Reconocimiento de frutas por imágenes

Aprendizaje Automático

Yago Fernández Rego	yago.fernandez.rego
Guillermo Fernández Sánchez	guillermo.fernandezs
Rodrigo Naranjo González	r.naranjo
Adrián Rodríguez López	adrian.rodriguez.lopez

14 de mayo de 2023

Índice

1. Introducción

Las frutas son una maravilla de la naturaleza que nos han acompañado desde tiempos inmemoriales. Además de ser deliciosas, son una excelente fuente de vitaminas, fibra y antioxidantes al igual que un elemento vital para una dieta saludable y equilibrada. A mayores, cuentan también con un importante papel en la cultura popular, la literatura y la mitología. El mundo de las frutas es grande y diverso, cuenta con una gran variedad de sabores, colores, texturas, formas, etc. Tiene una gran importancia tanto desde el punto de vista de la nutrición como de la cultura y economía.

La inteligencia artificial está revolucionando cada vez más el mundo de la agricultura y particularmente, el de las frutas, lo que está ayudando a mejorar la calidad y cantidad de la producción de frutas además de la reducción de los costes. Por ejemplo, el control de calidad de los productos frescos es un problema bastante complejo debido a la gran cantidad y diversidad de ellos que existen. Sin embargo se puede llevar a cabo con facilidad gracias a la inteligencia artificial.

En resumen, el uso de inteligencia artificial en el mundo de las frutas, y, concretamente, el uso un sistema de identificación de frutas puede ayudar en gran medida a los agricultores y productores de alimentos a optimizar la producción de frutas, mejorar la calidad de frutas, reducir el desperdicio, etc.

Nuestro objetivo es desarrollar un problema que identifique distintas frutas a partir de imágenes de las mismas. A pesar de que en un principio solo identificará frutas básicas, en un futuro puede ser de gran utilidad en la industria agrícola y alimentaria en aspectos como el control de calidad, la automatización de la clasificación y etiquetado de las frutas, además de otros posibles usos que puedan surgir en el futuro.

En esta memoria comenzaremos tratando el problema a resolver incluyendo una descripción del mismo, las distintas restricciones de la base de datos, y una breve definición de la misma. Posteriormente se lleva a cabo un breve análisis bibliográfico de artículos similares. Lo siguiente es definir las distintas aproximaciones del problema que se resuelve mediante los siguientes 4 método: RNA (Red de Neuronas Artificiales), SVM (Máquinas de soporte vectorial), KNN (k vecinos más próximos) y árboles de decisión.
Por último, terminaremos la memoria con una solución al problema y un apartado de conclusiones finales y trabajo futuro.

A continuación se muestran todas las siglas de términos que aparecerán a lo largo de este documento:

- **RGB**. Modelo de color aditivo que representa el color mediante la combinación de tres colores primarios: rojo, verde y azul.
- **RNA**. Red de Neuronas Artificiales.
- **SVM**. Support Vector Machines.
- **kNN**. k-Nearest Neighbours.
- **DT**. Decision Tree.

- **BD**. Base de datos.
- **SN**. Segmentation Network.
- **MAcc**. Average Accuracy.
- **MIoU**. Average Intersection Over Union.
- **CNN**. Convolutional Neural Network.
- **VGG**. Visual Geometry Group con 16 capas.
- **RBF**. Radial Basis Function, kernel de función de base radial

2. Problema a resolver

2.1. Descripción

En esta memoria, proponemos un sistema sencillo y eficiente de identificación y clasificación de frutas por medio de un sistema de inteligencia artificial configurado para esta tarea. El objetivo principal de este trabajo es el de aplicar técnicas y metodologías de aprendizaje automático a dos categorías de frutas: manzanas y plátanos. Con el método propuesto, se podrán reconocer las mejores características identificativas de una serie de imágenes preprocesadas, como color y forma, y lograr escalar el sistema para añadir otras frutas en el futuro.

Estas imágenes se han extraído a partir de otra base de datos [1] con un total de 225640 imágenes sobre 262 frutas diferentes, ocupando un total de 7GB. De esta base de datos, se han seleccionado las imágenes más significativas.

En cuanto a las métricas utilizadas, trabajaremos con precisión y F1. La precisión nos es útil para calcular la calidad del sistema realizado ya que se trata de una medida del numero de predicciones acertadas del sistema. Por otra parte el valor de F1 nos permite medir la proporción de positivos entre las clases.

También es importante mencionar la naturaleza RGB del color de las imágenes. Esto significa que se puede expresar mediante tres variables, cuyos valores están comprendidos entre 0 y 255 para el rojo, verde, y azul.

2.2. Restricciones

Para la base de datos propuesta, se han seleccionado las imágenes más significativas, eliminando aquellas que puedan dar lugar a confusión o que no mostraban a los sujetos de interés. Para las imágenes se han impuesto las siguientes restricciones:

- Las imágenes deben ser a color.
- Independientemente de su tamaño u orientación, todas las imágenes deben mostrar al sujeto aislado sobre un fondo blanco.
- No deben presentar daños visibles (deshidratación, plagas, moho, etc.) que puedan alterar la percepción natural de la fruta evaluada.

Figura 1: Ejemplo de imagen aceptada de la base de datos.

- Las frutas deben mostrarse completas, sin cortes o golpes que alteren su forma habitual.
- En caso de frutas que suelan aparecer en grupo (i.e. plátanos, uvas, etc.), se seleccionarán solo aquellas en las que aparezcan en unidades individuales

Para un ejemplo, véase la Figura 1.

2.3. Base de datos

La base de datos consta de 670 imágenes sobre las frutas que queremos distinguir (320 manzanas, 144 plátanos y 206 naranjas), todas ellas cumpliendo con las restricciones descritas anteriormente. Las imágenes están en formato JPG y tienen tamaños variables, con dos núcleos concentrados en los 7 y 30Kb. Con la resolución ocurre algo parecido, siendo la mayoría de 250x250 o 400x250. Más adelante se especificarán las características que se extraerán de las imágenes para su clasificación.

3. Análisis bibliográfico. Estado del arte

3.1. Trabajos relacionados

Como se ha mencionado anteriormente, es de interés el uso que se le puede dar a la inteligencia artificial al ámbito de la agricultura, por lo que no es de sorprender que existan numerosos artículos relacionados con el tema central de nuestra memoria, aunque con ligeras variaciones, como la especialización en identificación de una única fruta, donde se combina la arquitectura GSO (Operadores shuffle), y la SN HR-Net, para crear GSHR-Net, una SN de deep learning que producirá mejores resultados que otras SNs ya conocidas para las métricas MAcc y MIoU (98,835 y 0,8045 respectivamente), además de un coste computacional relativamente bajo;[2] o la puntuación de una fruta según sus características a ojos de clientes usando dos modelos de deep learning: Uno con 6 capas CNN, y otro reentrenando el modelo VGG-16. Los resultados obtenidos fueron una precisión de 99,49 y 99,75 sobre 100 para un dataset con imágenes claras, y de 85,43 y 96,75 respectivamente para otro dataset con imágenes más difíciles de clasificar. [3].

Ya existen artículos centrados en hacer un análisis exhaustivo del estado actual de la identificación y clasificación de frutas (y verduras) [4]. Se identifica una tendencia hacia el uso de redes neuronales convolucionales para la clasificación de frutas, como se puede observar en artículos [5], donde se resalta la utilidad de este tipo de redes neuronales para llevar a cabo la diferenciación de frutas, mas allá de las aparentes similaridades, otros [6] donde partiendo del uso de CNNs se intenta encontrar una arquitectura óptima, incluso otros que van más allá del diseño y entrenamiento del sistema inteligente para proponer implentaciones útiles de cara a la industria

[7]. Existen distintos focos de atención, especialmente en algo que nos interesa mucho, como es la extracción de características, mostrando en casos generales interés por colores, texturas y formas [8].

Referente a estas dos últimas características, observamos que su extracción y análisis suscitan interés por si mismos, fuera de nuestro ámbito de estudio. Para la extracción de texturas existen numerosos algoritmos que consiguen diferenciar que tan plana es la imagen en cuestión [9], utilizando Edge detection. Variaciones de estas técnicas también se utilizan para identificar el borde de las figuras en una imagen [10].

Pese a que el método tradicional que planteamos utilizar no suscita tanto interés, siguen apareciendo artículos analizando y documentando este proceso, centrándose en características como forma de la fruta, análisis del espectrograma o extracción de texturas. De este modo, se emplea una distribución de Laplace usando las características del color HSV para aproximar el tono de la fruta, y la distancia de Malahanobis como una medida de similaridad. Así, se obtienen unos resultados de entre un 93 y un 100 por ciento (dependiendo de la fruta) en un conjunto de 500 imágenes. [11].

Es innegable la utilidad e interés del estudio en esta ámbito específico, aunque si que la mayoría, al menos recientemente, se centra alrededor de las redes neuronales convolucionales.

4. Desarrollo

Al ser un problema amplio y con muchos factores, optaremos por un desarrollo incremental. De esta forma, comenzaremos con una BD reducida sobre la que realizaremos las primeras aproximaciones, incrementándola a medida que se desarrolle el sistema. Esto nos permitirá comenzar con problemas más simples, que podremos extrapolar hasta conseguir resolver el problema con su complejidad real.

4.1. Primera aproximación

4.1.1. Descripción

Tras el análisis en profundidad del problema y de la literatura relacionada, pudimos ver que las características que resultan más útiles para resolverlo son el color, y la forma. Por ello, en esta primera aproximación, optamos por comenzar a diferenciar las frutas por uno de esos rasgos: el color. Para comenzar, establecimos un subproblema acotado, donde nos limitamos a distinguir entre manzanas, de las que contábamos con 320 imágenes; y plátanos, con 142.

Partiendo de nuestra BD, el desarrollo consistía en extraer el color predominante de cada imagen, y añadir cada campo de su representación RGB como atributo que posteriormente analizaríamos. Para ello, para cada imagen seguíamos los siguientes pasos:

1. Primero, cargábamos la imagen a Julia y la convertíamos en una matriz de píxeles CHW (canal, altura, ancho).

2. Después, hacíamos que esa matriz pasara a ser bidimensional, ya que solo importa el píxel y el canal, pero no la distinción entre altura y anchura.

3. Luego, se eliminaban aquellas columnas cuyos valores en los tres canales de color superara 0.7. Esto se hizo para eliminar los píxeles que no contenían ninguna información útil de la imagen, como el fondo blanco.

4. Por último, se obtenía la media de la matriz filtrada, que contenía el vector RGB que indicaba el color dominante de la imagen.

Tras procesar todas las imágenes, obtuvimos un *dataset* similar al siguiente:

0.8754798,0.88267285,0.47707924,banana
0.6674146,0.5980989,0.24419715,banana
⋮
0.4417,0.5481607,0.1362472,manzana
0.46691656,0.5868383,0.1071849,manzana

Así, pudimos ver cómo se distribuyen los colores a lo largo de las frutas, y con eso intentar diferenciarlas. Además, como Julia trabaja con los vectores RGB en un rango del 0 al 1, no hizo falta normalizar las entradas a la hora de realizar las evaluaciones de los modelos.

Sin embargo, de esta aproximación surgieron los siguientes problemas con la BD, que necesitó de alteración. El primero fue que algunas imágenes se demostraban poco contrastadas, lo que causaba valores erráticos. Esto de arregló fácilmente aumentando el contraste de algunas imágenes manualmente, antes de procesarlas. El segundo problema fue que, debido al tamaño exagerado de algunas de las imágenes, los tiempos de procesamiento eran excesivamente altos. Para ello, se comprimieron y redujeron en tamaño las fotos que ocupaban más de 15kB, idealmente dejándolas en un tamaño menor o cercano a 10kB.

Como se puede ver en la Figura 2, donde cada barra representa el color dominante de su respectiva imagen, ambas frutas son diferenciables excepto en algunos casos anómalos que pueden dar lugar a confusión. Por ejemplo, una especie de plátanos de color rojo o plátanos muy maduros con tonos más marrones, o manzanas amarillentas o con un color apagado, más cercano a un gris que a un verde.

(a) Plátanos **(b)** Manzanas

Figura 2: Gráficos para visualizar los datos de esta aproximación.

A la hora de realizar las pruebas, el *dataset* se divide en una relación 90-10, en sets de entrenamiento-test, por medio de validación cruzada. De forma adicional, en la RNA se divide también el set de validación, quedando la relación en 70-20-10 para los sets de entrenamiento-validación-test, también con validación cruzada.

4.1.2. Resultados

Hemos comenzado haciendo una serie de experimentos con un sistema de Redes de Neuronas Artificiales con ocho arquitecturas diferentes, obtuvimos los resultados que se muestran en el Cuadro 1. De aquí podemos ver que los datos tienden a ser similares, probablemente debido a la sencillez de las imágenes, que carecen de ruido. Destacan las arquitecturas [50] y [50 20], con el menor valor F1 pero una precisión similar al resto. Esto se debe, de nuevo por la razón anterior, a que un número tan alto de neuronas intenta optimizar el conjunto de entrenamiento, y llega a clasificar la práctica totalidad de las imágenes como manzanas. Esto se puede ver reflejado en la Figura 3. Concluimos que la arquitectura más prometedora es la de 16 neuronas en una única capa oculta, ya que cuenta con la mayor precisión y el mayor valor F1. Para esta arquitectura, podemos ver la evolución de la precisión y la pérdida en un *fold* en la Figura 4.

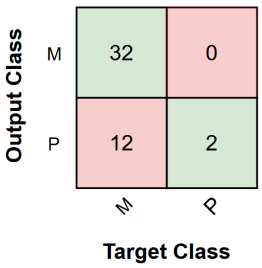

Figura 3: Matriz de confusión con arquitectura [50].

Arquitecturas	Precisión	*F1-Score*
[1]	74.43 % ($\sigma = 2{,}82$)	35.55 % ($\sigma = 6{,}54$)
[4]	75.27 % ($\sigma = 3{,}30$)	44.80 % ($\sigma = 6{,}48$)
[16]	75.75 % ($\sigma = 4{,}05$)	49.42 % ($\sigma = 8{,}23$)
[50]	69.14 % ($\sigma = 0{,}69$)	0.90 % ($\sigma = 1{,}25$)
[1 3]	72.60 % ($\sigma = 1{,}89$)	25.55 % ($\sigma = 5{,}80$)
[4 3]	74.61 % ($\sigma = 2{,}83$)	38.80 % ($\sigma = 5{,}06$)
[16 4]	74.95 % ($\sigma = 3{,}39$)	42.53 % ($\sigma = 7{,}70$)
[50 20]	69.38 % ($\sigma = 0{,}72$)	4.33 % ($\sigma = 2{,}18$)

Cuadro 1: Resultados de la RNA.

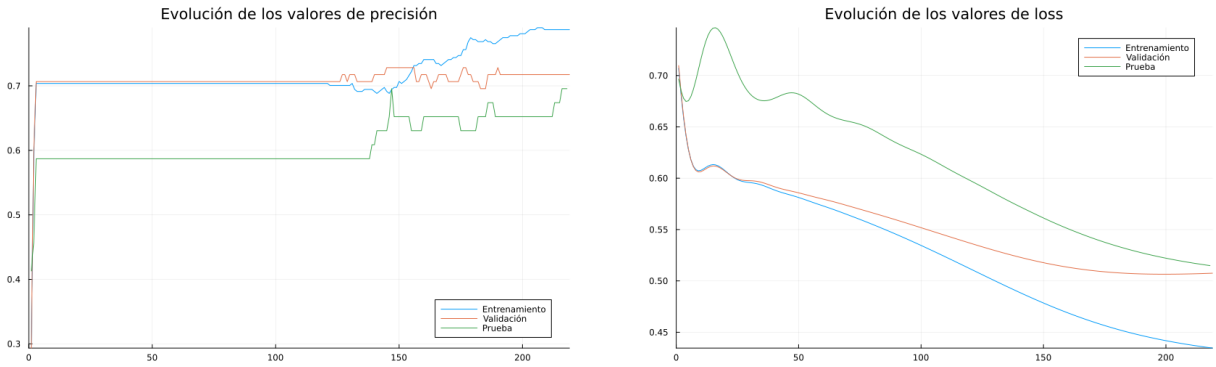

Figura 4: Gráficas de parámetros en la arquitectura [16].

Posteriormente, hemos probado diferentes configuraciones de Máquinas de Vectores de Soporte (SVM) *kernels* con distintos parámetros C y *gamma*. De estos datos, que podemos ver en el Cuadro 2, conc+luimos que los mejores *kernel* son el RBF y el polinómico. Lo sorprendente es que el *kernel* sigmoidal tenga tan mal rendimiento, ya que está mejor preparado para clases binarias. Esto puede deberse a que los parámetros elegidos para el modelo SVM no se ajusten adecuadamente para el *kernel* sigmoidal. Se tendrá esto en cuenta para siguientes aproximaciones.

Cuadro 2: Resultados de la SVM.

Kernel	***C***	γ	**Precisión**	***F1-Score***
rbf	1	2	81.15 %($\sigma = 4{,}74$)	70.64 % ($\sigma = 6{,}58$)
rbf	10	4	83.96 %($\sigma = 4{,}85$)	77.29 % ($\sigma = 5{,}35$)
linear	1	-	79.64 %($\sigma = 4{,}28$)	66.75 % ($\sigma = 6{,}26$)
linear	10	-	80.28 %($\sigma = 3{,}68$)	68.78 % ($\sigma = 6{,}99$)
poly	1	2	81.58 %($\sigma = 5{,}08$)	71.14 % ($\sigma = 7{,}15$)
poly	6	4	85.28 %($\sigma = 5{,}52$)	78.08 % ($\sigma = 7{,}61$)
sigmoid	1	2	46.78 %($\sigma = 7{,}41$)	4.84 % ($\sigma = 7{,}98$)
sigmoid	10	4	47.82 %($\sigma = 7{,}64$)	7.33 % ($\sigma = 7{,}66$)

En el caso del kNN hemos llevado a cabo varias pruebas con distintos valores de k, que varían entre 1 y 20, como se puede observar en el Cuadro 3.

Cuadro 3: Resultados del kNN.

k	**Precisión**	***F1-Score***
1	83.98 % ($\sigma = 4{,}93$)	74.41 % ($\sigma = 8{,}38$)
2	82.01 % ($\sigma = 3{,}92$)	74.77 % ($\sigma = 5{,}22$)
3	81.59 % ($\sigma = 3{,}78$)	69.83 % ($\sigma = 7{,}14$)
5	83.76 % ($\sigma = 4{,}16$)	73.35 % ($\sigma = 6{,}97$)
8	83.55 % ($\sigma = 5{,}011$)	75.18 % ($\sigma = 6{,}54$)
10	84.20 % ($\sigma = 5{,}61$)	76.48 % ($\sigma = 7{,}16$)
20	85.26 % ($\sigma = 5{,}44$)	78.34 % ($\sigma = 6{,}76$)

Por último hemos llevado a cabo un sistema de Árboles de decisión donde se han hecho pruebas con 6 profundidades distintas: 1, 4, 7, 10, 15, y 20, como puede verse en el Cuadro 4. El peor caso (como cabía esperar) es en el cual la profundidad es 1, lo que no permite que se tenga en cuenta la complejidad del problema. La mejor es 7, ya que a diferencia de la anteriormente mencionada, sí tiene la capacidad de responder ante problemas más complejos. Pero a diferencia de las mayores, evita el sobreentrenamiento.

Cuadro 4: Resultados del DT.

Profundidad	**Precisión**	***F1-Score***
1	69.02 % ($\sigma = 7{,}83$)	60.95 % ($\sigma = 5{,}92$)
4	81.81 % ($\sigma = 5{,}32$)	72.14 % ($\sigma = 7{,}43$)
7	82.04 % ($\sigma = 3{,}38$)	72.15 % ($\sigma = 5{,}50$)
10	81.17 % ($\sigma = 5{,}20$)	69.78 % ($\sigma = 7{,}54$)
15	80.96 % ($\sigma = 4{,}73$)	68.46 % ($\sigma = 6{,}83$)
20	80.96 % ($\sigma = 4{,}73$)	68.46 % ($\sigma = 6{,}83$)

4.1.3. Discusión

Tras haber llevado a cabo el sistema con los 4 métodos con distintos valores en cada uno, llegamos a la conclusión de que el mayor problema de nuestro sistema es la escasa diversi-

dad en las características, es decir, se deberían extraer mas características por cada imagen ya que, al extraer únicamente el color, las manzanas amarillas son habitualmente confundidas con plátanos. Viendo los resultados, podemos observar que la RNA es el sistema que peores resultados ofrece, siendo que con los otros tres métodos se obtienen unos resultados bastante similares entre si. Estos resultados son aceptables, aunque mejorables con una extracción de características mas exhaustiva.

Observando la matriz de confusión de la Figura 5, del caso con mayor precisión nos damos cuenta de que el mayor problema de nuestro sistema es la confusión de manzanas por plátanos. Ya que nuestras características se limitan a la obtención del color en las imágenes, las manzanas cuyo color se asemeje al de los plátanos suelen llevar a resultados no deseados. Para futuras aproximaciones procuraremos solucionar esto implementando nuevas características mas allá del color.

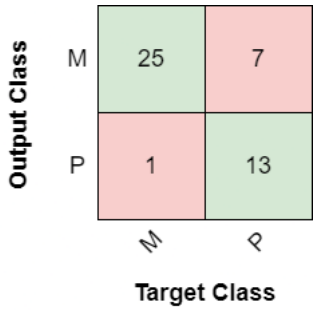

Figura 5: Matriz de confusión con el sistema SVM (kernel = poly; C=6; $\gamma = 4$)

Para una futura aproximación podríamos extraer nuevas características de las imágenes como puede ser, por ejemplo, la forma o simetría de las mismas ya que, en general, la forma de una fruta es una de las características mas distintivas de la misma.

4.2. Segunda aproximación

4.2.1. Descripción

En la primera aproximación vimos el resultado de intentar realizar la clasificación únicamente por el color más predominante. Esto no fue, sin embargo, una forma infalible de resolver el problema, ya que algunas frutas pueden variar de color en función de su madurez, variedad, o frescura. Por ello, en esta aproximación hemos extraído, de la matriz filtrada mencionada en la anterior aproximación, también otras características estadísticas [12] de cada canal de color de la imagen:

- *Desviación típica (σ)*: mide la variación de los valores de los píxeles en la imagen para ver cómo de dispersos son y obtener información sobre la distribución de colores en la imagen. Este valor es útil para detectar detalles en la imagen, como bordes o texturas.
- *Sesgo*: mide la asimetría de la distribución de los valores de los píxeles en la imagen. Si una imagen tiene un sesgo positivo, significa que la mayoría de los valores de los píxeles están en el lado izquierdo de la distribución. Si una imagen tiene un sesgo negativo, significa que la mayoría de los valores de los píxeles están en el lado derecho de la distribución. Un sesgo cercano a cero indica que la distribución es simétrica. Esto es útil para detectar la orientación de ciertos elementos y los patrones que pueda cumplir una fruta en concreto.

En total, se han extraído 9 características para esta aproximación, la media, σ, y sesgo para cada canal individual R, G, B. Al igual que en la anterior aproximación, estos valores no necesitan de normalización, pues Julia normaliza todos los canales a la hora de manipular las imágenes. Además, tuvimos nuevamente que ajustar algunas imágenes manualmente.

Para aumentar las posibles clasificaciones, introdujimos un nuevo tipo de fruta a clasificar: la naranja. Consideramos esta fruta una buena opción para probar estas, y futuras, características debido a su forma similar a la manzana, su color diferenciado, y su textura rugosa.

Como en la anterior aproximación, se realiza de igual manera validación cruzada para dividir el *dataset* en una relación 70-20-10 para los sets de entrenamiento-validación-test.

4.2.2. Resultados

Al realizar nuevos experimentos con las distintas arquitecturas de Redes de Neuronas Artificiales, obtuvimos los resultados que se muestran en el Cuadro 5. Se puede ver, en comparación a la anterior aproximación, que hemos evitado en mayor medida las arquitecturas de más de una capa oculta, pues no aportaban suficiente información útil y mostraban la peor precisión en todas las ocasiones. Este comportamiento se ve de nuevo reflejado en las arquitecturas [5 3] y [8 5], las únicas arquitecturas de dos capas con las que hemos experimentado. Ambas presentan una importante caída en precisión y F1 con respecto al resto de arquitecturas (a excepción de la [1], que es representada de forma simbólica). No creemos que el problema en su aproximación actual sea lo suficientemente complejo como para necesitar de más de una capa oculta, pero seguiremos experimentando a medida que se añadan nuevas características.

En cuestión al resto de arquitecturas, se puede ver que la mejor en cuestión de precisión y F1 es la [5]. Tiene, además, de las menores desviaciones típicas de la tabla, lo que indica una menor variabilidad en los resultados y sugiere a esta como la arquitectura más estable. Para esta arquitectura se han obtenido la matriz de confusión en la Figura 6, además de dos gráficas (Figura 7) que representan la evolución de los valores de pérdida y precisión a lo largo de un *fold*. También podemos ver una mejoría en estas gráficas con respecto a la Figura 4 de la anterior aproximación.

Por lo general, encontramos bastante homogeneidad a lo largo del resto de valores, que empeoran cuanto más se alejan de la arquitectura de cinco neuronas en una capa oculta, pero no de forma considerable.

Figura 6: Matriz de confusión con arquitectura [5].

Arquitecturas	Precisión	*F1-Score*
[1]	65.85 % ($\sigma = 2{,}81$)	57.56 % ($\sigma = 2{,}33$)
[4]	81.22 % ($\sigma = 4{,}32$)	81.04 % ($\sigma = 4{,}36$)
[5]	81.44 % ($\sigma = 4{,}00$)	81.32 % ($\sigma = 4{,}05$)
[6]	81.12 % ($\sigma = 4{,}27$)	80.97 % ($\sigma = 4{,}32$)
[8]	80.76 % ($\sigma = 4{,}41$)	80.59 % ($\sigma = 4{,}47$)
[16]	79.89 % ($\sigma = 4{,}40$)	79.70 % ($\sigma = 4{,}38$)
[5 3]	77.12 % ($\sigma = 3{,}72$)	75.64 % ($\sigma = 4{,}11$)
[8 5]	76.51 % ($\sigma = 4{,}57$)	75.43 % ($\sigma = 4{,}82$)

Cuadro 5: Resultados de la RNA.

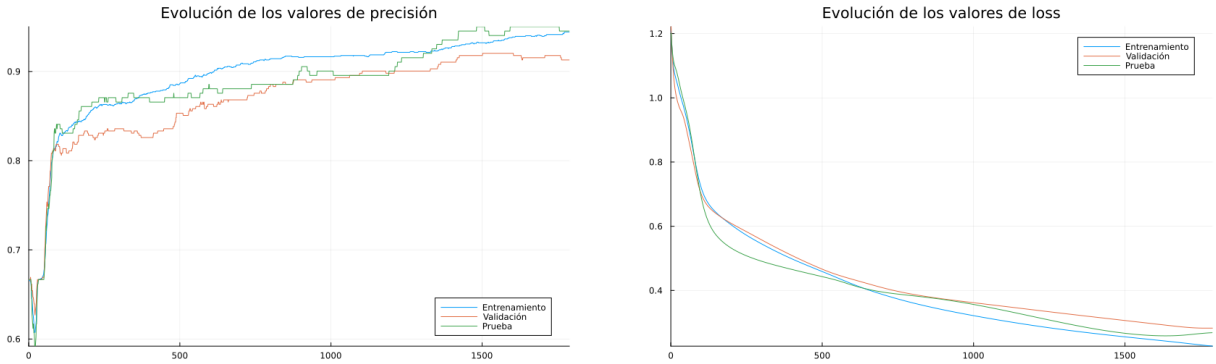

Figura 7: Gráficas de parámetros en la arquitectura [5].

En SVM, el mayor problema con el que nos encontramos en la primera aproximación era la obtención de valores de C y γ. En este caso para obtener una aproximación de dónde se encontraban los mejores valores de C y γ, hemos empezado probando una escala logarítmica entre 10E-3 y 10E3 de estos dos valores para cada uno de los kernels. Posteriormente se han ido probando valores, moviendo los parámetros en función de los resultados obtenidos. Los resultados finales se muestran en la siguiente tabla.

Cuadro 6: Resultados de la SVM.

Kernel	***C***	γ	**Precisión**	***F1-Score***
rbf	11	4	91,93 %($\sigma = 4{,}00$)	91.83 % ($\sigma = 4{,}21$)
rbf	10	4	91,93 %($\sigma = 3{,}64$)	91.85 % ($\sigma = 3{,}75$)
rbf	4	6	91,33 %($\sigma = 3{,}54$)	91.24 % ($\sigma = 3{,}80$)
linear	15	-	83.54 % ($\sigma = 4{,}80$)	83.76 % ($\sigma = 4{,}57$)
linear	24	-	83.84 % ($\sigma = 5{,}20$)	84.02 % ($\sigma = 5{,}04$)
linear	26	-	83.69 % ($\sigma = 5{,}23$)	83.83 % ($\sigma = 5{,}13$)
poly	6	4	89.09 % ($\sigma = 4{,}35$)	89.14 % ($\sigma = 4{,}52$)
poly	4	6	88.94 % ($\sigma = 4{,}02$)	89.07 % ($\sigma = 4{,}06$)

En general podemos apreciar que hemos obtenido unos mejores resultados que los obtenidos en la anterior aproximación. En cuanto a la precisión, las mayores mejoras se producen con los Kernels rbf (10 %) y poly (8 %), mientras que en el caso del F1, la mejora es similar en los 3 casos, con unos resultados un 18-20 % más altos. Debido a los malos resultados, hemos decidido prescindir del Kernel sigmoid ya que los resultados, aunque fueron algo mejores, fueron similares a la anterior aproximación, (Precisión - 47,91 % y F1 - 31,04 %). También debemos destacar el descenso de desviación típica en la Precisión y F1 de todos los casos (un 1-3 %), lo cual es positivo.

En el caso del sistema kNN hemos vuelto a realizar varias pruebas con 8 valores distintos de k que varían entre 1 y 21. En este caso hemos optado por probar mayoritariamente con valores impares ya que con ellos observamos mejores resultados que los pares, en general. Los resultados obtenidos para cada valor de k se pueden observar en el Cuadro 7. Viendo el Cuadro de resultados podemos llegar a la conclusión de que el mejor valor de k para nuestro sistema actualmente es 9.

Cuadro 7: Resultados del kNN.

k	Precisión	*F1-Score*
1	87.57 % ($\sigma = 3{,}71$)	87.35 % ($\sigma = 4{,}13$)
2	87.13 % ($\sigma = 4{,}27$)	87.51 % ($\sigma = 4{,}01$)
3	86.83 % ($\sigma = 4{,}75$)	86.75 % ($\sigma = 4{,}91$)
5	87.28 % ($\sigma = 4{,}42$)	87.17 % ($\sigma = 4{,}62$)
9	86.23 % ($\sigma = 3{,}73$)	86.25 % ($\sigma = 3{,}67$)
13	86.82 % ($\sigma = 4{,}39$)	86.78 % ($\sigma = 4{,}35$)
21	85.33 % ($\sigma = 4{,}04$)	85.38 % ($\sigma = 3{,}98$)

Sin embargo, si comparamos los resultados obtenidos con los de la anterior aproximación, comprobamos con facilidad que los valores de precisión del sistema actual son notablemente menores. En cambio en los valores de F1 si notamos una leve mejora respecto a la primera aproximación. Esto se puede deber al uso de una base de datos mas grande (debido a la inclusión de las naranjas).

Por ultimo, al igual que en la anterior aproximación,realizamos varios experimentos con un sistema de Árboles de Decisión, variando el valor de profundidad. Como puede verse en el Cuadro 8 se percibe una mejora con respecto a la aproximación anterior, incluso habiendo añadido una nueva fruta. Se puede apreciar que los patrones anteriores se mantienen. La peor arquitectura sigue siendo la primera. Al añadir un nuevo tipo de frutas, la arquitectura tiene aún menos capacidad para generar una solución que capture la complejidad del problema. A medida que aumenta la profundidad, vemos que los resultados son paulatinamente mejores, siendo aquella con profundidad 6 la mejor de entre las que se han probado. Al igual que en la anterior aproximación, con una profundidad máxima demasiado alta, el sistema tiende a sobre-entrenar y con una menor capacidad de generalización, obtienen peores resultados.

Cuadro 8: Resultados del DT.

Profundidad	Precisión	*F1-Score*
1	61.23 % ($\sigma = 4{,}44$)	55.16 % ($\sigma = 4{,}06$)
4	82.49 % ($\sigma = 3{,}77$)	82.56 % ($\sigma = 3{,}77$)
6	82.79 % ($\sigma = 4{,}41$)	82.84 % ($\sigma = 4{,}28$)
7	82.18 % ($\sigma = 3{,}60$)	82.32 % ($\sigma = 3{,}59$)
10	81.45 % ($\sigma = 4{,}29$)	81.40 % ($\sigma = 4{,}22$)
15	81.44 % ($\sigma = 4{,}39$)	81.36 % ($\sigma = 4{,}37$)

Como nota sobre los valores de profundidad utilizados, se eliminó el valor 20 a favor de añadir el valor 6. Se decidió este cambio ya que a partir de 10, los resultados empeoraban, mientras que en el rango [4, 7] parecía encontrarse el punto óptimo. Tanto es así que la mejor arquitectura de árboles de decisión es la que toma como valor de profundidad máxima 6.

4.2.3. Discusión

Observando los resultados obtenidos podemos notar una gran mejora respecto a la anterior aproximación, incluso añadiendo una nueva clase. Todos los sistemas implementados rondan

una precisión entre 70 % y 90 %. Analizando los diferentes datos de cada método, encontramos uno que destaca: se puede ver claramente que la SVM da los mejores resultados, concretamente con los siguientes parámetros:

- *Kernel*: rbf
- $C = 10$
- $\gamma = 4$

Alcanzando una precisión del 91.93 % y un F1 del 91.85 %. En la Figura 8 podemos ver una matriz de confusión de uno de los *folds* con estos parámetros. Podemos comprobar a simple vista que las naranjas las reconoce con facilidad, sin embargo el problema persiste entre las manzanas y los plátanos, donde en algunos casos no logra diferenciarlos, probablemente sean casos donde ambas frutas presentan un color muy similar.

Figura 8: Matriz de confusión con el sistema SVM (kernel = rbf; C=10; $\gamma = 4$)

Por otra parte, podemos percibir una notable mejora en los otros tres sistemas, tanto en RNA como en Árboles de Decisión y kNN, donde nos encontramos con mejoras de un 5 % en cada caso aproximadamente

Nuestro objetivo para la siguiente aproximación será extraer las características de los bordes y la textura de las frutas, características que, en muchos casos, las definen, y ver cómo afectan esos nuevos parámetros a la hora de obtener resultados a partir de un *dataset* más completo.

4.3. Tercera aproximación

4.3.1. Descripción

En esta tercera aproximación hemos extraído nuevas características por medio la obtención de los bordes y detalles de las imágenes. Para ello, hemos probado varios métodos y hemos optado por el siguiente:

- **Método de Laplace**: el método de Laplace es un operador de segundo orden que busca cambios en la curvatura de la imagen para así resaltar las regiones donde estos cambios se produzcan de forma súbita. Esto permite obtener los bordes y detalles de una imagen, de la que podremos extraer características sobre la estructura y las similitudes entre, en

nuestro caso concreto, frutas. Para ello, aplicamos a la imagen objetivo, y trabajando con ella en escala de grises, el siguiente *kernel* de convolución:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Este operador se conoce como operador Laplaciano de Gauss, y se deriva del operador de Laplace común. Se diferencia en que, a diferencia del segundo, este es anisotrópico y utiliza una matriz de pesos que sigue una distribución gaussiana que permite suavizar la imagen antes de realizar los cálculos. Optamos por este operador de detección ya que, a diferencia del operador de Sobel o Canny, se requiere de mayor sensibilidad, debido a la naturaleza de las imágenes, y es más rápido. Así, obtenemos la media y σ de la imagen resultante del proceso (Figura 9).

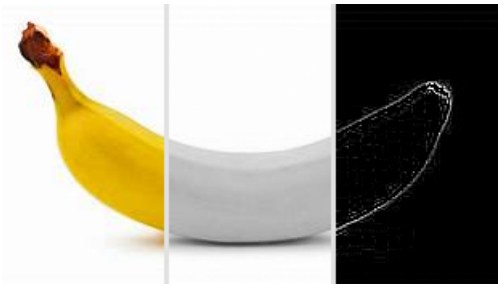

Figura 9: Fases de una imagen por el método de Laplace.

Con todo esto, y lo obtenido en anteriores aproximaciones, tenemos 11 características para resolver nuestro problema: la media, σ, y sesgo del canal RGB; y la media y σ de la imagen obtenida por el método de Laplace. Al igual que en otras aproximaciones, se mantiene la relación 70-20-10 para los sets de entrenamiento-validación-test obtenidos mediante validación cruzada. Los datos introducidos, al estar ya en un valor entre 0 y 1, no han necesitado normalización.

4.3.2. Resultados

Al realizar nuevos experimentos con las distintas arquitecturas de Redes de Neuronas Artificiales, obtuvimos los resultados que se muestran en el Cuadro 9. Hemos mantenido las mismas arquitecturas que en la anterior aproximación para poder realizar mejores comparaciones, y ver cómo han podido influir los nuevos atributos en el entrenamiento. Como se puede ver en el Cuadro, en esta aproximación han mejorado notablemente ambas métricas, aumentando todas las arquitecturas un 5 % de precisión.

Encontramos el primer caso en el que las arquitecturas de dos capas ocultas superan a las de una sola capa en la arquitectura [16 8], respondiendo de forma correcta al aumento de complejidad del problema y justificando esta vez la capa oculta adicional. Se puede ver la matriz de confusión resultante del set de test en la Figura 10. Comparada con la de la anterior aproximación (Figura 6), podemos constatar cómo han influido los nuevos atributos. Previamente, las naranjas causaban mucha confusión tanto para los plátanos como para las manzanas. Sin embargo, esta vez gozan de un 100 % de precisión. Lo que nos lleva a pensar que esto se da gracias a su distinguible textura, mostrando mayor media en la imagen resultante del método

de Laplace, y que permite diferenciarlas con mucha más facilidad del resto.

En cuestión al resto de arquitecturas, y como ya se ha mencionado, han mejorado todas. También se han acercado todas entre sí, aumentando la homogeneidad ya presente anteriormente. Esta vez, incluso las arquitecturas de dos capas ocultas se han mantenido en la línea del resto.

En próximas aproximaciones, probaremos más combinaciones de arquitecturas de dos capas y seguiremos experimentando con los valores y número de neuronas que, por lo que se ha podido ver en esta aproximación, han ido en aumento a lo largo de la investigación.

Figura 10: Matriz de confusión con arquitectura [16 8].

Arquitecturas	Precisión	*F1-Score*
[1]	72.01 % ($\sigma = 2{,}51$)	64.54 % ($\sigma = 2{,}17$)
[4]	85.89 % ($\sigma = 2{,}76$)	85.79 % ($\sigma = 2{,}77$)
[5]	86.45 % ($\sigma = 2{,}40$)	86.32 % ($\sigma = 2{,}43$)
[6]	86.64 % ($\sigma = 2{,}61$)	86.50 % ($\sigma = 2{,}70$)
[8]	86.95 % ($\sigma = 2{,}54$)	86.81 % ($\sigma = 2{,}60$)
[16]	87.67 % ($\sigma = 2{,}36$)	87.55 % ($\sigma = 2{,}49$)
[5 3]	85.93 % ($\sigma = 2{,}09$)	85.58 % ($\sigma = 2{,}33$)
[8 5]	87.63 % ($\sigma = 2{,}02$)	87.51 % ($\sigma = 2{,}14$)
[16 8]	87.89 % ($\sigma = 2{,}13$)	87.79 % ($\sigma = 2{,}24$)

Cuadro 9: Resultados de la RNA.

En SVM, hemos empleado la misma estrategia que en las aproximaciones anteriores, es decir, una escala logarítmica y posteriormente probar valores en función de los resultados que se van obteniendo. La tabla resultante es la siguiente.

Cuadro 10: Resultados de la SVM.

Kernel	*C*	γ	Precisión	*F1-Score*
rbf	3	4	92.20 %($\sigma = 2{,}45$)	92.16 % ($\sigma = 2{,}57$)
rbf	4	3	92.05 %($\sigma = 2{,}39$)	91.98 % ($\sigma = 2{,}56$)
rbf	2	5	92.05 %($\sigma = 2{,}15$)	91.99 % ($\sigma = 2{,}28$)
linear	22	-	84.88 % ($\sigma = 3{,}94$)	84.93 % ($\sigma = 3{,}75$)
linear	23	-	84.88 % ($\sigma = 3{,}94$)	84.92 % ($\sigma = 3{,}75$)
linear	24	-	84.73 % ($\sigma = 3{,}83$)	84.78 % ($\sigma = 3{,}63$)
poly	4	1	90.11 % ($\sigma = 2{,}54$)	89.97 % ($\sigma = 2{,}80$)
poly	9	1	90.11 % ($\sigma = 2{,}43$)	90.04 % ($\sigma = 2{,}58$)
sigmoid	8	1	64.48 % ($\sigma = 5{,}50$)	64.00 % ($\sigma = 6{,}17$)

Los resultados han mejorado ligeramente con respecto a la anterior aproximación, tanto la precisión como el F1. Los mejores resultados se han vuelto a obtener con el kernel rbf, con

una ligera mejora de un 0,3 %. En los kernels linear y poly la mejora ha sido de un 1 % en ambas métricas. También debemos destacar los resultados del kernel sigmoid, que aunque son inferiores al resto de kernels, han presentado mejoras sustanciales con respeto a la aproximación anterior, por lo que también debemos de tenerlos en cuenta. En cuanto a la desviación típica, volvemos a tener buenas noticias, ya que desciende de forma considerable en todos los casos (1-3 %), lo que significa que tenemos un sistema más estable.

A continuación hemos realizado, al igual que en las anteriores aproximaciones, experimentos con distintos valores de k en el sistema basado en el método de knn. En el Cuadro 11 se muestran los resultados obtenidos de dichos experimentos. Los valores resultantes no son muy distintos a los obtenidos en la anterior aproximación, aunque si ha habido una muy leve mejora en los valores de precisión y F1 de hasta un 1 % o 2 % en los mejores casos. También notamos una leve mejora en los valores de σ, que son ligeramente inferiores a los obtenidos en experimentos anteriores.

Cuadro 11: Resultados del kNN.

k	**Precisión**	***F1-Score***
1	87.43 % ($\sigma = 2{,}01$)	87.44 % ($\sigma = 1{,}93$)
3	87.60 % ($\sigma = 4{,}27$)	87.52 % ($\sigma = 4{,}24$)
4	88.15 % ($\sigma = 3{,}03$)	88.20 % ($\sigma = 2{,}96$)
5	88.68 % ($\sigma = 2{,}94$)	88.62 % ($\sigma = 2{,}95$)
7	87.97 % ($\sigma = 2{,}86$)	87.88 % ($\sigma = 2{,}94$)
9	86.72 % ($\sigma = 3{,}72$)	87.20 % ($\sigma = 3{,}72$)
13	87.25 % ($\sigma = 3{,}62$)	87.06 % ($\sigma = 3{,}71$)

Por ultimo, como siempre, realizamos varios experimentos con un sistema de Árboles de Decisión, variando el valor de profundidad. Como puede verse en el Cuadro 20, no hay cambios significativos con respecto a la aproximación anterior. Los resultados se mantienen con una variación no mayor de 1 % en ningún caso. Los patrones anteriores se mantienen idénticos: La peor arquitectura sigue siendo la primera, la arquitectura no tiene suficiente capacidad para generar una solución que capture la complejidad del problema. A medida que aumenta la profundidad, mejoran los resultados, siendo aquella con profundidad 6 la mejor de entre las que se han probado. Al igual que en las anteriores aproximaciones, con una profundidad máxima demasiado alta, el sistema tiende a sobre-entrenar y con una menor capacidad de generalización, obtienen peores resultados.

Cuadro 12: Resultados del DT.

Profundidad	**Precisión**	***F1-Score***
1	61.17 % ($\sigma = 3{,}7$)	55.22 % ($\sigma = 3{,}34$)
4	82.76 % ($\sigma = 3{,}42$)	82.46 % ($\sigma = 3{,}51$)
6	83.81 % ($\sigma = 5{,}10$)	83.97 % ($\sigma = 5{,}02$)
7	82.75 % ($\sigma = 4{,}12$)	82.85 % ($\sigma = 4{,}05$)
10	81.99 % ($\sigma = 4{,}27$)	81.99 % ($\sigma = 4{,}33$)
15	81.82 % ($\sigma = 4{,}11$)	81.80 % ($\sigma = 4{,}14$)

4.3.3. Discusión

Tras ver los resultados obtenidos mediante cada uno de los sistemas implementados llegamos a la conclusión de que hemos obtenido unos resultado muy similares a los de la anterior aproximación con leves mejoras en algunos casos. A continuación, la Figura 11 muestra una matriz de confusión del caso cuyo porcentaje de precisión es mas alto (SVM kernel = rbf; C=3; $\gamma = 4$). En dicha matriz de confusión vemos que el problema comentado en la anterior aproximación persiste, donde las naranjas son identificadas con exactitud sin embargo el sistema tiene problemas para distinguir manzanas y plátanos.

Figura 11: Matriz de confusión con el sistema SVM (kernel = rbf; C=3; $\gamma = 4$)

Viendo los resultados obtenidos tras implementar las nuevas características llegamos a la conclusión de que quizás no utilizamos el mejor método para extraer los bordes de las frutas. Para la siguiente aproximación procuraremos mejorar la extracción de características u optar por otro tipo de características como bien puede ser la simetría, ya que las manzanas presentan una simetría vertical y horizontal, pero, en cambio, los plátanos solo presentan simetría horizontal.

4.4. Cuarta aproximación

4.4.1. Descripción

Como cuarta aproximación, hemos comprobado la simetría horizontal y vertical de las imágenes, primero identificando su *bounding box* para después recortar el espacio sobrante alrededor de la ventana y quedarse con el sujeto centrado. Posteriormente, obtenemos esa imagen reflejada vertical y horizontalmente, para después calcular la media y σ de la diferencia entre la imagen original y cada uno de sus reflejos (véase la Figura 12).

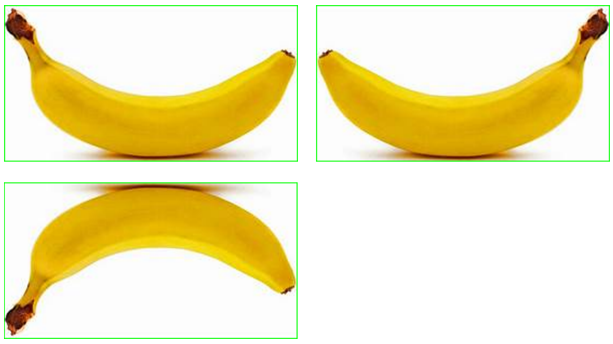

Figura 12: Representación gráfica de la *bounding box* del sujeto y sus imágenes reflejadas.

Por medio de validación cruzada se han obtenido los sets de entrenamiento-validación-test en un ratio 70-20-10, como hemos hecho ya anteriormente. Cabe notar que los datos introducidos en la RNA fueron normalizados usando ZeroMean, mientras que en el resto se ha realizado con MinMax, ya que utilizar este tipo de normalización en las RNA las hacía menos competitivas, con valores hasta un 20 % peores.

4.4.2. Resultados

Comenzando con las RNA, como de costumbre, encontramos, de nuevo, una clara mejoría con respecto a la anterior aproximación, como se puede ver en el Cuadro 13. Por lo general, las métricas crecen hasta un 6 % con respecto a los entrenamientos anteriores, y las desviaciones típicas bajan en su mayoría. También notamos la escasa variación entre arquitecturas, cuyos resultados son cada vez más homogéneos.

Como mencionamos en la aproximación anterior, hemos experimentado con un mayor número de arquitecturas de más de una capa. Pero los resultados varían muy poco, incluso en la arquitectura con tres capas ocultas. Siguen estas arquitecturas así la tendencia del resto de experimentos, con una variación de 0.5 % únicamente.

Esta vez la mejor arquitectura ya no es una de dos capas ocultas, si no que, al igual que en la primera aproximación, la mejor arquitectura es la de 16 neuronas en una única capa oculta. Como se puede ver en la Figura 13, se han reducido los errores al diferenciar plátanos y manzanas, ya que el plátano es simétrico horizontalmente pero la manzana es simétrica tanto horizontal como verticalmente. Comprobamos así la buena aplicación de los nuevos atributos y justificamos el éxito de esta aproximación en este algoritmo.

Sin embargo, la RNA falla al diferenciar entre una manzana y una naranja. Podemos suponer que la simetría tan similar entre una fruta y otra lleve a la confusión acerca de cuál es cuál. En siguientes aproximaciones intentaremos resolver estos problemas por medio del estudio más profundo de las texturas, extendiendo de esta forma las características extraídas en la aproximación anterior.

Figura 13: Matriz de confusión con arquitectura [16].

Arquitecturas	**Precisión**	***F1-Score***
[1]	71.00 % ($\sigma = 2{,}61$)	63.15 % ($\sigma = 2{,}31$)
[4]	91.80 % ($\sigma = 2{,}25$)	91.73 % ($\sigma = 2{,}34$)
[8]	92.87 % ($\sigma = 2{,}15$)	92.83 % ($\sigma = 2{,}22$)
[16]	93.21 % ($\sigma = 2{,}01$)	93.17 % ($\sigma = 2{,}07$)
[8 5]	92.77 % ($\sigma = 2{,}06$)	92.73 % ($\sigma = 2{,}12$)
[16 8]	92.85 % ($\sigma = 2{,}24$)	92.81 % ($\sigma = 2{,}27$)
[16 12]	92.73 % ($\sigma = 1{,}93$)	92.68 % ($\sigma = 2{,}00$)
[32 16]	92.43 % ($\sigma = 2{,}07$)	92.39 % ($\sigma = 2{,}15$)
[32 16 8]	92.26 % ($\sigma = 2{,}14$)	92.18 % ($\sigma = 2{,}20$)

Cuadro 13: Resultados de la RNA.

Para las SVM, hemos empleado una vez más la estrategia de las aproximaciones anteriores, es decir, elaborar una escala logarítmica y posteriormente probar valores en función de los resultados que se van obteniendo. La tabla resultante es la siguiente.

Cuadro 14: Resultados de la SVM.

Kernel	***C***	γ	**Precisión**	***F1-Score***
rbf	3	2	95.34 %($\sigma = 2{,}53$)	95.32 % ($\sigma = 2{,}53$)
rbf	7	2	95.19 %($\sigma = 2{,}66$)	95.19 % ($\sigma = 2{,}64$)
rbf	15	1	95.20 %($\sigma = 2{,}64$)	95.18 % ($\sigma = 2{,}66$)
linear	4	-	91.01 % ($\sigma = 3{,}04$)	90.91 % ($\sigma = 3{,}07$)
linear	5	-	90.70 % ($\sigma = 3{,}21$)	90.59 % ($\sigma = 3{,}23$)
linear	26	-	90.85 % ($\sigma = 3{,}03$)	90.75 % ($\sigma = 3{,}02$)
poly	1	1	94.16 % ($\sigma = 3{,}54$)	94.13 % ($\sigma = 3{,}60$)
poly	2	1	94.01 % ($\sigma = 2{,}53$)	94.00 % ($\sigma = 2{,}57$)

Esta vez notamos una clara mejoría en la precisión y F1 con respecto a la anterior aproximación. La mejora más destacable se da con el kernel linear, pasando de un 85 % a hasta un 91 %. Aún así, los mejores resultados se siguen dando con el kernel rbf, superando ya la barrera del 95 %. Sin embargo, no podemos decir lo mismo del kernel sigmoid, del que hemos vuelto a prescindir ya que nos ha dado malos resultados. En cuanto a la desviación típica, tenemos buenas y malas noticias: Buenas ya que en el caso del kernel linear disminuye en un 1 %, pero aumenta ligeramente en los kernels poly y rbf. Aún así, la valoración final de los resultados es positiva.

Una vez mas, llevamos a cabo los respectivos experimentos en el sistema basado en el método de knn. En el Cuadro 15 se muestran los resultados obtenidos de dichos experimentos. Los valores resultantes Muestran un notable aumento en el valor de precisión para todos los valores de k alcanzando un valor maximo de 93.78 %, aumentando un 4 % desde la anterior aproximación. Tambien destacamos un incremento similar en el valor de F1

Cuadro 15: Resultados del kNN.

k	Precisión	*F1-Score*
1	92.64 % ($\sigma = 2{,}60$)	92.58 % ($\sigma = 2{,}67$)
3	93.78 % ($\sigma = 2{,}52$)	93.02 % ($\sigma = 2{,}54$)
4	92.34 % ($\sigma = 3{,}20$)	92.33 % ($\sigma = 3{,}19$)
5	91.90 % ($\sigma = 2{,}66$)	91.75 % ($\sigma = 2{,}67$)
7	92.18 % ($\sigma = 2{,}86$)	92.03 % ($\sigma = 3{,}59$)
9	92.34 % ($\sigma = 3{,}24$)	92.17 % ($\sigma = 3{,}29$)
13	91.89 % ($\sigma = 3{,}50$)	91.69 % ($\sigma = 3{,}64$)

Por ultimo, como siempre, realizamos varios experimentos con un sistema de Árboles de Decisión, variando el valor de profundidad. Como puede verse en el Cuadro 20, existe una mejora relativa centrada en los mejores casos, llegando al 87 % en el mejor de los casos. La distribución de los resultados se mantiene, incluso volviéndose mas notable. Cabe mencionar que en todos los casos se aprecia una reducción de σ.

Cuadro 16: Resultados del DT.

Profundidad	Precisión	*F1-Score*
1	59.67 % ($\sigma = 2{,}62$)	52.82 % ($\sigma = 3{,}82$)
4	82.84 % ($\sigma = 3{,}19$)	86.45 % ($\sigma = 3{,}22$)
6	87.81 % ($\sigma = 3{,}5$)	87.64 % ($\sigma = 3{,}5$)
7	87.26 % ($\sigma = 3{,}26$)	87.19 % ($\sigma = 3{,}28$)
10	86.82 % ($\sigma = 3{,}9$)	86.79 % ($\sigma = 3{,}9$)
15	86.83 % ($\sigma = 4{,}21$)	86.80 % ($\sigma = 4{,}24$)

4.4.3. Discusión

Observando los resultados obtenidos podemos concluir con que esta aproximación supone una mejora importante sobre todas las arquitecturas con las que trabajamos. Siguen destacando las SVM como el sistema que mejor realiza este trabajo, aunque rondan valores superiores al 80 % Es esperable que en todos los sistemas los plátanos ahora sean más fácilmente distinguibles de las naranjas y las manzanas, habiendo entre estas dos últimas algún caso en el que debido a la similitud en forma se puedan confundir como podemos observar en la matriz de confusión de la Figura 14. Para evitar esta confusión planteamos la extracción de características relacionadas con la textura de las frutas.

Figura 14: Matriz de confusión con el sistema SVM (kernel = rbf; C=15; $\gamma = 1$)

4.5. Quinta aproximación

4.5.1. Descripción

Para esta última aproximación previa al uso de *deep learning*, hemos decidido solventar los problemas residuales que dejó la aproximación anterior por medio de un estudio más profundo de las texturas presentes en las frutas. Para ello, hemos utilizado matrices de co-ocurrencia de niveles de gris (GLCM, de sus siglas en inglés), una herramienta utilizada para analizar la textura de una imagen mediante una matriz que describe la relación espacial entre los píxeles de una imagen en términos de sus niveles de gris. Una entrada (i, j) en la matriz GLCM indica el número de veces que un píxel con un nivel de gris i se encuentra junto a un píxel con un nivel de gris j en una dirección específica $\theta = 16$. Para esta aproximación se han considerado cuatro direcciones: 0°, 45°, y 90°. Una vez construida la GLCM, extraemos el contraste, que mide la variación en los niveles de gris de la imagen; la homogeneidad, que mide la uniformidad de la imagen dando peso a los píxeles más cercanos; y la energía, similar a la homogeneidad pero dando el mismo peso a todos los píxeles. Por ejemplo, para la Figura 1, su GLCM en dirección 0° sería:

0	2	4	1	0	1	1	0	0	0	0	0	0	0	0	0
2	14	17	7	1	0	0	1	0	1	0	0	0	0	0	0
5	18	96	37	8	2	1	0	2	0	0	0	1	0	0	0
1	3	33	99	34	7	0	0	3	3	5	0	0	0	0	0
0	2	10	29	49	19	9	6	2	0	0	0	0	0	0	0
1	0	2	6	17	65	16	6	5	1	1	1	1	2	0	0
0	1	3	3	6	14	41	13	8	4	1	2	1	0	1	1
0	1	0	3	3	7	14	80	16	6	3	1	0	2	0	1
0	1	0	0	0	3	5	18	138	53	10	2	4	3	1	1
0	0	0	1	1	1	1	5	60	1394	201	10	7	4	4	1
0	1	3	0	1	2	1	2	1	211	7465	261	23	13	7	1
0	0	0	1	2	0	2	0	2	6	276	1873	124	28	16	5
0	0	1	0	0	1	2	0	0	3	16	148	1727	60	38	18
0	0	0	0	0	1	2	1	0	1	5	29	71	214	58	40
0	0	0	0	3	0	0	2	1	2	2	3	36	66	337	154
0	0	1	1	1	1	4	3	1	5	7	5	19	30	143	22880

Con esto en cuenta, tendríamos un *dataset* con 24 atributos. Además, se ha realizado una validación cruzada con un ratio 70-20-10 para dividir los sets de entrenamiento-validación-test, tal como se hizo en otras aproximaciones. También se han normalizado los datos por medio de MinMax en todos los métodos menos en la RNA, que se realizó con ZeroMean, como se mencionó también en otras aproximaciones.

4.5.2. Resultados

Para empezar, como se muestra en el Cuadro 17, esta aproximación ha mejorado nuevamente los resultados con respecto a la anterior. Encontramos también una mayor homogeneidad en las métricas, con una variación de apenas un 0.4 % entre la mejor y peor de las arquitecturas

(a excepción de la arquitectura [1], que es simbólica).

En esta aproximación optamos por experimentar con menos arquitecturas con más de una capa. Ya vimos anteriormente que los resultados no eran notables y no aportaban información más allá de ver las similitudes entre sus métricas. De aquellas que se muestran en el Cuadro identificamos la misma tendencia: entre las arquitecturas [16 8], [16 12], y [20 15], solo hay una variación del 0.15 %.

En esta ocasión, la mejor arquitectura es la de 15 neuronas en una única capa oculta. Como se puede ver en la Figura 15, que muestra la matriz de confusión de esta arquitectura, se han corregido los errores entre la identificación de manzanas y naranjas. Suponemos gracias a las características que se han aportado en esta aproximación. Siguen apareciendo errores entre manzanas y plátanos, pese a las características extraídas en la aproximación anterior. Investigaremos más sobre como resolver este problema, pero estamos por lo general contentos con los resultados que se muestran.

Figura 15: Matriz de confusión con arquitectura [15].

Arquitecturas	**Precisión**	***F1-Score***
[1]	71.79 % ($\sigma = 1{,}70$)	63.58 % ($\sigma = 1{,}43$)
[4]	93.82 % ($\sigma = 1{,}42$)	93.80 % ($\sigma = 1{,}41$)
[8]	93.86 % ($\sigma = 1{,}14$)	93.84 % ($\sigma = 1{,}14$)
[15]	94.10 % ($\sigma = 1{,}15$)	94.07 % ($\sigma = 1{,}13$)
[16]	94.01 % ($\sigma = 1{,}12$)	93.99 % ($\sigma = 1{,}11$)
[16 8]	94.00 % ($\sigma = 1{,}45$)	93.98 % ($\sigma = 1{,}44$)
[16 12]	93.85 % ($\sigma = 1{,}33$)	93.83 % ($\sigma = 1{,}32$)
[20 15]	93.85 % ($\sigma = 1{,}46$)	93.83 % ($\sigma = 1{,}44$)

Cuadro 17: Resultados de la RNA.

Para las SVM, hemos vuelto a usar la estrategia de las aproximaciones anteriores, es decir, elaborar una escala logarítmica y posteriormente probar valores en función de los resultados que se van obteniendo. La tabla resultante es la siguiente.

Cuadro 18: Resultados de la SVM.

Kernel	***C***	γ	**Precisión**	***F1-Score***
rbf	9	1	96.09 %($\sigma = 2{,}17$)	96.11 % ($\sigma = 2{,}13$)
rbf	8	1	95.95 %($\sigma = 2{,}03$)	95.97 % ($\sigma = 1{,}99$)
rbf	11	1	95.79 %($\sigma = 2{,}13$)	95.81 % ($\sigma = 2{,}08$)
linear	2	-	92.65 % ($\sigma = 1{,}83$)	92.58 % ($\sigma = 1{,}88$)
linear	7	-	92.65 % ($\sigma = 2{,}31$)	92.62 % ($\sigma = 2{,}32$)
linear	8	-	92.49 % ($\sigma = 2{,}48$)	92.47 % ($\sigma = 2{,}51$)
poly	1	1	95.80 % ($\sigma = 1{,}71$)	95.78 % ($\sigma = 1{,}75$)
poly	2	1	95.05 % ($\sigma = 1{,}73$)	95.04 % ($\sigma = 1{,}72$)

Esta vez notamos una ligera mejoría en la precisión y F1 con respecto a la aproximación anterior. Las mejoras más destacables se dan en los kernels linear y poly, llegando casi al 2 %. Los mejores resultados se siguen dando con el kernel rbf, superando el 96 %, pero a diferencia de las otras aproximaciones, ya le aparece un competidor, el kernel poly, situándose también cerca del 96 %. En cuanto a la desviación típica, tenemos buenas noticias ya que a diferencia de la aproximación anterior, en ésta si que conseguimos reducirla de forma significativa en todos los kernels, especialmente en el poly. La situación con el kernel sigmoid no ha cambiado con respecto a la aproximación anterior por lo que se ha vuelto prescindir de él. Aún así, la valoración final de los resultados es positiva.

En el cuadro 19 se contemplan los resultados obtenidos al aplicar los cambios en la extracción de características al sistema basado en el método kNN. Como es habitual realizamos experimentos con diferentes valores de k, sin embargo pese a haber aplicado los cambios en la extracción de características respecto a la aproximación anterior, notamos una leve disminución en los resultados obtenidos, donde pese a que son muy similares a los obtenidos en la cuarta aproximación, los valores son ligeramente inferiores. Por otra parte en este caso vemos como los valores son mas homogéneos, la mayoría ronda el 92 % independientemente del valor de k, hasta k=*9* donde empieza a disminuir notablemente la precisión

Cuadro 19: Resultados del kNN.

k	**Precisión**	***F1-Score***
1	92.19 % ($\sigma = 3{,}31$)	92.14 % ($\sigma = 3{,}35$)
2	91.90 % ($\sigma = 2{,}45$)	92.09 % ($\sigma = 2{,}34$)
3	92.65 % ($\sigma = 2{,}93$)	92.64 % ($\sigma = 2{,}93$)
5	92.65 % ($\sigma = 2{,}48$)	92.54 % ($\sigma = 2{,}49$)
7	92.35 % ($\sigma = 1{,}7$)	92.19 % ($\sigma = 1{,}97$)
9	91.75 % ($\sigma = 2{,}65$)	91.56 % ($\sigma = 2{,}88$)
13	90.10 % ($\sigma = 2{,}35$)	89.80 % ($\sigma = 2{,}57$)

Por ultimo, como siempre, realizamos varios experimentos con un sistema de Árboles de Decisión, variando el valor de profundidad. Como puede verse en el Cuadro 20. Con los nuevos cambios en las características extraídas podemos apreciar como los resultados empeoran muy levemente. Además, la mejor configuración pasa a ser la 7. Est es explicable debido a la complejidad del problema, actualmente se tienen 24 atributos y resulta lógico que sea necesaria mayor profundidad para encontrar el punto óptimo.

Cuadro 20: Resultados del DT.

Profundidad	**Precisión**	***F1-Score***
1	59.82 % ($\sigma = 2{,}10$)	52.31 % ($\sigma = 3{,}30$)
4	84.55 % ($\sigma = 3{,}92$)	84.40 % ($\sigma = 3{,}88$)
6	86.19 % ($\sigma = 2{,}69$)	86.14 % ($\sigma = 3{,}63$)
7	87.69 % ($\sigma = 3{,}31$)	87.69 % ($\sigma = 3{,}30$)
10	87.39 % ($\sigma = 2{,}87$)	87.36 % ($\sigma = 2{,}92$)
15	86.95 % ($\sigma = 2{,}48$)	86.89 % ($\sigma = 2{,}55$)

4.5.3. Discusión

Esta es la última aproximación basada en el desarrollo de sistemas mediante los cuatro métodos vistos hasta ahora. Gracias a las nuevas características extraidas en esta quinta aproximación alcanzamos un pico de 95.95 % en el valor de precisión (SVM: kernel = rbf; C=8; $\gamma = 1$)). En el sistema que utiliza Redes de Neuronas Artificiales notamos también un aumento en todas las métricas. Sin embargo en los otros dos sistemas, kNN y DT, vemos que los resultados empeoran si los comparamos con los de la aproximación anterior, esto se puede deber a la cantidad exhaustiva de características añadidas.
Si comprobamos la matriz de confusión en la Figura 16 vemos que las naranjas y manzanas son distinguidas a la perfección gracias a las diferencias en su textura sin embargo aun persiste algún error referente a la diferenciación de plátanos y manzanas, que se puede deber a una poca presencia en la base de datos de alguna clase como podrían ser los plátanos rojos. En la siguiente aproximación utilizaremos técnicas de *deep learning* para resolver nuestro problema. Con esto esperamos mejorar los resultados e intentar superar la barrera del 95 % en ambas métricas.

Figura 16: Matriz de confusión con el sistema SVM (kernel = rbf; C=8; $\gamma = 4$)

4.6. Sexta aproximación - Deep Learning

4.6.1. Descripción

Para esta última aproximación, hemos usado técnicas de *Deep Learning* para entrenar una Red de Neuronas Convolucionales (referida como CNN de aquí en adelante) que resuelva nuestro problema. Este tipo de redes de neuronas, inspiradas en los procesos biológicos [13] y la organización del sistema visual en el cerebro humano para los patrones de conectividad entre neuronas, están específicamente diseñadas para procesar datos de píxeles, y cuentan con un uso frecuente en análisis de imágenes. Las CNN utilizan capas convolucionales para aplicar filtros, o *kernels*, a pequeñas regiones de entrada para extraer características relevantes. Estos filtros se deslizan por la imagen y se convolucionan con las características locales, generando mapas de características.

La arquitectura de nuestra CNN consiste en una **capa de entrada** que recibe una imagen como un tensor tridimensional ancho x alto x canales (3, ya que la imagen es a color); las **capas convolucionales**, que aplican los filtros a pequeñas regiones de los datos de entrada para capturar patrones locales en la imagen; **capas de activación**, que aplican una función de activación no lineal, *Rectified Linear Unit* (ReLU) en esta aproximación, para que la red

aprenda relaciones y patrones complejos en los datos; **capas de *pooling***, que reducen la dimensión espacial de los mapas de características generados por las capas convolucionales; tras las capas convolucionales y las de *pooling*, se utiliza una **capa *flatten*** para transformar los mapas de características multidimensionales en un vector unidimensional antes de alimentarlos a las **capas totalmente conectadas**, que actúan como una capa oculta en un perceptrón multicapa tradicional; y, finalmente, la **capa de salida** a la que se le aplica una función de activación *softmax* para generar probabilidades de clasificación. Una representación gráfica de nuestra arquitectura se puede ver en la Figura 17.

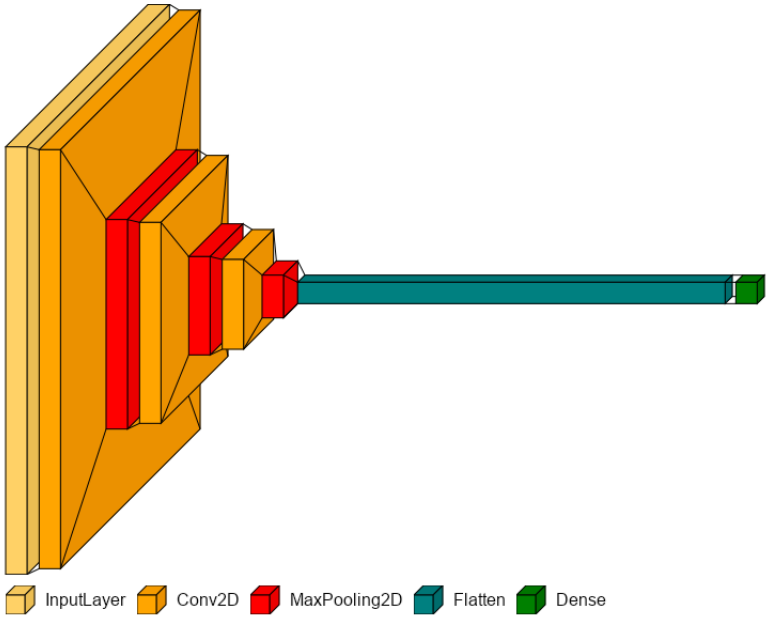

Figura 17: Arquitectura de la CNN.

Hemos tenido que redimensionar todas las imágenes para que tengan tamaños idénticos de 50x50 píxeles, en *batches* de 30 imágenes, y los datos no necesitaron de normalización. Además, se realizó una validación cruzada con un ratio de 0.1 para el set de test.

4.6.2. Resultados

Hemos de comenzar anunciando que los resultados han sido muy favorables, y todas las arquitecturas han rondado el 100 %, si es que no lo han alcanzado, como es el caso de la arquitectura que se muestra en el Cuadro 21. En algunas de las arquitecturas, se han modificado los canales de entrada y de salida de las capas de convolución (Cuadros 22, 23, 24), cosa que ha influido 5 % en la precisión final, tomando como referencia la primera arquitectura. En otras, se han modificado los filtros de convolución. En el caso del Cuadro 25, donde el filtro se cambió únicamente para la primera capa de convolución, el resultado fue desastroso y el entrenamiento se paró por no haber mejorado la precisión en el conjunto de entrenamiento durante 10 ciclos. Pero en el caso de los Cuadros 26, 27, y 28, donde se fue cambiando el filtro progresivamente de cada capa de convolución, los resultados fueron en los tres casos iguales.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (3,3) 3⇒ 16	(50, 50, 16)
MaxPool1	(25, 25, 16)
Conv2 (3,3) 16⇒ 32	(25, 25, 32)
MaxPool2	(12, 12, 32)
Conv3 (3,3) 32⇒ 32	(12, 12, 32)
MaxPool3	(6, 6, 32)
Flatten	(1152)
Dense	(3)
Precisión final (test)	100 %
Número de ciclos	50

Cuadro 21: Arquitectura 1.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (3,3) 3⇒ 16	(50, 50, 16)
MaxPool1	(25, 25, 16)
Conv2 (3,3) 16⇒ 32	(25, 25, 32)
MaxPool2	(12, 12, 32)
Conv3 (3,3) 32⇒ 64	(12, 12, 64)
MaxPool3	(6, 6, 64)
Flatten	(2304)
Dense	(3)
Precisión final (test)	96.96 %
Número de ciclos	42

Cuadro 22: Arquitectura 2.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (3,3) 3⇒ 16	(50, 50, 16)
MaxPool1	(25, 25, 16)
Conv2 (3,3) 16⇒ 16	(25, 25, 16)
MaxPool2	(12, 12, 16)
Conv3 (3,3) 16⇒ 32	(12, 12, 32)
MaxPool3	(6, 6, 32)
Flatten	(1152)
Dense	(3)
Precisión final (test)	96.96 %
Número de ciclos	76

Cuadro 23: Arquitectura 3.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (3,3) 3⇒ 32	(50, 50, 32)
MaxPool1	(25, 25, 32)
Conv2 (3,3) 32⇒ 64	(25, 25, 64)
MaxPool2	(12, 12, 64)
Conv3 (3,3) 64⇒ 128	(12, 12, 128)
MaxPool3	(6, 6, 128)
Flatten	(4608)
Dense	(3)
Precisión final (test)	95.45 %
Número de ciclos	146

Cuadro 24: Arquitectura 4.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (5,5) 3⇒ 16	(48, 48, 16)
MaxPool1	(24, 24, 16)
Conv2 (3,3) 16⇒ 32	(24, 24, 32)
MaxPool2	(12, 12, 32)
Conv3 (3,3) 32⇒ 32	(12, 12, 32)
MaxPool3	(6, 6, 32)
Flatten	(1152)
Dense	(3)
Precisión final (test)	62.12 %
Número de ciclos	49*

Cuadro 25: Arquitectura 5.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (1,1) 3⇒ 16	(52, 52, 16)
MaxPool1	(26, 26, 16)
Conv2 (3,3) 16⇒ 32	(26, 26, 32)
MaxPool2	(13, 13, 32)
Conv3 (3,3) 32⇒ 32	(13, 13, 32)
MaxPool3	(6, 6, 32)
Flatten	(1152)
Dense	(3)
Precisión final (test)	98.48 %
Número de ciclos	45

Cuadro 26: Arquitectura 6.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (1,1) 3⇒ 16	(52, 52, 16)
MaxPool1	(26, 26, 16)
Conv2 (1,1) 16⇒ 32	(28, 28, 32)
MaxPool2	(14, 14, 32)
Conv3 (3,3) 32⇒ 32	(14, 14, 32)
MaxPool3	(7, 7, 32)
Flatten	(1568)
Dense	(3)
Precisión final (test)	98.48 %
Número de ciclos	76

Cuadro 27: Arquitectura 7.

Capa	Salida
Entrada	(50, 50, 3)
Conv1 (1,1) 3⇒ 16	(52, 52, 16)
MaxPool1	(26, 26, 16)
Conv2 (1,1) 16⇒ 32	(28, 28, 32)
MaxPool2	(14, 14, 32)
Conv3 (1,1) 32⇒ 32	(16, 16, 32)
MaxPool3	(8, 8, 32)
Flatten	(2048)
Dense	(3)
Precisión final (test)	98.48 %
Número de ciclos	78

Cuadro 28: Arquitectura 8.

* Se paró el entrenamiento por no haber mejorado la precisión en el conjunto de entrenamiento durante 10 ciclos.

4.6.3. Discusión

Podemos confirmar que la mejor arquitectura es la primera, y la única que ha alcanzado el 100 % de precisión (Figura 18) en todo el experimento desde que se comenzó con la primera aproximación. Sin embargo, y a excepción de la quinta arquitectura, el resto de CNN,s presentaron buenos resultados, mejores incluso que en el resto de aproximaciones. Se muestra así el éxito de utilizar CNN,s para resolver este problema en concreto, y en general otros problemas que requieran de procesamiento de imágenes.

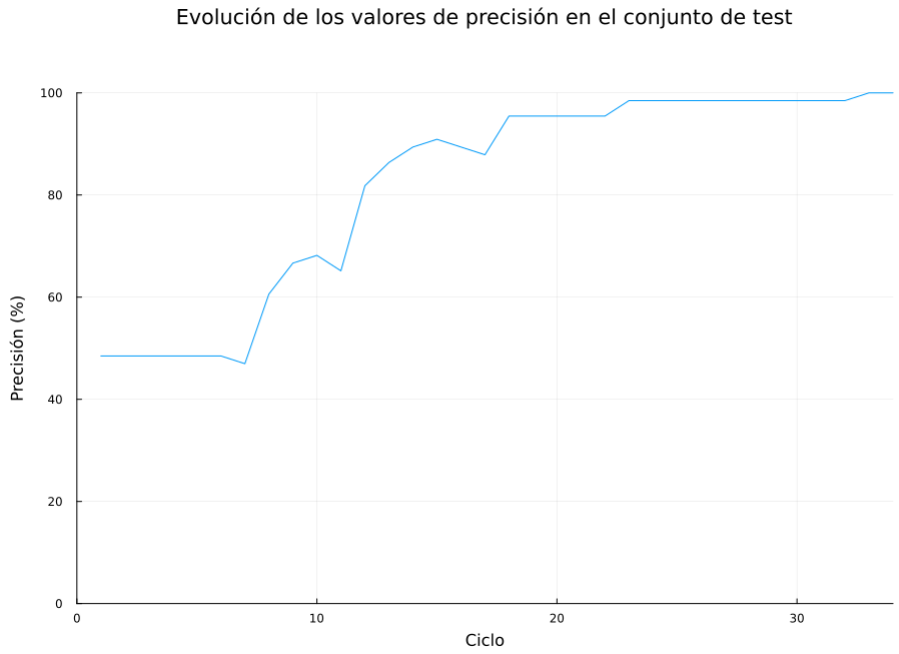

Figura 18

5. Conclusiones

Después de 5 aproximaciones en las que se han ido añadiendo nuevas características o incluso frutas y haciendo nuevos experimentos, llegamos a la conclusión de que el mejor sistema para resolver el problema de diferenciación de frutas, entre los cuatro primeros que hemos probado (RNA, SVM, kNN, DT), es, el sistema basado en Máquinas de Soporte Vectorial, donde alcanzamos un valor de precisión de casi un 96 %. Para alcanzar este valor las características que hemos extraído son el color, la forma, la simetría y la textura. A pesar de no haber alcanzado un 100 % de precisión, superamos el 95 %, lo que se podría considerar un ratio de acierto válido en este problema.

Tras la implementación del sistema, en la sexta aproximación, utilizando el método de *Deep Learning* basado en CNN,s hemos podido observar que los resultados son notablemente superiores a todos los obtenidos con anterioridad, llegando a valores de precisión en el conjunto de test de hasta un 100 %. Esto se debe a que las CNN,s son las arquitecturas que mejores resultados dan a la hora del análisis de imágenes ya que no dependen de una correcta extracción de características previa.

A lo largo del desarrollo de este sistema hemos tenido que enfrentar varias dificultades. Algunas de estas dificultades pueden incluir la selección de las frutas y la base de datos, el preprocesado de imágenes (filtración de la base de datos para mantener solamente aquellas imágenes que cumplen los requisitos establecidos), la correcta selección y extracción de características, etc.

Si nos enfocamos en las ventajas que podría tener el sistema se puede destacar que nos permite una detección rápida y precisa de las frutas en imágenes lo que puede ser útil en aplicaciones de clasificación y selección de frutas en industrias alimentarias.

En resumen, el mejor sistema entre los cuatro primeros con los que hemos realizado experimentos para solucionar este problema es la SVM, aunque es posible que exista algún sistema mejor fuera de estos cuatro o incluso, con una extracción de características más amplia, algún otro de estos métodos podría alcanzar un valor superior. Por otra parte, los resultados obtenidos del sistema basado en CNN,s son prácticamente perfectos, alcanzando valores de un 100 % de precisión, haciéndolo el método idóneo para implementar este sistema.

6. Trabajos futuros

Basándonos en los resultados obtenidos a lo largo de las distintas aproximaciones y en las conclusiones del proyecto podemos detallar algunas posibles lineas de trabajo futuras.

Principalmente, la inserción de nuevas frutas en el problema es totalmente compatible, como pudimos observar en la segunda aproximación con la inserción de las naranjas. Esto permitiría diferenciar entre un amplio rango de frutas de distintas formas, tamaños, colores, etc. Por otra parte sería una buena ampliación del sistema la adaptación del mismo para aceptar imágenes en diferentes condiciones de entorno, iluminación, estado de la fruta e incluso imágenes con varias frutas en ella.

Además, también existe la posibilidad de realizar experimentos con nuevos métodos de aprendizaje automático como por ejemplo las Redes Bayesianas o los árboles de clasificación aleatorios.

Resumiendo, encontramos varias oportunidades de trabajos futuros que podrían mejorar la eficacia de nuestro sistema de clasificación de frutas al igual que su alcance, pudiendo llegar a ser un gran sistema utilizado en la industria agrícola.

7. Bibliografía

Referencias

[1] Minut, Mihai: *Fruits 262*. https://www.kaggle.com/datasets/aelchimminut/fruits262, 2021. [Online; accessed 13-February-2023].

[2] Zulkifley, Mohd Asyraf, Asraf Mohamed Moubark, Adhi Harmoko Saputro y Siti Raihanah Abdani: *Automated Apple Recognition System Using Semantic Segmentation Networks with Group and Shuffle Operators*. Agriculture, 12(6), 2022, ISSN 2077-0472. https://www.mdpi.com/2077-0472/12/6/756.

[3] Hossain, M. Shamim, Muneer Al-Hammadi y Ghulam Muhammad: *Automatic Fruit Classification Using Deep Learning for Industrial Applications*. IEEE Transactions on Industrial Informatics, 15(2):1027–1034, Feb 2019, ISSN 1941-0050.

[4] Behera, Santi Kumari, Amiya Kumar Rath, Abhijeet Mahapatra y Prabira Kumar Sethy: *Identification, classification & grading of fruits using machine learning & computer intelligence: a review*. Journal of Ambient Intelligence and Humanized Computing, Mar 2020, ISSN 1868-5145. https://doi.org/10.1007/s12652-020-01865-8.

[5] Indira, D.N.V.S.L.S., Jyothi Goddu, Baisani Indraja, Vijaya Madhavi Lakshmi Challa y Bezawada Manasa: *A review on fruit recognition and feature evaluation using CNN*. Materials Today: Proceedings, 2021, ISSN 2214-7853. https://www.sciencedirect.com/science/article/pii/S2214785321051269.

[6] Huang, Ziliang, Yan Cao y Tianbao Wang: *Transfer Learning with Efficient Convolutional Neural Networks for Fruit Recognition*. En *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, páginas 358–362, March 2019.

[7] Yang, Ming, Pawan Kumar, Jyoti Bhola y Mohammad Shabaz: *Development of image recognition software based on artificial intelligence algorithm for the efficient sorting of apple fruit*. International Journal of System Assurance Engineering and Management, 13(1):322–330, Mar 2022, ISSN 0976-4348. https://doi.org/10.1007/s13198-021-01415-1.

[8] Jana, Susovan, Saikat Basak y Ranjan Parekh: *Automatic fruit recognition from natural images using color and texture features*. En *2017 Devices for Integrated Circuit (DevIC)*, páginas 620–624, March 2017.

[9] Idris, Bashir, Lili N. Abdullah, Abdul H. Alfian y Abdullah S. Mohd Taufik: *Comparison of Edge Detection Algorithms for Texture Analysis on Copy-Move Forgery Detection Images*. International Journal of Advanced Computer Science and Applications, 13(10), 2022. https://www.proquest.com/scholarly-journals/comparison-edge-detection-algorithms-texture/docview/2740752247/se-2.

[10] Martin, D.R., C.C. Fowlkes y J. Malik: *Learning to detect natural image boundaries using local brightness, color, and texture cues*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(5):530–549, May 2004, ISSN 1939-3539.

[11] He, Qiang, KangLi Xia y Hui Pan: *Fruit Recognition Using Color Statistics.* En *2022 International Conference on Automation, Robotics and Computer Engineering (ICARCE)*, páginas 1–4, Dec 2022.

[12] Aggarwal, N. y R. K. Agrawal: *First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images.* 3(2), 2012. https://www.scirp.org/journal/paperinformation.aspx?paperid=19553.

[13] Fukushima, Kunihiko: *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.* 36(193-202), 1980. https://doi.org/10.1007/BF00344251.