

Implementación de una Red Neuronal Multicapa en un Microcontrolador PIC para la Identificación de Color

¹Granados-Rojas Benito, ^{1*}Jiménez-Saucedo Mario A., ¹Vallejo-Alarcón Manuel A.,
¹González-Navarro Yesenia E., ²Villarreal-Cervantes Miguel G. y ¹Corona-Ramírez Leonel G.

¹Departamento de Tecnologías Avanzadas, UPIITA-IPN.
Av. Instituto Politécnico Nacional 2580, C.P. 07340, México, D.F.
²Sección de Mecatrónica, Departamento de Posgrado, CIDETEC-IPN.
Juan de Dios Bátiz s/n, C.P. 07700, México, D.F.
*Correo electrónico de contacto: rollover_bethoven@hotmail.com

Resumen

En este trabajo se presenta la implementación de una red neuronal multicapa con entrenamiento fuera de línea para el reconocimiento de cinco colores en objetos. Se implementa el modelo elemento lineal adaptativo (ADALINE en inglés) de red neuronal con cinco entradas, dos capas ocultas y cinco salidas.

La identificación de los pesos óptimos de la red neuronal se realiza fuera de línea y una vez obteniéndose dichos pesos, se posibilita a la red neuronal para la identificación de los colores en línea.

El sistema electrónico del sensor de color consiste de tres diodos emisores de luz (rojo, verde y azul), un convertidor de intensidad luminosa a voltaje y un microcontrolador en donde se programa la red neuronal. Resultados experimentales muestran el desempeño de la red neuronal implementado en un microcontrolador.

Palabras clave: ADC, algoritmo de entrenamiento, microcontrolador PIC, RGB, Red neuronal, sensor de color.

1. Introducción

En la actualidad la identificación y clasificación de colores da pie a varias aplicaciones industriales como pueden ser diversos sistemas autónomos para clasificación de etiquetas, clasificación de materia prima, control de calidad de

telas o tintes entre otras, que resultan de gran utilidad en industrias como la textil, plásticos, alimentos, etc.

Un sistema autónomo de clasificación requiere de la obtención de una señal de entrada que represente las características que se están evaluando (en este caso, las componentes luminosas de rojo, verde y azul de un objeto); de un dispositivo que transforme esa señal de entrada en una señal eléctrica, y un sistema de procesamiento que transforme dicha señal eléctrica en una señal de datos o control dirigida a los actuadores o subsistemas que sean pertinentes de acuerdo a la aplicación final del sistema.

Cabe mencionar que cuando se trata de clasificación de color, la señal de entrada puede tener variaciones y verse afectada directamente por factores externos como la intensidad de luz, haciendo que una exacta clasificación se torne complicada, sin embargo, las redes neuronales artificiales presentan características convenientes para este problema como son su flexibilidad y su tolerancia a variaciones.

Una red neuronal artificial es un modelo computacional inspirado en sistemas biológicos que consiste en elementos de procesamiento (neuronas) y conexiones entre ellos, así como de algoritmos iterativos de entrenamiento [1]. Las redes neuronales se han utilizado en el reconocimiento del iris del ojo humano [2] obteniendo una exactitud de 93.33% en iris del mismo color. En [3], las redes neuronales se han utilizado para la identificación de minerales con base en su color.

En este trabajo se propone la solución al problema de clasificación de colores mediante un sistema de bajo costo conformado por un sensor fotorreflexivo y un microcontrolador que se valen de

una red neuronal artificial multicapa para la clasificación de color.

El entrenamiento de la red se llevó a cabo fuera de línea en MATLAB mediante el algoritmo de retro-propagación y una vez que la red neuronal fue entrenada, se implementó en un microcontrolador PIC para que este evaluara la respuesta del sensor con respecto a la red neuronal, determinando así el color presente.

2. Red Neuronal

Como elemento unitario de la red se utiliza la neurona de tipo ADALINE (Adaptative Linear Element) [4] con función de activación "pureline" la cual da como resultado 1 cuando la activación es positiva y -1 cuando la activación es negativa. En la Figura 1 se muestra la representación de una neurona tipo ADALINE.

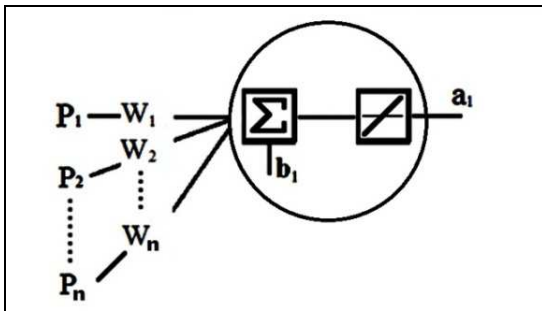


Fig. 1. Estructura de una neurona tipo adaline.

Esta neurona se caracteriza por la siguiente ecuación:

$$a = \hat{w}^T \hat{p} + b \quad (1)$$

Donde $a \in R$, es el valor de la salida de la neurona, $\hat{w} = [w_1, w_2, \dots, w_n]^T \in R^n$ es el vector de los pesos de la neurona asociados al vector de entrada $\hat{p} = [p_1, p_2, \dots, p_n]^T \in R^n$ y $b \in R$ es el valor de polarización.

El problema de clasificación en este caso involucra conjuntos de datos obtenidos de la medición de color cuya distribución no es lineal y debido a que una red neuronal sin capas ocultas o intermedias sólo puede clasificar conjuntos linealmente separables, se utiliza una arquitectura multicapa.

La red neuronal artificial que se muestra en la Figura 2 queda entonces conformada por una red de 4 capas con 5 neuronas en cada una con conexiones

completas, es decir todas las entradas se conectan a todas las neuronas dentro de una misma capa.

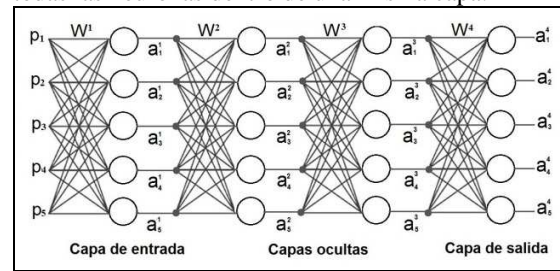


Fig. 2. Esquema de la red neuronal multicapa. Los superíndices denotan el número de capa y los subíndices el número de elemento.

2.1 Capa de entrada

Los objetos a identificar el color consisten de cubos de madera pintados de diferentes colores (cian, rojo, amarillo, verde y magenta) cada uno con acabado mate.

Para el problema de identificación de cinco colores en un objeto mediante una red neuronal se define como vector de entrada al conjunto de luz reflejada en el objeto a detectar. Para el caso particular, se define al vector de entrada como a la cantidad de luz reflejada con la luz ambiente, luz blanca, luz roja, luz verde y luz azul, es decir, $\hat{p} = [p_1, \dots, p_5]^T = [\text{luz ambiente, luz blanca, luz roja, luz verde, luz azul}]^T$.

El sistema de detección de colores implementa un sensor fotorreflexivo en formato rojo-verde y azul (RGB en inglés). Este sistema se elige por ser pequeño y más barato que una cámara o algún otro dispositivo de visión artificial. El formato de colores RGB incorpora tres colores primarios, el Rojo, el Verde y el Azul. Combinaciones de los colores primarios dan como resultado los colores secundarios, como por ejemplo: amarillo, magenta y cian los cuales son combinaciones de los colores primarios rojo-verde, rojo-azul y verde-azul, respectivamente.

El sensor a utilizar es el ColorPal [5] del fabricante PARALLAX el cual usa tres diodos emisores de luz roja, verde y azul, respectivamente (LED RGB) para iluminar el objeto a detectar el color. La iluminación que provee el sensor puede ser tanto de colores primarios como secundarios y mediante un convertidor de flujo luminoso a voltaje (fototransistor) se mide la luz reflejada por el objeto hacia el sensor. La cantidad de luz reflejada por el objeto a detectar el color bajo la iluminación del LED rojo, verde y azul es usada para determinar su color. El sensor de luz utilizado en el ColorPal es un TAOS

TSL13T que tiene la curva de sensibilidad espectral mostrada en la Figura 3.

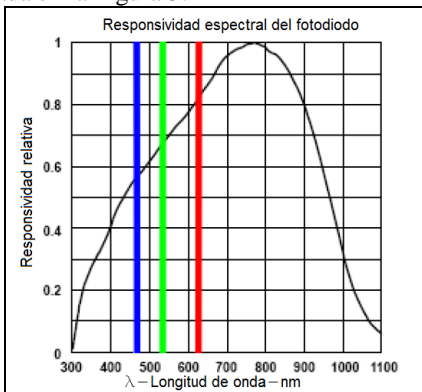


Fig. 3. Curva de respuesta espectral del sensor ColorPal.

En el punto de salida del sensor se obtiene una tensión proporcional a la luz que toma, ponderada por la curva de la figura 3 de tal manera que cuando un objeto es iluminado con el LED en rojo solamente, el sensor responderá con un voltaje proporcional a la componente en rojo del color del objeto. De manera análoga ocurre con el verde y el azul.

Puesto que el sensor ColorPal está hecho para interactuar con otros productos de la misma marca como lo son tarjetas de desarrollo y tarjetas de entrenamiento, el sensor viene integrado junto con un microcontrolador AVR ATtiny13A del fabricante Atmel. Este dispositivo tiene precargado un programa con las instrucciones y el protocolo de comunicación necesario (comunicación serial de una sola línea) para interconectar al sensor con una PC mediante dichas tarjetas y su respectivo software, así, el usuario ejecutará comandos predefinidos para generación y detección de colores. Esta arquitectura del sensor implica una menor velocidad de adquisición de datos, razón por la cual se modificó el sensor retirando el AVR y conectando a la salida del sensor un microcontrolador PIC18F2550 para el libre manejo de la información adquirida por el sensor y una mayor velocidad de adquisición de datos.

En el PIC, la salida del sensor pasa por un convertidor analógico digital (ADC Analog to Digital Converter por sus siglas en inglés) interno de 10 bits de resolución. Las lecturas obtenidas por el convertidor son agrupadas para posteriormente ingresarlas al sistema clasificador como vector de entrada a la red neuronal que se encuentra cargada en el mismo PIC.

2.2 Capa de salida

La salida de la red neuronal consiste en un vector con las asignaciones numéricas referentes a los colores que se desean identificar. En este caso, los colores a identificar son el cian, rojo, amarillo, verde y magenta, por lo que el vector de salida del sistema es $\hat{a} = [a_1, \dots, a_5]^T = [\text{cian}, \text{rojo}, \text{amarillo}, \text{verde}, \text{magenta}]^T$. Si el elemento del vector a está en el estado activo ($a_i = 1 \forall i = 1, 2, 3, 4, 5$), el color identificado corresponderá a la salida activa. Así, habrá cinco posibles combinaciones para identificar el color del objeto. Cabe mencionar que la red neuronal da salidas de 1 y -1, sin embargo, para efectos del cómputo de la red neuronal, los -1 se han convertido a 0 de tal forma que si el vector de salida es $\hat{a} = [1, 0, 0, 0, 0]^T$ el color detectado es cian, si $\hat{a} = [0, 1, 0, 0, 0]^T$ el color detectado es rojo, etc.

3. Entrenamiento de la Red Neuronal

La idea general del entrenamiento de una red neuronal es obtener mediante la solución de un problema de optimización un conjunto de vectores de pesos w y un conjunto de valores b de tal forma que se minimice el error entre la salida de la red neuronal con el valor deseado o conocido del sistema a modelar por la red.

El problema de ajustar estos valores para las neuronas de las capas de entrada hacia las capas ocultas de la red se puede solucionar partiendo de la idea de que el error en la capa de salida de la red se puede retro-propagar hacia las capas intermedias de la red.

Por tal motivo, se opta por utilizar el algoritmo de retro-propagación descrito en [6], el cual es un algoritmo de aprendizaje supervisado que se puede ver como un método de optimización que se vale de la técnica de gradiente de descenso para ajustar los pesos en la red. El algoritmo consiste en minimizar un error cuadrático el cual está definido por la siguiente ecuación:

$$E = (t(k) - a(k))^2 \quad (2)$$

Donde t es el valor objetivo que representa el valor deseado para la salida de la red, a es la salida de la red y k es el número de iteración. Si la red tiene varias salidas:

$$E = (t(k) - a(k))^T (t(k) - a(k)) \quad (3)$$

Así los valores para w y b están dados por:

$$w_{ij}^m(k+1) = w_{ij}^m(k) - \alpha \frac{\partial E}{\partial w_{ij}^m} \quad (4)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial E}{\partial b_i^m} \quad (5)$$

Donde el superíndice m indica el número de la capa, el subíndice i indica el número de neurona en la capa y el subíndice j indica el número de entrada a la neurona. El término α corresponde al factor de aprendizaje de la red neuronal, este valor puede encontrarse entre 0 y 1, entre más alto sea, la red se entrena más rápido es decir, con menos iteraciones, sin embargo un α alto puede hacer que el sistema oscile y se vuelva inestable.

Dado que éste es un algoritmo de entrenamiento supervisado, a la red neuronal se le dan 2 conjuntos de valores principales: el vector de entrenamiento que consiste de una serie de entradas patrón y un vector objetivo que es el vector de salidas deseadas para dichas entradas; el vector objetivo debe tener la misma cantidad de elementos que el vector de salidas ya que como se observa en la ecuación (2) se restan elemento por elemento para ir obteniendo el error. Un ejemplo de vector de objetivo puede ser el color cian ($t_1 = [1.0, 0.0, 0.0]^T$).

4. Implementación en un microcontrolador

La estructura de la red neuronal y los valores para W y b que resultaron del entrenamiento se ingresaron a un microcontrolador PIC18F2550 [7]. Este microcontrolador se eligió principalmente porque tiene la capacidad de manejar datos de tipo flotante.

Entre las características principales del microcontrolador se encuentran:

- Convertidores A/D (10 bits de resolución).
- Canales con modulación por ancho de pulso (PWM).
- Temporizadores de 8 y 16 bits (hasta 3 temporizadores por dispositivo).
- Oscilador interno de 8MHz.

El microcontrolador fue programado en lenguaje C utilizando el compilador CCS y el programador Pickit2. Una vez declaradas las variables a utilizar, el programa se ejecuta como lo muestra el siguiente pseudocódigo.

1. Se cargan las matrices de pesos y los vectores de polarizaciones obtenidos por el algoritmo

de retro-propagación: $W^1, W^2, W^3, W^4, \hat{b}^1, \hat{b}^2, \hat{b}^3$ y \hat{b}^4 .

2. Se apagan los leds del sensor para tomar lectura con luz ambiente y la lectura se guarda en p_1 .
3. Se encienden los 3 leds para formar luz "blanca", se ilumina el objeto a detectar su color, se toma la lectura del sensor y se guarda en p_2 .
4. Se ilumina el objeto a detectar su color con luz roja (se enciende el led rojo), se toma lectura y se guarda en p_3 .
5. Se ilumina el objeto a detectar su color con luz verde (se enciende el led verde), se toma lectura y se guarda en p_4 .
6. Se ilumina el objeto a detectar su color con luz azul (se enciende el led azul), se toma lectura y se guarda en p_5 .
7. Se apagan los leds.
8. Se genera el vector de entradas $\hat{p} = [p_1, p_2, p_3, p_4, p_5]$.
9. Se multiplica \hat{p} por W^1 y se guarda el resultado en \hat{a}^1 .
10. Se suma \hat{b}^1 a \hat{a}^1 y se guarda el resultado en \hat{a}^1 .
11. Se multiplica \hat{a}^1 por W^2 y se guarda el resultado en \hat{a}^2 .
12. Se suma \hat{b}^2 a \hat{a}^2 y se guarda el resultado en \hat{a}^2 .
13. Se multiplica \hat{a}^2 por W^3 y se guarda el resultado en \hat{a}^3 .
14. Se suma \hat{b}^3 a \hat{a}^3 y se guarda el resultado en \hat{a}^3 .
15. Se multiplica \hat{a}^3 por W^4 y se guarda el resultado en \hat{a}^4 .
16. Se suma \hat{b}^4 a \hat{a}^4 y se guarda el resultado en \hat{a}^4 .
17. Se toman \hat{a}^4 como el vector de resultados de la red neuronal artificial.

5. Análisis de resultados

Las mediciones de la salida de sensor de intensidad luminosa necesarias para el entrenamiento de la red neuronal se llevaron a cabo en el interior de un salón con iluminación artificial, intentando de esta manera conseguir condiciones de iluminación constantes. Sin embargo, al igual que en muchos casos de visión artificial [8], la iluminación del espacio de trabajo presenta variaciones que afectan directamente al sistema de visión. Por tal motivo se tomaron mediciones de la cantidad de iluminación reflejada por los cubos siendo iluminados por luz blanca (combinación de luz roja, luz verde y luz azul) y con la luz ambiental además de considerar los tres colores primarios (luz roja, luz verde, luz azul), con el propósito de hacer al sistema de identificación de colores no tan sensible a variaciones de luz en el ambiente.

Para cada color se realizaron 5 mediciones bajo cada tipo de iluminación proporcionada por el sensor de color, es decir, 5 mediciones bajo luz ambiental, 5 bajo luz “blanca”, 5 bajo luz roja, 5 bajo luz verde y 5 bajo luz azul.

Con un total de 125 mediciones (25 por cada color), se obtuvo una tabla con valores promedio de la respuesta del sensor para los 5 colores iluminados con los diferentes tipos de luces. Dichos valores se muestran en la Tabla 1. Esta tabla se utilizó para formar vectores de entrenamiento para la red neuronal.

El programa de entrenamiento de la red neuronal se ejecutó fuera de línea mediante MATLAB con los datos promedio de la Tabla 1 como vector de entrenamiento, con distinta cantidad de épocas (iteraciones) y distintos valores de α a fin de determinar un valor adecuado para estos parámetros. La razón de aprendizaje se propuso $\alpha=1.88 \times 10^{-9}$. El α tan pequeño propuesto para el entrenamiento se debe a que en general, si se observa la Tabla 1, los valores para los diferentes colores son pequeños (en milivolts) y a que en algunos casos a pesar de tener resolución de 10 bits, la salida del ADC (cuyo rango va de 0 a 5V) dio valores cercanos para diferentes colores con diferente tipo de

iluminación proporcionada por el sensor de color.

En consecuencia la red neuronal requirió de más iteraciones para completar su entrenamiento y reconocer efectivamente los colores deseados.

El entrenamiento tomó 10 millones de épocas en completarse y dio como resultado 4 matrices (W^1 , W^2 , W^3 y W^4) de 5×5 , una matriz por cada capa y 4 vectores (\hat{b}^1 , \hat{b}^2 , \hat{b}^3 y \hat{b}^4) de 5 elementos, un vector por cada capa. Los elementos de estas matrices representan los valores de los pesos para la capa a la que corresponde la matriz. Cada matriz contiene los 5 valores de pesos para cada una de las 5 neuronas en una capa. Los elementos de los vectores representan los valores de polarización para cada una de las neuronas que conforman una capa.

Los valores de estas matrices y vectores se muestran a continuación.

$$W^1 = \begin{bmatrix} -0.587 & 0.242 & 0.368 & -0.035 & -0.540 \\ -0.618 & -0.386 & -0.386 & 0.011 & 0.185 \\ 0.174 & -0.477 & -0.304 & -0.532 & 0.301 \\ -0.809 & -0.713 & -0.566 & -0.429 & -0.174 \\ 0.183 & -0.671 & -0.430 & -0.041 & -0.887 \end{bmatrix} \quad (6)$$

$$W^2 = \begin{bmatrix} -0.267 & -0.568 & 0.417 & -0.190 & 0.163 \\ -0.546 & -0.363 & -0.235 & 0.084 & -0.227 \\ 0.393 & -0.061 & -0.189 & 0.173 & 0.620 \\ -0.773 & -0.153 & -0.305 & -0.075 & 0.019 \\ -0.127 & -0.457 & -1.173 & 0.317 & -0.555 \end{bmatrix} \quad (7)$$

$$W^3 = \begin{bmatrix} -0.567 & -0.709 & -0.697 & -0.740 & 0.266 \\ -0.698 & -0.140 & -0.511 & -0.102 & -0.058 \\ 0.161 & -0.427 & -0.959 & -0.276 & -0.182 \\ -0.447 & -0.387 & -0.859 & -0.125 & -0.516 \\ 0.138 & -0.207 & -0.059 & -0.534 & 0.042 \end{bmatrix} \quad (8)$$

$$W^4 = \begin{bmatrix} 0.118 & -0.457 & -0.312 & 0.067 & -0.507 \\ -0.101 & -1.074 & -0.614 & -0.486 & -0.338 \\ 0.686 & -0.613 & -1.052 & -0.872 & -0.839 \\ -0.714 & -1.302 & -1.397 & -0.400 & -1.006 \\ -0.237 & -0.447 & -0.435 & -0.032 & -0.385 \end{bmatrix} \quad (9)$$

$$\hat{b}^1 = [0.939 \quad 0.916 \quad 0.087 \quad 0.711 \quad 0.269] \quad (10)$$

$$\hat{b}^2 = [0.439 \quad 0.542 \quad 0.919 \quad 0.418 \quad 0.951] \quad (11)$$

$$\hat{b}^3 = [0.222 \quad 0.672 \quad 0.698 \quad 0.606 \quad 0.747] \quad (12)$$

$$\hat{b}^4 = [0.675 \quad 0.225 \quad -0.007 \quad 0.998 \quad 0.282] \quad (13)$$

Tabla 1. Tabla de lecturas promedio en mV del sensor ColorPal bajo diferentes tipos de luz aproximadamente a 1cm de distancia.

Cubo	Cian [mV]	Rojo [mV]	Amarillo[mV]	Verde [mV]	Magenta[mV]
Luz Ambiente	25.62079899	27.61160431	28.39061509	27.69816107	29.51585289
Luz Blanca	1223.046925	921.6563094	1328.646164	573.4384908	894.8237159
Luz Roja	287.4549778	813.720038	788.7051363	253.7844008	482.8135701
Luz Verde	421.1851617	90.79803424	489.5649968	244.2631579	134.9419784
Luz Azul	681.807546	128.4502219	219.1616994	170.6899176	327.1845276

Una vez que la red neuronal artificial estuvo entrenada, los valores de los pesos y los valores de polarización se introdujeron directamente al programa del microcontrolador.

La implementación física del dispositivo se realizó utilizando tecnología de montaje superficial en componentes como el microcontrolador lo que permitió que el tamaño del dispositivo fuera compacto y delimitado por el tamaño del sensor ColorPal (3.8x2.3cm x 2.5cm de profundidad). En la Figura 4 se muestra el dispositivo completo; el microcontrolador se encuentra oculto en la parte posterior de la tarjeta electrónica de la que sólo se aprecian sus conectores y un par de capacitores.



Fig. 4. Sistema completo de reconocimiento de colores mediante red neuronal artificial.

Las pruebas de funcionamiento se realizaron clasificando grupos de 5 cubos, uno de cada color, colocando el cubo frente al sensor a una distancia de aproximadamente 1cm. Después de varias pruebas, el sistema de clasificación falló uno de cada 25 intentos de clasificación obteniendo así un margen de error de tan solo 4%.

6. Conclusiones

En este trabajo se presenta el desarrollo de un sistema basado en redes neuronales artificiales para la clasificación de colores mediante el cual se han obtenido resultados satisfactorios con un porcentaje de efectividad del 96%. El sistema de clasificación se ha empleado con éxito como subsistema de un manipulador móvil cuya tarea fue clasificar y recolectar cubos de colores colocados de manera aleatoria dentro de un área de trabajo.

Se concluye que las Redes Neuronales Artificiales representan una técnica adecuada para la resolución del problema de clasificación de colores con condiciones de iluminación variables, sin embargo, se hace énfasis en que el uso de redes neuronales puede extenderse a otros campos de aplicación por lo que el sistema desarrollado puede modificarse o complementarse para trabajos futuros

ya sea sobre la misma problemática o tomar el procesamiento de redes neuronales como método de solución a otros problemas e implementarlo para alguna otra aplicación.

Reconocimiento

Se agradece el apoyo otorgado de la COFAA y la SIP del Instituto Politécnico Nacional a través del proyecto con número SIP-20110165 y al SNI del Consejo Nacional de Ciencia y Tecnología.

Referencias

- [1] Kasabov N. *“Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering”*, The MIT Press, USA, 2º edición, 1998.
- [2] Sibai F., Hosani H., Naqbi R., Dhanhani S., Shehhi S. *“Iris recognition using artificial neural networks”*, Expert Systems with Applications, 38, pp. 5940-5946, 2011..
- [3] Baykan N. A., Yilmaz N. *“Mineral identification using color spaces and artificial neural networks”*, Computers & Geosciences, 36, pp. 91-97, 2010.
- [4] Freeman J., Skapura D. *“Neural Networks Algorithms, Applications, and Programming Techniques”*, Addison-Wesley Publishing Company, USA, 1º edición, 1991.
- [5] PARALLAX [en línea]: *“ColorPAL (#28380) Color and Ambient Light Sensor and Color Generator”* <<http://www.parallax.com/Portals/0/Downloads/docs/prod/sens/28380ColorPAL.pdf>> [Consulta: 26 Agosto, 2011].
- [6] Hagan M., Howard B., Beale M. *“Neural Network Design”*, PWS, USA, 1º edición, 1995.
- [7] Microchip [en línea]: *“PIC18F2455/2550/4455/4550 Data Sheet”* <<http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>> [Consulta: 26 Agosto, 2011].
- [8] O’Gorman L., Sammon M., Seul M. *“Practical Algorithms for Image Analysis”*, Cambridge University Press, USA, 2º edición, 2008.