

COMPUTACIONAL LEARNING AND IMAGING RESEARCH

Avances en el diseño de un algoritmo para la clasificación en tiempo real de actividades físicas usando un sistema embebido

Geenkel Coss Lara, Mildred González Hau, Carlos Brito Loeza y Anabel Martin González

November 2016.

To cite this version: Geenkel Coss Lara, Mildred González Hau, Carlos Brito Loeza y Anabel Martin González, 16° Congreso Nacional de Ingeniería Eléctrica, Electrónica y Biomédica del Mayab, CONIEEM 2016, ISSN: 1665-0271.

ABSTRACT

This paper describes the design of an algorithm and its realization on an embedded device for recognizing physical activities in real time such as running, walking, jumping, doing push-ups and doing sit-ups. The device has an accelerometer and a gyroscope, it is worn on the user's wrist and the sensors retrieve information about the user's movement. The collected data is used as input for a classifier algorithm which recognises the activities being done by the user. The classifier algorithms tested in this work are K nearest neighbour and Artificial Neuronal Networks. This device is composed by a microcontroller MSP430, an inertial measurement unit, a Micro SD slot, push buttons and LEDs.

AVANCES EN EL DISEÑO DE UN ALGORITMO PARA LA CLASIFICACIÓN EN TIEMPO REAL DE ACTIVIDADES FÍSICAS USANDO UN SISTEMA EMBEBIDO

Geenkel Coss Lara, Mildred González Hau, Carlos Brito Loeza y Anabel Martin González

Universidad Autónoma de Yucatán, Facultad de Matemáticas

Laboratorio de Aprendizaje Automático y Visión Computacional, www.clir-lab.org

mcoss002, miltire@gmail.com, carlos.brito, amarting@correo.uady.mx

RESUMEN

En este trabajo se describe el diseño de un algoritmo y su implementación en un dispositivo embebido para el reconocimiento en tiempo real de actividades físicas como correr, saltar, caminar, realizar lagartijas o abdominales. El dispositivo cuenta con un acelerómetro y un giroscopio, se coloca en la muñeca del usuario y con los sensores se obtiene información acerca del movimiento del individuo. Esta información sirve como entrada para un algoritmo clasificador que determina la actividad que se está realizando. Los algoritmos clasificadores que se probaron en este trabajo son K Vecinos más Cercanos y Redes Neuronales Artificiales. El dispositivo consiste principalmente en un microcontrolador MSP430, una unidad de medición inercial, una ranura para tarjetas Micro SD, botones y diodos leds.

ABSTRACT

This paper describes the design of an algorithm and its realization on an embedded device for recognizing physical activities in real time such as running, walking, jumping, doing push-ups and doing sit-ups. The device has an accelerometer and a gyroscope, it is worn on the user's wrist and the sensors retrieve information about the user's movement. The collected data is used as input for a classifier algorithm which recognises the activities being done by the user. The classifier algorithms tested in this work are K nearest neighbour and Artificial Neuronal Networks. This device is composed by a microcontroller MSP430, an inertial measurement unit, a Micro SD slot, push buttons and LEDs.

1. INTRODUCCIÓN

En años recientes se ha incrementado la importancia del ejercicio en la vida diaria de las personas, ya sea como una forma de entretenimiento o para combatir o prevenir enfermedades como la obesidad y problemas cardíacos entre otros. Conocer que actividades físicas se realizan durante el día, durante cuánto tiempo se realiza cada actividad y la intensidad con la que se realiza ayuda a tener una mejor estimación del gasto energético. Con esta información, se puede recomendar un entrenamiento personalizado eficiente o una dieta alimenticia apropiada con el consecuente beneficio de una mejor salud y por lo tanto mejor calidad de vida. En este trabajo se propone un dispositivo portable capaz de reconocer qué actividad física realiza una persona.

En la literatura podemos encontrar una gran cantidad de trabajos en los que se aborda el tema del reconocimiento de actividades, sin embargo, son pocos en los que la clasificación de la actividad se realiza en un dispositivo embebido, ya que, por lo general, esta tarea requiere de poder

computacional. En la tabla 1 tomada del artículo [6] se encuentra una lista resumida de los tipos de actividades que han sido clasificadas.

Tabla I

Grupo	Actividades
Ambulación	Cambiar, correr, permanecer sentado o de pie, subir o bajar escaleras.
Transporte	Usar el autobús, montar bicicleta, manejar.
Vida diaria	Comer, beber, ver TV, lavarse los dientes, usar la PC, leer un libro.
Ejercicio	Abdominales, lagartijas, spinning.
Social	Platicar.

Tabla I. Tipos de actividades que se han detectado [Lara y Labrador, 2013]

Para realizar el reconocimiento automático de actividades es necesario obtener primero, información acerca de los movimiento o gestos que realice el usuario. Para esto se han utilizado diferentes sensores, siendo los más comunes los sensores inerciales, en especial los acelerómetros, ya que se ha visto que son los que proporcionan información más relevante. Además de estos sensores, también se han utilizado micrófonos para reconocer actividades como platicar o ver la televisión [9], sensores GPS [8] para actividades como conducir un auto o montar en bicicleta, sensores de temperatura, humedad y luz ambiental [9] para detectar si una actividad se realiza de día o de noche. También se han utilizado cámaras de video [2], sin embargo, éstas tienen la desventaja de que incomodan al usuario al invadir su privacidad.

En la mayoría de los trabajos de la literatura se han utilizado técnicas de aprendizaje supervisado para realizar la clasificación de las actividades ya que generalmente es fácil obtener muestras etiquetadas y se obtienen mejores resultados. Entre las técnicas utilizadas están árboles de decisión, clasificadores Bayesianos, métodos basados en instancias, redes neuronales, máquinas de soporte vectorial, lógica difusa, métodos de regresión o combinaciones de varios métodos.

Para realizar la detección de actividades en tiempo real se han utilizado computadoras portátiles [3], teléfonos celulares [4] y dispositivos *wearables* [9][7]. Los dispositivos *wearables* o vestibles son aquellos dispositivos electrónicos que una persona puede llevar encima, en alguna parte de su cuerpo o la ropa, interactuando continuamente con él, sin afectar sus actividades normales. En general, estos dispositivos se caracterizan por ser pequeños, tener una capacidad computacional limitada y además funcionan con batería. De manera comercial, ya se ha realizado el reconocimiento de actividades físicas utilizando este tipo de dispositivos. Por ejemplo, un teléfono de la compañía Samsung puede detectar las actividades de caminar, correr y montar bicicleta. Por otra parte, en la literatura encontramos dos trabajos que han realizado la clasificación en dispositivos *wearables*. Sus características se resumen en la tabla II.

Tabla II

	ewatch [Maurer et al, 2006]	Mobile Sensig Plataform (MSP) [Choudhury et al,2008]
Procesador	Philips LPC2106 a 60Mhz.	Intel PXA271 de 32-bit en tarjeta iMote y Atmega128 de 8-bits en tarjeta de sensores.
Medidas	50mm x 48mm x 17.5mm.	51mm x 36mm.
Batería	Capacidad de 700mAh y duración de 1 día.	Capacidad de 1800 mAh y duración de 10 horas.
Sensores	Acelerómetro de 2 ejes colocado en la muñeca.	Acelerómetro de 3 ejes colocado en la cintura.
Actividades	Caminar, correr, estar sentado o de pie, subir o bajar escaleras.	Caminar, estar sentado o de pie, subir o bajar escaleras, subir o bajar en elevador, cepillarse los dientes.
Precisión del reconocimiento	87.10%	93.80%

Tabla II Comparación entre dispositivos wearables encontrados en la literatura.

2. DISEÑO DE HARDWARE

Un dispositivo *wearable* normalmente se caracteriza por ser de tamaño pequeño, no deben ser estorbosos y generalmente funcionan con batería. Por esta razón, para el diseño se seleccionaron componentes pequeños y de bajo consumo de energía bajo para que este pueda funcionar de manera prolongada sin necesidad de recargar la batería.

El diseño se puede dividir en 6 módulos: Microcontrolador, Sensor inercial, Memoria F-RAM, Memoria Micro SD, Botones y diodos LED, Batería. En la figura 1 se ilustran los módulos que componen el sistema. El microcontrolador controla todos los demás módulos del sistema.

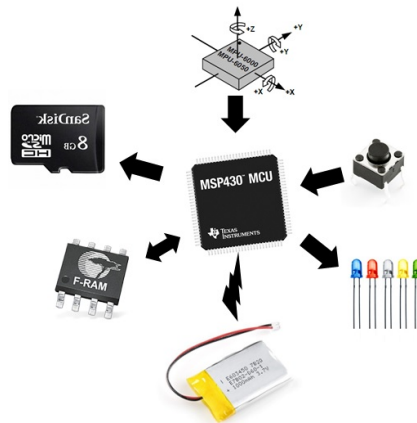


Figura 1. Módulos del dispositivo de detección de actividades

2.1 Microcontrolador

El microcontrolador que se añadió al sistema fue el MSP430F5529 de *Texas Instruments*, principalmente debido a que tiene un consumo muy bajo de energía, lo que permite una mayor autonomía del dispositivo. El MSP430F5529 es un microcontrolador de 16 bits y tiene una velocidad de hasta 25Mhz. Una de sus principales características es que permite configurar de manera flexible y en tiempo de ejecución la velocidad de las fuentes del reloj para el procesador y los periféricos y además éste puede operar en varios modos de energía. Con estas características se puede obtener un rendimiento óptimo según la tarea que se realice. El microcontrolador puede configurarse también para operar a baja velocidad para ahorrar energía o de manera análoga, puede configurarse para funcionar rápido en tareas más demandantes. El MSP430 cuenta con los módulos de comunicación SPI e I2C necesarios para controlar el sensor, la memoria FRAM y la memoria Micro SD.

2.2 Sensor inercial

El sensor MPU-9150 incluido en el sistema, es una unidad de medición inercial, la cual cuenta con un acelerómetro, un giroscopio y un magnetómetro integrados en el mismo chip. Todos ellos pueden funcionar en los 3 ejes x, y y z. Para este proyecto solo se utilizarán los sensores acelerómetro y giroscopio dejando para pruebas futuras incluir información del sensor magnetómetro. Las principales características de los sensores acelerómetro y giroscopio se enlistan a continuación.

- Acelerómetro
 - Ejes x, y y z.
 - $\pm 16g$ de rango a escala completa.
 - ADC de 16 bits.
 - Detección de toques.

- Giroscopio
 - Ejes x, y y z.
 - $\pm 2000^\circ/\text{sec}$ de rango a escala completa.
 - ADC de 16 bits.
 - Filtro digital programable.

2.3 Memoria F-RAM

La memoria F-RAM FM24VN10 de Cypress es una memoria no volátil que se caracteriza por tener una velocidad de escritura muy rápida a diferencia de otras memorias no volátiles como las EEPROM y las memorias flash. La F-RAM se añadió para almacenar en primera instancia las muestras provenientes de los sensores en formato binario. El microcontrolador no necesita esperar a que se escriban los datos en la memoria ya que la escritura ocurre de manera instantánea. La memoria utiliza el protocolo I2C para comunicarse con el microcontrolador a una velocidad de hasta 3.4 Mhz. Además, consume solamente 175uA mientras escribe y 5uA mientras está en reposo.

2.4 Memoria Micro SD

Para transferir los datos obtenidos de los sensores a la computadora, se añadió al sistema una ranura para tarjetas Micro SD. Las bibliotecas existentes para controlar memorias micro SD para el microcontrolador MSP430 facilitan la programación del almacenamiento de los datos en ésta. La memoria Micro SD se añadió para almacenar en un archivo de texto los datos provenientes de los sensores que se almacenaron primero en la memoria F-RAM.

2.5 Botones y diodos led.

Se añadieron al dispositivo botones y diodos led para interactuar con el usuario de una manera simple. En total, se añadieron 4 botones y 3 diodos LED. Dos de los botones se utilizan para indicarle al dispositivo a qué actividad pertenecen los datos que se van a recolectar para posteriormente indicar en el archivo de texto almacenada en la Micro SD el número de actividad a la que corresponden los datos. Un tercer botón sirve para iniciar o suspender la recolección de datos, y el último botón se utiliza para reiniciar el funcionamiento de la tarjeta. Los LED indican al usuario la actividad seleccionada en formato binario, es decir, un LED encendido indicará un uno, y apagado indicará cero.

2.6 Batería

Para alimentar al dispositivo se utiliza una batería Li-Po. Este tipo de baterías se caracterizan por ser ligeras y pueden fabricarse con casi cualquier forma o tamaño, lo que las hace ideales para dispositivos *wearables*. La batería seleccionada tiene 420mAh de capacidad.

2.7 Diseño del PCB

El diseño del PCB del dispositivo se realizó en el programa CadSoft EAGLE. En la figura 2 se encuentra el esquemático del circuito electrónico.

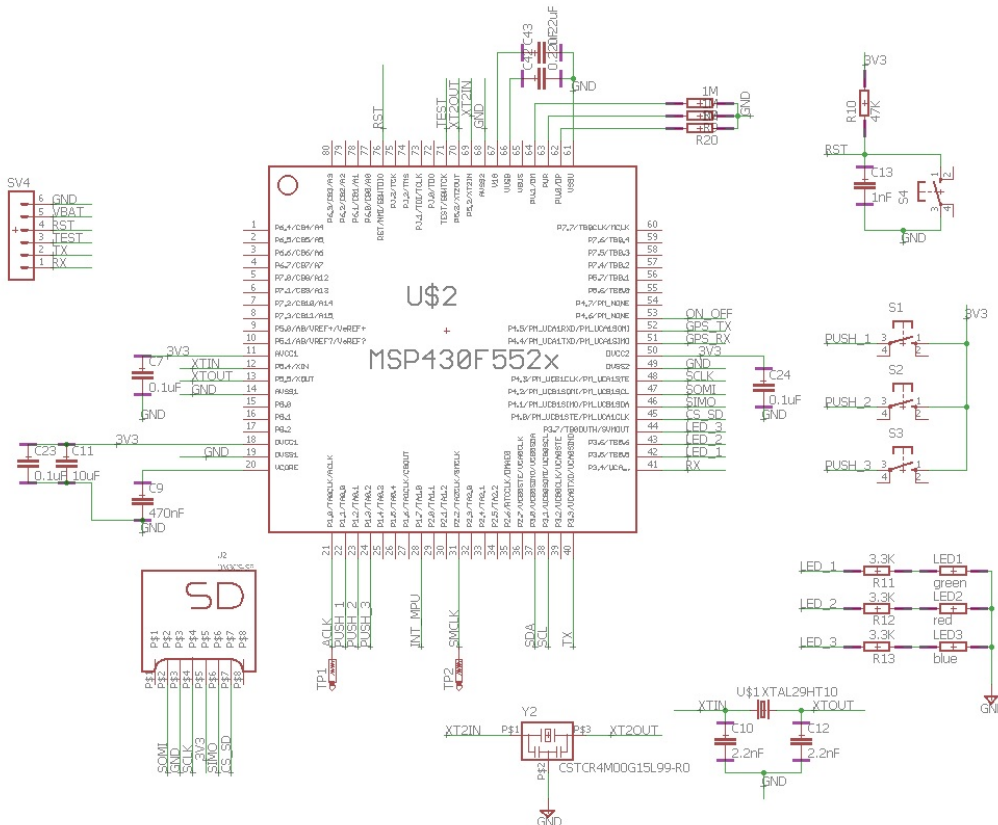


Figura 2. Esquemático del dispositivo de detección de actividades.

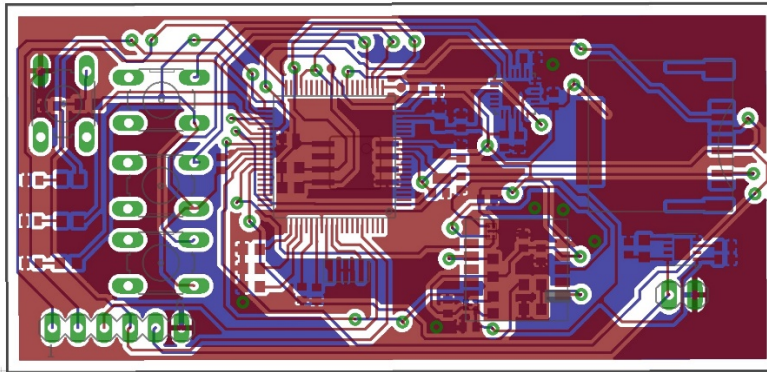


Figura 3. Diseño del PCB.

En la figura 4 se puede observar la tarjeta fabricada. Esta mide 7.5cm por 3.8cm y tiene un consumo de corriente de 11mA aproximadamente, medido con el microcontrolador adquiriendo muestras de los sensores a 5Hz.

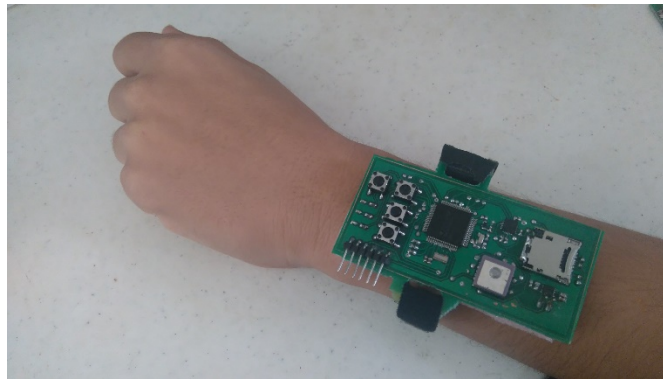


Figura 4 Dispositivo de detección de actividades.

3. CLASIFICACIÓN DE ACTIVIDADES

Para realizar la detección de actividades se utilizó aprendizaje automático, en específico, se usaron dos técnicas de aprendizaje supervisado: k vecinos más cercanos, también conocido como K-NN por sus siglas en inglés y redes neuronales artificiales. Las dos técnicas consisten en algoritmos clasificadores que identifican la clase a la que pertenece una muestra de entrada utilizando conocimiento previamente adquirido mediante una fase de entrenamiento. Estos dos algoritmos se entrenaron para clasificar 6 clases: caminar, correr, estar en reposo, saltar, realizar abdominales y realizar lagartijas.

3.1 Conjunto de entrenamiento

K-NN y redes neuronales son técnicas de aprendizaje supervisado, es decir, necesitan de ejemplos de entrenamiento etiquetados con la clase a la que pertenecen. Los ejemplos de entrenamiento están formados por un conjunto de características obtenidas de los datos de los sensores. A la lista de vectores de características etiquetados con la clase a la que pertenecen se le conoce como conjunto de entrenamiento. La clasificación está basada en el conocimiento adquirido a través de los ejemplos en el conjunto de entrenamiento.

Los vectores de características contienen información proveniente de los sensores. Debido a que la aceleración y la velocidad angular medida en un instante de tiempo proporcionan muy poca información acerca de alguna actividad física, se utilizan características con modos estadísticos de secuencias de estas mediciones. Por ejemplo, la media o la varianza de varias muestras consecutivas. El vector de características seleccionado está conformado de 16 elementos:

1. La aceleración en el eje x.
2. La aceleración en el eje y.
3. La aceleración en el eje z.
4. La velocidad angular en el eje x.
5. La velocidad angular en el eje y.
6. La velocidad angular en el eje z.
7. El módulo del vector de aceleración.
8. El módulo del vector de la velocidad angular.

Las siguientes 8 características se obtienen a partir de un número N de muestras consecutivas:

9. El máximo del módulo de aceleración.
10. El máximo del módulo de la velocidad angular.
11. El mínimo del módulo de aceleración.
12. El mínimo del módulo de la velocidad angular.
13. La media del módulo de aceleración.
14. La media del módulo de velocidad angular.
15. La varianza del módulo de aceleración.
16. La varianza del módulo de la velocidad angular.

Para formar el conjunto de entrenamiento se programa la tarjeta para obtener muestras de los sensores, almacenarlos en la F-RAM y por último escribir éstas en un archivo de texto en la Micro SD. En la figura 5 se encuentra el diagrama de flujo del programa. Como primer paso el microcontrolador configura los puertos, e inicializa las configuraciones para controlar la Micro SD, además de escribir las configuraciones para el funcionamiento del MPU-9150 a 5Hz de frecuencia de muestreo. Después, el microcontrolador queda a la espera de que el usuario por medio de 2 botones seleccione el número de actividad de la cual se recolectarán datos. Los diodos LED despliegan en formato binario el número. Al presionar el botón de inicio, el microcontrolador comienza a obtener datos y después de leer cada muestra, la almacena en la memoria F-RAM en formato binario. En este momento, el microcontrolador entra en un estado de bajo consumo para ahorrar energía hasta que la siguiente muestra esté lista para ser leída. Una vez que se hayan obtenido un número N de muestras, el microcontrolador crea un archivo de texto en la Micro SD en donde escribe en cada línea la lista de valores. El primer carácter de la línea es el número seleccionado a través de los botones e indica la actividad que se estaba realizando al obtener los datos.

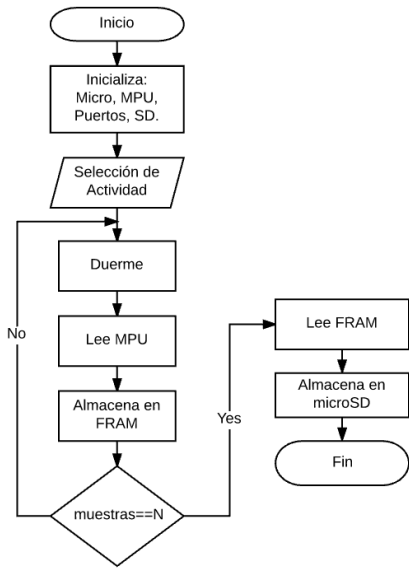


Figura 5. Diagrama de flujo del programa.

Después de obtener un archivo de texto en la Micro SD con los datos de los sensores, éste se transfiere a la computadora en donde un programa en MATLAB toma como parámetros el archivo de texto y el número N de muestras para formar los vectores de características. El programa almacena en otro archivo los vectores de características etiquetados.

En los experimentos, se colocó la tarjeta en la muñeca izquierda de un voluntario, del cual se obtuvieron muestras de los sensores de aceleración y giroscopio mientras realizaba las 6 actividades. De las muestras obtenidas, se construyeron un total de 1200 vectores de características etiquetados que sirven como conjunto de entrenamiento para los algoritmos de clasificación.

3.2 K vecinos más cercanos.

El algoritmo K-NN clasifica una nueva muestra en la clase más frecuente a la que pertenecen sus K vecinos más cercanos. En el caso más simple, cuando K=1, la muestra se clasifica en la clase a la que pertenece el vecino más cercano. La función de distancia utilizada en este trabajo fue la de distancia euclidiana dada por la siguiente ecuación:

$$d(P, Q) = \sqrt{\sum (p_i - q_i)^2} \quad (1)$$

En donde P es la muestra sin clasificar, Q es la muestra ejemplo, p_i y q_i son las características de P y Q.

Para medir el rendimiento se clasificó cada vector de características en la base de datos de 1200 vectores utilizando los restantes como conjunto de entrenamiento y se contabilizaron los vectores clasificados correctamente. El rendimiento se define como el número de muestras clasificadas correctamente entre el total de muestras clasificadas.

Para hallar el valor óptimo de K se calificaron las 1200 muestras varias veces incrementando el valor de K. En la gráfica de la figura 6 se encuentran los resultados obtenidos, de los que se concluye que el valor de K no afecta significativamente el rendimiento, por lo tanto, el valor de K=1 puede seleccionarse como el mejor ya que la complejidad es menor.

Seleccionando $K=1$ se realizaron varias pruebas variando el tamaño de ventana dado por el valor N de muestras consecutivas. Los resultados se encuentran en la gráfica de la figura 7. En la gráfica se observa que en general, al incrementar el tamaño de ventana se incrementa el rendimiento de la clasificación, sin embargo, incrementa también la complejidad al momento de clasificar.

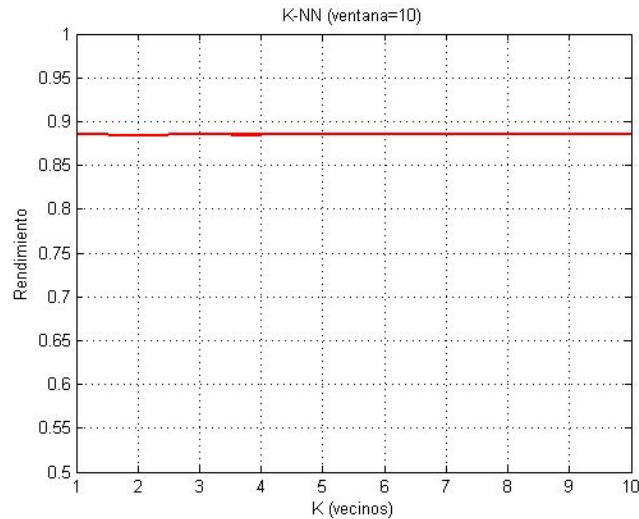


Figura 6. Rendimiento del algoritmo K-NN variando K.

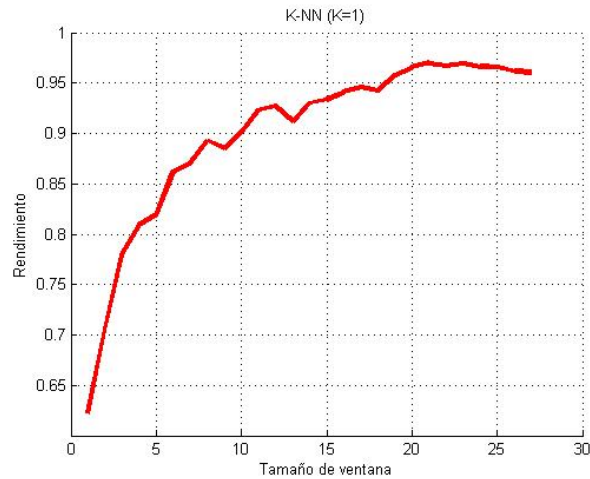


Figura 7. Rendimiento del algoritmo K-NN variando el tamaño de ventana.

3.3 Redes neuronales artificiales.

Al igual que con el algoritmo K-NN, también se hicieron pruebas utilizando redes neuronales artificiales. Se utilizó una topología sin ciclos de propagación hacia adelante utilizando una capa de neuronas de salida y una o varias capas de neuronas ocultas. Las redes probadas fueron entrenadas utilizando el método de propagación de error hacia atrás. Para encontrar la topología óptima se realizaron varias pruebas fijando los siguientes parámetros: se tomó una ventana de tamaño 10, se entrenó a la red 500 épocas utilizando 200 ejemplos por cada actividad, se utilizó validación cruzada con 5 particiones y se utilizó la función tangente hiperbólica como función de activación de las neuronas.

Las topologías probadas y el rendimiento obtenido en cada una se encuentran en la tabla III.

Tabla III

Topología de red	Rendimiento
1-6	0.4675
2-6	0.6122
3-6	0.8143
4-6	0.9174
5-6	0.9542
8-6	0.9686
10-6	0.9706
3-2-6	0.6703
5-2-6	0.6696
6-6-6	0.972
10-7-6	0.9849
10-10-6	0.9767

Tabla III. Comparación de topologías de red.

De la tabla III se concluyó que la mejor topología es la que está formada por dos capas ocultas de tamaño 3 y 2, la cual tuvo un rendimiento de 0.95. El incremento en el rendimiento en las topologías con más neuronas no es suficiente para compensar el incremento en la complejidad de la clasificación. Se midió el rendimiento de la topología seleccionada variando el tamaño de ventana con el que se construyen los vectores de características. Los resultados obtenidos se grafican en la figura 8, en la que se puede observar que a partir de una ventana de tamaño 5 ya se logra un rendimiento mayor 0.9.

Figura

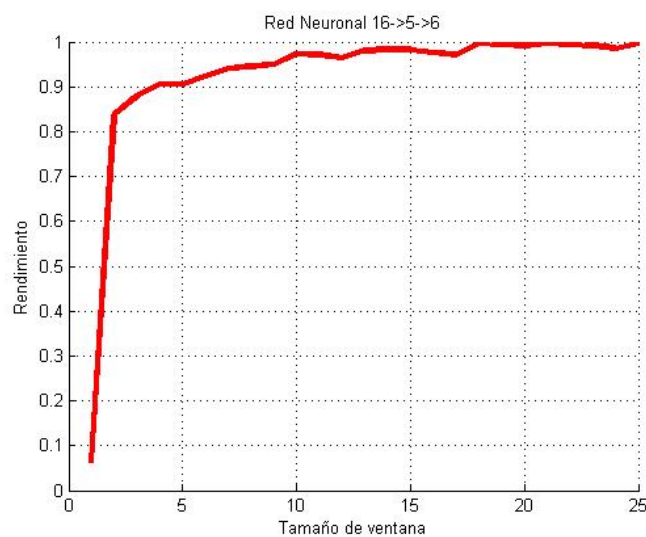


Figura 8. Rendimiento de la Red Neuronal variando el tamaño de ventana.

4. CONCLUSIONES.

De los resultados obtenidos de las pruebas con los algoritmos K-NN y redes neuronales muestran evidencia que apunta a que los datos de aceleración y del giroscopio tomados del sensor inercial colocado en la muñeca proporcionan suficiente información acerca de la actividad física que se esté realizando. Los porcentajes de rendimiento obtenidos con el vector de características elegido y los algoritmos clasificadores son aceptables para la aplicación considerada. Es importante remarcar que dichos vectores y clasificadores de baja complejidad fueron elegidos dadas las restricciones de poder computacional y consumo energético del dispositivo.

En la tabla IV se comparan los rendimientos medidos de los dos métodos para cada una de las 6 actividades y se puede observar que la Red Neuronal tiene un mayor rendimiento que el algoritmo K-NN, además que la actividad de lagartijas fue la más difícil de clasificar para los dos algoritmos.

Tabla IV

Actividad	Red Neuronal	K-NN
Reposo	1	1
Caminar	1	0.89
Correr	1	0.81
Saltar	1	0.83
Abdominales	0.71	0.65
Lagartijas	1	0.97

Tabla IV. Comparación de los métodos probados.

El estudio presentado utiliza sin embargo pocos datos de entrenamiento por lo que el trabajo futuro inmediato es incrementar el tamaño de la base de datos de entrenamiento y validar con muchos individuos. A corto plazo, se efectuarán pruebas con vectores de características más complejos y se estudiará en detalle el algoritmo K-NN ya que con este es posible reentrenar en tiempo de ejecución.

5. REFERENCIAS

1. N. Bicocchi, M. Mamei, and F. Zambonelli, "Detecting activities from body-worn accelerometers via instance-based algorithms," *Pervasive and Mobile Computing*, vol. 6, no. 4, pp. 482–495, (2010).
2. C. N. Joseph, S. Kokulakumaran, K. Srijevanthyan, A. Thusyanthan, C. Gunasekara, and C. Gamage, "A framework for whole-body gesture recognition from video feeds," in *International Conference on Industrial and Information Systems (ICIIS)*, pp. 430–435, (2010).
3. L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5250–5253, (2008).
4. T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and*

Ambient Assisted Living, vol. 5518 of Lecture Notes in Computer Science, pp. 796–799, Springer Berlin / Heidelberg, (2009).

5. C.-C. Yang and Y.-L. Hsu, “A review of accelerometry-based wearable motion detectors for physical activity monitoring.” Sensors (Basel, Switzerland), vol. **10**, pp. 7772–88, (2010).

6. O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Communications Surveys Tutorials, vol. **15**, pp. 1192–1209, (2013).

7. U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, “Activity recognition and monitoring using multiple sensors on different body positions,” in International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06), pp. 4 pp.–116, (2006).

[8] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, “Using mobile phones to determine transportation modes,” ACM Trans. Sen. Netw., vol. 6, pp. 13:1– 13:27, Mar. 2010.

[9] Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., ... Wyatt, D. (2008). The mobile sensing platform: An embedded activity recognition system. IEEE Pervasive Computing, 7(2), 32-41. [4487086]. DOI: