

# Aplicando redes BPN para determinar áreas no deseadas en quinotos: resultados gráficos.

Jonatán E. Collazo Silvestre, Guillermo Sampallo y Arturo González Thomas

Facultad Regional Resistencia  
Universidad Tecnológica Nacional  
French 414, 3500 Resistencia, Chaco  
jonacollazo@gmail.com

**Resumen** El bajo costo de las cámaras digitales hace atractivo el uso de la segmentación de imágenes a color para evaluar la calidad de alimentos como ser los citrus. Para implementar un sistema automático de clasificación se requiere explicitar el conocimiento presente en las personas y caracterizar la variación de color en áreas de similar aceptabilidad atendiendo la profundidad de color con que se capturan las imágenes.

En este trabajo evaluaremos algunas redes BPN aplicadas a la diferenciación de áreas con colores deseables de aquellas con colores que corresponden a defectos como ser manchas o áreas no lo suficientemente maduras. Se pondrá énfasis en la aplicabilidad de la técnica y se profundizará en la descripción del espacio de decisión a aprender, los efectos de las representaciones en espacio de color  $RGB$  y  $L^*a^*b^*$  y la calidad de los resultados obtenidos.

Concluiremos que el enfoque es viable y perfeccionable atendiendo la discretización de las salidas de la red, la variedad de entradas usadas y el interés en disminuir cada tipo de error y la complejidad de la red usada.

**Palabras claves:** segmentación supervisada, segmentación imagen color, clasificación automática fruta, red BPN.

## 1. Introducción

Tanto la agricultura como la industria de alimentos requieren analizar imágenes de productos para realizar tareas de interés económico. Roy Davies en [1] hace un compendio del uso del *procesamiento de imágenes* en la industria de la alimentación y lo compara con la performance humana. Comúnmente la tarea de determinar la calidad, defectos o enfermedades en ciertos productos es propensa a no mantener la coherencia y sufrir limitaciones propias del accionar humano al ser realizada en forma no automática por un experto empleando solo sus sentidos y experiencia.

Uno de los problemas en el procesamiento de imágenes y la visión por computadora es el determinar automáticamente las regiones en una imagen que poseen características de interés y que de su análisis, resalte, diferenciación o relación con el resto que permita derivar alguna conclusión ya sea con o sin intervención humana. Trabajos como el presente en [2] por Jimenez et al. sobre la recolección automática de frutas y en [3] por Sun et al. presentan algunos resultados y problemas a considerar al momento de diferenciar objetos en *agricultura* y la *industria de alimentos*.

Peterson et al. en [4] analizan el uso de *redes neuronales* en problemas propios del procesamiento de imágenes. Los *perceptrones multicapa* resultan atractivos por su aplicación exitosa en la segmentación realizada a nivel de píxel y su facilidad de uso al ser aplicables como *cajas negras* que aprenden por sí mismas. De estas cajas negras posteriormente se podrá extraer el *conocimiento adquirido* tanto para hacer una descripción formal como para dar pautas de estudios futuros, tal como se hace en inteligencia de negocio con la extracción de reglas y el análisis estadístico.

Nos propusimos aplicar redes BPN (Backpropagation Neural Networks) para ubicar áreas con manchas en quinotos como tarea previa a su evaluación de calidad asumiendo que la red puede aprender la manera en que un experto discrimina entre los colores propios de la superficie de un quinoto maduro de excelente calidad, de la de aquellos con manchas, verdes o en mal estado. Hicimos un estudio sobre su eficacia a la hora de separar los colores deseables de los no deseables y complementaremos los resultados tradicionales con un análisis de los patrones a aprender y los errores cometidos.

Nuestro trabajo se organiza como sigue: en las secciones 2 y 3 presentaremos el marco teórico utilizado. En la sección 4 describiremos el diseño y el criterio de evaluación de los experimentos realizados; y en la sección 5, sus resultados. Por último, en la sección 6 concluiremos.

## 2. Segmentación de imágenes

Un problema de *segmentación* implica establecer *una partición del conjunto de puntos  $F$  que forman una imagen digital* para poder realizar tareas de mayor grado de abstracción. Esto es [5], establecer subconjuntos de puntos, regiones de interés o *ROIs (Region of Interest)* ( $S_1, S_2, \dots, S_n$ ) según algún criterio de tal modo que  $\bigcup_{i=1}^n S_i = F$  siempre que  $S_i \cap S_j = \emptyset$  para todo  $i \neq j$ .

Métodos de segmentación que se reseñan en [6] diferencian en una imagen de nivel de grises aquellos píxels pertenecientes al fondo, que no interesan, de aquellos pertenecientes al primer plano, que sí interesan. Entre ellos, la *umbralización* o *thresholding* basa su funcionamiento en clasificar los píxels según sus niveles de gris sean menores o mayores que un valor umbral  $L$ . Mayor información sobre métodos para determinar este valor y los resultados que se obtienen al aplicarlos en distintas imágenes de interés puede encontrarse en [7].

### 2.1. Segmentación de imágenes a color

Un *color* se acostumbra a describir en un espacio tridimensional con componentes vinculados a la elección de tres colores primarios y un blanco de referencia. Monitores y cámaras de video identifican, en general, los colores en el *espacio de color RGB*, por ser un espacio que facilita las tareas al momento de reproducir colores mediante la adición de haces rojos, verdes y azules.

Numerosos espacios de color se han definido sin lograr que ninguno sea capaz de describir el color con el que se percibe un objeto en cualesquiera condiciones de observación[8]. Al estandarizar espacios como el *RGB*, el *XYZ* y el  *$L^*a^*b^*$*  la CIE (International Commission on Illumination) puso énfasis en conseguir representaciones que privilegiaran diferentes aspectos.

El espacio de color  $L^*a^*b^*$  busca lograr que la noción de distancia euclídea entre puntos tenga una correspondencia con la noción de diferencia de color percibida por el hombre. Las coordenadas  $L^*$ ,  $a^*$  y  $b^*$  fueron definidas tal que la claridad, el tono y el cromatismo de un color puedan ser expresados, respectivamente, como  $L^*$ ,  $\arctan \frac{b^*}{a^*}$  y  $\sqrt{b^{*2} + a^{*2}}$ , y se pueda lograr un mapeo a un espacio que distinga al color propiamente dicho de su intensidad y pureza, y que sea consistente con la definición del espacio  $HSI$ (Hue-Saturation-Intensity)[5].

El uso de estos y otros espacios de color en la segmentación de imágenes color es explorado en [5]. Son cuestiones a tener en cuenta para extender las nociones de segmentación por niveles de gris: el modo en que se va a particionar el espacio de colores y la elección del espacio de color con el que se van a expresar los datos de una imagen expresada en  $RGB$ . Señala asimismo que el  $RGB$  no es bueno para la segmentación de imágenes, y que modelos como el  $HSI$  y el  $L^*a^*b^*$  si bien son no lineales respecto al primero permiten controlar de forma independiente información sobre el color y la intensidad.

### 3. Redes neuronales BPN

Las *redes BPN* son uno de los modelos usados para la aproximación de funciones y reconocimiento de patrones. Se caracterizan por la agrupación de neuronas en capas, las conexiones hacia adelante y la propagación de errores hacia atrás.

Cada neurona realiza una combinación lineal de sus entradas y bias y le aplica una función de transferencia  $f$  para generar su salida. La combinación lineal puede expresarse como  $net = \sum_{i=0}^n w_i x_i$  donde cada  $w_i$  representa el peso asignado a cada entrada  $x_i$ ,  $w_0 = bias$  y  $x_0 = 1$  y la función de transferencia puede ser, entre otras, lineal  $f(net) = net$  o sigmoideal  $f(net) = \frac{1}{1+e^{-net}}$ . Las neuronas luego se agrupan en una *capa de entrada*, una *capa de salida* y una o más *capas ocultas* para formar una *red neuronal* que se utiliza como *caja negra*. Según [9] puede establecerse cuál es la *complejidad máxima* de la superficie que separa las clases según la cantidad de neuronas que integran las capas y la cantidad de capas utilizadas.

Las BPN, según [10], pueden describirse como funciones que realizan un *mapeo*  $F = \mathfrak{R}^n \rightarrow \mathfrak{R}^m$  con  $n$  igual a la cantidad de entradas o neuronas de entrada y  $m$ , la cantidad de salidas o neuronas de salida; capaz de *aproximar tan precisamente como se desee* cualquier función real continua o con un número finito de discontinuidades entre dos conjuntos compactos y que puede ser expresado sin perder detalle como un *sistema de reglas con lógica difusa*.

El aprendizaje en la red neuronal se logra ajustando los pesos de la red para minimizar una medida de error escogida como la dada por el promedio en una época de las diferencias al cuadrado entre las salidas obtenidas y las deseadas también conocida como  $MSE$ (Mean Squared Error). El ajuste de los pesos puede realizarse utilizando el algoritmo de *Levenberg-Marquardt* que tiene un comportamiento que varía entre el algoritmo de descenso más pronunciado y el de Gauss-Newton según un parámetro  $\mu$  manejado por el propio algoritmo y que en general presenta una mejor performance al compararlo con el primero [11,12].

## 4. Propuesta de solución

### 4.1. Metodología

Una manera de determinar la presencia de manchas en alguna fruta es definir en qué grado los píxeles que componen esa región de la imagen cumplen con un patrón de color deseado. En nuestro caso, consideramos como *colores aceptables* a aquellos presentes en frutas maduras de buen aspecto; y como *colores no aceptables* a aquellos presentes en frutas verdes, en mal estado o bien en áreas con manchas visibles.

Nuestro enfoque se basó en *utilizar una red neuronal que diferencie los colores deseables de aquellos no deseables usando su representación como puntos en un determinado espacio de color*. Sobre una red BPN se aplican los *componentes de color de un píxel* expresados bien sea en el *espacio color RGB* o bien en el  $L^*a^*b^*$  para obtener un *valor continuo centrado entre 0 y 1* que indique de manera creciente su *deseabilidad* y que luego se discretizó a un *nivel 0 o 1 según fuera inferior o superior a algún valor límite de corte* para *distinguir* entre los puntos no aceptables (nivel 0) de los aceptables (nivel 1).

Matemáticamente, la entrada a la red es  $\mathfrak{R}^n = colorPixel = (c_1, c_2, c_3)$  con  $c_i$ , el valor de cada componente color. En caso de usar RGB, la terna es  $(R, G, B)$ ; y, en caso de usar  $L^*a^*b^*$ ,  $(L^*, a^*, b^*)$ . La salida de la red es  $\mathfrak{R}^m = aceptabilidad = \mathfrak{R}$ .

A fin de obtener los pares de entrenamiento (*colorPixel*, *aceptabilidad*) se evaluaron en forma manual con un  $1$  (*deseable*) o un  $0$  (*no deseable*) los píxeles que forman parte de las imágenes color de las frutas escogidas como ejemplo. La evaluación se realizó tal como lo haría un experto que determina áreas en la fruta que no son aceptables; esto es, a nivel de una única fruta o de regiones en ella y no a nivel global del conjunto de frutas elegidas. Se incluyeron 14 quinotos entre 5 maduros y con algunas manchas pequeñas, 5 verdes y 4 en mal estado de conservación o rechazables por su excesivo contenido de manchas.

### 4.2. Espacio de decisión a considerar

A fin de observar cuál es la complejidad del problema planteado y dar respuestas a ciertos interrogantes derivados de los resultados presentamos un resumen de los datos presentes en el conjunto de entrenamiento.

En el cuadro 1 se pone en evidencia la cantidad de ejemplos obtenidos con la poca cantidad de frutas utilizadas, los colores que hacen al concepto “*color aceptable*” y la redundancia presente en las muestras utilizadas. Del conjunto de pares de entrenamiento obtenidos se tabulan por cada clase a diferenciar la cantidad de ejemplos presentados y la cantidad de colores encontrados en cada una de ellas.

La figura 1 muestra cuáles son los colores que forman parte del conjunto de entrenamiento particularizando las diferencias de representación proporcionadas por los espacios  $RGB$  y  $L^*a^*b^*$ , como así también ubica en ambos espacios los colores que entendemos son aceptables y aquellos que no los son. Resulta llamativa la presencia de colores similares (que se aprecian mejor observándolos en el monitor) y, en ciertos casos, iguales en las clases supuestamente excluyentes de colores. De hecho, colores aceptables como los presentes en las áreas de sombra de frutas maduras estaban también

presentes como colores no aceptables en frutas en mal estado o con excesiva cantidad de manchas.

La figura 2 resalta la forma que debería adoptar una superficie de decisión que separe las clases de colores.

Tanto la figura 1 como la 2 muestran las clases con una agrupación principal de puntos caracterizada gráficamente porque existen puntos que no pueden unirse con una línea recta que esté contenida en esa región, como así también puntos aislados. A priori se puede observar que el espacio de color  $L^*a^*b^*$  minimiza la complejidad de una posible superficie de separación entre clases al poder expresarla con menor cantidad de vértices.

Tipo de píxel	Total ejemplos	Colores presentes
Aceptable	237.030	25.948
No aceptable	481.900	59.215

Cuadro 1: Patrones de entrenamiento

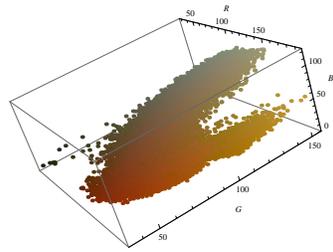
### 4.3. Redes neuronales a evaluar y parámetros de entrenamiento

Se probaron redes neuronales con *capas ocultas sigmoidales* puesto que poseen derivadas elevadas en el origen de coordenadas, y luego de una evaluación previa, se observó que finalizaban el entrenamiento con menor error; y con *capa de salida lineal* para mejorar la distribución de los valores de salida y evitar su concentración en los extremos con una sola neurona. En la *capa de entrada* se utilizó una función identidad con tres neuronas.

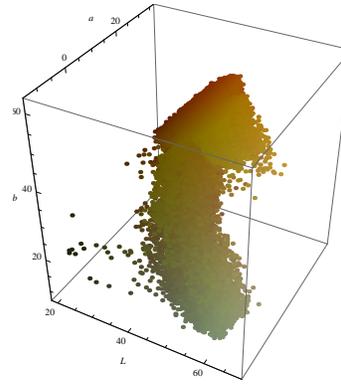
Redes de *una y dos capas ocultas* fueron probadas. La *primer capa oculta* con 20, 40 o 60 neuronas. La segunda capa, de existir, con 5 neuronas.

En el entrenamiento se usó el algoritmo *Levenberg-Marquardt* y una división aleatoria del conjunto de entrenamiento para evitar sobreajuste de datos con 60% para entrenamiento y 20% para validación con los parámetros que siguen:

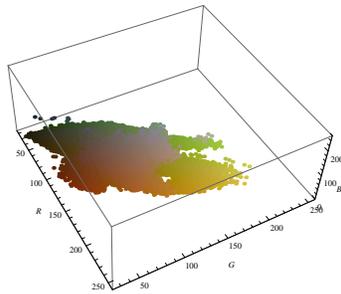
*Épocas:* 50  
*Factor de aprendizaje:* 0,025  
 $\mu$  *máximo:*  $1 \times 10^{10}$   
*Gradiente mínimo:*  $1 \times 10^{-10}$   
*Épocas con incremento del error de validación:* 6  
*Medida de performance:* MSE



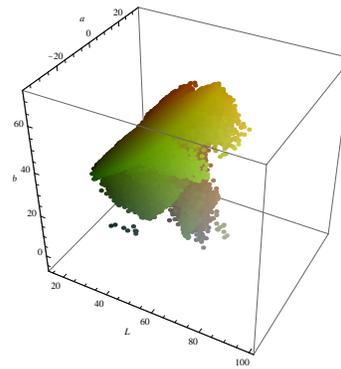
(a) Colores aceptables en  $RGB$



(b) Colores aceptables en  $L^*a^*b^*$ .

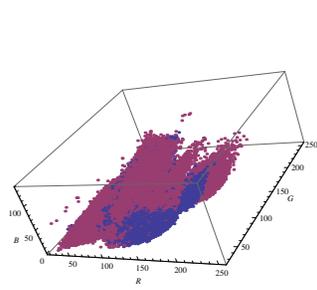


(c) Color no aceptables en  $RGB$

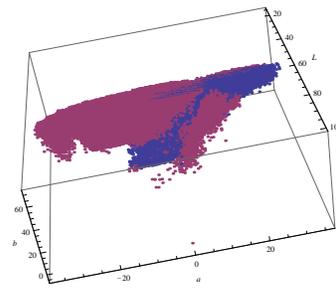


(d) Colores no aceptables en  $L^*a^*b^*$

Figura 1: Colores de entrada presentes en los pares de entrenamiento en distintos espacios de color según la clase de píxel a la que corresponden.



(a) En  $RGB$



(b) En  $L^*a^*b^*$

■ Aceptables    ■ No aceptables

Figura 2: Colores a diferenciar.

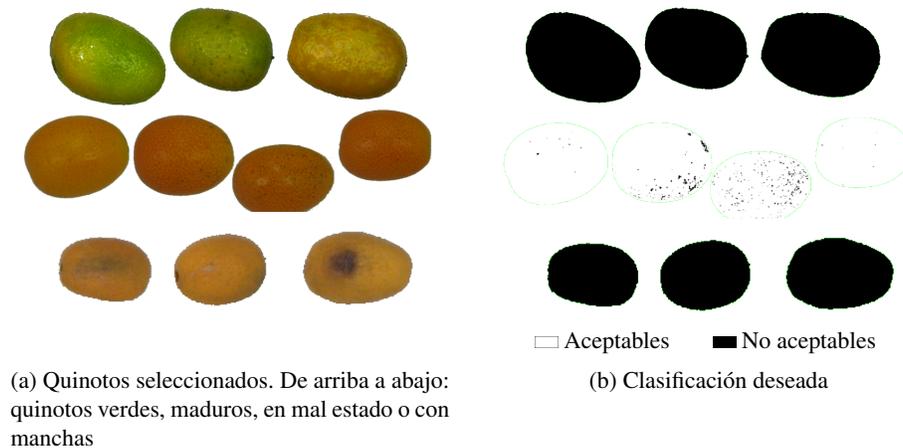


Figura 3: Imágenes de evaluación

## 5. Resultados

### 5.1. Criterio de evaluación

La eficacia de la propuesta para detectar áreas con manchas se evaluó en dos etapas.

En la primera etapa se comparó en *forma general* las distintas configuraciones de redes BPN junto a los espacios color sobre los que trabajaban. El aprendizaje efectuado se evaluó *según la performance* obtenida al finalizar el entrenamiento.

En la segunda etapa, atendiendo que la medida de performance es un promedio, se evaluaron los resultados proporcionados por una red neuronal y el proceso de binarización sobre un conjunto (*colorPixel, aceptabilidad*) diferente al empleado en el entrenamiento. Se dispuso de prototipos de quinotos verdes, maduros y en mal estado cuyas imágenes y clasificación deseada (realizada de la misma forma que para obtener el conjunto de entrenamiento) están presentes en las figuras 3a y 3b. Por cada clase de quinotos se obtuvieron los errores de clasificación y las probabilidades de obtener determinados valores en las salidas.

### 5.2. Performance de las redes neuronales al finalizar entrenamiento

El cuadro 2 presenta los parámetros usados para determinar el fin de entrenamiento en cada red al momento de finalizar la época 50. En todos los casos la finalización se produjo por finalización de épocas estipuladas y no por error de validación.

Las redes neuronales con una capa oculta (cuadro 2a) presentan peores rendimientos que aquellas con dos capas ocultas (cuadro 2b). En ambos casos puede verse que hay una leve diferencia en el rendimiento según la cantidad de neuronas en la primer capa oculta: a mayor cantidad, mejores resultados. Cuando se utilizan datos expresados en  $L^*a^*b^*$  se obtienen resultados con una pequeña mejoría respecto a cuando se usa  $RGB$ .

neuronas en 1er capa oculta	$L^*a^*b^*$	RGB	neuronas en 1er capa oculta	$L^*a^*b^*$	RGB
20	0,0447 (0,06; 0,1)	0,0463 (0,29; 1)	20	0,0370 (0,95; 0,01)	0,0400 (2,86; 0,1)
40	0,0421 (0,05; 1)	0,0455 (0,19; 1)	40	0,0369 (0,28; 0,1)	0,0396 (0,344; 1)
60	0,0409 (0,07; 1)	0,0427 (0,03; 1)	60	0,0367 (1,149; 0,01)	0,0387 (0,76; 1)

(a) 1 capa oculta

(b) 2 capas ocultas

Cuadro 2: Variables de control de entrenamiento a su finalización. Datos en forma *performance* (gradiente,  $\mu$ )

### 5.3. Resultados al utilizar la red 3-60-5-1 con colores expresados en $L^*a^*b^*$

Por presentar el menor error de aprendizaje al finalizar el entrenamiento se escogió la red 3-60-5-1 con entradas expresadas en espacio  $L^*a^*b^*$ .

*Salida de la red* Los datos presentes en las figuras 4a y 4b permiten diferenciar entre los quinotos maduros, y aquellos verdes o con manchas. Los quinotos maduros tienen salidas predominantemente altas; en tanto aquellos verdes o en mal estado, predominantemente bajas.

En la figura 4a se forman picos en torno a los valores 0 y 1. En los quinotos maduros una proporción considerable de píxeles no aceptables presentan salidas elevadas.

*Clasificación* Usando un valor de corte de 0,68 para garantizar que un 90% de los puntos clasificados manualmente como aceptables en los quinotos maduros sean clasificados como aceptables se obtiene la clasificación y los errores de clasificación presentes en la figura 5b y en el cuadro 5a.

Son muy llamativos la presencia de falsos no aceptables en los bordes de los quinotos maduros y el error cometido en la clasificación de los píxeles no aceptables en los quinotos maduros.

*Palabras comunes* Tanto a la salida de la red como en la imagen de clasificación se distinguen defectos presentes en las frutas y reflejos de luz clasificadas como áreas no deseables. Una ampliación de un sector con manchas, como el presente en la figura 6, revela que en gran cantidad de ellas existe al menos un pixel identificado correctamente como no aceptable.

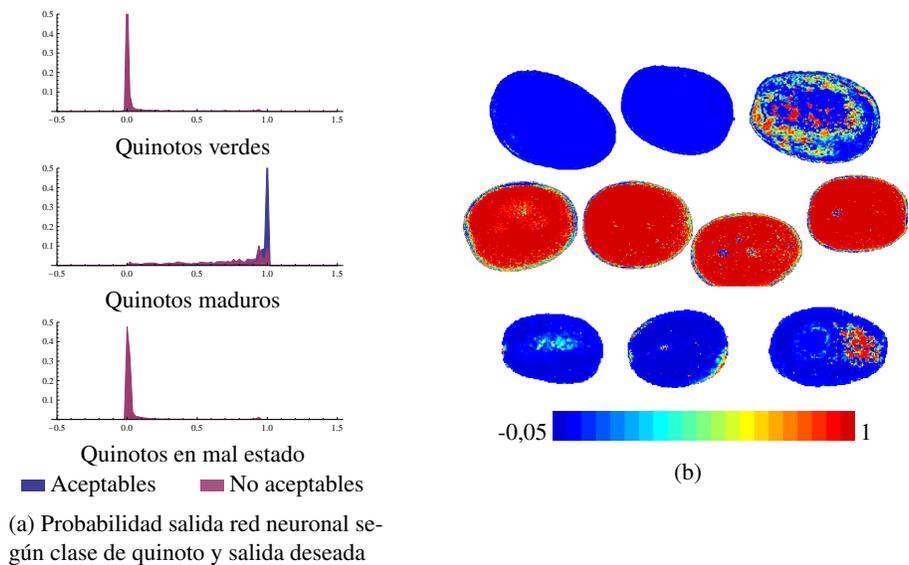


Figura 4: Salida red neuronal

Obtenido/ Deseado	Aceptables	No aceptables
Aceptables	-	-
No aceptables	5,8 %	94,2 %
Quinotos verdes		
Obtenido/ Deseado	Aceptables	No aceptables
Aceptables	90,5 %	9,5 %
No aceptables	57,9 %	42,1 %
Quinotos maduros		
Obtenido/ Deseado	Aceptables	No aceptables
Aceptables	-	-
No aceptables	3,6 %	96,4 %
Quinotos en mal estado		

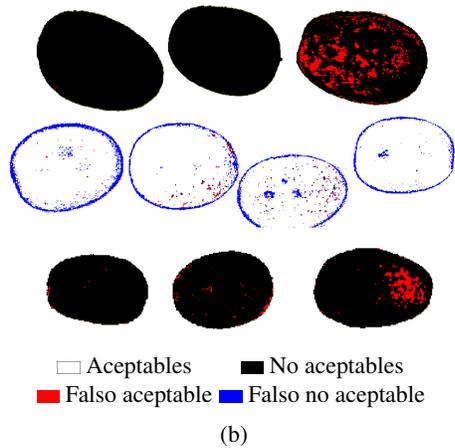


Figura 5: Clasificación de los píxeles con un valor de corte igual a 0,68

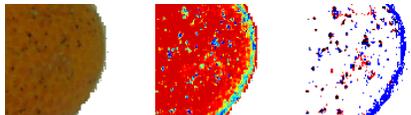


Figura 6: Ampliación de las figuras 3a, 4b y 5b sobre un quinoto maduro

## 6. Conclusiones

Atendiendo a la relativa arbitrariedad con la que se separaron los colores aceptables de aquellos no aceptables, es posible determinar la ubicación de manchas en las frutas y decidir en términos generales si una fruta es aceptable o no usando redes BPN. Siempre que existe una diferencia significativa de color, la clasificación es acertada.

La presencia de píxeles incorrectamente clasificados en las áreas de sombra en frutas maduras puede explicarse por el hecho que esos colores están presentes en mayor medida en áreas no aceptables pertenecientes a frutas en mal estado. La distribución de salidas en las frutas maduras indica que pueden obtenerse otros resultados según los valores de corte empleados, siempre haciendo concesiones según la importancia de los errores a ignorar.

De los resultados obtenidos y la agrupación de los colores en las entradas puede suponerse que con redes neuronales más pequeñas aplicadas sucesivamente con criterios de decisión menos complejos sobre menor variedad de entradas se generarían representaciones más compactas y con mejores resultados. Se podría pensar en la aplicación sucesiva de redes explotando la similitud de colores tal que cada etapa diferencie las áreas maduras de las verdes, las maduras de las pasadas, las sin manchas de las con manchas, y entre las manchas, por ejemplo, aquellas negras de las blanquecinas.

Restaría profundizar en el uso de la teoría de segmentación aplicada sobre la imagen en nivel de grises construida con las salidas de la red, la calidad de resultados empleando distintos espacios color, el empleo de lógica difusa para analizar las salidas y la calidad de salidas obtenibles según las dimensiones de la red.

## Referencias

1. E. Roy Davies. *Image Processing for Food Industry*. World Scientific, 2000.
2. J.L. Jiménez, Ar; Ceres R. y Pons. A survey of computer vision methods for locating fruit on trees. *Transaction of the ASAE*, 2000.
3. DW Du, CJ; Sun. Learning techniques used in computer vision for food quality evaluation: a review. *Journal of Food Engineering*, 2006.
4. D. de Ridder b y H. Handelsc M. Egmont-Petersena. Image processing with neural networks - a review. *Pattern Recognition*, 2002.
5. X. H.; Sun Y. y Jing Li Wang Cheng, H. D. ; Jiang. Color image segmentation: Advances & prospects. *Pattern Recognition*, 2001.
6. González R. y Woods R. *Digital Image Processing*. Prentice Hall, 2002.
7. S. Sezgin, M; Bülent. Survey over image thresholding techniques and quatitative performance evaluation. *Journal of Electronic Imaging*, 13, 2004.
8. P.; Pujol J. y Pujol R. J. Artigas Verde, J. M; Capilla Perea. *Tecnología del color*. Universitat de València, 2002.
9. J.R.; Martínez Hernando V. J. Hilera González. *Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*. Ra-Ma, 1995.
10. J. L. Castro y I. Requena J. M. Bentez. Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, 1997.
11. Matlab neural network toolbox(tm) 6 user's guide. [http://www.mathworks.com/access/helpdesk/help/pdf\\_doc/nnet/nnet.pdf](http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf). Visitado el 11 de junio de 2009.
12. Levenberg-marquardt algorithm. [http://en.wikipedia.org/wiki/Levenberg-Marquardt\\_algorithm.html](http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm.html). Visitado el 20 de junio de 2009.