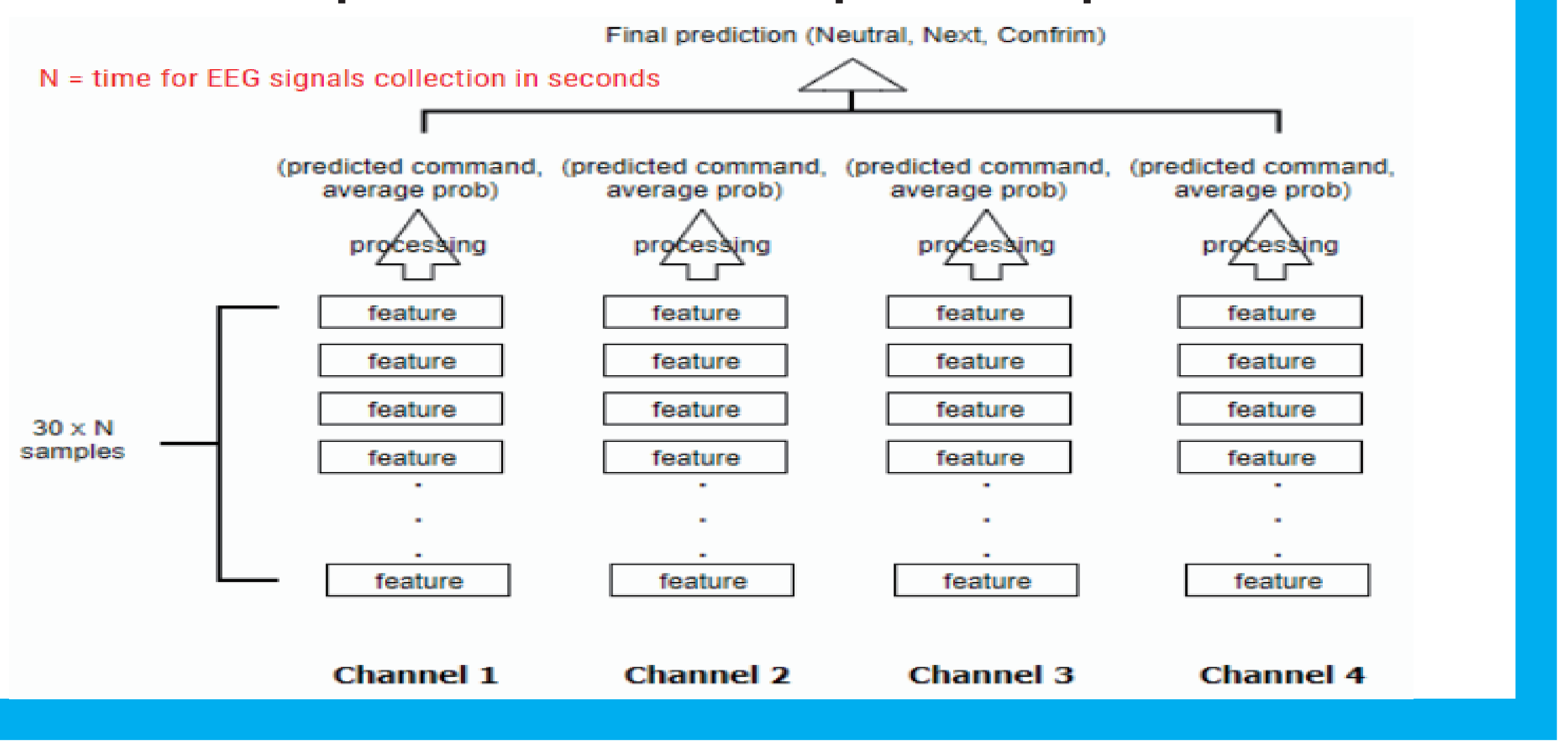
1. User Windows UI Automation to inspect the coordinates and size of GUI elements on Desktop
2. Eliminates all the elements covered by the foreground windows or those inside the minimized windows
3. Scoring each visible element according to the distance and angle from top-left corner of the selected area
4. Sort the elements by scores and choose the first quarter of the elements to be in subdivided area 1
5. Divide the remainging area into 3 rectangular areas so that the remaining elements are distributed as even as possible



Prediction Algorithm

The current approach predict the mental command per each second

1. Get the FFT features from the EEG signals in a sampling rate of 30/s
2. After collecting each second of FFT features, feed into the NN model and conclude a final predicted command in these 30 predictions by voting
3. Four results get from step 2 since there are 4 EEG channels with distinct signals
4. Conclude the four results by voting and also compare the prediction probabilities to get a single final predicted command
5. Three consecutive “Next” or “Confirm” predictions from step 4 are required to execute the command


Final prediction (Neutral, Next, Confrim)
N = time for EEG signals collection in seconds
(predicted command, average prob)
(predicted command, average prob)
(predicted command, average prob)
(predicted command, average prob)
processing
processing
processing
processing
feature
30 x N samples
Channel 1
Channel 2
Channel 3
Channel 4