ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

MODELO DE RECONOCIMIENTO EN TIEMPO REAL DE GESTOS DE LA MANO UTILIZANDO TÉCNICAS DE DEEP LEARNING Y SEÑALES ELECTROMIOGRÁFICAS

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

EDISON ALEJANDRO CHUNG LIU
edison.chung@epn.edu.ec

DIRECTOR: MARCO E. BENALCÁZAR, PHD.
marco.benalcazar@epn.edu.ec

Quito, julio de 2018

AVAL

Certifico que el presente trabajo fue desarrollado por Edison Alejandro Chung Liu, bajo mi supervisión.

MARCO ENRIQUE BENALCÁZAR, PHD.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo Edison Alejandro Chung Liu, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

EDISON ALEJANDRO CHUNG LIU

AGRADECIMIENTO

A mis padres por su apoyo incondicional.

Al director de mi trabajo de titulación, Marco Benalcázar, por haberme guiado durante la realización de este trabajo, y por haberme permitido formar parte del proyecto de investigación PIJ-16-13.

A mis compañeros que forman parte del proyecto de investigación PIJ-16-13 de la Escuela Politécnica Nacional por la ayuda brindada durante la realización de este trabajo.

Edison Alejandro Chung Liu

ÍNDICE DE CONTENIDO

RESUMEN

El reconocimiento de gestos tiene múltiples dominios de aplicación como medicina, ingeniería, y robótica. Asimismo, nos permite desarrollar nuevas maneras más naturales de abordar la interacción humano-máquina. El reconocimiento de gestos de la mano en tiempo real consiste en identificar un gesto realizado por la mano en un momento dado. En este trabajo, proponemos un modelo para reconocimiento de gestos de la mano en tiempo real, el cual toma como entrada las señales electromiográficas (EMG) de la actividad muscular en el antebrazo, medidas mediante el sensor *Myo Armband*. Utilizamos un *autoencoder* para extracción automática de características, y una red neuronal artificial (ANN, por sus siglas en inglés) basada en la función de transferencia unidad lineal rectificada (ReLU, por sus siglas en inglés) para clasificación. El modelo propuesto puede reconocer los mismos 5 gestos que el sistema de reconocimiento propietario del *Myo Armband* (*fist*, *wave left*, *wave right*, *fingers spread* y *double tap*), logrando una exactitud de reconocimiento promedio de 85.08% $\pm$ 15.21%, con un tiempo de respuesta promedio de 3 $\pm$ 1 ms. El modelo propuesto es general, esto implica que puede reconocer los gestos de cualquier usuario, aun cuando sus datos no hayan sido incluidos en el conjunto de entrenamiento formado por 50 usuarios. Esta característica representa una ventaja para el modelo propuesto ya que no requiere de entrenamiento por cada usuario de este modelo a diferencia de modelos específicos por usuario que requieren un entrenamiento usando los datos de la persona que utilizará el modelo.

PALABRAS CLAVE: reconocimiento de gestos de la mano, EMG, tiempo real, Myo Armband, aprendizaje profundo, extracción automática de características, red neuronal artificial, autoencoder, ReLU,

ABSTRACT

Gesture recognition has multiple application domains such as medicine, engineering and robotics. It also allows us the development of new and more natural approaches to human-machine interaction. Real-time hand gesture recognition consists of identifying a given gesture performed by the hand at any given moment. In this work, we propose a model for real-time hand gesture recognition, which takes as input the electromyographic (EMG) signals from the muscular activity in the forearm, measured with the Myo Armband sensor. We use an autoencoder for automatic feature extraction, and an artificial neural network (ANN) based on the rectified linear unit (ReLU) transfer function for classification. The proposed model can recognize the same 5 gestures as the proprietary recognize the same 5 gestures as the proprietary recognition system of the Myo Armband (fist, wave left, wave right, fingers spread and double tap), achieving an average recognition accuracy of 85.08% $\pm$ 15.21%, with an average response time of 3 $\pm$ 1 ms. The proposed model is general, this implies that it can recognize the gestures from any user, even when their data were not included in the training dataset composed of 50 users. This feature represents an advantage for the proposed model since it does not require training for each user of this model, unlike user-specific models which require training using the data from the person that will use the model.

KEYWORDS: hand gesture recognition, EMG, real-time, Myo Armband, deep learning, automatic feature extraction, artificial neural network, autoencoder, ReLU.

1. INTRODUCCIÓN

El reconocimiento de gestos tiene múltiples dominios de aplicación como medicina, ingeniería, y robótica. Asimismo, nos permite desarrollar nuevas maneras más naturales de abordar la interacción humano-máquina. El reconocimiento de gestos de la mano consiste en identificar un movimiento dado de la mano, y el instante en el que ocurre dicho movimiento [1]. El reconocimiento de gestos de la mano permite desarrollar nuevas formas de interacción humano-máquina más naturales y centradas en el ser humano. Resolver el problema de reconocimiento de gestos involucra varios retos. Por ejemplo, dos personas distintas pueden realizar el mismo gesto de manera diferente. Los datos requeridos para el reconocimiento de gestos pueden ser adquiridos mediante varios tipos de sensores: guantes, cámaras, sensores de ultrasonido, etc., cada uno con sus propios retos a superar. Por ejemplo, las cámaras son sensibles a variaciones en la luz y tienen problemas con la oclusión y el cambio de distancia entre la mano y la cámara [2]. Los guantes pueden no ser de la talla correcta para las manos del usuario y pueden resultar incómodos para algunos usuarios dependiendo de la aplicación y del tiempo de uso [1]. En los sensores de ultrasonido se puede dar el efecto *cross-talking*, que ocurre cuando un sensor recibe la onda de otro sensor o una onda propia producto de un disparo previo o de dispersiones.

En este proyecto proponemos el uso de sensores electromiográficos para capturar las señales eléctricas producidas por la actividad muscular en el antebrazo. Estas señales eléctricas se llaman señales electromiográficas, o simplemente electromiografías (EMG) [3]. La clasificación de estas señales puede ser utilizada en múltiples dominios de aplicación, incluyendo: interfaces para videojuegos, robótica, traducción de lenguaje de señas a texto o voz, y biónica [4, 5]. Las EMGs son señales ruidosas que pueden variar en amplitud y frecuencia debido al ruido en el ambiente circundante, inducción electromagnética, o incluso la interacción entre las señales eléctricas de diferentes tejidos. Sin embargo, las señales EMG reflejan los comandos enviados por el cerebro para contraer un músculo esquelético y así producir fuerza y/o movimiento. Por lo tanto, las señales EMG son una representación directa de la intención de movimiento de un usuario [6]. Por esta razón, las señales EMG son un buen candidato para ser la entrada de un sistema de reconocimiento de gestos de la mano.

El modelo de reconocimiento de gestos de la mano propuesto en este proyecto utiliza el *Myo Armband* para recolectar las señales EMG del antebrazo del usuario. Implementamos una red neuronal multicapa *feed-forward* para clasificar las señales EMG. Utilizamos este modelo de aprendizaje de máquina debido a que este tipo de red neuronal tiene el potencial de volverse una máquina de aprendizaje universal [7]. En otras palabras, esta arquitectura

de red tiene el potencial de implementar cualquier frontera de decisión. La arquitectura de la red propuesta consta de 4 capas: una capa de entrada, dos capas ocultas y una capa de salida. La capa de entrada toma la señal EMG rectificada. La primera capa oculta se utiliza para la extracción automática de características. La segunda capa oculta, junto con la capa de salida, implementan el clasificador.

El modelo propuesto es capaz de procesar y clasificar señales EMG en tiempo real, y de reconocer, con una mayor exactitud, los mismos gestos que el sistema propietario y cerrado del *Myo Armband*. Los gestos reconocidos en este trabajo se muestran en la Figura 1. El modelo propuesto funciona para cualquier usuario y no requiere repeticiones de entrenamiento, ni ajustes antes de usarlo, puesto que ya ha sido entrenado como un modelo de reconocimiento general.

Figura 1 Gestos reconocidos por el sistema

1.1. Marco teórico

Estado del arte

Varios investigadores han propuesto distintos sistemas de reconocimiento de gestos para múltiples propósitos. Utilizando señales EMG y una variedad de clasificadores, reportan una alta exactitud para un bajo número de gestos. Sin embargo, se trata de modelos específicos para cada usuario que requieren de entrenamiento específico para la persona que los utilizará. Por ejemplo, en [8], usando una máquina de vectores de soporte (SVM, por sus siglas en inglés) como clasificador y señales EMG, el sistema propuesto, de bajo consumo energético, logra una exactitud de 88% para 3 gestos con solo 5 repeticiones de entrenamiento. Este sistema específico por usuario funciona en tiempo real. En [9], el sistema propuesto, también específico por usuario, puede reconocer hasta 4 gestos en tiempo real con una exactitud de reconocimiento de 87%, utilizando un conjunto de electrodos, un clasificador SVM, y 10 repeticiones de entrenamiento. En [10], proponen un sistema para conducir un carro a control remoto utilizando un solo electrodo EMG y un conjunto de 4 gestos, con una exactitud de 94% al combinar dos clasificadores lineales simples. Al igual que los anteriores, este sistema es específico por usuario, funciona en tiempo real, pero requiere de 20 repeticiones de entrenamiento.

También se ha propuesto sistemas de reconocimiento que utilizan redes neuronales artificiales (ANNs, por sus siglas en inglés). En [11], proponen un sistema capaz de obtener una exactitud de 100% para 4 gestos, utilizando un conjunto de electrodos EMG y un perceptrón multicapa que contiene una capa oculta, y utilizando el algoritmo de retropropagación del error (BP, por sus siglas en inglés) para entrenamiento. Sin embargo, este sistema no funciona en tiempo real. En [12], el modelo propuesto también utiliza una ANN para clasificar las señales de 4 gestos utilizando 2 electrodos EMG, con una exactitud de 83.5%. Este modelo general obtiene una exactitud más baja que los anteriormente descritos, con el mismo número de gestos. En [13], proponen un sistema para controlar un helicóptero quadrotor con 4 electrodos EMG y un conjunto de 4 gestos. Este sistema en tiempo real usa una red neuronal BP para obtener una exactitud de reconocimiento de 93%, pero es específico por usuario y requiere de 20 repeticiones para su entrenamiento. En [14], propusieron un sistema que puede reconocer 3 gestos en tiempo real mediante análisis de una señal EMG de un solo canal. Usando la transformada wavelet y una ANN, obtuvieron una exactitud de reconocimiento promedio de 93.25%. En [15], entrenaron una red neuronal *feed-forward* utilizando BP para reconocer 6 gestos en tiempo real a partir de una señal EMG de 3 canales, y obtuvieron una exactitud de reconocimiento de 71%.

Tipos de modelos para reconocimiento de gestos

Existen dos formas para entrenar un modelo de reconocimiento de gestos. La primera es que el usuario realice un cierto número de repeticiones para cada gesto, y, a partir de ellas, entrenar el modelo. El resultado es un modelo específico para el usuario, el cual solo puede reconocer los gestos cuando son realizados por ese usuario en particular. La segunda manera es entrenar el modelo con anterioridad, utilizando un conjunto de datos que contiene las repeticiones de los gestos realizados por una variedad de usuarios. El resultado es un modelo general capaz de reconocer estos gestos cuando son realizados por cualquier usuario, incluso, si los datos de ese usuario no fueron utilizados para el entrenamiento del modelo.

Tipos de sensores EMG

Existen dos tipos de sensores utilizados para capturar las señales EMG: invasivos y no invasivos. Los sensores invasivos, usualmente agujas, requieren un ambiente controlado, son más caros e incómodos de usar para el usuario. Los sensores no invasivos, usualmente electrodos superficiales, son menos precisos, pero más versátiles para aplicaciones cotidianas y de ingeniería. En este proyecto, proponemos el uso del *Myo Armband* de *Thalmic Labs*. El *Myo Armband* es un sensor que nos permite capturar las

señales EMG del antebrazo de manera no invasiva, sin la necesidad de geles que mejoren la conductividad entre los electrodos y la piel. Además, el *Myo Armband* es más barato, ligero y pequeño que otros sensores comerciales disponibles para el mismo propósito.

1.2. Pregunta de investigación

El modelo general de reconocimiento de gestos de la mano, en tiempo real y para uso cotidiano, propuesto en este trabajo, podrá lograr una exactitud de reconocimiento superior al 85%, que corresponde al promedio obtenido por sistemas de alta tecnología de última generación [8].

1.3. Objetivo general

Desarrollar un modelo de reconocimiento en tiempo real de gestos de la mano usando técnicas de aprendizaje profundo y señales EMG del brazo humano.

1.4. Objetivos específicos

- Desarrollar un modelo general de clasificación de señales EMG que pueda ser utilizado por cualquier usuario.
- Reconocer 6 clases (*relax*, *fist*, *wave left*, *wave right*, *fingers spread*, *double tap*) de gestos realizados con la mano, utilizando el sensor *Myo Armband*, en tiempo real, es decir, con un tiempo de respuesta menor a 300ms.
- Obtener una exactitud de reconocimiento para el modelo propuesto superior al 85%.

1.5. Hipótesis

El uso de técnicas de aprendizaje profundo y señales EMG permitirá implementar un sistema de reconocimiento de gestos de la mano que funcione en tiempo real (que responda en menos de 300 ms [16]) y que tenga una exactitud mayor al 85%, considerando que la dificultad del problema se incrementa con el reconocimiento de 6 clases: *relax*, *fist*, *waveIn*, *waveOut*, *fingersSpread*, *doubleTap*.

Este trabajo, se ha organizado en 4 capítulos, incluido el presente (Capítulo I). En el capítulo II, se presenta la metodología propuesta, en el cual se describen detalladamente los materiales y métodos utilizados. En el capítulo III, presentamos los resultados obtenidos de la evaluación del modelo propuesto, así como la discusión de los resultados obtenidos. En el capítulo IV, presentamos las conclusiones de nuestro trabajo, así como lineamientos para trabajo futuro. Finalmente, presentamos los materiales bibliográficos que han sido revisados en el desarrollo de este trabajo.

2. METODOLOGÍA

2.1. Materiales

En esta sección, describimos las características del *Myo Armband*, y la naturaleza de las muestras de gestos grabadas en la base de datos que utilizamos.

Myo Armband

El *Myo Armband*, mostrado en la Figura 2, es un dispositivo de control mediante gestos y movimiento de la mano, fabricado por *Thalmic Labs*. También permite la medición de las señales eléctricas y la orientación del antebrazo. Pesa aproximadamente 93g y consta de 8 sensores EMG que operan a una frecuencia de muestreo de 200Hz, junto con una unidad de medición inercial (IMU, por sus siglas en inglés) que funciona a una frecuencia de 50 Hz. La IMU contiene un giroscopio, un acelerómetro y un magnetómetro. En este proyecto, sin embargo, solo utilizamos los datos EMG. El *Myo Armband* utiliza Bluetooth para conectarse a la computadora. También viene con un software propietario, de caja negra, que reconoce los cinco gestos mostrados en la Figura 1, y permite que la computadora utilice estos gestos y la orientación del brazo para controlar una variedad de aplicaciones. Gracias al kit de desarrollo de software *Myo* liberado por *Thalmic Labs*, y la comunidad activa de desarrolladores para el *Myo Armband*, el número de aplicaciones disponibles que utilizan este dispositivo crece cada día.

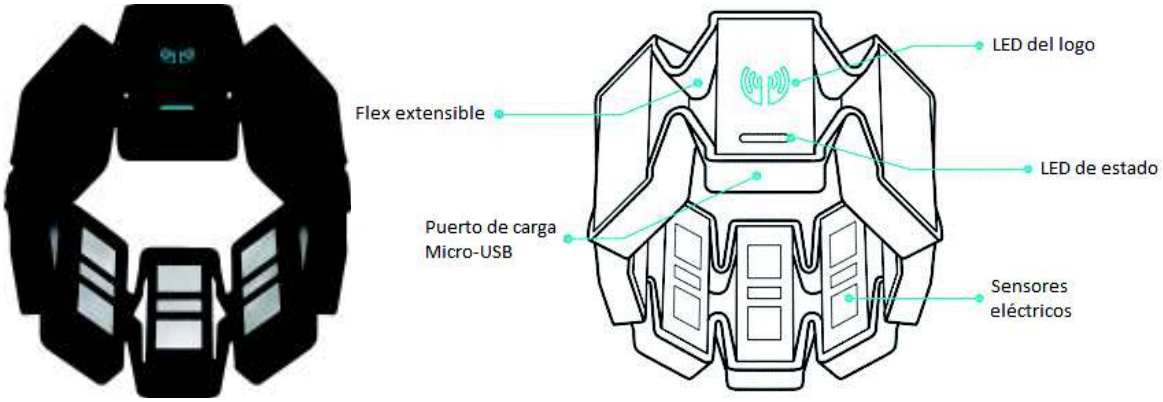

Figura 2 Myo Armband

Base de datos de gestos

Utilizamos el *Myo Armband* para grabar las señales EMG de los 5 gestos mostrados en la Figura 1. Estos gestos fueron realizados por 120 usuarios diferentes. Grabamos 50 muestras de cada gesto por usuario, con cada muestra contenida en una ventana de 5 segundos. Estas 50 muestras fueron divididas en 2 subconjuntos de 25 muestras cada uno,

uno para entrenamiento y el otro para evaluación. También grabamos 10 muestras del brazo en una posición relajada. En total, grabamos 37200 muestras.

A todos los usuarios se les pidió utilizar el *Myo Armband* en su brazo derecho, con el indicador LED sobre el brazo y apuntando hacia la mano, y con su palma apuntando hacia el suelo, como se muestra en la Figura 3. Para cada muestra, se pidió al usuario empezar en la posición de relajación, realizar el gesto solo una vez cuando el programa de grabación dé la señal de inicio del gesto, y luego regresar a la posición de relajación, todo dentro de 5 segundos.

Figura 3 Posición del *Myo Armband*

También guardamos información básica sobre el usuario, incluyendo género (75% hombres, 25% mujeres), edad (17-29 años), mano dominante (9% zurdos, 81% diestros), y si sufrieron o no alguna lesión en su brazo derecho (16% sufrió una lesión). Los datos grabados incluyen la EMG para cada gesto, el nombre de cada gesto, la respuesta del software propietario del *Myo Armband*, y las lecturas de la IMU.

Los datos EMG grabados para una sola muestra de cualquier gesto constan de 8 canales, uno por cada sensor del *Myo Armband*, con sus valores dentro del intervalo [-1, 1]. A una frecuencia de muestreo de 200Hz, a lo largo de 5 segundos, obtenemos 1000 valores por canal. Sin embargo, la duración del gesto en sí es menor a 5 segundos.

Utilizamos los datos de 60 de los 120 usuarios para el desarrollo de este modelo. Estos 60 usuarios fueron seleccionados al azar. De estos 60 usuarios, 50 fueron utilizados para entrenar el modelo y 10 fueron separados para selección del modelo. Los datos utilizados para entrenar el modelo propuesto corresponden a todas las 50 muestras EMG de entrenamiento y de evaluación de cada gesto grabado de los 50 usuarios con sus etiquetas respectivas. Los datos correspondientes a las muestras EMG de evaluación de los 60 usuarios restantes fueron utilizados para la evaluación final del modelo completamente entrenado.

2.2. Métodos

El desarrollo del modelo de reconocimiento de gestos propuesto en este proyecto consta de 2 fases: entrenamiento y evaluación.

Fase de entrenamiento

La fase de entrenamiento consta de 4 etapas:

1) **Etapa preliminar**: En esta etapa, recolectamos información sobre el estado del arte en la detección y preprocesamiento de señales EMG, el reconocimiento de gestos basado en señales EMG, la operación del *Myo Armband*, y técnicas de aprendizaje profundo.
2) **Análisis y Diseño**: En esta etapa, analizamos el funcionamiento en tiempo real de las posibles técnicas de aprendizaje profundo disponibles para el desarrollo del modelo de reconocimiento de gestos. Escogimos un enfoque basado en extracción automática de características con una ANN. Finalmente, diseñamos nuestro algoritmo de reconocimiento de gestos.
3) **Implementación**: En esta etapa, desarrollamos el código necesario para entrenar y para evaluar el modelo de reconocimiento de gestos.
4) **Ajuste fino**: En esta etapa, probamos el modelo con el conjunto de datos para selección de modelo. Previo al ajuste fino, estimamos una exactitud de reconocimiento promedio de 83.28% sobre este conjunto de datos. Para entrenar completamente el modelo, realizamos un ajuste fino basándonos en los resultados de estas pruebas, de manera que el modelo cumpla las restricciones de tiempo y exactitud de reconocimiento definidas anteriormente, obteniendo el modelo de reconocimiento final, con una exactitud de reconocimiento promedio de 87.44% sobre el conjunto de datos para selección de modelo.

Fase de evaluación

Esta fase consta de una única etapa de evaluación final. En esta etapa, obtenemos y analizamos los resultados de la evaluación del modelo propuesto: tiempo de respuesta y exactitud de reconocimiento.

Estructura del modelo de reconocimiento

La estructura del modelo de reconocimiento consta de 5 pasos, que se muestran en la Figura 4. En la figura 5, se muestra la arquitectura final de la ANN del modelo propuesto en este trabajo.

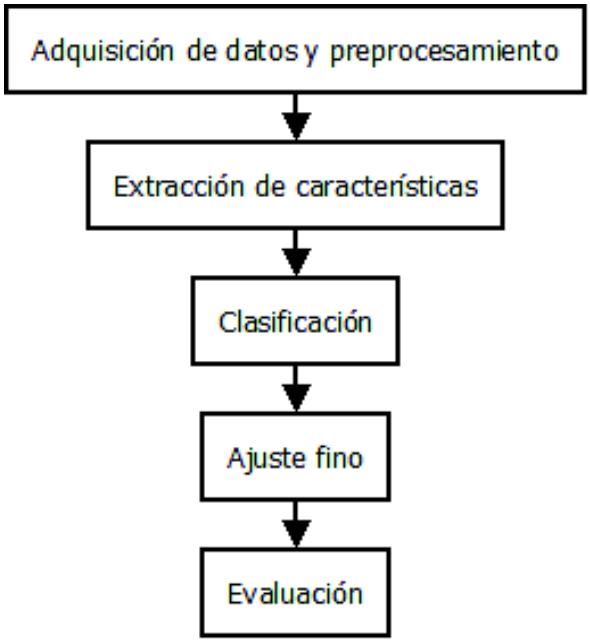

Figura 4 Estructura del modelo de reconocimiento.

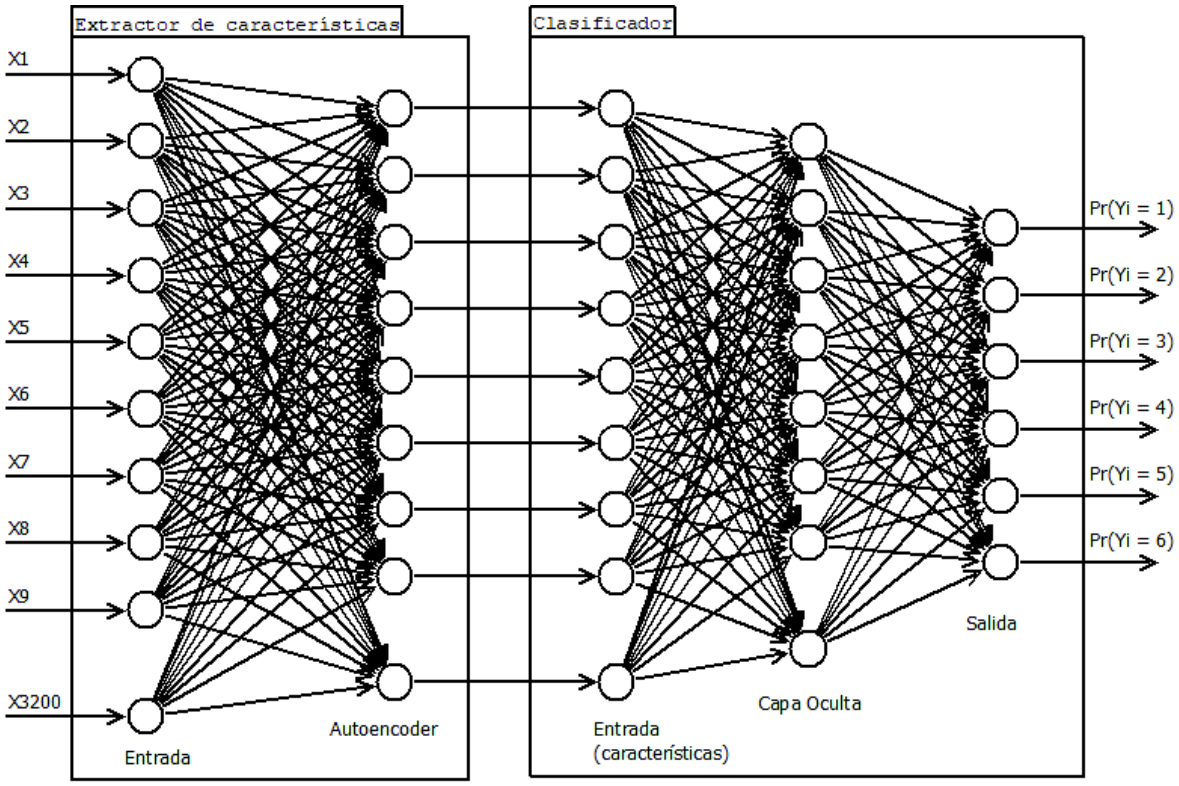

Figura 5 Arquitectura ANN del modelo de reconocimiento de gestos de la mano

Paso 1. Adquisición de datos y preprocesamiento: El preprocesamiento para cada muestra empieza con la rectificación de la señal, mediante el cálculo del valor absoluto en todos los canales. Luego, utilizamos el detector de actividad muscular propuesto en [1] para segmentar la región de la señal EMG correspondiente a la contracción del músculo, como se muestra en la Figura 6. En otras palabras, utilizamos el detector para eliminar la cabecera y la cola de cada muestra, las cuales no contienen información útil sobre el gesto.

Esto se debe a que las muestras de cada gesto empiezan y terminan en la posición de relajación. Si dentro de la muestra no existe una región correspondiente a la contracción de un músculo, como por ejemplo en muestras de relajación, utilizamos todos los puntos de la muestra.

Después, para simular cómo un sistema en tiempo real funcionaría, utilizamos una ventana deslizante, obteniendo varias observaciones de ventana. Encontramos que los mejores resultados de exactitud se obtuvieron utilizando una ventana deslizante con un tamaño de 400 valores, y con una longitud de salto entre dos ventanas consecutivas de 10 valores. La longitud del salto es directamente proporcional al tiempo de respuesta del modelo. Si el tamaño de la muestra es menor que el tamaño de la ventana, se agregan ceros antes y después de la muestra para llenar la longitud de la ventana. Esto puede resultar en una sola observación de ventana para una muestra. Cada observación de ventana obtenida es transformada en un solo vector mediante la concatenación de los 8 canales, empezando por los valores del primer canal, luego los valores del segundo canal y así sucesivamente, como se muestra en la Figura 7. Todos los vectores obtenidos de esta manera de todas las muestras son recogidos en una sola matriz.

El resultado es una matriz con 779653 filas y 3200 columnas, en la cual cada fila corresponde a una sola observación de ventana, como se muestra en la Figura 8. Luego, la matriz de datos de los 50 usuarios para entrenamiento es dividida: 50% de las observaciones componen un conjunto de entrenamiento no supervisado, 25% un conjunto de entrenamiento supervisado, y el restante 25% un conjunto de validación. Finalmente, cada uno de estos conjuntos es permutado al azar.

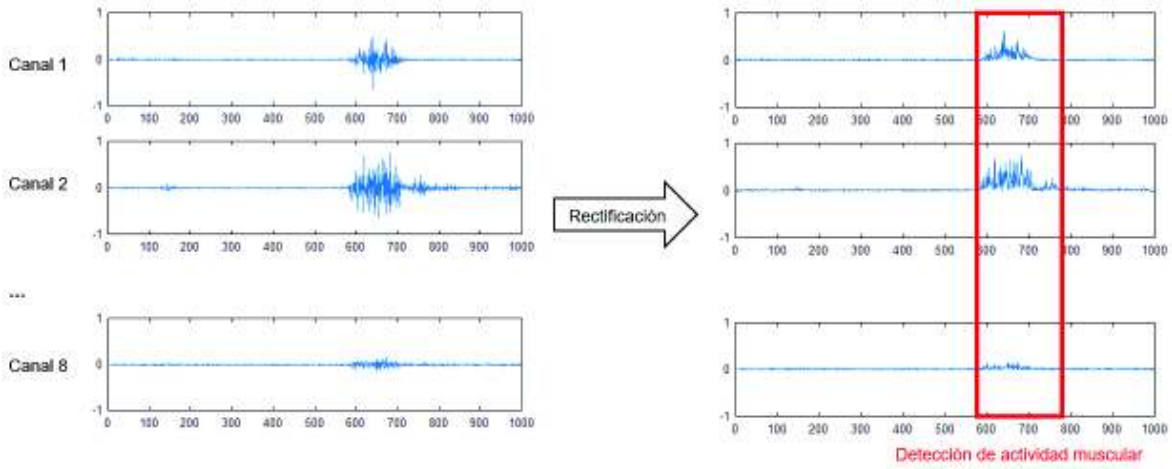

Figura 6 Rectificación de la señal EMG y detección de la actividad muscular

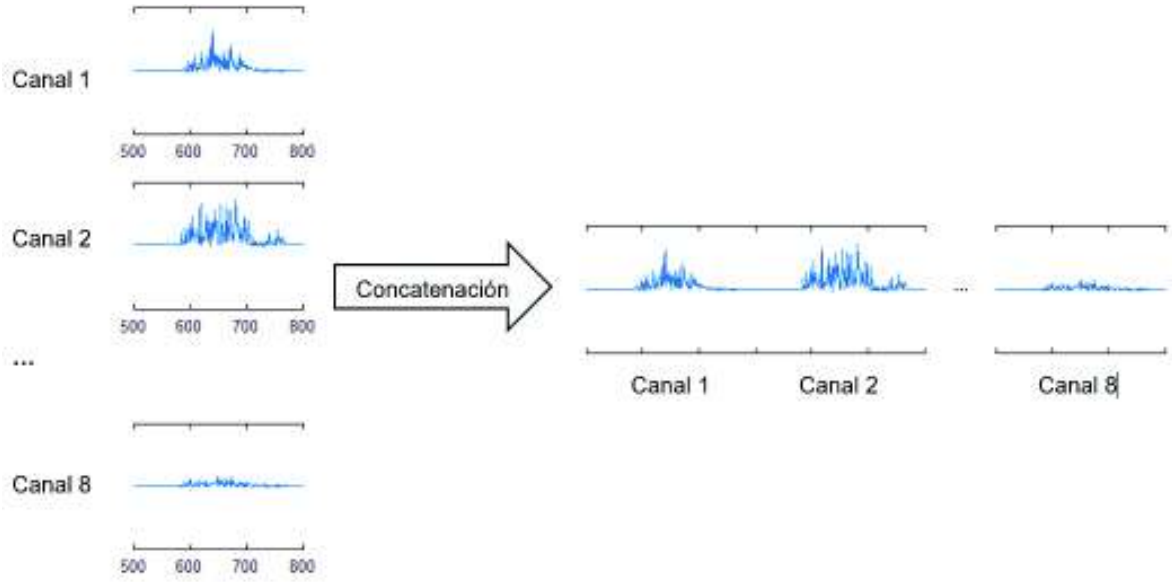

Figura 7 Concatenación de los canales de una observación de ventana

	Valores del canal 1	Valores del canal 2	…	Valores del canal 8
Observación 1			…	
Observación 2			…	
Observación 3			…	
…	…	…	…	…
Observación n			…	

Figura 8 Estructura de la matriz de observaciones

Paso 2. Extracción de características: En este paso, entrenamos un *autoencoder*, utilizando el conjunto de entrenamiento no supervisado, para la extracción automática de características. Un *autoencoder* es una red neuronal cuyo objetivo es aprender la función identidad, de manera que su salida sea igual a su entrada. Al reducir el tamaño de su capa

oculta, el *autoencoder* debe aprender una representación comprimida de la entrada, es decir, codifica la entrada. Esta representación comprimida contiene las características que le permiten al *autoencoder* reconstruir la entrada al aplicar el proceso de decodificación [17].

Para entrenar el *autoencoder*, utilizamos el método L-BFGS para optimizar la función de costo, la cual corresponde al error cuadrático medio entre la entrada y la salida del *autoencoder*. El *autoencoder* entrenado es utilizado para codificar las muestras del conjunto de entrenamiento supervisado, dando como resultado un conjunto de vectores de características que será utilizado para entrenar el clasificador junto con sus etiquetas respectivas. El tamaño de cada vector es igual al tamaño de la capa oculta del *autoencoder*.

Después de haber probado varias configuraciones, incluyendo doble y triple *autoencoder*, determinamos que la mejor configuración para nuestro proyecto consta de un solo *autoencoder*, con un tamaño de 200 neuronas en la capa oculta, un parámetro de decaimiento de peso de 3*10-3, un peso de penalización de dispersión de 0.5, y una proporción de dispersión de 0.1.

Paso 3. Entrenamiento del clasificador: Entrenamos el clasificador utilizando el conjunto de vectores de características obtenidos en el paso anterior a partir del conjunto de entrenamiento supervisado. Este clasificador está basado en una ANN de 3 capas: la capa de entrada toma el vector de características que es la salida del *autoencoder*, y la capa oculta junto con la capa de salida implementan el módulo de clasificación, el cual retorna un vector columna con 6 filas que contiene las probabilidades de que la entrada pertenezca a cada clase. Definiendo un valor de 0.8 como el umbral de probabilidad condicional, se toma la clase con la más alta probabilidad y se la asigna a la muestra respectiva. Si todas las probabilidades son menores que el umbral, se etiqueta la muestra como clase 1 (no gesto o relajación).

Las funciones de transferencia que probamos incluyen la unidad lineal rectificada (ReLU, por sus siglas en inglés), tanh, *log-sigmoid*, *softplus*, y la unidad lineal exponencial (ELU, por sus siglas en inglés). De todas estas funciones, los mejores resultados se obtuvieron utilizando una capa oculta de 150 neuronas con la función ReLU, y un factor de regularización de 0.1 para el entrenamiento de la ANN.

Paso 4. Ajuste fino: La arquitectura final del modelo completo corresponde a una ANN con 4 capas: entrada, *autoencoder*, capa oculta ReLU, y salida. Realizamos un ajuste fino de esta ANN haciendo varias iteraciones, en un rango bajo (de 1 a 10) y en un rango alto (de 250 a 500), con las entradas y etiquetas del conjunto de entrenamiento no supervisado,

con los datos del conjunto de entrenamiento supervisado, y con los datos de ambos conjuntos combinados. El mejor resultado sobre el conjunto de selección del modelo se obtuvo con 500 iteraciones y los datos solo del conjunto de entrenamiento supervisado. Este proceso incrementó significativamente la exactitud de reconocimiento promedio en el conjunto de selección del modelo de 83.28% a 87.44%.

Paso 5. Evaluación: En este paso, estimamos la exactitud de reconocimiento del modelo propuesto. Para ello, utilizamos los datos de los 60 usuarios del conjunto de datos de evaluación, el cual no fue utilizado para entrenamiento, ni para selección del modelo. Debido a que no necesitamos entrenar el modelo general con los datos de cada usuario, no necesitamos utilizar las muestras de entrenamiento de estos usuarios. En total, se realizó la evaluación sobre 7500 muestras, 2500 por gesto. Cada muestra es preprocesada de la misma manera que se describió en el primer paso de esta sección, excepto que esta vez no se utiliza el detector de actividad muscular, y cada muestra se clasifica por separado.

El objetivo de la evaluación de reconocimiento es obtener, para cada muestra de un gesto, una sola clase correspondiente al gesto. Sin embargo, la salida del clasificador es un vector de resultados, denotado como R, en el cual cada elemento corresponde a la clase de una observación de ventana de la muestra. Idealmente, R empezaría con etiquetas correspondientes a la clase 1 (no gesto o relajación), luego tendría algunas etiquetas correspondientes a la clase del gesto reconocido dentro de la muestra, y terminaría con etiquetas de la clase 1. En la práctica, R puede tener valores atípicos, es decir, etiquetas de otros gestos, como resultado de clasificaciones erróneas.

Para eliminar estos valores atípicos, cada clasificación $C_i \in R$, con $i > 1$, es comparada con la clasificación anterior C_{i-1}. Si C_i es diferente del C_{i-1} original, C_i es reemplazado por una clasificación de relajación, como se muestra en la Figura 9.

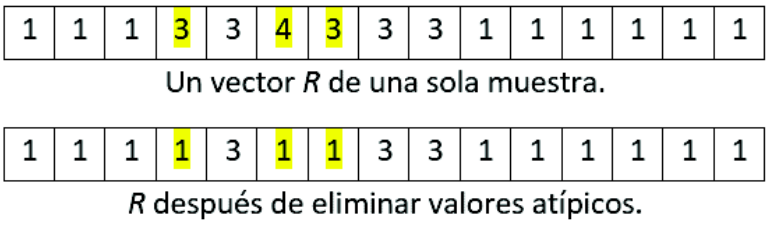

Figura 9 Eliminación de valores atípicos del vector R

Para obtener una sola clase para toda la muestra, primero se eliminan las clasificaciones de relajación de R. Después, definimos tres métodos diferentes para obtener esta clase: el

primer método es etiquetar R con el gesto más frecuente que aparezca en este vector, método que será referido como la moda, "*Mode*". El segundo método es etiquetar R con el primer gesto diferente de relajación que aparezca en este vector, método que será referido como la primera transición, "*First Transition*". El tercer método es obtener un vector ordenado de elementos únicos que aparecen en R, denotado como U, y tomar el primer gesto de U como la clase de R, método que será referido como único, "*Unique*".

Una vez que todas las muestras son procesadas y clasificadas, agrupamos todas las clasificaciones de las muestras en tres vectores, según el método utilizado para obtenerlas. Entonces, comparamos cada uno de estos vectores con un vector que contiene las etiquetas originales, obteniendo así la exactitud de reconocimiento para cada método descrito anteriormente.

3. RESULTADOS Y DISCUSIÓN

3.1. Resultados

Exactitud de reconocimiento

Para los 60 usuarios de evaluación, obtuvimos una exactitud de reconocimiento promedio de 85.08 ± 15.21% utilizando el método *Mode*, una exactitud de reconocimiento promedio de 82.29 ± 14.71% utilizando el método *First Transition*, y una exactitud de reconocimiento promedio de 82.32 ± 15.43% utilizando el método *Unique*. A continuación, se presentan las matrices de confusión para cada método en las Tablas 1, 2 y 3, respectivamente. Para el método *Mode*, la mayor y menor sensibilidad ocurren para los gestos *wave left* (87.3%) y *fingers spread* (78.8%), respectivamente. La mayor y menor precisión ocurren para los mismos gestos, con 93.1% para *wave left* y 82.5% para *fingers spread*. Para el método *First Transition*, la mayor y menor sensibilidad ocurren para los gestos *wave left* (85.1%) y *fingers spread* (77.3%), respectivamente. La mayor y menor precisión ocurren para los mismos gestos, con 89.9% para *wave left* y 83.2% para *fingers spread*. Para el método Unique, la mayor y menor sensibilidad ocurren para los gestos *fist* (88.8%) y *fingers spread* (75.2%), respectivamente. La mayor y menor precisión ocurren para los gestos *wave left* (90.2%) y *fist* (81.7%), respectivamente.

Tabla 1 Matriz de confusión para el método *Mode*

		Objetivos					**% PRECISIÓN** % ERROR
		FIST	*WAVE LEFT*	*WAVE RIGHT*	*FINGERS SPREAD*	*DOUBLE TAP*	
Predicciones	NO GESTO	**0** 0.0%	**35** 0.5%	**8** 0.1%	**82** 1.1%	**29** 0.4%	**0.0%** 100%
	FIST	**1296** 17.3%	**81** 1.1%	**1** 0.0%	**43** 0.6%	**66** 0.9%	**87.2%** 12.8%
	WAVE LEFT	**73** 1.0%	**1310** 17.5%	**3** 0.0%	**3** 0.0%	**18** 0.2%	**93.1%** 6.9%
	WAVE RIGHT	**22** 0.3%	**68** 0.9%	**1287** 17.2%	**112** 1.5%	**17** 0.2%	**85.5%** 14.5%
	FINGERS SPREAD	**17** 0.2%	**3** 0.0%	**167** 2.2%	**1182** 15.8%	**64** 0.9%	**82.5%** 17.5%
	DOUBLE TAP	**57** 0.8%	**30** 0.4%	**31** 0.4%	**78** 1.0%	**1306** 17.4%	**87.0%** 13.0%
	%SENSIBILIDAD % ERROR	**86.4%** 13.6%	**87.3%** 12.7%	**85.8%** 14.2%	**78.8%** 21.2%	**87.1%** 12.9%	**85.08%** 14.92%

Tabla 2 Matriz de confusión para el método *First Transition*

		Objetivos					**% PRECISIÓN** % ERROR
		FIST	*WAVE LEFT*	*WAVE RIGHT*	*FINGERS SPREAD*	*DOUBLE TAP*	
Predicciones	NO GESTO	**50** 0.7%	**22** 0.3%	**46** 0.6%	**135** 1.8%	**59** 0.8%	**0.0%** 100%
	FIST	**1226** 16.3%	**92** 1.2%	**8** 0.1%	**42** 0.6%	**60** 0.8%	**85.9%** 14.1%
	WAVE LEFT	**96** 1.3%	**1277** 17.0%	**18** 0.2%	**2** 0.0%	**28** 0.4%	**89.9%** 10.1%
	WAVE RIGHT	**32** 0.4%	**60** 0.8%	**1255** 16.7%	**98** 1.3%	**30** 0.4%	**85.1%** 14.9%
	FINGERS SPREAD	**25** 0.3%	**5** 0.1%	**136** 1.8%	**1160** 15.5%	**69** 0.9%	**83.2%** 16.8%
	DOUBLE TAP	**71** 0.9%	**44** 0.6%	**37** 0.5%	**63** 0.8%	**1254** 16.7%	**85.4%** 14.6%
	%SENSIBILIDAD % ERROR	**81.7%** 18.3%	**85.1%** 14.9%	**83.7%** 16.3%	**77.3%** 22.7%	**83.6%** 16.4%	**82.29%** 17.71%

Tabla 3 Matriz de confusión para el método *Unique*

		Objetivos					**% PRECISIÓN** % ERROR
		FIST	*WAVE LEFT*	*WAVE RIGHT*	*FINGERS SPREAD*	*DOUBLE TAP*	
Predicciones	NO GESTO	**50** 0.7%	**22** 0.3%	**46** 0.6%	**135** 1.8%	**59** 0.8%	**0.0%** 100%
	FIST	**1332** 17.8%	**140** 1.9%	**9** 0.1%	**65** 0.9%	**85** 1.1%	**81.7%** 18.3%
	WAVE LEFT	**62** 0.8%	**1255** 16.7%	**35** 0.5%	**4** 0.1%	**36** 0.5%	**90.2%** 9.8%
	WAVE RIGHT	**14** 0.2%	**59** 0.8%	**1261** 16.8%	**110** 1.5%	**38** 0.5%	**85.1%** 14.9%
	FINGERS SPREAD	**8** 0.1%	**3** 0.0%	**123** 1.6%	**1128** 15.0%	**84** 1.1%	**83.8%** 16.2%
	DOUBLE TAP	**34** 0.5%	**21** 0.3%	**26** 0.3%	**58** 0.8%	**1198** 16.0%	**89.6%** 10.4%
	% SENSIBILIDAD % ERROR	**88.8%** 11.2%	**83.7%** 16.3%	**84.1%** 15.9%	**75.2%** 24.8%	**79.9%** 20.1%	**82.32%** 17.68%

Los histogramas de la exactitud de reconocimiento para cada método se presentan en las Figuras 10, 11 y 12, respectivamente.

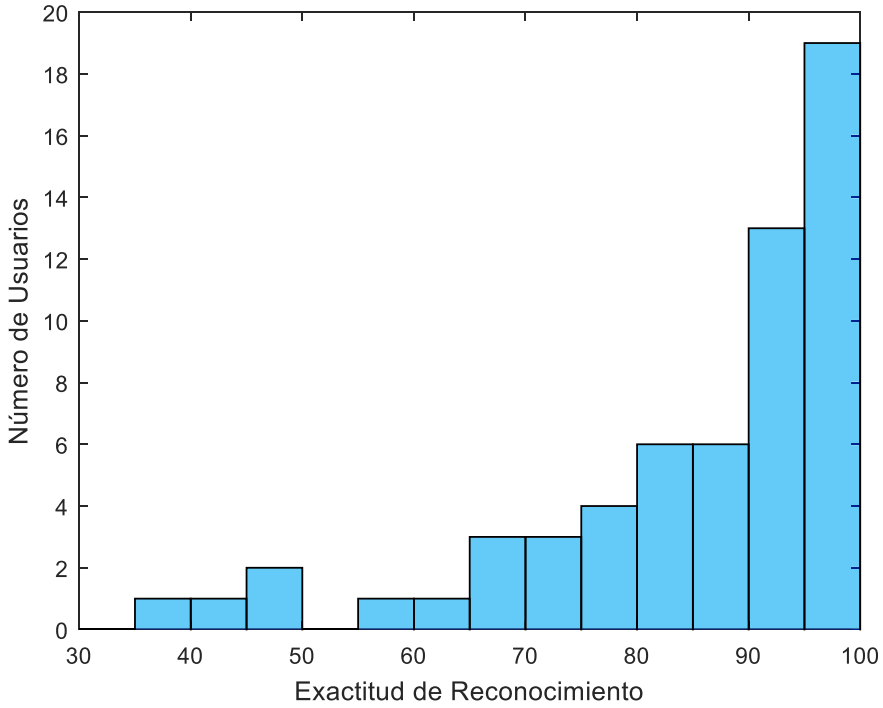

Figura 10 Histograma de las exactitudes de reconocimiento obtenidas con el método Mode, cada intervalo tiene una longitud de 5%.

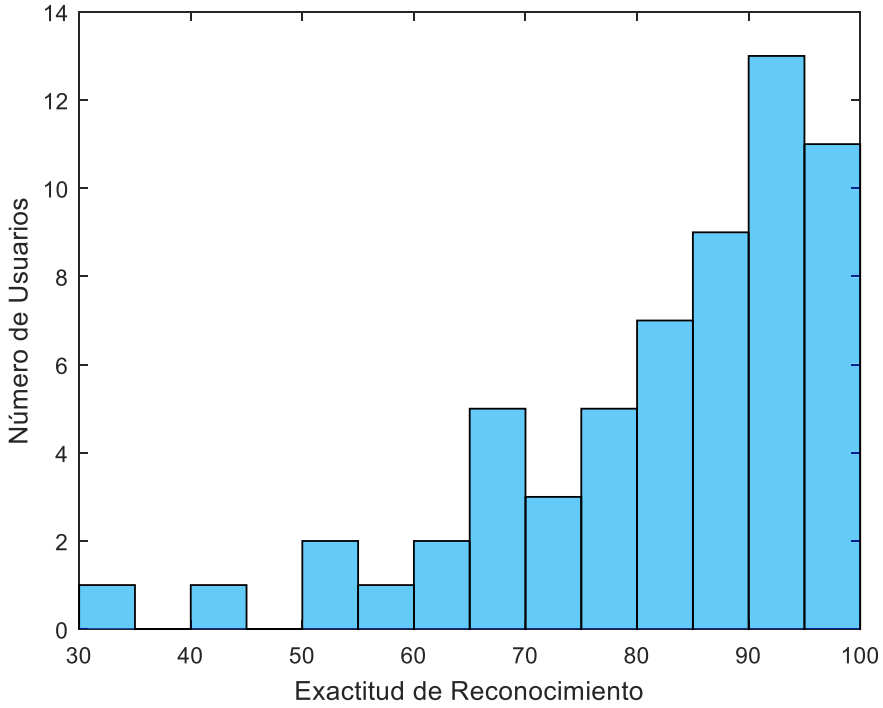

Figura 11 Histograma de las exactitudes de reconocimiento obtenidas con el método First Transition, cada intervalo tiene una longitud de 5%.

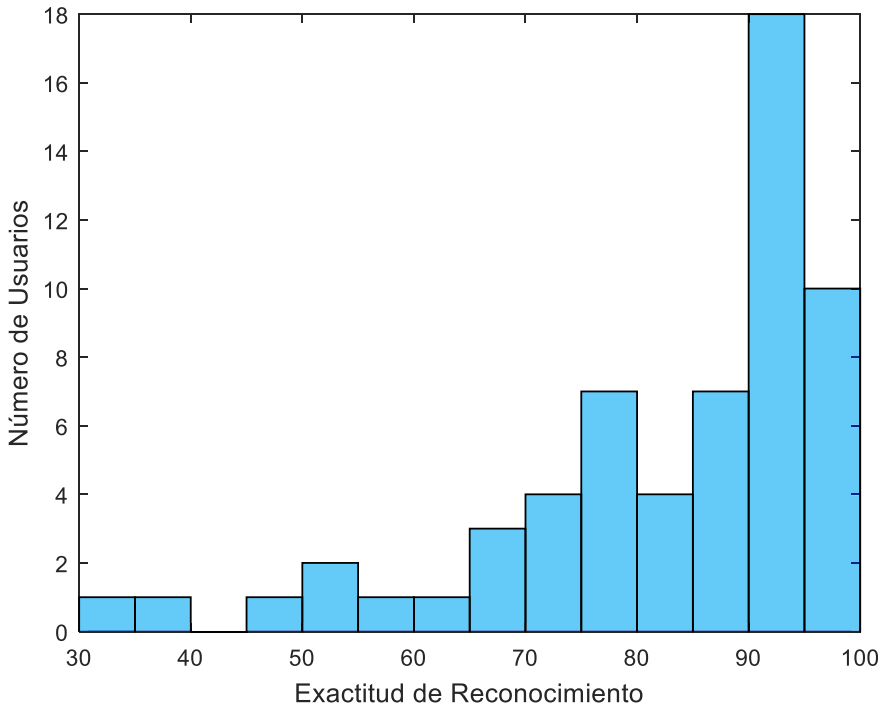

Figura 12 Histograma de las exactitudes de reconocimiento obtenidas con el método Unique, cada intervalo tiene una longitud de 5%.

Operación en tiempo real

Para evaluar la operación en tiempo real del modelo propuesto, implementamos un programa simple que utiliza una ANN ya entrenada y se conecta con el *Myo Armband*. Este programa muestra las señales EMG que están siendo procesadas y el gesto reconocido por el modelo.

Las pruebas del tiempo de procesamiento del modelo propuesto se ejecutaron en una computadora de escritorio con un procesador Intel® Core™ i7-3770S y 8GB de RAM. A partir de estas pruebas, determinamos que el tiempo requerido para procesar y clasificar una sola observación de ventana es en promedio 3 ± 1 ms. El histograma correspondiente a los resultados de estas pruebas se muestra en la Figura 13.

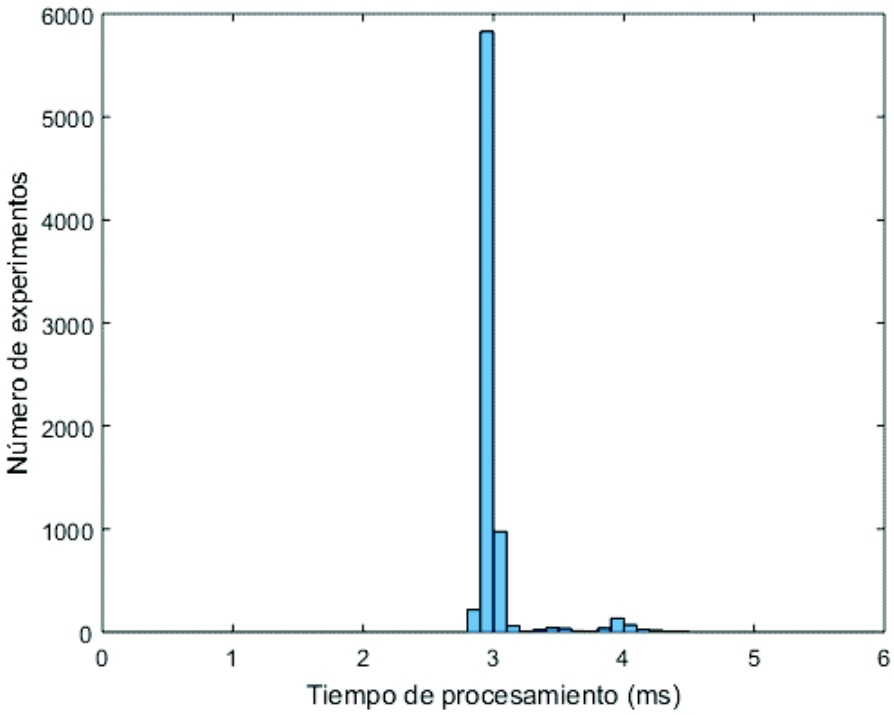

Figura 13 Histograma de las tiempos de procesamiento obtenidos.

3.2. Discusión

Exactitud de reconocimiento

Al inspeccionar las matrices de confusión, podemos notar que el modelo propuesto, con cualquier método de evaluación, falla más frecuentemente al discriminar entre los gestos *fingers spread* y *wave right*. Con el método *Mode*, el modelo propuesto no reconoce ningún gesto en 2.1% de los casos. Tanto con el método *First Transition* como con el método *Unique*, el modelo no reconoce ningún gesto en 4.2% de los casos.

La alta desviación estándar en los resultados de la evaluación del modelo propuesto puede deberse a que cada usuario produce señales EMG diferentes. El modelo puede estar mejor adaptado para reconocer los gestos de usuarios con señales EMG similares a las del conjunto de datos de entrenamiento.

Al analizar los histogramas, podemos notar que, para la mayoría de los usuarios, el modelo propuesto tiene una exactitud de reconocimiento mayor al 85%, específicamente 38 de los 60 usuarios cuando son evaluados con el método *Mode*, 33 de los 60 usuarios cuando son evaluados con el método *First Transition*, y 35 de los 60 usuarios cuando son evaluados con el método *Unique*.

Operación en tiempo real

En teoría, con una longitud de salto entre dos ventanas consecutivas de 10 valores a 200Hz, el tiempo total para dar una respuesta es de 50 ms, de los cuales el sistema solo requiere en promedio 3 ms para clasificar la observación de ventana.

Sin embargo, existe un corto retraso entre el momento en que el usuario empieza a realizar el gesto y el momento en el cual el sistema reconoce que el usuario está realizando el gesto. Esto se debe al uso de la ventana deslizante. Debe existir un cierto solapamiento entre la ventana deslizante y la señal del gesto para que el sistema pueda reconocer con exactitud el gesto. El sistema podría no reconocer correctamente el gesto hasta que el usuario lo haya completado, es decir, hasta que el usuario haya vuelto a la posición de relajación.

4. CONCLUSIONES

En este trabajo, hemos presentado un modelo de reconocimiento de gestos de la mano en tiempo real basado en técnicas de aprendizaje profundo y señales EMG adquiridas con el *Myo Armband*. Las técnicas de aprendizaje profundo son utilizadas para extracción automática de características y para clasificación.

El modelo propuesto puede reconocer 5 gestos diferentes: *fist*, *wave left*, *wave right*, *fingers spread* y *double tap*; así como una sexta clase correspondiente a la relajación o no gesto. Se trata de un modelo general, es decir, no es necesario entrenarlo para cada usuario, y puede ser utilizado por cualquier persona. El modelo propuesto funciona en tiempo real, debido a que el procesamiento y clasificación de las señales EMG toma en promedio 3 ± 1 ms, un tiempo de respuesta mucho menor que el propuesto como objetivo de 300ms. El modelo propuesto alcanza una exactitud de reconocimiento promedio de $85.08 \pm 15.21\%$ sobre un conjunto de datos de 60 usuarios.

El modelo solo puede ser utilizado con el *Myo Armband* en el antebrazo derecho y en la posición correcta con el indicador LED sobre el brazo y apuntando hacia la mano, y con su palma apuntando hacia el suelo, como se muestra en la Figura 3. Dependiendo del usuario, el modelo puede tener resultados diferentes, como lo evidencia la alta desviación estándar de la exactitud de reconocimiento.

En particular, de este trabajo podemos concluir que la extracción automática de características de las señales EMG utilizando un *autoencoder* es un enfoque viable para el reconocimiento de gestos. Asimismo, podemos concluir que las redes neuronales pueden ser utilizadas en aplicaciones de reconocimiento de gestos en tiempo real, debido al bajo tiempo que requieren para procesamiento y clasificación.

En trabajos futuros, se debe considerar el uso de redes neuronales recurrentes, debido a que pueden utilizar su estado interno para procesar una secuencia de entradas. Su estado interno les permite utilizar reconocimientos anteriores para predecir el actual con una mayor exactitud, y dentro de las mismas restricciones de tiempo.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] M. E. Benalcázar *et al*., "Real-time hand gesture recognition using the Myo armband and muscle activity detection," *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Salinas, 2017, pp. 1-6.

[2] G. Marin, F. Dominio y P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, 2014, pp. 1565-1569.

[3] A. Ganiev, H. Shin y K. Lee., "Study on Virtual Control of a Robotic Arm via a Myo Armband for the Self-Manipulation of a Hand Amputee," *International Journal of Applied Engineering Research*, 2016, vol. 11, no 2, pp. 775-782.

[4] G. Luh, H. Lin, Y. Ma y C. J. Yen, "Intuitive muscle-gesture based robot navigation control using wearable gesture armband," *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, Guangzhou, 2015, pp. 389-395.

[5] A. Norali, M. Som y J. Kangar-Arau, "Surface electromyography signal processing and application: A review," *Proceedings of International Conference on Man-Machine Systems (ICoMMS),* 2009, pp. 11-13.

[6] L. Tagliapietra, M. Vivian, M. Sartori, D. Farina y M. Reggiani, "Estimating EMG signals to drive neuromusculoskeletal models in cyclic rehabilitation movements," *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, 2015, pp. 3611-3614.

[7] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, 1991, vol. 4, pp. 251-257.

[8] S. Benatti *et al*., "A sub-10mW real-time implementation for EMG hand gesture recognition based on a multi-core biomedical SoC," *2017 7th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, Vieste, 2017, pp. 139-144.

[9] Z. Wu y X. Li, "A wireless surface EMG acquisition and gesture recognition system," *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Datong, 2016, pp. 1675-1679.

[10] J. Kim, S. Mastnik, y E. André, "EMG-based Hand Gesture Recognition for Realtime Biosignal Interfacing," *13th International Conference on Intelligent User Interfaces - IUI '08*, Islas Canarias, 2008, pp. 30-39.

[11] V. T. Gaikwad y M. M. Sardeshmukh, "Sign language recognition based on electromyography (EMG) signal using artificial neural network (ANN)," *International Journal of Industrial Electronics and Electrical Engineering*, 2014, vol. 2, no 6, pp. 73-75.

[12] E. H. Shroffe y P. Manimegalai, "Hand Gesture Recognition based on EMG Signals using ANN," *International Journal of Computer Application*, 2013, vol. 2, no 3, pp. 31-39.

[13] H. Li, X. Chen y P. Li, "Human-computer interaction system design based on surface EMG signals," *Proceedings of 2014 International Conference on Modelling, Identification & Control*, Melbourne, VIC, 2014, pp. 94-98.

[14] S. M. Mane, R. A. Kambli, F. S. Kazi y N. M. Singh, "Hand Motion Recognition from Single Channel Surface EMG Using Wavelet & Artificial Neural Network," *Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15)*, 2015, vol. 49, pp. 58-65.

[15] J. Wang, H. Ren, W. Chen y P. Zhang, "A portable artificial robotic hand controlled by EMG signal using ANN classifier," *2015 IEEE International Conference on Information and Automation*, Lijiang, 2015, pp. 2709-2714.

[16] H. Mizuno, N. Tsujiuchi y T. Koizumi, "Forearm motion discrimination technique using real-time EMG signals," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, 2011, pp. 4435-4438.

[17] A. Ng, J. Ngiam, C. Foo, Y. Mai y C. Suen, "Autoencoders and Sparsity - Ufldl", *ufldl.stanford.edu*, para. 1-2, abr. 7, 2013. [En línea]. Disponible: http://ufldl.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity. [Consultado: ago. 11, 2018].