

## DETECCIÓN DE LA ONDA P300 CON RED NEURONAL CONVOLUCIONAL IMPLEMENTADA EN SISTEMA EMBEBIDO PORTABLE

José M. Macías Macías, Juan A. Ramírez Quintana, Alma D. Corral Sáenz, Mario I. Chacón Murguía  
Tecnológico Nacional de México/I.T. Chihuahua  
Laboratorio de Percepción Visual con Aplicaciones en Robótica  
División de Estudios de Posgrado e Investigación  
Ave. Tecnológico No. 2909, 31310, Chihuahua, Chih., México, Tel. (614)2026511  
jose08m3@gmail.com, jarq1910@gmail.com, mchacon@ieee.org, mcie@itchihuahua.edu.mx

### RESUMEN.

Los potenciales relacionados a eventos (ERP) son cambios en la actividad cerebral cuando se expone un sujeto a un estímulo aleatorio. El ERP más popular es la onda P300 ya que permite desarrollar aplicaciones de comunicación para personas con discapacidad motriz y del habla. No obstante, los métodos de análisis de P300 con buenos desempeños reportados en la literatura son muy complejos y no se diseñan para sistemas embebidos que permitan esquemas de computación ubicua enfocada al paciente. En el presente trabajo se propone un método de análisis de la onda P300 basado en redes neuronales convolucionales diseñado para una Raspberry Pi que funciona como un sistema embebido portable. Dicho sistema permite el análisis de P300 con un procesamiento ubicuo en vez de utilizar una computadora. Se realizaron 48 sesiones de extracción de muestras y el resultado de desempeño es de 93.8% para validación y 100% para prueba.

**Palabras Clave:** Interfaz cerebro computadora, OpenBCI, Raspberry Pi 3, P300, redes neuronales convolucionales.

### ABSTRACT.

Event related potentials ERP are changes in brain activity when a subject is exposed to a random stimulus. The most popular ERP is the P300 wave because it allows the development of communication applications for people with motor and language disabilities. However, the reported methods in literature about P300 analysis with good performance are very complex and are not designed to work in embedded systems that allows ubiquitous computing schemes. Therefore, in present work we propose a P300 wave analysis method based on convolutional neural network designed for Raspberry Pi that works as a portable embedded system. This system allows a P300 wave analysis based on ubiquitous processing instead of standard computer. For the experiments, 48 sample extraction sessions were made and the performance result are 93.8% for training and 100% for testing.

**Keywords:** Brain computer interface, OpenBCI, Raspberry Pi, P300, convolutional neural networks.

### 1. INTRODUCCIÓN

Los potenciales evocados (EP) son exploraciones del sistema nervioso utilizados para evaluar alguna función sensorial (Acústica, visual, somatosensorial o cognitiva) por medio de respuestas provocadas frente a un estímulo específico, conocido y normalizado [1]. Estos potenciales se pueden obtener de

señales electroencefalográficas (EEG) sometiendo a un sujeto de pruebas a una serie de estímulos sensoriales que se presentan en forma aleatoria. Para ello, el sujeto debe cumplir un perfil clínico basado en el cumplimiento de condiciones generales tales como no haber consumido fármacos o no contar con historial de enfermedades neurológicas [2].

Los EP pueden ser divididos en potenciales evocados de estado estable (SSEP) y potenciales relacionados a eventos (ERP) [3]. Entre ellos, el ERP más estudiado en la literatura es la onda P300 por su capacidad para generar mensajes de sujetos de pruebas a partir de señales EEG por diversos canales sensoriales. La onda P300 es un ERP que consiste en una deflexión positiva que se presenta en la actividad cerebral 300 ms después de la exposición a un estímulo. La latencia de esta onda puede variar de los 250 ms a los 750 ms [4]. En la literatura, la detección de la onda P300 tiene diversas aplicaciones relacionadas a trastornos psiquiátricos [5], delirios [6], sistemas de control [7] y análisis de los estados de vigilancia [8].

En el procesamiento de P300 las redes neuronales convolucionales (CNN) han tenido resultados con buenos desempeños. Por ejemplo, en [9] se propuso un análisis con los 64 electrodos del sistema internacional 10/10. En este caso, la entrada a la CNN es una representación de las señales EEG en 3D y la red realiza un análisis con diferentes ventanas de tiempo. Los resultados tienen desempeños por debajo del 80%. En [10] se propuso otro método basado en CNN utilizando la base de datos BCI Competition Data set II y la misma configuración de electrodos que [9], pero se obtuvo un desempeño de 95.5%. En [11] se generó una nueva base de datos basada en el dispositivo OpenBCI® y una CNN que procesa señales EEG con P300. El desempeño fue de 92% pero utilizando únicamente cinco electrodos. Adicional a estos trabajos, existen otros métodos que también analizan la onda P300, pero utilizan diferentes cantidades de electrodos que varían de 8 a 64.

Sin embargo, la cantidad de electrodos utilizados para el análisis, así como los métodos resultantes han generado algoritmos muy complejos que solo se pueden utilizar en sistemas de computación genéricos. Esto causa que los sistemas de análisis de la onda P300 sean poco factibles, ya que en una aplicación real es necesario que un sujeto se comunique en cualquier parte de su entorno sin necesidad de llevar consigo

una computadora de escritorio o laptop. Por lo tanto, el propósito de este trabajo es elaborar un método con un modelo de redes neuronales convolucionales (CNN) para la detección de la onda P300 en un sistema embebido portable que permita un esquema de computación ubicua centrado en la persona (es decir, el procesamiento se realice con un dispositivo que la persona pueda portar fácilmente). La adquisición de señales se realiza con el dispositivo de adquisición OpenBCI® y la CNN se implementa en una tarjeta de desarrollo Raspberry Pi 3 modelo B. Este esquema brinda la posibilidad de crear un sistema BCI que procesa la onda P300 con las propiedades de ser embebido, escalable y portable.

El artículo se estructura de la siguiente manera: en la Sección 2 se describe el esquema general del sistema BCI embebido. La Sección 3 describe una base de datos desarrollada con el sistema embebido. La Sección 4 muestra el preprocesamiento de las señales EEG. La Sección 5 describe la forma de agrupar señales y en la Sección 6 la etapa de clasificación dada por la CNN. La Sección 7 presenta los resultados y finalmente en la Sección 8 las conclusiones y trabajo futuro.

## 2. SISTEMA BCI EMBEBIDO

### 2.1. Sistema interfaz cerebro computadora.

La tecnología relacionada con la medición de la actividad cerebral en una computadora se le denomina interfaz cerebro computadora (BCI por sus siglas en inglés) [3]. Como se muestra en la Figura 1, el sistema BCI propuesto se compone de las etapas de adquisición de señales, procesamiento de señales y una aplicación.

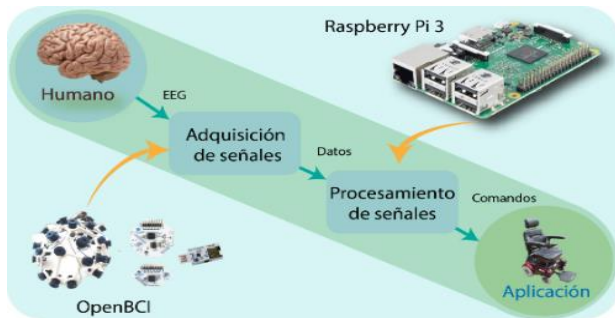


Figura 1 Sistema BCI propuesto.

La etapa de adquisición utiliza el dispositivo OpenBCI® como dispositivo de adquisición EEG para la obtención de señales cerebrales. La etapa de procesamiento de señales se implementa en una Raspberry Pi 3 modelo B, su utilidad es analizar las señales EEG mediante una CNN para detectar la presencia o no de la onda P300. La aplicación del sistema toma como entrada la detección del P300 para mover una silla de ruedas, cambiar los estados de posición de una camilla, encendido o apagado de aparatos eléctricos, pasar la información a Internet, etc.

### 2.2. Adquisición de señales.

La adquisición de las señales EEG se realizó mediante el dispositivo de código libre OpenBCI®, el cual cuenta con electrodos de copa secos y una tarjeta de adquisición[12]. Para el desarrollo del proyecto se utilizaron 3 electrodos de copa, situados según el sistema internacional 10-20 en las posiciones Oz, A1 (referencia) y Fpz (tierra) como se muestra en la Fig. 2, utilizados para obtener una señal de un canal del electrodo Oz. La tarjeta de adquisición trabaja con una frecuencia de muestreo de 250 Hz en la toma de muestras.

Para evocar un ERP es necesaria la presentación aleatoria de estímulos que en el caso de este proyecto serán figuras visuales configuradas como recuadros que cambian de color aleatoriamente. Para ello, se realizó una interfaz visual por medio del paquete de interfaces graficas Tkinter en Python [13]. Como se ve en la Fig. 3a, la interfaz es un escenario con el diseño típico de una terapia virtual con 4 recuadros en la parte inferior que van de 1 a 4. Estos recuadros serán utilizados como estímulos que cambian de color de negro a blanco como se ve en el estímulo 2 en la Fig. 3a y 3b. Cuando el recuadro está en blanco, se dice que el estímulo se intensificó.

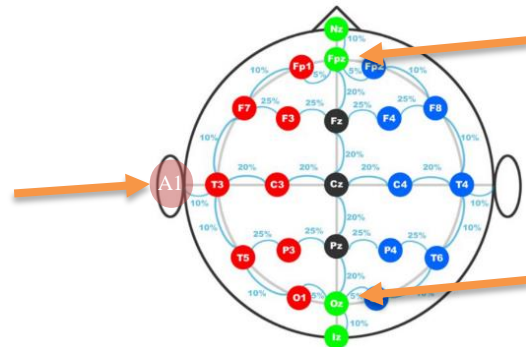


Fig. 2 Posicionamiento de electrodos de acuerdo al sistema internacional 10/20.

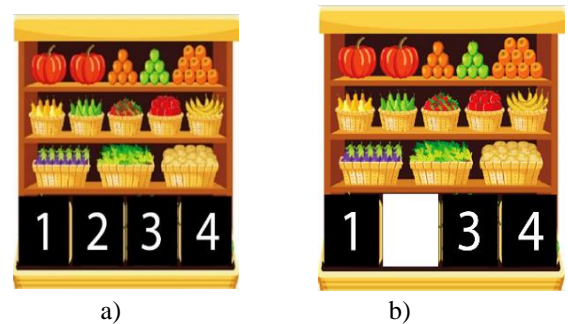


Figura 3 Interfaz visual para la toma de muestras a) Ningún estímulo intensificado. b) Estímulo 2 intensificado.

Se conoce que entre más inesperado sea el estímulo visual, se evoca un mayor potencial en las neuronas del cerebro [14]. Por esta razón, en la etapa de adquisición de señales se realizan intensificaciones aleatorias de un recuadro a la vez, tal como se muestra en la Figura 3b. Para establecer los tiempos de

cambios de color en los recuadros, se utiliza el diagrama de tiempos de la base de datos BCI Competition III dataset II [15]. Dicho diagrama se puede observar en la Figura 4 donde el tiempo de intensificación (uno de los cuatro recuadros es color blanco)  $T_H$  es de 100 ms, mientras que el tiempo donde todos los recuadros permanecen en negro es de  $T_L$  de 72 ms. Para asegurar que no se repitan más de 3 intensificaciones seguidas en el mismo recuadro, se generan las intensificaciones en cada recuadro dentro del grupo de los 4 totales, 15 veces el mismo proceso.

Para extraer la onda P300 de un sujeto, se debe realizar una serie de pruebas que consisten en que el sujeto de prueba mantenga la mirada fija en uno de los 4 recuadros como se muestra en la Figura 4, mientras se realiza una toma de 4550 muestras (18.2 segundos dada la frecuencia de muestreo). Cuando se intensifique el recuadro que el sujeto está observando, se debe registrar una deflexión positiva en la señal EEG del canal Oz 300 ms después que se generó la intensificación.

En las sesiones que se hacen para cada sujeto, la toma de muestras se divide en tres etapas: 1000 muestras para estabilizar el filtro de preprocesamiento que se verá en la sección 4, 3300 muestras en las que se presentan las intensificaciones en los recuadros (60 cambios en total) y finalmente 250 muestras que equivalen a 600 ms con el fin de tener una ventana para grabar el estímulo P300 de la última intensificación presentada (en caso de ser evocado).

Si cada cuadro contiene un mensaje (por ejemplo, la Figura 3 tiene números), entonces se podrá conocer con la onda P300 lo que un sujeto trata de decir con solo detectar cual es el recuadro al que se presta atención.

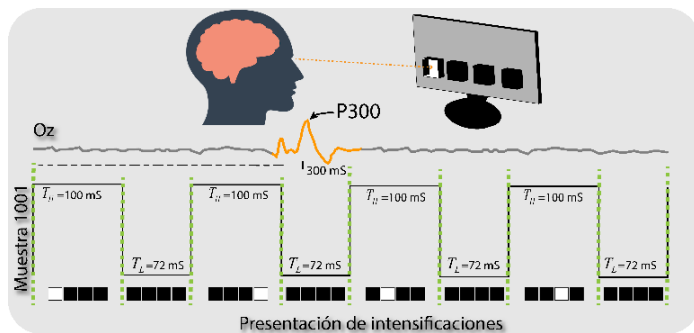


Figura 4 Tiempo de intensificación y su efecto en la señal EEG.

Una vez recolectadas las señales de cada estímulo, se guardan en un archivo con extensión .csv, los cuales forman una señal de la forma:

$$s(n) \quad , \quad n = 1, 2, \dots, 4550 \quad (1)$$

Donde  $s(n)$  corresponde a la señal del electrodo Oz y  $n$  son las muestras tomadas durante la exposición al estímulo visual de los recuadros que corresponde a 18.2 segundos dada la frecuencia de muestreo de 250 Hz.

### 3. BASE DE DATOS

Para realizar el diseño de algoritmos, se creó una base de datos compuesta de muestras obtenidas bajo un protocolo que inicia con un cuestionario creado para documentar y validar que el sujeto de pruebas cumpla el perfil clínico para estas pruebas [2]. Una vez validados los potenciales evocados, se colocan los electrodos Oz, Fpz y A1 del OpenBCI® con gel Ten20. Posteriormente, se le dan indicaciones al sujeto de prueba de permanecer quieto en una silla cómoda fijando la mirada solamente en uno de los recuadros (estímulo) de la interfaz durante la sesión de toma de muestras.

Con este protocolo, se realizaron 48 sesiones de toma de muestras, las cuales generaron 48 señales de la forma  $s(n)$ . Tal como se muestra en la Figura 5, las 48 sesiones fueron grabadas por estímulo, en el orden de estímulo 1 a estímulo 4. De esta manera se compone la base de datos de apilar el conjunto de señales de la forma  $s(n)$  por cada sesión de toma de muestras de cada estímulo. Para la sección 6 de clasificación, se destinan 28 señales para entrenamiento y 20 para validación, todas ellas con su respectivo preprocesamiento y agrupación de las secciones 4 y 5.



Figura 5 Composición de la base de datos.

### 4. PREPROCESAMIENTO

El preprocesamiento de las señales consiste en un filtrado pasa bandas, que por medio de experimentación se optó por utilizar un filtro Butterworth pasabandas de orden 3 con frecuencias de corte de 1 a 15 Hz. De esta manera se obtiene la señal:

$$H(w) : s \rightarrow sf \quad (2)$$

A partir del filtro:

$$|H(w)|^2 = \frac{1}{1 + (w/w_c)^{2N}} \quad (3)$$

Donde  $N$  es el orden el filtro,  $w_c$  es la frecuencia de corte y  $H(w)$  es la respuesta a la frecuencia del filtro. El filtro se aplica en la señal debido a que la información importante donde se genera la onda P300 se encuentra dentro de este rango de frecuencia [16]. El filtro presenta distorsión de fase en las primeras muestras, por lo que el análisis a partir del filtrado se realiza en el intervalo  $1001 < n \leq 4550$ , con el fin de eliminar las muestras a las que afecta la distorsión de fase del filtro.

Una vez filtrada la señal se realiza una normalización, con la finalidad de que todas las señales se encuentren en el mismo rango de valores sin importar el offset de cada señal  $s(n)$ . La normalización de la señal  $sf(n)$  se define por la técnica Z-Score, donde se busca tener media cero y varianza unitaria, como se muestra en la siguiente ecuación:

$$sn(n) = \left( sf(n) - \mu_{sf} \right) / \sigma_{sf} \quad (4)$$

donde  $sn(n)$  es la señal normalizada,  $\mu_{sf}$  la media de la señal filtrada y  $\sigma_{sf}$  la desviación estándar.

## 5. AGRUPAMIENTO DE SEÑALES POR INTENSIFICACIONES DE RECUADROS

Para esta etapa se realiza un agrupamiento que consiste en tomar 15 fragmentos de cada señal  $sn(n)$  (1 por cada intensificación) y promediarlas según el inicio de la intensificación de la Figura 4. Para ello, cada señal  $sn(n)$  se descompone en fragmentos de 400 ms, los cuales van desde los 100 ms a 500 ms después de cada intensificación. Se consideró esta ventana porque el potencial evocado se presenta 300 ms después de la presentación del estímulo [16]. Dada la frecuencia de muestreo del dispositivo de adquisición, cada señal fragmento se compone de 100 muestras. De esta manera la señal  $P_{ij}(n)$  son cada uno de los fragmentos, siendo  $i=1, \dots, 15$  el número de fragmento y  $j=1, \dots, 4$  el número de estímulo al que pertenece. El agrupamiento se realiza de la siguiente manera:

$$X_{Ejl}(n) = \frac{1}{15} \sum_{i=0}^{15} P_{ij}(n) \quad (5)$$

Dado que el agrupamiento se realiza por cada señal  $s(n)$ , para las  $l=1, \dots, 48$  señales que conforman la base de datos, se realiza la sumatoria para cada fragmento  $i$  de cada estímulo  $j$ . Para luego obtener la señal  $X_{Ejl}(n)$ , la cual se conforma por los promedios de los fragmentos luego de la presentación de cada estímulo.

## 6. CLASIFICACIÓN

La etapa de clasificación se basa en la implementación de una red neuronal convolucional, cuyo modelo es una modificación al modelo propuesto de la red CNN300 en [11]. No obstante, para adaptar este modelo en la tarjeta Raspberry, se realizaron dos modificaciones:

- En el presente trabajo se utiliza solamente un electrodo en la posición Oz. Por está, razón la entrada a la red será una señal con longitud de 1x100.
- Se eliminó una capa convolucional y una unidad rectificadora lineal, pues por medio de experimentación se obtuvieron los mismos resultados con una capa convolucional que con dos. Esto se debe a que la composición de la señal es de un solo canal y son menos los mapas de características que necesita la red para clasificar.

El modelo CNN300 se adaptó por medio de Python 3.4 para ser implementado en la Raspberry Pi 3 modelo B. Los pesos y el sesgo de la red fueron modelados con un entrenamiento en una computadora Dell Precision T7500 8 GB de RAM, procesador Intel Xeon E5620 de 2.4 GHz y GPU Nvidia GTX 970.

Este entrenamiento se llevó a cabo mediante el algoritmo gradiente estocástico descendente con momento utilizado en [11], el cual minimiza el error mediante la actualización de pesos en los filtros  $F_k$  de convolución. Se destinaron  $l=1, \dots, 28$  señales de estímulos  $s(n)$  para la etapa de entrenamiento, y  $l=1, \dots, 20$  para validación de la base de datos realizada en este trabajo. Los objetivos utilizados para el entrenamiento se obtienen de la base de datos, dado que se conoce el orden del estímulo en el que el sujeto fijó la mirada.

La clasificación utilizando la red neuronal convolucional se compone de dos secciones. En la primera sección la red neuronal predice el grado de existencia  $\alpha_{Ejl}$  y no existencia  $\beta_{Ejl}$  de la onda P300 en cada señal  $j$  dentro de  $X_{Ejl}(n)$ . Con base en  $\alpha_{Ejl}$ , la segunda sección busca encontrar en cuál señal del estímulo  $j$  se presenta con mayor probabilidad la onda P300.

### 6.1. Arquitectura de la CNN para detectar la existencia $\alpha_{Ejl}$ y no existencia $\beta_{Ejl}$ de la onda P300

La CNN propuesta se puede observar en la Figura 6. La red se compone de dos subetapas. Una es extracción de características que consta la capa de convolución y la capa de rectificación lineal (ReLU). La otra subetapa es la clasificación y se compone por una capa totalmente conectada (Fully Connect) y la última capa es una exponencial normalizada denominada soft max [17].

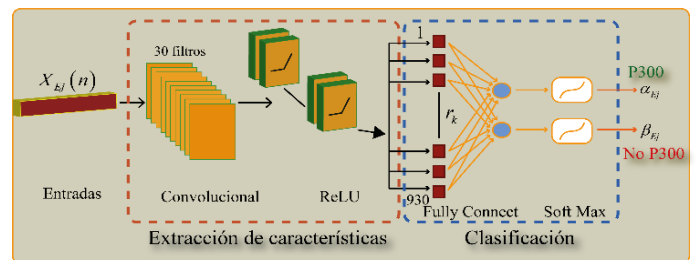


Figura 6 Topología de la red neuronal utilizada para un solo canal.

La capa convolucional de esta etapa está formada por un número de neuronas  $c_k$ , que mapean cada promedio de la señal de entrada  $X_{Ejl}(n)$  de longitud  $q=1, \dots, 100$  en conjuntos de características dadas por la misma capa convolucional. Esto se realiza mediante 30 filtros lineales de longitud  $(k=1, \dots, 30)$  de  $1 \times 70$ , que utilizan conjuntos de pesos  $w_k$  (máscaras de convolución). Lo anterior se define como [18]:

$$F_k(q) = w_k(q) + B_k \quad (6)$$

$$k = 1, \dots, 30$$



Donde cada filtro  $F_k$  se compone de los pesos  $w_k$  y su polarización  $B_k$ . El valor de estos pesos se obtiene durante la etapa de entrenamiento de la red. De esta manera, las salidas de las neuronas  $c_k(p)$  para cada filtro  $k$  se definen como:

$$c_k(p) = F_k(q) * X_{Ejl} \quad (7)$$

$$= \sum_{q=1}^{100} w_k(q) X_{Ejl}(p-q) + B_k$$

Es decir, se realiza la convolución entre los filtros  $F_k(q)$  y cada promedio de  $X_{Ejl}$ . Cada neurona  $c_k(p)$  tendrá una composición de  $1 \times 31$  lo cual es resultado de la convolución. La siguiente capa de la extracción de características, está dada por una función de activación denominada rectificador lineal (ReLU). La función característica de esta capa es la siguiente expresión:

$$r_m(i) = \max(0, c_k(p)) \quad (8)$$

Esta etapa ajusta los valores de  $c_k$  para tomar solo aquellos que estimulan a la red.  $r_m(p)$  tendrá la dimensión de 30 veces  $c_k(p)$  ( $m=1, \dots, 930$  valores) y de acuerdo a (8), se toma el valor de  $c_k(p)$  si es positivo, o cero si  $c_k(p)$  es negativo. Para la etapa de clasificación se toman los valores  $r_m(p)$  y cada uno se conecta a una neurona en la capa totalmente conectada (Fully Connect), de esta manera, se obtienen  $p=1, \dots, 930$  valores, dados por la salida de los  $k=30$  filtros, cada salida de longitud  $1 \times 31$ . En esta capa se lleva a cabo la multiplicación de  $r_m(p)$  por los pesos de la neurona  $wf_m(p)$  para finalmente sumar la polarización  $bf_m$  de cada neurona, lo que se caracteriza por la siguiente ecuación:

$$\varphi_o = r_m(p) \cdot wf_m(p) + bf_m \quad (9)$$

Donde  $\varphi_o$  contiene los valores de salida de la capa totalmente conectada, y  $o=0,1$ , un valor por cada neurona en la capa totalmente conectada. Finalmente, la última capa de esta etapa mapea los valores  $\varphi_o$  a un grado de probabilidad de pertenencia a una salida dada (clases) mediante la función soft max, descrita en la siguiente ecuación:

$$\Phi(e^{\varphi_o}) = \frac{e^{\varphi_o(i)}}{\sum_{l=0}^1 e^{\varphi_l(i)}} \quad (10)$$

$\Phi$  se compone de dos clases dadas las dos neuronas en la capa totalmente conectada, estas clases son  $\alpha_{Ej}$  y  $\beta_{Ej}$ , las cuales dan la probabilidad de pertenencia de una entrada dada a una clase, tales clases serian presencia de la onda P300 y no P300.

## 6.2. Predicción del estímulo seleccionado

Tomando en cuenta que la señal  $X_{Ejl}(n)$  son las 4 agrupaciones de cada estímulo  $j$  por cada señal  $l$ , una vez que la red neuronal convolucional en la subsección 6.1 de extracción de características ha dado los valores de pertenencia a cada clase según una entrada dada  $X_{Ejl}(n)$ , se determina cuál estímulo  $j$

cuenta con mayor probabilidad de presencia de la onda P300. Para ello se utiliza el parámetro  $\alpha_{Ej}$  del que se toma el valor máximo de probabilidad de los estímulos  $j=1, \dots, 4$  de la siguiente manera:

$$e = \max(\alpha_{Ej}) \quad (11)$$

$e$  determinará el número del estímulo  $j$  al que se fijó la mirada durante la sesión de toma de muestras.

## 7. RESULTADOS

Para obtener los resultados, se entrenó la CNN con un sujeto y se volvieron a realizar pruebas con los mismos estímulos visuales de la interfaz para que la red predijera cuál estímulo estaba observando. Los resultados se presentan de dos formas. La primera subsección muestra los resultados de la etapa de entrenamiento y validación en forma de matrices de confusión. Posteriormente, en la segunda subsección, se reportan los comentarios de las pruebas del sistema en línea.

### 7.1. Resultados de entrenamiento y validación de la red neuronal convolucional

En esta subsección se presenta el desempeño de la CNN en la detección de la onda P300 en 4 estímulos presentados durante la sesión de toma de muestras. Se tienen 192 agrupaciones en total dentro de  $X_{Ejl}(n)$ , pues son 4 señales promedio de cada señal  $s(n)$ , una por cada estímulo. Del total de las agrupaciones solamente 48 presentan P300, uno por cada sesión de toma de muestras. En la partición del conjunto de entrenamiento y validación, resultarían un total de 112 agrupaciones para entrenamiento, de las cuales solamente 28 contienen la onda P300 presente, mientras que para validación se utilizan 80 agrupaciones, de las que solamente 20 presentan la onda P300.

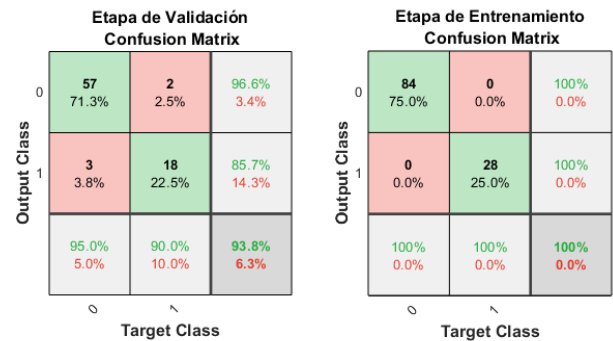


Figura 7. Matrices de confusión para entrenamiento y validación de la red neuronal convolucional.

En la Figura 7 se muestran los resultados de las etapas de entrenamiento y validación, siendo la clase 0-  $\beta_{Ej}$  y la clase 1-  $\alpha_{Ej}$ . Durante el entrenamiento, se observa que la red logra un desempeño del 100%, esto es debido a que la red se ajusta

correctamente a las señales de entrada presentadas, donde clasifica a 84 señales como no P300 y 28 con P300. Durante la validación la red clasifica correctamente 18 de 20 señales con P300 y 57 de 60 como No P300. El desempeño de esta etapa es de un 93.8%.

## 7.2. Pruebas en línea con la implementación del modelo CNN

En esta sección se llevaron a cabo las pruebas en línea del sistema, simulando una aplicación real de una persona discapacitada para que el sistema embebido predijera a cuál estímulo se fijó la mirada, colocándolo de color rojo al final de la sesión. Para ello se tomaron un total de 20 muestras en las mismas condiciones que se utilizaron para la base de datos de entrenamiento y prueba. Del total de 20 muestras, la red mostró un 100% de desempeño en la clasificación. Los resultados de la última sección pueden ser validados en [19].

## 8. CONCLUSIONES

En este artículo se presenta la implementación de una red neuronal convolucional para la detección de P300 basado en el modelo CNN300 propuesto en [11], pero adaptada a un sistema embebido BCI portable. Se utilizó el dispositivo de adquisición OpenBCI® operando en una Raspberry Pi 3 modelo B. Para lograr la implementación de la CNN en la Raspberry fue necesario realizar algunas modificaciones del modelo CNN300, las cuales implicaron utilizar un canal EEG y eliminar una capa convolucional de la CNN.

Los resultados de la implementación fueron satisfactorios, ya que al momento hacer pruebas en línea, la red clasifica correctamente las sesiones de toma de muestras. En lo referente al desempeño, los resultados son comparables a los métodos populares en la literatura para analizar el P300 con CNN. Además, el número de canales en este trabajo está muy por debajo del número utilizado en métodos populares en la literatura, permitiendo un algoritmo más simple que el resto de los métodos y es más cómodo para un sujeto el uso del sistema. También es importante destacar que el sistema se desarrolló en una Raspberry Pi 3, lo cual representa la posibilidad de generar computación ubicua, sin la necesidad de llevar consigo una computadora genérica. En conclusión, el método propuesto detecta con un desempeño aceptable la onda P300 en un sistema ubicuo BCI embebido que se puede posicionar con tres electrodos.

Como trabajo futuro se planea crear mayor rigidez estadística, ya que la etapa de entrenamiento siempre se realiza con las primeras 28  $s(n)$  señales, y la validación con las últimas 20 restantes. Este procedimiento puede favorecer a los resultados dados.

## AGRADECIMIENTOS

Los autores agradecen a Tecnológico Nacional de México por el apoyo TecNM 6418.18-P para la realización de este trabajo.

## 9. REFERENCIAS.

- [1] P. A. U. G. I. Bayarri, *Potenciales evocados somatosensoriales*. 2015.
- [2] E.-M. Lee, "Evoked potential: basic requirements and guidelines for writing reports", 2018.
- [3] S. N. Abdulkader, A. Atia, y M. S. M. Mostafa, "Brain computer interfacing: Applications and challenges", *Egypt. Informatics J.*, vol. 16, núm. 2, pp. 213–230, 2015.
- [4] M. Yu, N. Kaongoen, y S. Jo, "P300-BCI-based authentication system", *4th Int. Winter Conf. Brain-Computer Interface, BCI 2016*, pp. 2–5, 2016.
- [5] A. Baskaran, R. Milev, y R. S. McIntyre, "The neurobiology of the EEG biomarker as a predictor of treatment response in depression", *Neuropharmacology*, vol. 63, núm. 4, pp. 507–513, 2012.
- [6] L. F. Nicolas-Alonso y J. Gomez-Gil, "Brain computer interfaces, a review", *Sensors*, vol. 12, núm. 2, pp. 1211–1279, 2012.
- [7] E. AKMAN AYDIN, O. F. BAY, y I. Guler, "P300 Based Asynchronous Brain Computer Interface for Environmental Control System", *IEEE J. Biomed. Heal. Informatics*, vol. 2194, núm. c, pp. 1–1, 2017.
- [8] S. Samima, M. Sarma, y D. Samanta, "Correlation of P300 ERPs with visual stimuli and its application to vigilance detection", *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 2590–2593, 2017.
- [9] E. Carabez, M. Sugi, I. Nambu, y Y. Wada, "Convolutional Neural Networks with 3D Input for P300 Identification in Auditory Brain-Computer Interfaces", *Comput. Intell. Neurosci.*, vol. 2017, 2017.
- [10] H. Cecotti y A. Gr, "Convolutional neural networks for P300 Detection with Application to Brain-Computer Interfaces", vol. 33, núm. 3, pp. 433–445, 2011.
- [11] L. Madrid H., "Diseño de algoritmos para extracción de características de señal P300 en interfaces cerebro computadora", Instituto Tecnológico de Chihuahua, 2018.
- [12] "OpenBCI - Open Source Biosensing Tools (EEG, EMG, EKG, and more)", 2016. .
- [13] J. W. Shipman, "Tkinter 8.5 reference: a GUI for Python", pp. 1–118, 2013.
- [14] E. Donchin y L. A. Farwell, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials", *Electroencephalogr. Clin. Neurophysiol.*, vol. 70, núm. 6, pp. 510–523, 1988.
- [15] O. B. Blankertz, "Documentation Wadsworth BCI Dataset ( P300 Evoked Potentials ) BCI Competition III Challenge 2004", *Interface*, pp. 1–8, 2004.
- [16] A. Rakotomamonjy y V. Guigue, "BCI competition III: Dataset II-ensemble of SVMs for BCI P300 speller", *IEEE Trans. Biomed. Eng.*, vol. 55, núm. 3, pp. 1147–1154, 2008.
- [17] P. Vamplew, R. Dazeley, y C. Foale, "Softmax exploration strategies for multiobjective reinforcement learning", *Neurocomputing*, vol. 263, pp. 74–86, 2017.
- [18] T. Pamula, "Classification Based on Multilevel Filtering of Image Content Using Convolutional Neural Networks", *IEEE Intell. Transp. Syst. Mag.*, vol. PP, p. 12, 2018.
- [19] J. M. Macias M., "Interfaz Raspberry Pi 3 OpenBCI utilizando red neuronal CNN - YouTube", 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=FWzbBzJPd8I&t=109s>. [Consultado: 14-jun-2018].