

Modelo de Detección de Frutas Frescas con Red Neuronal Convolutiva VGG16

Johana Catalina Ríos Torres
jcatalina.rios@udea.edu.co

Luis Felipe Cadavid Chica
felipe.cadavid@udea.edu.co

Simulación de Sistemas y Laboratorio
Prof. Antonio Tamayo Herrera.
Departamento de Ingeniería de Sistemas, Universidad de Antioquia
Colombia

Resumen— Este artículo presenta un estudio detallado sobre el diseño, análisis y simulación de un modelo de aprendizaje profundo para la detección y clasificación de frutas frescas o descompuestas. El problema de detección de frutas es un desafío significativo en la industria agrícola, donde la clasificación rápida y precisa de las frutas puede facilitar la selección y el empaquetado, mejorando la eficiencia y reduciendo el desperdicio.

El trabajo se desarrolló en varias etapas. Primero, se recopiló un conjunto de datos de imágenes de frutas frescas. Luego, se preprocesaron las imágenes para su uso con el modelo de aprendizaje profundo. A continuación, se diseñó y entrenó un modelo basado en la arquitectura de Red Neuronal Convolutiva VGG16, pre-entrenadas en ImageNet, también se utilizó ResNet, InceptionV3 y DenseNet121.

El modelo VGG16 tiene varias restricciones, incluyendo la necesidad de grandes cantidades de datos de entrenamiento, el tiempo de entrenamiento computacionalmente intensivo, el riesgo de sobreajuste y la dificultad con variaciones complejas en las imágenes de frutas.

Para validar el primer modelo, se utilizó un enfoque de validación de retención (holdout validation). El número de pasos por época para el entrenamiento y la validación se calcula como el número total de muestras en el conjunto de entrenamiento o validación dividido por el tamaño del lote, se utilizaron varias métricas como: Accuracy, los, la Precisión, Recall (sensibilidad), F1-Score y la matriz de confusión.

Las conclusiones obtenidas indican que el modelo CNN ResNet50 demostró un rendimiento superior en comparación con CNN VGG16 y con los métodos tradicionales de visión por computadora. Sin embargo, se incluyó la exploración de otras técnicas de preprocesamiento de imágenes.

Keywords— Detección frutas frescas, Convolutional Neural Network, VGG16, Deep learning, Procesamiento imágenes, Pooling, Data Augmentation, GPU computing, Keras, TensorFlow, Metrics, Optimización.

I. INTRODUCCIÓN

La predicción de frutas frescas es un problema significativo en la industria agrícola y de alimentos. La capacidad de clasificar y predecir la calidad de las frutas frescas rápidamente y precisa

puede afectar la eficiencia de la cadena de suministro de alimentos, desde la cosecha hasta el empaquetado y la distribución. Además, puede ayudar a reducir el desperdicio de alimentos al permitir una mejor gestión de las existencias y una planificación más precisa.

Hay 6 categorías de frutas: manzanas frescas, naranjas frescas, plátanos frescos, manzanas podridas, naranjas podridas y plátanos podridos, donde el objetivo es clasificar las imágenes de frutas en diferentes categorías basadas en su tipo y calidad. En este contexto, las variables de entrada al sistema son las imágenes de las frutas, que son procesadas y transformadas utilizando técnicas de aprendizaje profundo.

El proceso general empieza cuando las etiquetas de las clases se transforman usando one-hot encoding. Durante el entrenamiento, la CNN produce una distribución de probabilidad sobre las clases, usando una capa softmax al final. Luego la función de pérdida categorical_crossentropy se utiliza para comparar las predicciones del modelo (distribución de probabilidades) con las etiquetas reales (en forma de vectores one-hot).

El campo de aplicación de este problema es amplio, las técnicas desarrolladas también pueden ser aplicables en otros contextos donde se requiere la clasificación de imágenes, como la medicina, la seguridad y la visión por computadora en general.

II. ANTECEDENTES

Artículo 1:

En los últimos años, el uso de técnicas de aprendizaje profundo para la detección de la calidad de las frutas y vegetales ha ganado una atención significativa. La madurez y frescura de las frutas son factores críticos que afectan directamente su valor de mercado y su aceptabilidad para el consumidor.

Sin embargo, la fruta de mora no muestra rasgos visibles de madurez obvios y confiables cuando está madura. La mora madura, para el ojo humano, es negra antes, durante y después de la maduración. Para abordar este problema, este artículo 1 propone la configuración de VGG16 (Simonyan y Zisserman, 2014) [1] cambia con 1) el tipo de entrada a la red que en este

caso son las imágenes estéreo (biespectrales). Se considera un tamaño de imagen de $32 \times 32 \times 3$ para minimizar el costo computacional, los hiperparámetros de múltiples entradas se optimizan mediante búsqueda de cuadrícula y validación cruzada estratificada de k veces [1]. Para compilar el algoritmo de entrenamiento, se prefiere la función de activación de unidad lineal rectificadora (ReLU) [2] (Agarap, 2018) porque supera el problema del gradiente de fuga.

Para determinar la mejor función de optimización para CNNvgg16 de múltiples entradas, los experimentos incluyeron comparar varias funciones de optimización. En la figura 2 se muestran los hyperparametros y las funciones de optimización del optimizador Adam (Kingma y Ba, 2014) [3], el descenso de gradiente estocástico SGD (Sutskever et al., 2013) [4] y el gradiente adaptativo AdaGrad (Duchi et al., 2011) [5] con su tasa de aprendizaje establecida en 0,01, 0,01, 0,001 y 0,001, también se evaluaron diferentes modelos para detección de madurez de la mora, ver figura 1 respectivamente.

Model	Depth	Image Size	FC Layer	Precision	Recall	F1-Score	Test Accuracy
DenseNet	242	$32 \times 32 \times 3$	$N^{112,256}$	0.872	0.735	0.772	0.870
InceptionV3	189	$75 \times 75 \times 3$	$N^{286,64}$	0.906	0.829	0.797	0.829
Xception	81	$71 \times 71 \times 3$	$N^{112,256}$	0.777	0.683	0.712	0.683
MobileNet	55	$75 \times 75 \times 3$	$N^{112,256}$	0.873	0.876	0.875	0.870
Vgg16	16	$32 \times 32 \times 3$	$N^{1024,1024}$	0.898	0.902	0.897	0.902

Fig. 1. Evaluacion de diferentes modelo de detecccion madures de la mora. [1]

Table 2
The multi-input CNN_vgg16 classifier hyperparameters.

Hyperparameter	Description	Initial Values
FC Layer	The fully connected (FC) layer defines the number of neurons connected to each input from the last convolutional layer of the pre-trained VGG16 network for the weights of the target task to be learned.	$N^{1024,1024}$
Optimization Function	An algorithm that minimizes the loss function of the network during training.	Adam
Batch Size	The number of samples utilized in one iteration.	10
Epoch	Defines the number of passes made through the entire training dataset during training.	20

Fig. 2. Tabla hiperparemetros del clasificador CNN [2]

Artículo 2:

Este artículo propone un método novedoso que fusiona diferentes modelos de aprendizaje profundo para extraer las imágenes de frutas y verduras y la correlación entre varias áreas de la imagen, para detectar la frescura de frutas y verduras de manera más eficiente.

Los resultados experimentales muestran que el modelo de aprendizaje profundo CNN_BiLSTM, que fusiona CNN y BiLSTM, se combina con el procesamiento de optimización de parámetros para lograr una precisión máxima del 97,76% en la detección de la frescura de frutas y verduras. Los resultados de la investigación muestran que este método puede extraer automáticamente características profundas que caracterizan la frescura de frutas y verduras, y es adecuado para la detección de frescura de verduras y frutas. [1]

Este estudio seleccionó el optimizador Adam que combina el método de impulso y la tasa de aprendizaje adaptativo para optimizar el modelo de aprendizaje profundo CNN_BiLSTM y se utiliza como función de pérdida cross-entropy con las ventajas de alta eficiencia y estabilidad.

En la Figura 3, el aprendizaje profundo CNN_BiLSTM, el modelo propuesto en este estudio tiene la mayor detección de frescura de frutas y verduras Acc, Pre, Rec y F1, que son 97,76%, 95,63%, 97,17% y 0,97 respectivamente.

Table 2
The comparison of model performance in fruit and vegetable freshness detection.

Model	Acc	Pre	Rec	F1
CNN	96.07%	92.16%	95.62%	0.94
BiLSTM	94.62%	92.49%	93.27%	0.92
CNN_LSTM	97.13%	94.86%	96.13%	0.96
CNN_BiLSTM	97.76%	95.63%	97.17%	0.97

Fig. 3. Tabla comparación de rendimientos en detección de frutas y vegetales . [3]

Artículo 3:

Uno de los temas en la industria hortícola es la clasificación de la fruta, para superar este problema se requiere un sistema automatizado que pueda clasificar diferentes tipos de frutas sin necesidad de ningún esfuerzo humano. En este estudio se utilizaron 26.149 imágenes de 40 tipos diferentes de frutas para la experimentación. El conjunto de entrenamiento y prueba se recreó aleatoriamente y se dividió en una proporción de 3:1. El experimento introduce un cabezal personalizado de cinco capas diferentes en la arquitectura MobileNetV2. La capa de clasificación del modelo MobileNetV2 se reemplaza por el encabezado personalizado, que produjo la versión modificada de MobileNetV2 llamada TL-MobileNetV2. [2]

Además, el modelo TL MobileNetV2 obtuvo un 99 % de precisión, un 99 % de recuperación y una puntuación F1 del 99 %. Se puede concluir que el aprendizaje por transferencia juega un papel importante en el logro de mejores resultados y la técnica de abandono ayuda a reducir el sobreajuste en el aprendizaje por transferencia.

Presenta el rendimiento del modelo propuesto en términos de training accuracy, training loss y validation accuracy, El modelo propuesto fue entrenado durante 100 iteraciones. [2]

Artículo 4:

El valor agregado en un fruto puede aumentarse con un buen manejo postcosecha. La clasificación en diferentes parámetros es de las operaciones más importantes. En las pequeñas empresas se realizan de forma manual obteniendo deficiencias en la calidad del producto. Estos problemas se podrían resolver o disminuir con la implementación de algoritmos inteligentes que en este caso incluyen visión artificial y redes neuronales artificiales. En este proyecto se presenta la clasificación de manzanas mediante un algoritmo inteligente, usando una red neuronal convolucional (CNN), la cual se desarrolla utilizando librerías de código abierto (OpenCV, Tensorflow y Keras) en Python con una estructura de diferentes capas convolucionales y MaxPooling. Para 2.800 imágenes de 128x128 píxeles, de las

que el 80% se usaron para entrenamiento y el 20% para hacer una prueba de la red, obteniendo una precisión del 98,3% y del 95,36%, respectivamente. Después del entrenamiento, se realizó una clasificación con un vídeo en tiempo real, obteniendo una precisión del 92,25%. [3]

III. EXPERIMENTOS

La biblioteca Keras ha demostrado ser una herramienta invaluable en el campo del aprendizaje profundo, especialmente en la implementación de redes neuronales convolucionales que se expone en este artículo, permitiendo a los usuarios centrarse en el diseño y la experimentación de modelos. Keras proporciona clases para definir capas convolucionales (Conv2D para imágenes 2D, Conv3D para imágenes 3D), capas de pooling (MaxPooling, AveragePooling), y capas completamente conectadas (Dense).

Para la ejecución de estos modelos, se requiere una cantidad significativa de capacidad de cálculo y memoria. Aquí es donde entran en juego las Unidades de Procesamiento Tensorial (TPU) y la alta capacidad de RAM disponibles en Google Colab. Las TPU son aceleradores de hardware diseñados para acelerar las operaciones de tensor, fundamentales en el aprendizaje profundo.

A continuación, se detallan los pasos que se siguieron para desarrollar y evaluar los modelos de detección de frutas frescas:

1. El Dataset

El conjunto de datos proviene de [Kaggle](#). La estructura del conjunto de datos se encuentra en la carpeta `data/fruits`. Hay 6 categorías de frutas: manzanas frescas, naranjas frescas, plátanos frescos, manzanas podridas, naranjas podridas y plátanos podridos. Esto significa que el modelo requerirá una capa de salida de 6 neuronas para realizar la categorización con éxito. También se necesita compilar el modelo con `categorical_crossentropy`, ya que tenemos más de dos categorías.

2. Cargar el modelo base de ImageNet

Además, se diseñó y entrenó con Keras un modelo basado en la arquitectura de red neuronal convolucional VGG16, pre-entrenada en ImageNet, es una colección de imágenes de frutas de alta calidad tomadas en diferentes condiciones de iluminación y orientación. Estos datos incluyen imágenes de 60 tipos de frutas, con 30 000 imágenes. Cada imagen está etiquetada con el tipo de fruta y la calidad de la fruta (fresco y sobre maduro), lo que nos permite entrenar y validar nuestro modelo para estas tareas de clasificación. Recientemente, estos modelos de detección de objetos funcionan mejor que los humanos, con una precisión superior al 95%. Si no se decide utilizar VGG16, se tendría que elegir otro modelo pre-entrenado o entrenar uno desde cero.

Se carga este modelo con los pesos correctos, se establece una forma de entrada y se elige eliminar las últimas capas del modelo. Estas imágenes tienen tres dimensiones: alto, ancho y varios canales. Debido a que estas imágenes están en color, habrá tres canales para rojo, verde y azul.

3. Congelar modelo base

A continuación, se congela el modelo base, esto se hace para que todo el aprendizaje del conjunto de datos de ImageNet no se destruya en el entrenamiento inicial.

4. Agregar capas al modelo

Ahora es el momento de agregar capas al modelo previamente entrenado. Hay que prestar mucha atención a la última capa densa y asegurarse de que tenga el número correcto de neuronas para clasificar los diferentes tipos de fruta.

5. Compilar modelo

Ahora es el momento de compilar el modelo con opciones de métricas y pérdidas. Hay que recordar que estamos entrenando en varias categorías diferentes, en lugar de un problema de clasificación binaria.

6. Cargar conjunto de datos

Ahora es el momento de cargar el train y los conjuntos de datos de validación. Se elijen las carpetas correctas, así como el `target_size` correcto de las imágenes (debe coincidir con la altura y el ancho ingresados del modelo que ha creado).

7. Entrenar al modelo

Para validar el primer modelo, se utilizó un enfoque la validación de retención (holdout validation). El número de pasos por época para el entrenamiento y la validación se calcula como el número total de muestras en el conjunto de entrenamiento o validación dividido por el tamaño del lote.

Se pasan los iteradores `train` y `valid` a la función `fit`, además de establecer el número deseado de épocas.

8. Descongelar modelo para realizar ajustes finos

Si se alcanzó el 92% de precisión de validación, el siguiente paso es opcional. De lo contrario, se ajusta el modelo con una tasa de aprendizaje muy baja.

9. Evaluar el modelo

Ahora se tiene un modelo que tiene una precisión de validación del 92% o más. De lo contrario, volver atrás y ejecutar más épocas de entrenamiento o ajustar el aumento de datos.

Una vez que se esté satisfecho con la precisión de la validación, la función de evaluación devolverá una tupla, donde el primer valor es su pérdida y el segundo valor es su precisión. Para aprobar, el modelo deberá tener un valor de precisión del "92% o superior".

Las instrucciones para lanzar el modelos estudiados en este artículo con Docker, visitar el proyecto con nombre fruit_sorter_project y leer el README de Github.

IV. EVALUACION DEL MODELO

Examinamos el rendimiento de varios modelos de redes neuronales convolucionales (CNN) en la tarea de detectar el estado de maduración de las frutas. Nuestro enfoque principal

fue el modelo VGG16, Sin embargo, para asegurar una evaluación equilibrada, también consideramos otros modelos de CNN prominentes en el campo del aprendizaje profundo, incluyendo ResNet, InceptionV3 y DenseNet121. Cada modelo se entrenó y validó utilizando el mismo conjunto de datos, generamos un reporte de clasificación que incluye precisión, recall, f1-score support y accuracy, como se ve en la figura 5. tambien se generaron matrices de confusión que veremos acontinuacion:

Se presenta la matriz de confusion del Modelo Vgg16 y se observalo siguiente:

La diagonal principal que presenta las predicciones está destacadamente alta, con valores 384, 381, 371, 586, 529 y 400 para cada una tipo de fruta. Esto indica que el modelo tiene una alta precisión en la clasificación correcta de las imágenes en sus respectivas clases. Los valores fuera de la diagonal sugiere que los errorer de clasificacion son relativamente bajos, ver la figura 4.

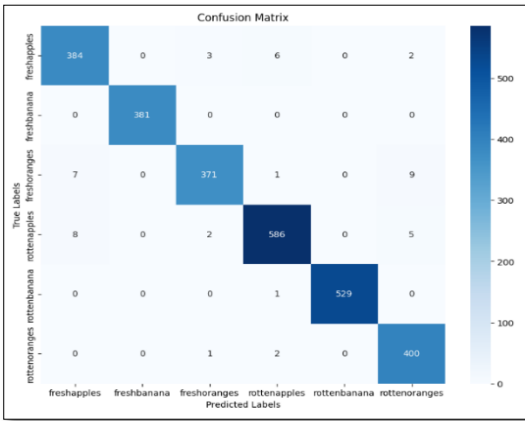


Fig. 4. Matriz de confusión Modelo Vgg16

La exactitud (Accuracy) general del modelo es del 987%, lo que se calcula como la suma de las predicciones correctas (sumatoria de la diagonal) dividida por el total de instancias, ademas muestra la precision REcall y F1-Score para cada clase, ver figura 5.

Classification Report:				
	precision	recall	f1-score	support
freshapples	0.96	0.97	0.97	395
freshbanana	1.00	1.00	1.00	381
freshoranges	0.98	0.96	0.97	388
rottenapples	0.98	0.98	0.98	601
rottenbanana	1.00	1.00	1.00	530
rottenoranges	0.96	0.99	0.98	403
accuracy			0.98	2698
macro avg	0.98	0.98	0.98	2698
weighted avg	0.98	0.98	0.98	2698

Fig. 5. Reporte de clasificacion modelo CNN VGG16

La matriz de confusión del Modelo ResNet50 de la figura 6, Sorprendentemente, alcanzó un Accuracy del 99%, superando significativamente al modelo VGG16. No hay errores de clasificación significativos, todas las clases tienen una precisión cercana al 100%. ResNet50 supera a VGG16 en todas las métricas de evaluación para cada clase.

La consistencia de las métricas (Precisión, Recall y F1-Score) en ResNet50 es más alta y uniforme. Ver figura 7.

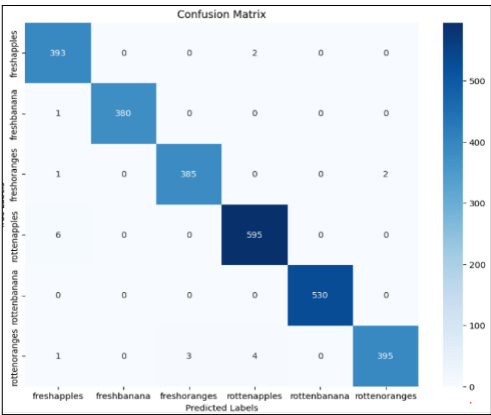


Fig. 6. Matriz de confusión Modelo ResNet50

Classification Report:				
	precision	recall	f1-score	support
freshapples	0.98	0.99	0.99	395
freshbanana	1.00	1.00	1.00	381
freshoranges	0.99	0.99	0.99	388
rottenapples	0.99	0.99	0.99	601
rottenbanana	1.00	1.00	1.00	530
rottenoranges	0.99	0.98	0.99	403
accuracy			0.99	2698
macro avg	0.99	0.99	0.99	2698
weighted avg	0.99	0.99	0.99	2698

Fig. 7. Reporte de clasificación modelo CNN ResNet50

A continuación, la matriz de confusión obtenida del modelo CNN *MobileNet*, la diagonal principal muestra que tiene valores altos, indicando que la mayoría de las instancias son clasificadas correctamente. Hay errores significativos en ciertas clases. La Clase rottenapples, en particular, muestra 41 instancias clasificadas erróneamente como Clase freshorange. Esta confusión sugiere que el modelo tiene dificultades para distinguir entre las clases rottenapples, y freshorange, posiblemente debido a características similares en los datos, ver figura 8.

Exactitud (Accuracy) global del modelo es del 93%, calculada como la suma de las predicciones correctas (sumatoria de la diagonal) dividida por el total de instancias, ver figura 9.

Se puede notar que las clases freshbanana y rottenbanana de los primeros tres modelos mencionados, la Precision, el Recall, f1-score es de casi 100%.

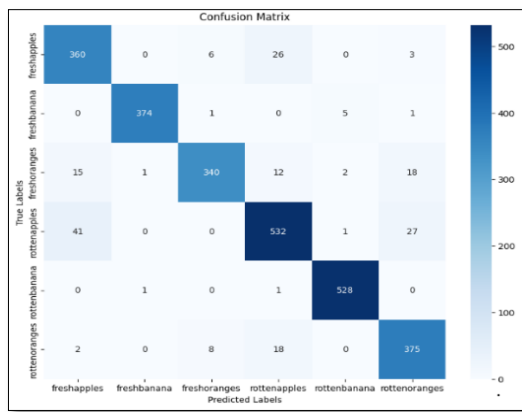


Fig. 8. Matriz de confusion modelo MobileNet.

Classification Report:				
	precision	recall	f1-score	support
freshapples	0.86	0.91	0.89	395
freshbanana	0.99	0.98	0.99	381
freshoranges	0.96	0.88	0.92	388
rottenapples	0.90	0.89	0.89	601
rottenbanana	0.99	1.00	0.99	530
rottenoranges	0.88	0.93	0.91	403
accuracy			0.93	2698
macro avg	0.93	0.93	0.93	2698
weighted avg	0.93	0.93	0.93	2698

Fig. 9. Reporte de clasificacion modelo CNN MobileNet.

Para el caso del modelo CNN InceptionV3, La diagonal principal está desvanecida, indicando que muchas instancias no están clasificadas correctamente en sus respectivas clases. La columna correspondiente a la Clase 'rottenbanana' muestra valores altos en todas las filas, indicando que muchas instancias de diferentes clases están siendo clasificadas erróneamente como 'rottenbanana', esto sugiere una tendencia del modelo a clasificar incorrectamente muchas instancias en esta categoría. Ver figura 10

El modelo tiene una precisión global (accuracy) del 37%, lo cual es bastante bajo y refleja problemas significativos en la capacidad del modelo para clasificar correctamente las instancias, sin hablar de las metricas de evaluación. Ver figura 11.

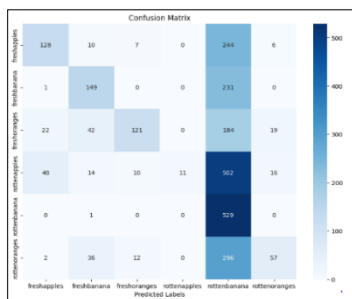


Fig. 10. Matriz de confusion Modelo InceptionV3

Classification Report:				
	precision	recall	f1-score	support
freshapples	0.64	0.32	0.43	395
freshbanana	0.59	0.39	0.47	381
freshoranges	0.81	0.31	0.45	388
rottenapples	1.00	0.02	0.04	601
rottenbanana	0.27	1.00	0.42	530
rottenoranges	0.58	0.14	0.23	403
accuracy			0.37	2698
macro avg	0.65	0.36	0.34	2698
weighted avg	0.65	0.37	0.32	2698

Fig. 11. Reporte de clasificacion modelo CNN InceptionV3

Para finalizar, el modelo CNN DenseNet121, la diagonal principal tiene valores relativamente altos, lo que indica que la mayoría de las instancias están siendo clasificadas correctamente. Sin embargo, hay cierta confusión en algunas clases.

La clase rottenapples, por ejemplo, tiene 593 instancias correctamente clasificadas, mientras que la clase rottenbanana tiene 492 instancias correctamente clasificadas. La columna correspondiente a la clase 'rottenapples' muestra valores altos, indicando que muchas instancias de diferentes clases están siendo clasificadas erróneamente como 'rottenapples'. Ver figura 12.

El modelo tiene una precisión global (accuracy) del 72%, lo cual indica que hay un número significativo de errores de clasificación.

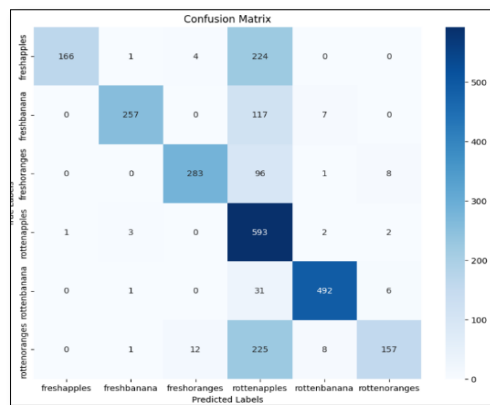


Fig. 12. Matriz de confusion Modelo DenseNet121

Classification Report:				
	precision	recall	f1-score	support
freshapples	0.99	0.42	0.59	395
freshbanana	0.98	0.67	0.80	381
freshoranges	0.95	0.73	0.82	388
rottenapples	0.46	0.99	0.63	601
rottenbanana	0.96	0.93	0.95	530
rottenoranges	0.91	0.39	0.55	403
accuracy			0.72	2698
macro avg	0.88	0.69	0.72	2698
weighted avg	0.85	0.72	0.72	2698

Fig. 13. Reporte de clasificacion modelo CNN DenseNet121

Los modelos VGG16 y ResNet50 destacaron por su alta precisión, superando el 98% en la tarea de clasificación multiclase. Ambos modelos muestran valores altos en la diagonal principal, indicando que la mayoría de las instancias fueron clasificadas correctamente, figura 14. ResNet50 no tiene errores fuera de la diagonal, mientras que VGG16 tiene unos pocos errores mínimos, como en la Clase ‘freshorange’ y Clase ‘rottenapples’, ver figura 4 y 6.

TABLE 1

Modelo	Evaluación de diferentes modelos CNN para detección frutas frescas			
	Precision	Recall	F1-Score	Accuracy
Vgg16	0.98	0.98	0.98	0.98
ResNet50	0.99	0.99	0.99	0.99
MobileNet	0.93	0.93	0.93	0.93
InceptionV3	0.37	0.37	0.37	0.37
DenseNet121	0.72	0.72	0.72	0.72

Fig. 14. Evaluación de los modelo entrenados en este artículo.

V. DISCUCIÓN

En el ámbito de la detección de frutas frescas, la literatura existente destaca varios modelos de redes neuronales convolucionales (CNN) que han sido utilizados con éxito. Entre los estudios consultados, el Artículo 1 (*Convolutional neural network ensemble learning for hyperspectral imaging-based blackberry fruit ripeness detection in uncontrolled farm environment*) resulta especialmente relevante, ya que evaluó los modelos VGG16, DenseNet, InceptionV3 y MobileNet, concluyendo que VGG16 tenía el mejor rendimiento en términos de precisión y métricas generales. Se debe aclarar que este articulo utilizó un dataset de frutas diferente al modelo que se entrenó.

Por otro lado, el mejor modelo entrenado en este proyecto, con las mejores métricas los tiene el modelo ResNet50, ha demostrado ser significativamente más efectivo que el modelo VGG16 en la tarea de clasificación multiclase en este conjunto de datos. La precisión casi perfecta y la ausencia de errores de clasificación destacables subrayan la superioridad de ResNet50 en términos de rendimiento.

En el artículo 3 (*Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique*) el modelo TL MobileNetV2 obtuvo un 99 % de precisión, un 99 %. [2]
 En el artículo 2 (*An innovative approach to detecting the freshness of fruits and vegetables through the integration of convolutional neural networks and bidirectional long short-term memory network*) modelo de aprendizaje profundo

CNN_BiLSTM, que fusiona CNN y BiLSTM, se combina con el procesamiento de optimización de parámetros para lograr una precisión máxima del 97,76% [1]
 Por otra parte, los modelos de redes neuronales convolucionales (CNN) han demostrado ser extremadamente efectivos en la detección y clasificación de imágenes. Sin embargo, también presentan ciertas limitaciones que pueden afectar su rendimiento y aplicabilidad, los modelo CNN son computacionalmente intensivo y requieren gran cantidad de memoria y procesamiento, debido a su complejidad pueden tardar mucho tiempo en entrenarse, si se utilizan grandes conjuntos de datos.
 La recopilación y etiquetado de estos datos puede ser costosa y consumir mucho tiempo. Adicionalmente pueden ser propensos al sobreajuste, especialmente si el conjunto de datos de entrenamiento es pequeño.

REFERENCIAS

[1] K. &. Z. Simonyan, «Very deep convolutional networks for large-scale image recognition,» *arXiv preprint arXiv:1409.1556.*, p. 2, 2014.

[2] A. F. Agarap, «Deep learning using rectified linear units (relu),» *arXiv preprint arXiv:1803.08375.*, 2018.

[3] D. P. &. B. J. Kingma, «Adam: A method for stochastic optimization,» *arXiv preprint arXiv:1412.6980.*, 2014.

[4] I. M. J. D. G. &. H. G. Sutskever, «On the importance of initialization and momentum in deep learning. In International conference on machine learning,» pp. (pp. 1139-1147), 2013, May.

[5] J. H. E. &. S. Y. Duchi, «Adaptive subgradient methods for online learning and stochastic optimization,» *Journal of machine learning research*, p. 12(7), 2011.

[6] *. J. C. b. K. P. c. A. A. d. Yue Yuan a, «An innovative approach to detecting the freshness of fruits and vegetables,» *ScienceDirect*, p. 1, 2024.

[7] Y. Gulzar, «Fruit Image Classification Model Based on MobileNetV2 with,» p. 12, 2023.

[8] C. J.-G. C. y. O.-R. J. C. Mota-Delfin, «Clasificación de manzanas utilizando visión artificial y redes neuronales artificiales,» *Revista Ingeniería y Región*, p. 1, 2018.

