

Manual de usuario de los programas de clasificación de etapas del sueño

Departamento de Electrónica, Mecatrónica y Biomédica

Clasificación de señales usando base de datos

Programas necesarios para su funcionamiento:

- Extracción_de_Caracteristicas.m
- Min_Max_Distance.m
- Zero_Crossing.m
- SVM.m
- Red_Neural.m
- KNN.m

Ejecución del programa:

Una vez teniendo todos los archivos en la misma ubicación se realiza el siguiente procedimiento para poder ejecutar los clasificadores.

1. Como primer paso si aun no se tienen datos de entrenamiento es necesario descargarlos utilizando la herramienta de **SleepEDFToolBox** disponible en el siguiente enlace junto con su respectivo instructivo para la instalación: <https://github.com/anasimtiz/sleep-edfx-toolbox>
2. Ya teniendo los datos de prueba se pasa a utilizar el primer programa el cual es el de extracción de características en el cual se cargan los datos obtenidos en el paso anterior en la sección comentada del inicio y luego se ejecuta y así se generan los vectores de características para poder implementar dentro de los clasificadores.
3. Por ultimo se ejecuta cualquiera de los tres clasificadores que se desee utilizar y automáticamente se generaran las matrices de confusión y el porcentaje de error mediante validación cruzada.

Clasificación de señales en tiempo real

Programas necesarios para su funcionamiento:

- Extracción_de_Caracteristicas.m
- Min_Max_Distance.m
- Zero_Crossing.m
- SVM.m
- Red_Neuronal.m
- KNN.m
- Live_Data.m
- Openbci_matlab.py

Ejecución del programa:

Una vez teniendo todos los archivos en la misma ubicación se realiza el siguiente procedimiento para poder ejecutar los clasificadores.

1. Para poder ejecutar este programa es necesario tener conectado el Electro – Cap (Si no se conoce como utilizarlo leer el manual de usuario disponible en el repositorio de GitHub: <https://github.com/larivera-UVG/Etapas-de-sueno-pulsos-binaurales/tree/master/Reconocimiento%20de%20Patrones/Documentacion>)
2. Ya teniendo en funcionamiento el Electro – Cap se pasa a ejecutar el código llamado Openbci_matlab.py el cual comenzará a obtener los datos del puerto serial y lo transmitirá a Matlab para poder ser procesado.
3. Como ultimo paso se ejecuta el archivo Live_Data.m el cual ejecuta el entrenamiento del clasificador seleccionado, obtendrá los datos de Matlab y realizará la separación de épocas, extrayendo las diferentes características y clasificándolas.