Predicción automática de la carga frutal de olivos empleando UAV y redes convolucionales

P. Asensio Jiménez, D. M. Martínez Gila, S. Satorres Martínez, E. Estévez Estévez, J. Gómez Ortega y J. Gámez García
Grupo de Robótica, Automática y Visión por Computador
Universidad de Jaén, Campus Las Lagunillas s/n, ES-23071, Jaén (España)
dmgila@ujaen.es

Resumen

El sector del aceite de oliva y de la aceituna de mesa representan ya el 3% del PIB total de Andalucía. Teniendo en cuenta las cifras que este hecho supone, predecir la cosecha campaña tras campaña es clave para definir estrategias de marketing. Dada la gran superficie de olivar existente resulta interesante la integración de tecnologías emergentes que puedan facilitar esta tarea de predicción. En este trabajo se estudia la viabilidad del uso de cámaras de visión por computador de espectro visible embarcadas en UAVs para valorar de forma cualitativa la carga frutal de los olivos de una plantación. Las imágenes adquiridas fueron etiquetadas y posteriormente utilizadas para entrenar tres arquitecturas CNN (AlexNet, GoogLeNet, y ResNet) por el método de transferencia de aprendizaje. La arquitectura que mejor rindió fue GoogLeNet, que posteriormente fue optimizada obteniendo finalmente una tasa de éxito del 90% a la hora de clasificar imágenes que mostraban regiones de olivos con carga alta, media, baja y descarte (no olivo).

Palabras clave: UAV, olivar, aceitunas, redes convolucionales, procesado de imagen, aprendizaje máquina.

1 INTRODUCCIÓN

En los últimos diez años la superficie de olivar de España ha crecido paulatinamente produciéndose el mayor auge en los últimos tres años con un aumento en torno al 4,2% y alcanzando la cifra de más de 2,5 millones de hectáreas. La mayor parte de esta superficie corresponde a olivar para almazara y las regiones con mayor producción son Andalucía, Castilla-La Mancha y Extremadura. El precio del aceite de oliva producido es variable campaña tras campaña y depende de la cantidad de aceitunas recolectadas. Para planificar las estrategias comerciales de las campañas futuras todos los años se realiza un aforo del olivar previo a los meses de recolección. En Andalucía los encargados de realizar esta tarea son los técnicos de la Consejería de Agricultura que cuantifican bajo su criterio la carga frutal de un número determinado de árboles. Por ejemplo, en 2017 se inspeccionaron más de 8.000 olivos. La elevada cantidad de tiempo que conlleva esta metodología hace que no sea válida para ser mantenida a lo largo de los meses más próximos a la campaña y este hecho conlleva a errores de predicción. Para hacerse una idea de la dimensión de este error de predicción solo cabe observar el aforo oficial de la campaña oleícola 2020/21, cuando se estimó una producción de aceite de oliva en Andalucía de 1.348.200 toneladas. En febrero, la Consejería de Agricultura tuvo que ajustar a la baja la previsión inicial como consecuencia de la falta de lluvias y las altas temperaturas del otoño. La Agencia de Información y Control Alimentarios (www.aica.gob.es) apuntó a una producción de 384.500 toneladas en Jaén, mientras que el aforo auguraba 670.000. Por otro lado, un correcto aforamiento podría ser útil para la planificación de los procesos industriales a los que se someten las aceitunas después de la recolección.

Figura 1: Imagen de la copa de un olivo con aceitunas.

El concepto de agricultura de precisión contempla el empleo de vehículos aéreos no tripulados (UAV) para automatizar tareas agrícolas. Las diferentes aplicaciones sobre su uso están siendo muy documentadas en los últimos años y por ejemplo en [1] clasifican estas aplicaciones en tareas relacionados con riego, fertilización, uso de pesticidas, manejo de malezas, monitorización del crecimiento de plantas, manejo de enfermedades de cultivos y fenotipado a nivel de campo. El empleo de UAV en el olivar tradicional (o de montaña) es especialmente

 https://doi.org/10.17979/spudc.9788497498418.0956

interesante donde el acceso con vehículos terrestres puede estar limitado. Su uso ha sido documentado en [2] y en [3], donde los autores emplearon un sistema multiespectral embarcado en un UAV para la estimación del estado nutricional de diferentes plantaciones de olivos. Por otro lado, en [4] los autores emplearon imágenes cenitales RGB adquiridas desde un UAV para medir el área de la copa de diferentes olivos y a partir de ésta predecir la producción del árbol.

El empleo de técnicas relacionadas con *deep learning* y aplicadas sobre imágenes hortofrutícolas para la detección de frutos en árbol y la estimación de su aforo ha sido documentado en diferentes trabajos de investigación [5]. Por ejemplo, en [6] los autores emplearon redes neuronales convolucionales (CNN) para detectar y cuantificar el número de frutas (naranjas y manzanas) a partir de imágenes capturadas por un UAV. Otro ejemplo es el documentado en [7] en el que los autores adaptaron una arquitectura de red CNN denominada ResNet para cuantificar tomates.

El objetivo del trabajo que se documenta en este artículo consiste en evaluar diferentes arquitecturas CNN para la cuantificación de la carga frutal de olivos a partir de imágenes RGB (Figura 1) capturadas desde un UAV. La estructura que se ha seguido para presentar el trabajo realizado ha sido la siguiente: En la Sección 2 se expone la metodología seguida para la adquisición y etiquetado de imágenes, y la experimentación llevada a cabo para el entrenamiento y evaluación de las CNNs seleccionadas; la Sección 3 contiene los resultados obtenidos de la experimentación realizada; finalmente, la Sección 4 contiene las conclusiones obtenidas y los trabajos futuros que se plantean.

2 MATERIALES Y MÉTODOS

En esta sección se detalla la metodología llevada a cabo para la integración de arquitecturas CNN pre-entrenadas en la tarea de detección automática de la carga frutal de olivos. En primer lugar, se describe el procedimiento de adquisición y etiquetado de imágenes. Posteriormente, se detalla el procedimiento seguido para la parametrización de las arquitecturas CNN y la evaluación de los modelos de clasificación. Por último, se describe la técnica de optimización de hiperparámetros empleada antes de describir la sección de resultados.

2.1 ADQUISICIÓN DE IMÁGENES

Durante el mes de octubre de 2021 (un mes antes del comienzo de la última campaña de recolección), se adquirió un conjunto de 275 imágenes en un olivar situado en Torrequebradilla (37°55'47.0"N 3°39'37.8"W), provincia de Jaén (Andalucía, España). Para la realización de este estudio se seleccionaron 10 árboles con aceitunas de la variedad Picual. La selección se llevó a cabo visualmente en base a la carga frutal de los olivos buscando la máxima variación disponible en cuanto a carga frutal.

Las imágenes se adquirieron de día con iluminación natural. La captura de imágenes se realizó mediante una cámara ZENMUSE X5 equipada con una óptica DJI MFT de 15 mm y embarcada en un UAV DJI Inspire 1 (Figura 2). Dicha cámara monta un sensor CMOS de 16 Mpx y se configuró para capturar imágenes con una resolución de 4.000 x 2.250 píxeles.

Figura 2. Imagen del UAV empleado para la experimentación que se documenta en el trabajo.

Se realizaron un total de 3 vuelos en días diferentes, cada uno con una duración aproximada de 1 hora, a lo largo del mes de octubre de 2021. Para cada árbol se capturaron imágenes lateralmente a una distancia de separación del árbol que osciló entre 1 m y 2 m y a una altura de vuelo comprendida entre 1 m y 3 m. Estas distancias se seleccionaron en función de la las características de cada árbol y con el objetivo de ajustar la copa del árbol dentro del campo de visión de la cámara (Figura 3).

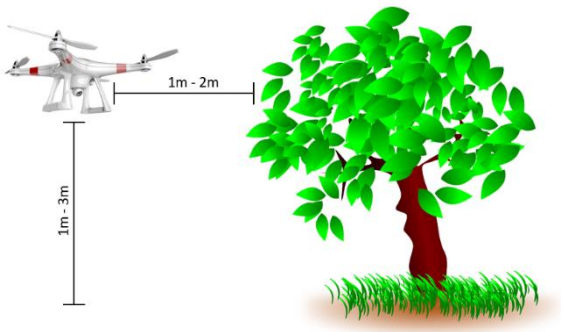

Figura 3. Metodología de vuelo.

Se decidió realizar los vuelos de forma lateral ya que de forma cenital las aceitunas son ocluidas por las hojas. Este mismo problema se reporta en [8] donde los autores solo contabilizaron el 27% de las manzanas contenidas en los árboles inspeccionados.

https://doi.org/10.17979/spudc.9788497498418.0956

2.2 ETIQUETADO DE IMÁGENES

Para la realización de este estudio se elaboró un conjunto de datos propio. Para tal fin, cada imagen adquirida se dividió en 8 recortes, cada uno con un tamaño de 1.000 x 1.125 píxeles, con el objetivo de poder identificar zonas de la superficie del olivo con diferente carga de aceituna, obteniendo un total de 2200 imágenes. Las imágenes se etiquetaron manualmente en cuatro clases, siguiendo los siguientes criterios:

- Carga alta: El olivo ocupa más de un 50% de la imagen y los frutos ocupan un 60% o más de la superficie visible del olivo.
- Carga media: El olivo ocupa más de un 50% de la imagen y los frutos ocupan entre un 40% y un 60% de la superficie visible del olivo.
- Carga baja: El olivo ocupa más de un 50% de la imagen y los frutos ocupan un 40% o menos de la superficie visible del olivo.
- Descarte: El olivo ocupa menos de un 50% de la imagen.

La Figura 4 presenta imágenes de muestra del conjunto de datos elaborado. Después de la exclusión de algunas imágenes con errores y baja calidad, se procedió a realizar un submuestreo seleccionando las 100 imágenes más representativas de cada clase, descartando por completo las imágenes restantes, con el objetivo de equilibrar el número de muestras.

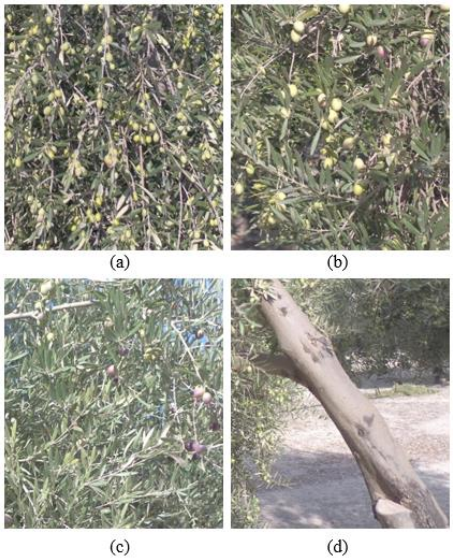

Figura 4. Ejemplo de imágenes etiquetadas dónde (a) pertenece a la clase “Carga alta”, (b) a “Carga media”, (c) a “Carga baja” y (d) a “Descarte”.

Posteriormente, el conjunto de datos se dividió en dos grupos, un primer grupo para el entrenamiento de las redes convolucionales y un segundo grupo para la validación de las mismas. Con el objetivo de evaluar la robustez de los modelos se probaron diferentes divisiones del conjunto de datos: 80-20 (80% del conjunto de datos para entrenamiento y 20% para validación), 60-40 y 50-50.

2.3 AUMENTO DE DATOS

El aumento de datos [9] es una técnica utilizada para evitar el sobreajuste de los modelos de clasificación supervisados. Consiste en aumentar artificialmente el tamaño del conjunto de imágenes de entrenamiento aplicando diversas transformaciones tales como desplazamientos, rotaciones, cambios de escala, etc. Esto obliga al modelo a ser más tolerante a las variaciones en la posición, orientación y tamaño de los objetos en las imágenes.

Se estudió el efecto que tiene el uso de aumento de datos durante el entrenamiento de la red. Para ello, se entrenó el modelo sin aplicar aumento de datos y, posteriormente, aplicando aumento de datos. El aumento de datos se realizó mediante la aplicación de las siguientes transformaciones a las imágenes de entrenamiento:

- Aumento y disminución de la escala entre un 0% y un 10%.
- Desplazamiento entre -30 y 30 pixeles tanto en el eje vertical como en el eje horizontal.
- Rotación en el rango entre -90º y 90º.
- Volteo vertical.

2.4 ARQUITECTURAS CNN EVALUADAS

Las redes neuronales convolucionales (ConvNets o CNN) son un tipo especializado de redes neuronales profundas que están diseñadas para trabajar con imágenes. En este estudio se evaluaron los resultados obtenidos con tres arquitecturas que ya se encuentran pre-entrenadas: AlexNet [10], GoogLeNet [11] y ResNet [12]. Estas redes se diseñaron en el contexto del "Desafío de reconocimiento visual a gran escala" (ILSVRC) para el conjunto de datos de ImageNet [13].

AlexNet está formada por un total de ocho capas. Las primeras cinco capas son capas convolucionales, seguidas por tres capas totalmente conectadas. Utiliza ReLU como función de activación y aplica capas de max-pooling después de las capas convolucionales para mejorar la eficiencia del tiempo de entrenamiento.

GoogLeNet introdujo el módulo de inicio para procesar las operaciones requeridas en paralelo. El módulo de inicio actúa como un eficiente extractor de funciones de varios niveles y hace que la red sea considerablemente más pequeña y más rápida. La arquitectura consta de 22 capas de profundidad, con 27 capas de pooling incluidas. Está formada por 9 módulos de inicio apilados linealmente.

https://doi.org/10.17979/spudc.9788497498418.0956

Por otra parte, ResNet consta de varios bloques residuales básicos que proporcionan una conexión de acceso directo entre capas. Esta conexión de acceso directo permite entrenar cientos o más capas mientras se logra un rendimiento mejorado. ResNet está diseñado principalmente para el análisis de datos a gran escala y se desarrolla con muchos números diferentes de capas. Para este estudio se utilizó ResNet-50, la cual está compuesta por 50 capas convolucionales, incluida una capa totalmente conectada.

Estas tres redes tienen una gran cantidad de parámetros entrenables. En particular, AlexNet cuenta con un total de 61 millones de parámetros, GoogLeNet con 7 millones y, por último, ResNet50 con 25.6 millones.

2.5 ESTRATEGIA DE ENTRENAMIENTO

La estrategia de aprendizaje por transferencia (TL) hace que las CNNs funcionen de manera efectiva en muchas tareas de clasificación visual, incluso si sus conjuntos de datos tienen un tamaño insuficiente o limitado.

Se analizó el rendimiento de las tres arquitecturas mencionadas en el apartado anterior sobre el conjunto de datos elaborado y para cada una de ellas se compararon dos técnicas de aprendizaje por transferencia: En primer lugar, utilizando la CNN como extractor de características y, posteriormente, utilizando la técnica de ajuste fino o "fine-tuning". La Figura 5 muestra el flujo de trabajo para ambas estrategias de entrenamiento.

En el primer caso, se selecciona un modelo previamente entrenado y se sustituye la parte clasificadora del modelo (es decir, la última capa totalmente conectada) por otra nueva que contenga la misma cantidad de nodos que la cantidad de clases nuevas (cuatro en el caso de este estudio). Posteriormente, se "congela" la parte extractora de características del modelo (es decir, las capas de convolución y de pooling anteriores) estableciendo sus tasas de aprendizaje a cero para que los parámetros de dichas capas no se actualicen. Por último, se realiza un nuevo entrenamiento del modelo para que aprenda e identifique las características relacionadas con las nuevas imágenes y categorías. De esta forma, solo se entrena el nuevo clasificador agregado. En el segundo caso, el proceso es el mismo, pero con una diferencia: Se descongela la parte extractora de características y se entrena conjuntamente con la capa de clasificación agregada. Esto permite volver a entrenar los pesos de las capas anteriores para ajustar las características extraídas de orden superior.

Para comparar los resultados de las dos configuraciones experimentales se utilizaron los mismos hiperparámetros en todos los experimentos. Se utilizó el descenso de gradiente estocástico con un momento de 0.9 como algoritmo de optimización, se estableció una tasa de aprendizaje de 0.0001, se estableció un tamaño de lote de 16 y el entrenamiento se ejecutó durante un total de 100 épocas. La elección de estos hiperparámetros se basó en la observación empírica de que, en todos los experimentos realizados, el entrenamiento siempre convergió bien dentro de las 100 épocas.

Los pesos finales utilizados para la evaluación de las redes corresponden a la última iteración de entrenamiento, es decir, los pesos obtenidos tras la época 100.

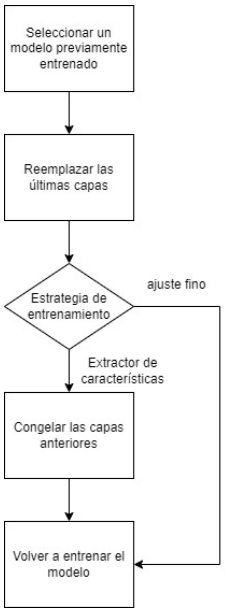

Figura 5. Flujograma con las fases de las dos estrategias de entrenamiento evaluadas.

Las métricas empleadas para evaluar el rendimiento de los modelos tanto en fase entrenamiento como en fase de validación fueron la precisión (ACC) definida como el número de muestras correctamente clasificadas entre el número de muestra totales, la función de pérdida basada en entropía cruzada (LCE), la tasa de verdaderos positivos (TPR) definida como la proporción de verdaderos positivos entre el número total de positivos detectados y el valor predictivo positivo (PPV) definido como la proporción de verdaderos positivos entre el número real de positivos.

Las matrices de confusión obtenidas tras aplicar los modelos de clasificación a las muestras de validación se presentarán en la sección de Resultados según la Figura 6, donde C_1, C_2, …, C_n son las distintas clases del problema de clasificación y TP_x, PPV_x y TPR_x son el número de verdaderos positivos, el valor predictivo positivo y la tasa de verdaderos positivos de la clase x respectivamente.

https://doi.org/10.17979/spudc.9788497498418.0956

		Predicted class			
		C_1	C_2	...	C_n
True class	C_1	TP_1			
	C_2		TP_2		
	...			...	
	C_n				TP_n

TPR_1	$1- TPR_1$
TPR_2	$1- TPR_2$
...	...
TPR_n	$1- TPR_n$

PPV_1	PPV_2	...	PPV_n
$1- PPV_1$	$1- PPV_2$	...	$1- PPV_n$

Figura 6. Formato empleado para presentar las matrices de confusión.

2.6 OPTIMIZACIÓN DE HIPERPARÁMETROS

Una vez finalizados los experimentos anteriores se analizaron los resultados obtenidos con el objetivo de seleccionar la configuración experimental que proporcionaba los mejores resultados para las muestras empleadas. Posteriormente, se realizó un proceso de optimización de los hiperparámetros de entrenamiento con el objetivo de encontrar los valores que generaran el mejor rendimiento en términos de precisión. Encontrar esta configuración óptima es un proceso iterativo en el que los parámetros se inicializan y ajustan varias veces.

Para el proceso de optimización se decidió utilizar la técnica de optimización bayesiana. En términos generales, la optimización bayesiana de hiperparámetros consiste en crear un modelo probabilístico en el que el valor de la función objetivo es la métrica de validación del modelo (precisión, pérdida…). El algoritmo de optimización itera hasta llegar a la condición de parada para encontrar los valores de hiperparámetros que maximizan o minimizan, según el caso, la función objetivo.

Los hiperparámetros que se optimizaron fueron la tasa de aprendizaje y el momento. Para la tasa de aprendizaje se especificó un rango búsqueda entre 1e-5 y 0.1 y para el momento se especificó un rango de búsqueda entre 0.8 y 0.99. Por último, se seleccionó la métrica de precisión en el conjunto validación como función objetivo a maximizar.

3 RESULTADOS

Para la realización de los experimentos se utilizó la "Deep Learning Toolbox" de Matlab, la cual proporciona un framework para diseñar e implementar redes neuronales profundas con modelos previamente entrenados. En resumen, se analizaron un total de 36 configuraciones experimentales, las cuales varían según la arquitectura empleada (AlexNet o Google Net), según si la estrategia de entrenamiento empleada es con extracción de características (EC) o ajuste fino (AF), según la aplicación de la estrategia de aumento de datos (CAD) o no (SAD) y en base a la elección de la distribución del conjunto de entrenamiento y validación.

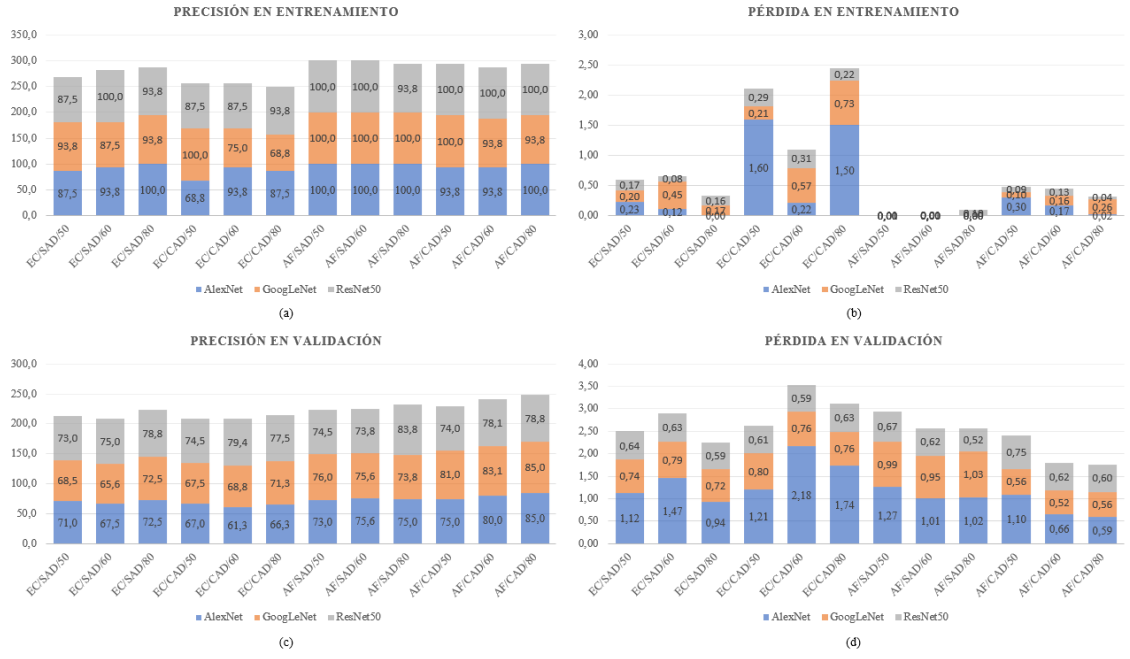

Figura 7. La figura muestra los resultados obtenidos para los diferentes experimentos. Éstos han sido identificados con la secuencia AA/BBB/CC donde AA indica la estrategia de entrenamiento, BBB si se ha empleado o no aumento de datos y CC el porcentaje de muestra usadas para entrenamiento. La subfigura (a) muestra el porcentaje de imágenes correctamente clasificadas en entrenamiento, la (b) el factor de pérdida en entrenamiento, la (c) el porcentaje de imágenes bien clasificadas en validación y la (d) el factor de pérdida en validación.

https://doi.org/10.17979/spudc.9788497498418.0956

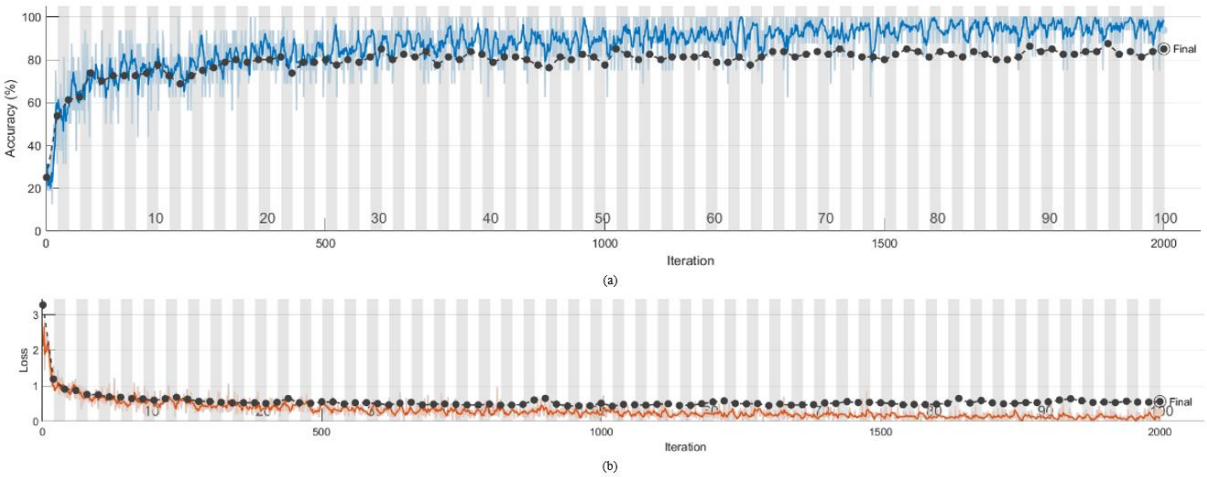

Figura 8. Evolución de la precisión y del factor de pérdida durante el proceso de entrenamiento y validación de la red GoogLeNet configurada como AF/CAD/80. En las gráficas de precisión (a), el conjunto de datos de entrenamiento está representado en azul y el conjunto de datos de validación en negro. De forma similar, en las gráficas de pérdida (b), el conjunto de datos de entrenamiento está representado en naranja y el conjunto de datos de validación en negro.

La Figura 7 presenta la precisión obtenida en cada una de las configuraciones experimentales evaluadas. En primer lugar, en relación a la distribución del conjunto de datos, se puede observar una ligera mejora de la precisión conforme aumenta el tamaño del conjunto de datos de entrenamiento. Por otra parte, en relación a la estrategia de entrenamiento, se observa como la técnica de ajuste fino proporciona mejores resultados que la técnica de extractor de características.

Por último, en relación con el aumento de datos, se puede diferenciar dos casos; Cuando se utiliza extractor de características como estrategia de entrenamiento, el uso de aumento de datos proporciona peores resultados; Sin embargo, cuando se utiliza ajuste fino, el uso de aumento de datos proporciona mejores resultados. En general las tres arquitecturas se ajustan bien a los datos de entrenamiento con precisiones en la fase de entrenamiento del 100%. En la fase de validación, la arquitectura que mejor predice la clase a la que pertenece la imagen inspeccionada es GoogLeNet entrenada con la técnica de ajuste fino, utilizando aumento de datos y empleando un 80% del conjunto de datos para el entrenamiento y un 20% para la validación (Figura 8). Ésta logra el mejor rendimiento en precisión en el conjunto de validación, obteniendo un 85% de imágenes correctamente clasificadas. Adicionalmente, logra una menor pérdida en comparación con el resultado obtenido en AlexNet utilizando la misma configuración. La matriz de confusión obtenida se puede ver en la Figura 9.

Partiendo del modelo que proporcionó el mejor rendimiento en los experimentos anteriores se aplicó optimización de hiperparámetros según la metodología descrita en el punto 2.6. La Figura 10 muestra la evolución del proceso de optimización a lo largo de 50 iteraciones. La mayoría de los resultados obtenidos tras el procedimiento arrojaron una tasa de éxito superior al 80% obteniendo un valor máximo en la iteración 13, donde la precisión alcanza un 90% (tasa de aprendizaje del 0.01 y momento 0.8003).

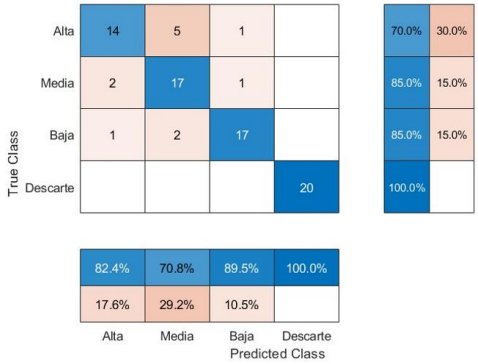

Figura 9. Matriz de confusión obtenida en la fase de validación del experimento AF/CAD/80 realizado con la arquitectura GoogLeNet.

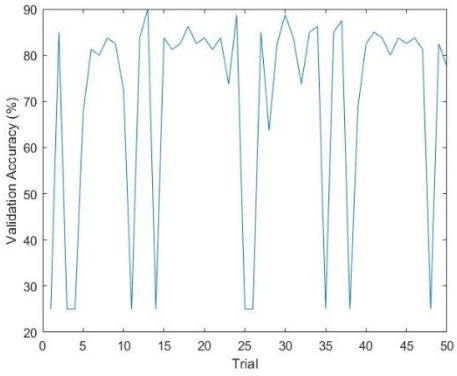

Figura 10. Resultados obtenidos durante el procedimiento de optimización bayesiana de la arquitectura GoogLeNet.

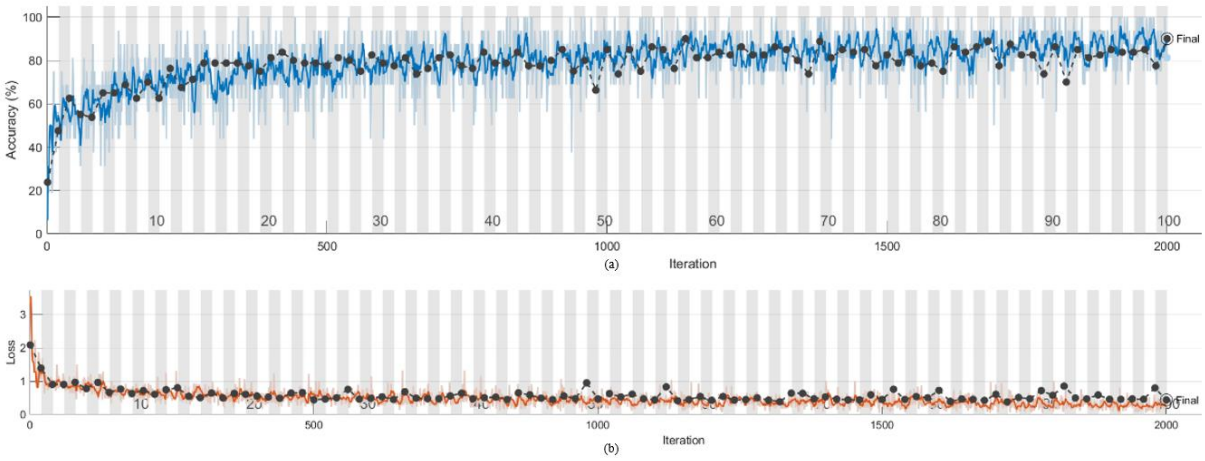

Figura 11. Gráficos de precisión (a) y pérdida (b) para el modelo GoogLeNet optimizado con la configuración AF/CAD/80.

La Figura 11 muestran las gráficas de precisión y pérdida correspondientes al entrenamiento y validación del modelo optimizado y la Figura 12 la matriz de confusión obtenida en la fase de validación del modelo optimizado. En la figura se puede observar que la mayoría de las imágenes mal clasificadas se predijeron entre clases adyacentes. Esto puede ser debido a posibles errores en el proceso inicial de etiquetado manual.

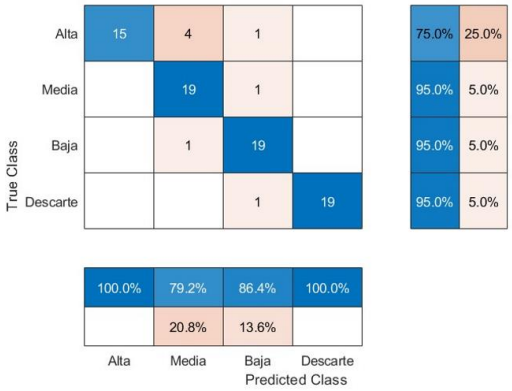

Figura 12. Matriz de confusión obtenida en la fase de validación del experimento AF/CAD/80 realizado con la arquitectura GoogLeNet optimizada.

Finalmente, para evaluar de forma gráfica los resultados del modelo optimizado, se desarrolló en Matlab una interfaz gráfica en la que dada la imagen de la copa de un olivo se obtiene de forma cualitativa la carga frutal de las diferentes regiones del olivo (Figura 13).

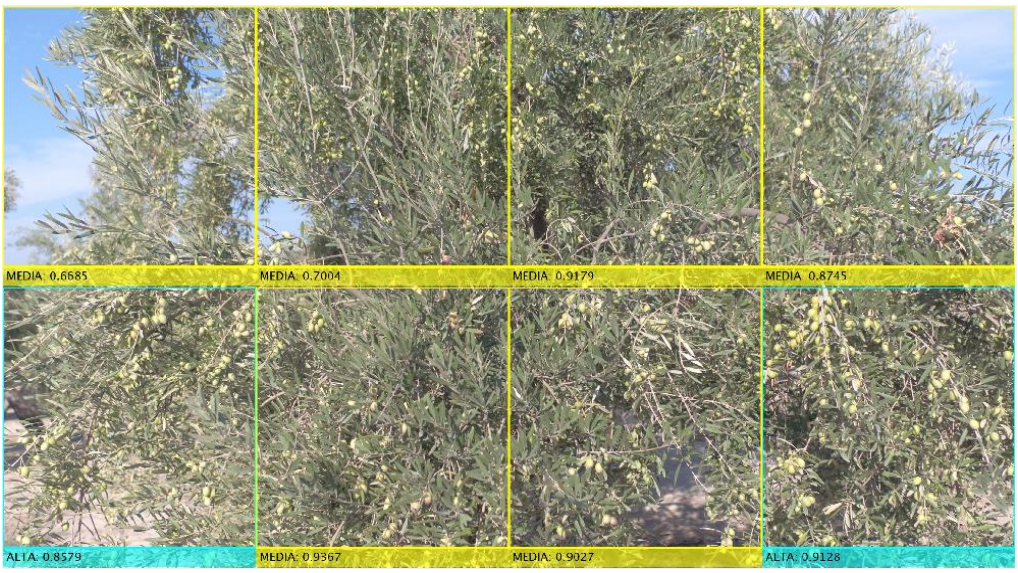

Figura 13. Interfaz gráfica desarrollada para etiquetar la carga frutal de las diferentes regiones de un olivo.

https://doi.org/10.17979/spudc.9788497498418.0956

4 CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se ha planteado una metodología basada en UAV y CNNs para la adquisición, procesado y clasificación cualitativa de imágenes para la predicción de la carga de aceitunas de olivos. Se han evaluado tres arquitecturas (GoogLeNet, ResNet y AlexNet) y se ha comprobado que la configuración óptima se obtiene con GoogLeNet entrenada con la técnica de ajuste fino, utilizando aumento de datos y empleando un 80% del conjunto de datos para el entrenamiento y un 20% para la validación, alcanzando una tasa de acierto del 90% de imágenes correctamente clasificadas. Esta metodología puede ser útil para valorar de forma rápida la cantidad de aceituna que va a producir una plantación de olivos en los meses previos a la campaña de recogida.

Como trabajo futuro se plantea ampliar el estudio contrastando las estimaciones hechas según la metodología propuesta y la producción en kilogramos de aceitunas. Además, se considera interesante el desarrollo de una arquitectura software que permita integrar esta información con los sistemas de gestión de las almazaras.

Agradecimientos

Este trabajo de investigación ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación de España bajo el proyecto PID2019-110291RB-I00 y el proyecto I+D+i en el marco de la cooperativa FEDER- Andalucía con el código FEDER A1123060E00010 y la referencia 1380776.

Referencias

[1] A. D. Boursianis *et al.*, "Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review," *Internet of Things*, vol. 18, p. 100187, May 2022,

[2] P. C. Marchal, D. M. M. Gila, S. I. Rico, J. G. Ortega, and J. G. García, "Assessment of the Nutritional State for Olive Trees Using UAVs," in *CONTROLO*, 2021, pp. 284–292.

[3] M. Noguera *et al.*, "Nutritional status assessment of olive crops by means of the analysis and modelling of multispectral images taken with UAVs," *Biosyst. Eng.*, vol. 211, pp. 1–18, Nov. 2021,

[4] L. Ortenzi *et al.*, "Early Estimation of Olive Production from Light Drone Orthophoto, through Canopy Radius," *Drones 2021, Vol. 5, Page 118*, vol. 5, no. 4, p. 118, Oct. 2021,

[5] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning – Method overview and review of use for fruit detection and yield estimation," *Comput. Electron. Agric.*, vol. 162, pp. 219–234, Jul. 2019,

[6] S. W. Chen *et al.*, "Counting Apples and Oranges with Deep Learning: A Data-Driven Approach," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 781–788, Apr. 2017,

[7] M. Rahnemoonfar and C. Sheppard, "Deep Count: Fruit Counting Based on Deep Simulated Learning," *Sensors 2017, Vol. 17, Page 905*, vol. 17, no. 4, p. 905, Apr. 2017,

[8] O. E. Apolo-Apolo, M. Pérez-Ruiz, J. Martínez-Guanter, and J. Valente, "A Cloud-Based Environment for Generating Yield Estimation Maps From Apple Orchards Using UAV Imagery and a Deep Learning Technique," *Front. Plant Sci.*, vol. 11, p. 1086, Jul. 2020,

[9] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019,

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017,

[11] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, pp. 1–9, Oct. 2015,

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, Dec. 2015,

[13] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015,

© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC-SA 4.0 license (https://creativecommons.org/licenses/by-nc-sa/4.0).