

# Clasificación de la manzana royal gala usando visión artificial y redes neuronales artificiales

Gustavo Andrés Figueredo Avila

Universidad Pedagógica y Tecnológica de Colombia,  
Escuela de Ingeniería de Sistemas y Computación,  
Tunja, Boyacá,  
Colombia

gustavoandres.figueredo@uptc.edu.co

**Resumen.** La clasificación de una fruta según su estado de maduración es complicada debido a todas las variables que influyen en ese cambio. Se propuso la construcción de una red neuronal artificial multicapa backpropagation (BPNN) con aprendizaje supervisado entrenada con cuatro diferentes algoritmos de aprendizaje (TRAINSCG, TRAINBFG, TRAINBR, TRAINLM) de manera independiente. De un conjunto de 30 manzanas Royal Gala, se adquirieron un total de 4200 fotos durante 35 días, que fueron procesadas y redimensionadas a un tamaño de 256x456. Las fotos segmentadas se convirtieron a modelo de color RGB, HSV y Lab, con el fin de extraer los valores de la media, la varianza y la desviación estándar. El mejor rendimiento de entrenamiento se obtuvo con el algoritmo TRAINBR (99.3%), y el mejor rendimiento en pruebas se obtuvo con TRAINSCG (67.7%). Finalmente, las comparaciones entre los diferentes resultados de cada algoritmo muestran que el mejor algoritmo de clasificación fue TRAINSCG.

**Palabras clave:** Procesamiento de imágenes digitales, red neuronal artificial Backpropagation (BPNN), Red Neuronal Artificial (RNA), algoritmo de entrenamiento, Mean Squared Error (MSE).

## Classification of Royal Gala Apple Using Computer Vision and Artificial Neural Networks

**Abstract.** The fruit classification according to their state of maturation is complicated because of all variables that influence this change. In this paper a backpropagation multilayer artificial neural network (BPNN) is proposed, it was trained with supervised learning with four different learning algorithms (TRAINSCG, TRAINBFG, TRAINBR, TRAINLM) independently. During 35 days, a total of 4200 pictures were acquired from a set of 30 Royal Gala apples, which were processed and resized to a size of 256x456. The segmented pictures were converted to RGB, HSV and Lab color models, in order to extract the values of the mean, variance and standard deviation. The best training performance was obtained with the algorithm TRAINBR (99.3%) and the best test performance was obtained with TRAINSCG (67.7%). Finally, comparisons between different results from each algorithm show that the best classification algorithm was TRAINSCG.

**Keywords:** Digital image processing, Backpropagation Artificial Neural Network (BPNN), Artificial Neural Network (ANN), training algorithm, Mean Squared Error (MSE).

## **1. Introducción**

La identificación del comportamiento y cambio constante del proceso de maduración de una fruta durante un periodo determinado de tiempo, conlleva el análisis de diferentes variables que afectan dichos cambios y que de antemano se les atribuye un comportamiento impredecible o no lineal [1]. El análisis técnico de los principales cambios físicos de una fruta a lo largo de su vida útil, normalmente se realiza utilizando técnicas de espectrometría, que abarcan los cambios más significativos relativos al color; o análisis destructivo para medir las variaciones de líquidos, alcohol, almidón y azúcares [2].

Usualmente, la determinación del estado de madurez de una fruta se encuentra en manos de los humanos, que son capaces de entrar a considerar un número significativo de cambios en la estructura física de las frutas, y dar un resultado cualitativo respecto a la calidad, aunque al igual que las técnicas tecnológicas tradicionales, dicha clasificación suele ser provechosa en periodos de tiempo relativamente largos.

Las Redes Neuronales Artificiales (RNA) han sido principalmente usadas para la clasificación de datos, reconocimiento de patrones y predicción de comportamientos, debido a su intención de emular la función de las redes neuronales biológicas, a las que se les atribuye el proceso de aprendizaje mediante ejemplos [3, 4,5]. Para el caso de clasificación y reconocimiento de patrones en frutas, usualmente se ha venido usando la arquitectura de red neuronal denominada perceptrón multicapa backpropagation, la cual puede ser entrenada con diferentes algoritmos. En la literatura se menciona el uso de redes neuronales artificiales para clasificar manzanas según el color [6], para identificar diferentes tonalidades de color, magulladuras o daños en la superficie de la manzana [7], o para clasificar diferentes tipos de frutas según ciertas características extraídas de la imagen [8].

Para esta investigación se usó la Red Neuronal Backpropagation (BPNN), entrenada por separado con algoritmos Gradiente Conjugado Escalado (TRAINSCG), BFGS Quasi Newton (TRAINBFG), Regularización Bayesiana (TRAINBR) y Levenberg-Marquardt (TRAINLM), con una arquitectura de 3 capas, un total de 30 neuronas de entrada, 28 neuronas en la capa oculta y 5 neuronas de salida.

Esta investigación incluye la sección 2, en donde se describen los materiales y métodos utilizados para el experimento, la sección 3, en donde se especifican los pasos en el procesamiento de las imágenes, en la sección 4 se especifica la arquitectura de la red neuronal, en la sección 5, los resultados, y en la sección 6 las conclusiones y trabajos futuros.

## **2. Materiales y métodos**

Para el desarrollo de esta investigación, se adquirieron un total de 30 manzanas Royal Gala recién llegadas al supermercado, las cuales fueron dispuestas diariamente sobre y delante una superficie blanca por un periodo de 35 días. A cada manzana se le

tomó una foto por cada una de las 4 caras de la manzana previamente marcadas, bajo la presencia de luz natural. Diariamente se obtuvieron un total de 240 fotografías.

Las fotografías fueron adquiridas en formato JPG (Grupo Conjunto de Expertos en Fotografía) bajo el modelo RGB (Red, Green, Blue – Rojo, Verde, Azul) con una dimensión de 2592x1456 y una resolución de 72ppp. No se aplicó flash ni zoom digital.

A cada manzana se le tomaron datos iniciales de peso, altura y diámetro ecuatorial, durante los primeros días del experimento, y posteriormente se retomaron en los últimos días, con el propósito de conocer la variación por cada día transcurrido.

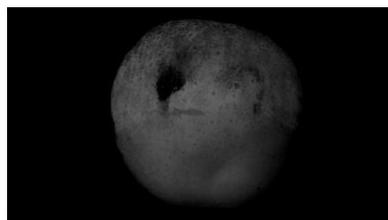
### 3. Procesamiento de imagen

#### 3.1 Pre-Procesamiento

Cada imagen adquirida (Ver Figura 1 (a)) fue inicialmente modificada eliminando los valores del canal verde y azul con el fin de obtener la forma de la manzana (Ver Figura 1 (b)). Posteriormente se le aplicó un filtro de Gauss que permitiera difuminar la imagen y eliminar la mayor cantidad de ruido presente sobre la misma, y a su vez suavizar los bordes (Ver Figura 1 (c)).



a) Imagen Original (2592x1456)



b) Imagen Canal Rojo menos Verde y Azul.

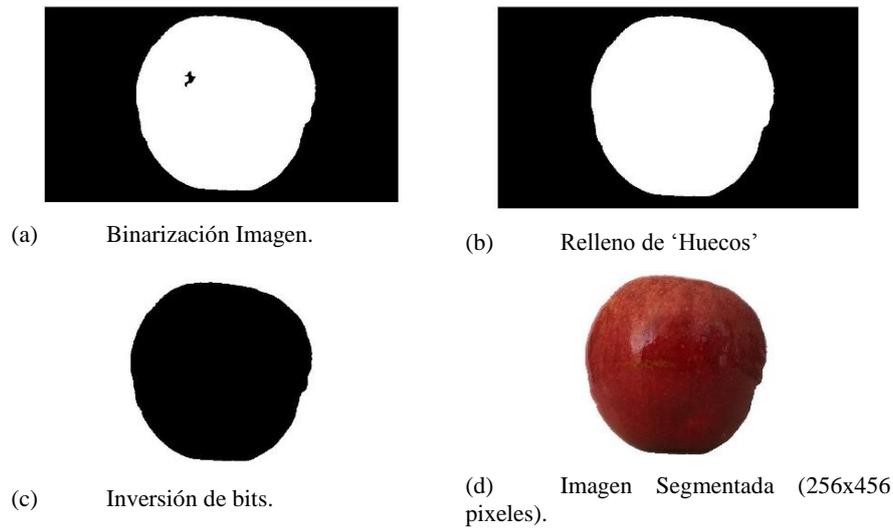


c) Imagen Filtro Gauss.

**Fig. 1.** Detección de forma y eliminación de ruido

Una vez obtenida la forma de la manzana, se realizó el proceso de binarización, con el cual se obtuvo una imagen representada únicamente en dos valores (0 y 1) (Ver Figura 2 (a)). En algunos casos la imagen binarizada presentó la presencia de ‘Huecos’, los cuales son bits no pertenecientes al conjunto de bits predominantes en un área, al ser eliminados se obtuvo la forma completa de la manzana (Ver Figura 2 (b)).

Paso siguiente, se invirtieron los valores de los bits de la imagen binarizada (Ver Figura (c)), para superponer la imagen original sobre la máscara binaria creada (Ver Figura 2 (d)). La imagen final se redimensionó a 256x456 pixeles para reducir el tiempo de procesamiento en la obtención de datos.



**Fig. 2.** Creación de máscara binaria y superposición de imagen original

### 3.2 Obtención de datos

Las imágenes segmentadas y redimensionadas originalmente en modelo RGB fueron transformadas a los modelos de color HSV (Hue, Saturation, Value – Matiz, Saturación, Valor) y Lab (Luminosidad, Rojo-Verde, Gradiente Azul), para extraer la información de cada uno de los canales calculando la media, la varianza y la desviación estándar, generando de esta forma una matriz de 9x4200 por cada modelo de color, en donde el 9 representa la concatenación de los 3 valores estadísticos calculados por cada canal de color, y los 4200 datos son el resultado de 30 manzanas x 4 caras x 35 días.

La media se calcula mediante la ecuación 1, en donde P representa cada uno de los tres canales de cada modelo de color RGB, HSV, Lab:

$$\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i. \quad (1)$$

La varianza se calcula mediante la ecuación 2, en donde P representa cada uno de los tres canales de cada modelo de color RGB, HSV, Lab y  $\bar{P}$  es la media calculada de cada canal:

$$\sigma_p^2 = \frac{1}{n} \sum_{i=1}^n (P_i - \bar{P})^2. \quad (2)$$

La desviación estándar se calcula mediante la ecuación 3:

$$\sigma_p = \sqrt{\sigma_p^2}. \quad (3)$$

Posteriormente se calculó el valor de la media de las cuatro caras que corresponden a cada manzana, con lo cual se redujo el tamaño de la matriz a 27 x 1050, que representan las 9 entradas de cada modelo de color por 30 manzanas durante un periodo de 35 días.

Los datos de entrada correspondientes a los tres modelos de color seleccionados, proporcionaron información sobre el cambio en la coloración, matiz, saturación e iluminación en la superficie de la manzana. Se complementaron los datos de entrada asociados al color, con los valores de peso, altura y diámetro ecuatorial, resultados de un cálculo interpolado basado en datos adquiridos en los primeros días del experimento y al final del experimento.

Finalmente se consolidaron los datos en una matriz de 30 x 1050 que se usa para el entrenamiento de la red neuronal.

## 4. Arquitectura de red neuronal

### 4.1 Red neuronal

La red neuronal artificial planteada consta de un total de tres capas (la capa de entrada, la capa oculta y la capa de salida). La capa de entrada está compuesta por 30 neuronas (representan la media, la varianza y la desviación estándar por cada canal (3 canales) de cada modelo de color (3 modelos)), la capa oculta por 28 neuronas y la capa de salida por 5 neuronas (Representan cinco semanas, cada una de siete días).

La red es entrenada inicialmente por una matriz de 30x1050 de los cuales el 75% de los datos se usaron como entrenamiento, el 15% como validación y el 15% como pruebas. Posteriormente se realizó otro entrenamiento únicamente con una matriz de 30x750, dejando una matriz de 30x350 como matriz de prueba independiente. La arquitectura de la red se puede apreciar en la Figura 3.

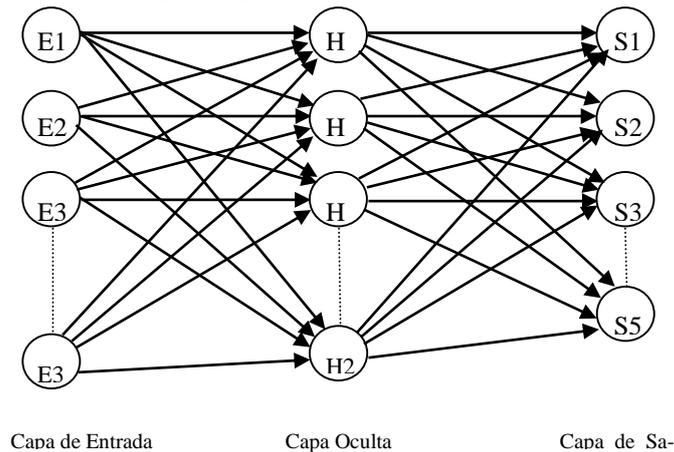


Fig. 3. Arquitectura Red Neuronal Multicapa

## 4.2 Algoritmos de entrenamiento

Los Algoritmos de entrenamiento modifican los pesos de cada neurona para identificar a lo largo de una serie de iteraciones, cual es el patrón que mejor se ajusta a los objetivos. Se calcularon los valores de rendimiento MSE (Mean Squared Error) para cada algoritmo, los cuales indican el error mínimo alcanzado para el entrenamiento o prueba.

En esta investigación se propuso utilizar como algoritmos de entrenamiento para la red neuronal backpropagation los siguientes:

- Gradiente Conjugado Escalado (**TRAINSCG**),
- BFGS Quasi Newton (**TRAINBFG**),
- Regularización Bayesiana (**TRAINBR**),
- Levenberg-Marquardt (**TRAINLM**).

El tipo de aprendizaje de la red neuronal es supervisado, lo cual conlleva a que se establezca previamente que valores son los deseados para cada conjunto de datos proporcionados en la capa de entrada.

## 5. Resultados y discusión

El entrenamiento de la red neuronal artificial propuesta en este estudio, se establece sobre la herramienta Matlab. Se crearon dos redes neuronales, las cuales fueron entrenadas por cada uno de los algoritmos de entrenamiento propuestos en la sección anterior. La primera red fue entrenada con un total de 30 x 1050 datos, de los cuales se tomaron aleatoriamente el 75% para entrenamiento, 15% para validación y 15% para pruebas. La segunda red fue entrenada con 30 x 700 datos, de los cuales 60% fueron a entrenamiento, 20% a validación y 20% a pruebas, sobre esta última red se simuló la clasificación con una matriz de pruebas conformada por 30 x 350, datos que no fueron incluidos en el entrenamiento a diferencia de la primera red.

En la Tabla 1 se presenta el rendimiento de la red neuronal (BPNN) entrenada con cada uno de los algoritmos y validada con el 15% de los datos aleatorios.

**Tabla 1.** Rendimiento de Algoritmos de entrenamiento medido en MSE.  
Conjunto de datos 30x1050

| <b>Algoritmo /RedPerformance (MSE)</b> |        |
|--|--------|
| TRAINBFG                               | 0.0915 |
| TRAINBR                                | 0.0196 |
| TRAINLM                                | 0.0590 |
| TRAINSCG                               | 0.0675 |

En la Tabla 2 se presenta el rendimiento de la red neuronal (BPNN) entrenada con cada uno de los algoritmos, validada con el 20% de los datos aleatorios, y validada con un conjunto de datos diferentes a los de entrenamiento.

**Tabla 2.** Rendimiento de Algoritmos de entrenamiento medido en MSE. Conjunto de datos 30x700

| Algoritmo/Red | Performance Train (MSE) | Performance Test (MSE) |
|---------------|-------------------------|------------------------|
| TRAINBFG      | 0.0993                  | 0.1276                 |
| TRAINBR       | 0.0233                  | 0.1808                 |
| TRAINLM       | 0.0598                  | 0.1615                 |
| TRAINSOG      | 0.0820                  | 0.1221                 |

Como se aprecia en la Tabla 1 y 2, el algoritmo TRAINBR proporcionó el mejor rendimiento en cuanto a entrenamiento de la red neuronal, pero el mejor rendimiento en las pruebas, lo proporcionó el algoritmo TRAINSOG. Esencialmente esto se debe a que los algoritmos TRAINBFG, TRAINSOG y TRAINLM realizaron un proceso de validación para que finalice el proceso de entrenamiento cuando encuentre 6 desaciertos, lo que hace que la red no se sobre-entrene, mientras que el algoritmo TRAINBR realiza un entrenamiento hasta cumplir la totalidad de las iteraciones (1000) o hasta que el valor del gradiente tienda a 0.

En la Tabla 3 se presenta la duración de cada algoritmo en el proceso de entrenamiento con datos de prueba independiente incluidos y sin datos de prueba independiente incluidos.

**Tabla 3.** Duración de entrenamiento de la Red Neuronal

| Algoritmo/Red | Duración sin prueba incluida (Segundos) | Duración con prueba incluida (Segundos) |
|---------------|---|---|
| TRAINBFG      | 141                                     | 93                                      |
| TRAINBR       | 755                                     | 569                                     |
| TRAINLM       | 58                                      | 37                                      |
| TRAINSOG      | 1                                       | 0                                       |

En la Tabla 3 se puede ver que el algoritmo TRAINSOG convergió bastante rápido debido a las validaciones que lleva a cabo, lo que frena el proceso de entrenamiento a tan solo un segundo. El algoritmo TRAINBR tardó bastante tiempo en el proceso de entrenamiento debido a que ejecuta todo el número de iteraciones (para este caso 1000).

En la Tabla 4 se presenta el número de iteraciones que fueron necesarias para que la red neuronal convergiera y asegurara buenos resultados de entrenamiento y pruebas.

**Tabla 4.** Numero de iteraciones realizadas por cada algoritmo

| Algoritmo/Red | Iteraciones sin prueba (Nº/Total) | Iteraciones con prueba (Nº/Total) |
|---------------|-----------------------------------|-----------------------------------|
| TRAINBFG      | 52/1000                           | 34/1000                           |
| TRAINBR       | 1000/1000                         | 1000/1000                         |
| TRAINLM       | 17/1000                           | 16/1000                           |
| TRAINSOG      | 71/1000                           | 43/1000                           |

El menor número de iteraciones lo realizó el algoritmo TRAINLM, valor que representa la efectividad de la red en converger rápido evitando exceso de procesamiento de la máquina.

En la Tabla 5 se aprecia el porcentaje de asertividad de cada algoritmo de entrenamiento con datos de pruebas aleatorios.

**Tabla 5.** Efectividad de entrenamiento y prueba de los algoritmos de entrenamiento. Conjunto de datos 30x1050

| Algoritmo/Red | Entrenamiento (%) | Test (%) |
|---------------|-------------------|----------|
| TRAINBFG      | 69.3              | 53.2     |
| TRAINBR       | 99.3              | 66.5     |
| TRAINLM       | 87.2              | 62.7     |
| TRAINS CG     | 80.5              | 67.7     |

En la Tabla 6 se muestra el porcentaje de asertividad de cada algoritmo de entrenamiento con datos de pruebas independientes.

**Tabla 6.** Efectividad de entrenamiento y prueba de los algoritmos de entrenamiento. Conjunto de datos 30x700

| Algoritmo/Red | Entrenamiento (%) | Test(%) incluido | Test(%) independiente |
|---------------|-------------------|------------------|-----------------------|
| TRAINBFG      | 64.3              | 52.9             | 51.1                  |
| TRAINBR       | 99.8              | 65.0             | 49.1                  |
| TRAINLM       | 95.2              | 56.4             | 47.4                  |
| TRAINS CG     | 71.7              | 64.3             | 54.0                  |

En la Tabla 5 y 6, se aprecia que el algoritmo TRAINBR, proporcionó un alto porcentaje de acierto en el entrenamiento y en pruebas con datos incluidos y elegidos aleatoriamente, pero en la clasificación de pruebas independientes al conjunto de datos de entrenamiento, es el algoritmo TRAINSCG el que proporcionó mejores resultados.

Tomando en cuenta cada una de las características evaluadas en cada algoritmo de entrenamiento y sabiendo que las redes neuronales artificiales en el momento en que empiezan a aprender dejan de converger, se atribuye los mejores resultados al algoritmo de entrenamiento TRAINSCG, por su rapidez y buenos resultados de pruebas tanto incluidas en el conjunto de datos, como externas a estos.

En la Figura 4 se visualiza la matriz de confusión para la red entrenada con algoritmo TRAINSCG con datos de prueba incluidos y seleccionados aleatoriamente.

La Figura 4 expresa la diferencia entre el objetivo de la clasificación y los valores obtenidos del entrenamiento de la red neuronal, la cual fue capaz de clasificar un total de 67.7% de casos correctamente y obtuvo un promedio general de 76.3%. Los casos clasificados incorrectamente en realidad expresan la semana de maduración potencial de cada manzana en relación al cambio de sus características evaluadas, esto esencialmente porque cada fruta presenta un proceso de maduración diferente entre sí, lo que lleva a pensar que mientras una manzana debería ser clasificada en la cuarta semana de maduración, puede que sus características en realidad le atribuyan una clasificación en la primera semana en donde aún no hay cambio significativo de color, ni pérdida de peso, altura o diámetro.

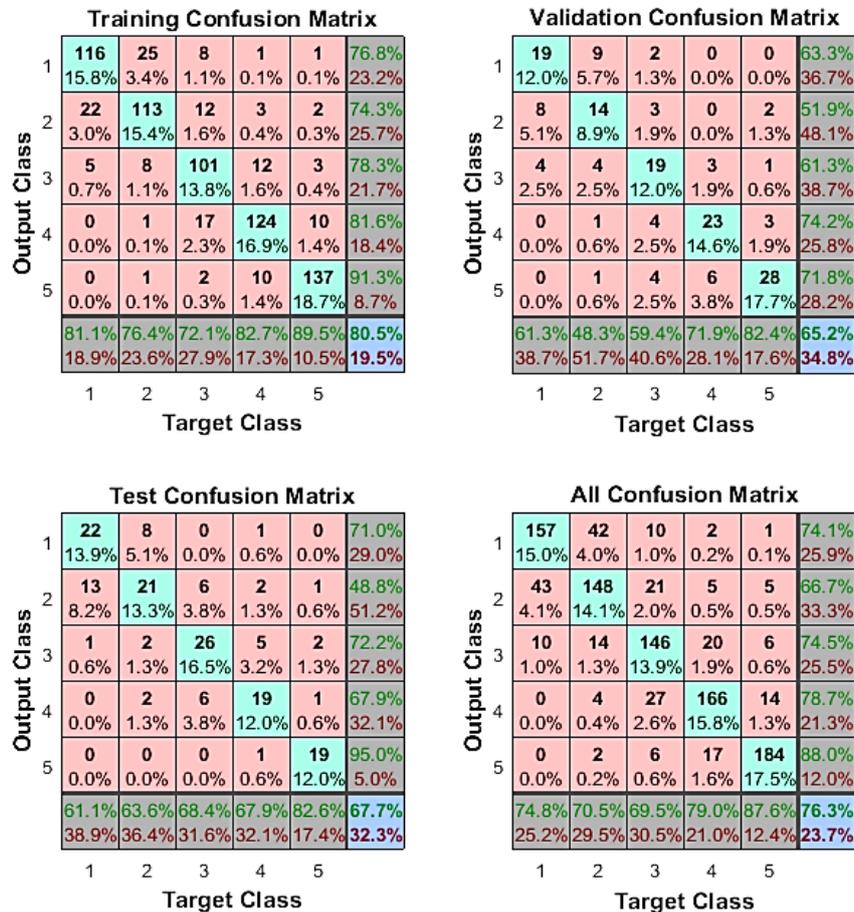


Fig. 4. Matriz de confusión algoritmo TRAINSCG

## 6. Conclusiones y trabajo futuro

En este trabajo se propuso la creación de una red neuronal backpropagation multicapa con aprendizaje supervisado, entrenada con cuatro diferentes algoritmos de backpropagation (TRAINBFG, TRAINBR, TRAINLM y TRAINSCG), bajo dos esquemas diferentes, un conjunto de datos que incluyó un total de 30x1050 ejemplos de los cuales se extrajeron un 75% para entrenamiento, 15% para pruebas y 15% para validación, y otro con 30x700 datos de entrenamiento y 30x350 datos para pruebas independientes.

Los datos adquiridos para entrenamiento de la red provinieron de un previo tratamiento de las imágenes de 30 manzanas Royal Gala, equivalentes a 4200 imágenes tomadas durante 35 días y posteriormente transformadas a los modelos de color RGB, HSV y Lab.

Se analizaron los resultados obtenidos por el proceso de entrenamiento de cada algoritmo de aprendizaje tales como los valores de MSE, tiempo de ejecución, número de iteraciones, exactitud en el entrenamiento y en las pruebas. Los mejores resultados para la aplicación de esta red neuronal se obtuvieron con el algoritmo de Gradiente Conjugado Escalado (TRAINSCG)

Investigación posterior puede incluir la utilización de otras arquitecturas de red neuronal y otros algoritmos de entrenamiento. Complementar la clasificación teniendo de base características como los colores principales, las enfermedades de la fruta, y la calidad. Mejorar el rendimiento de los algoritmos de entrenamiento y extender su aplicación a otras frutas y/o áreas similares.

## **Referencias**

1. Buitrago, M.A.: Manejo de Manzanas en Cosecha y Poscosecha. Bioplasma. Vol. Documentos, Núm. IV, p. 12 (1991)
2. Brezmes Llecha, J.J.: Diseño de una nariz electrónica para la determinación no destructiva del grado de maduración de la fruta. Universitat Politècnica de Catalunya (2002)
3. Chen, G., Dong, X.: From chaos to order: perspectives, methodologies and applications. Singapore. World Scientific (1998)
4. Fausett, L.V.: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice-Hall (1994)
5. Lin, C.T., Lee, C.S.: Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. Prentice-Hall, Inc. (1996)
6. Ben-Hanan, U., Peleg, K., Gutman, P.O.: Classification of fruits by a Boltzmann perceptron neural network. Automatica, Vol. 28, No. 5, pp. 961–968 (1992)
7. Nakano, K.: Application of neural networks to the color grading of apples. Computers and Electronics in Agriculture, Vol. 18, No. 2-3, pp. 105–116 (1997)
8. Zhang, Y., Wang, S., Ji, G., Phillips, P.: Fruit classification using computer vision and feed-forward neural network. Journal of Food Engineering, 143, pp. 167–177 (2014)