ISSN: 1692-7257 - Volumen X – Número XX - 20XX

Revista Colombiana de Tecnologías de Avanzada

Recibido: xx de mes de 20xx
Aceptado:

CONSTRUCCIÓN DE UNA BASE DE DATOS DE FRUTAS CÍTRICAS A PARTIR DEL AUMENTO DE DATOS PARA UN MODELO DE CLASIFICACIÓN CON APRENDIZAJE PROFUNDO

CONSTRUCTION OF A CITRUS FRUIT DATASET USING DATA AUGMENTATION FOR A DEEP LEARNING CLASSIFICATION MODEL

Jacson F. Cárdenas Torres, Kevin J. Jiménez Delgado
Cesar H. Valencia, Deisy C. Páez Casas

Universidad Santo Tomás, Facultad de Ingeniería Mecatrónica
Calle 9 No. 18-27, Bucaramanga, Santander, Colombia.
PBX: +57 (7) 6985858 ext. 6542.
E-mail: {jacson.cardenas, kevin.jmenez, cesar.valencia, deisy.paez}@ustabuca.edu.co

Resumen: Cada día millones de imágenes son almacenadas, ya sea por páginas web o redes sociales, generando base de datos con etiquetas. Sin embargo, no siempre son representativas, de gran variedad o de dominio público. En este artículo se presenta la construcción de una base de datos particular de frutas cítricas distribuidas en Colombia que se encuentran escasamente en repositorios, siendo estas informaciones los registros de entrada a un modelo de clasificación de única etiqueta con aprendizaje profundo. Las imágenes fueron obtenidas por medio de la opción ráfaga en una cámara de 48 MP (f2,0 con tecnología Quad Píxel, 1,6 µm), se variaron condiciones de exposición tales como la posición por medio de la rotación y la iluminación en la fruta con el fin de obtener el aumento de registros en la base de datos, previa definición de las características que identifican cada una de las frutas se realiza la selección de una cantidad limitada representativa. Mediante la utilización de diferentes técnicas se aplica un redimensionamiento espacial en las imágenes a nivel de software. De esta forma son obtenidos 20738 registros pertenecientes a 8 clases de frutas cítricas, que luego se separan implementando la regla de división 90% a 10%, en dos directorios llamados entrenamiento y prueba.

Palabras clave: Base de datos, aumento de datos, frutas cítricas, clasificación, aprendizaje profundo.

Abstract: Every day millions of images are stored, either by websites or by social networks, generating datasets with labels. However, they are not always representative, of great variety or in the public domain. This article presents the construction of a particular dataset of citrus fruits distributed in Colombia that are sparsely found in repositories, these information are the input records to a single-label classification model with deep learning. The images were obtained through the burst option on a 48 MP camera (f2.0 with Quad Pixel technology, 1.6 µm). Exposure conditions such as position were evaluated through rotation and lighting of the fruit with the purpose to obtain the increase of records in the dataset, previous defining the features that identify each of the fruits, the selection of a representative limited quantity is made. Through the use of different techniques, a spatial resizing is applied to the images at the software level. In this way, 20738 samples belonging to 8 classes of citrus fruits are obtained and they are separated by implementing the division rule 90% to 10%, in two directories called; train and test.

Keywords: Dataset, data augmentation, citrus fruits, classification, deep learning.

1. INTRODUCCIÓN

La clasificación de diferentes frutas sigue siendo una tarea importante y tediosa en la industria de procesamiento de alimentos, no solo es un tema relevante en el campo de la investigación académica sino también en aplicaciones industriales. Las personas tienden a recoger las frutas por sí mismas para asegurarse de que se ajusten perfectamente a su dieta y de acuerdo con sus gustos (Dandavate y Patodkar, 2020), pero si aparte se les informara que variedad es y qué beneficios tiene, sería muy útil, además de que esta información sería de valor agregado a los vendedores para dar un contenido adicional a los productos.

Las primeras variedades e híbridos de cítricos fueron el resultado de un largo proceso de identificación, colecta y reproducción de plantas silvestres (Roldan y Salazar, 2002).

Aunque parezca difícil de creer existen muchos tipos de limones que se ignora completamente, cada uno con diferentes características. Las variedades más importantes de limones en Colombia en el sector económico son: *Citrus latifolia* (limón Tahití), *Citrus limonia* (lima *rangpur* o limón mandarino), *Citrus jambhiri* (limón rugoso) y *Citrus aurantifolia* (Limón común, limón pajarito o también llamado limón criollo) (Arrieta y Rico, 2005). A diferencia de otros cítricos como las naranjas, no se pueden distinguir tan claramente unos de otros, pero es importante conocerlos a fondo puesto que se trata de una fruta de gran utilidad (Santomera, 2017).

Un caso puede ser en los usos caseros. Por ejemplo, el limón criollo para uso medicinal sirve para la irritación y las ronchas por las picadas del mosquito (Sabelotodo, 2010), el limón mandarino evita futuros resfríos, así mismo ayuda a prevenir que la piel se reseque y controla la obesidad (Ecured, 2020), y por último el limón Tahití reduce el riesgo de enfermedades cardiacas y derrames cerebrales y permite bajar la presión arterial (Persian, 2019), todo esto es información que se ignora a la hora de comprar una fruta.

En este caso el aprendizaje profundo es el estado del arte para tareas de visión computacional (Chollet, 2020), lo que se convierte en el algoritmo perfecto para la clasificación de este tipo de frutas con características muy similares entre sí a partir, de una base de datos, A diferencia de los algoritmos clásicos implementados por (González y Ojeda, 2003) para el reconocimiento de objetos se requiere de un trabajo manual para definir el conjunto de reglas de dicho modelo ya que, solo permitirá el reconocimiento de una pequeña cantidad de objetos en condiciones muy restringidas.

A la hora de abordar problemas con el aprendizaje automático, el conjunto de la base de datos es una de las secciones más importantes que se debe tener en cuenta ya que, según distintas encuestas realizadas a científicos de datos, en torno a un 80% del tiempo de trabajo se centra en obtener, limpiar y organizar los datos mientras que tan solo el 3% del tiempo se dedica a construir modelos de aprendizaje automático (Xeridia, 2020).

Para la construcción de una base de datos de frutas según una variedad seleccionada, se debe considerar muestras que tengan la mayor cantidad posible de características representativas que definan la clase. Por ejemplo, la clase a considerar tenga muestras con diferentes colores, maduraciones, tamaños, formas, entre otras, esto para que el modelo aprenda todas las características generales, ya que estas son variables que representan cada una de las clases, permitiendo que el modelo generalice a nuevas muestras (Muresan y Oltean, 2018; Dimitrie, 2021; Olsson y Femling, 2018).

El presente artículo muestra la construcción de una base de datos para el entrenamiento, y testeo de un modelo de clasificación de única etiqueta para 8 clases con 20738 imágenes de cítricos. En la sección 2 se describe a detalle esta metodología con los respectivos resultados y discusiones en la sección 3. Se trató una discusión con la generación de una base de datos única y representativa a partir del aumento de datos para entrenar modelos de clasificación de frutas cítricas con aprendizaje por transferencia.

2. METODOLOGÍA

La presente propuesta muestra las etapas aplicadas en la metodología, como se presenta en la figura 1.

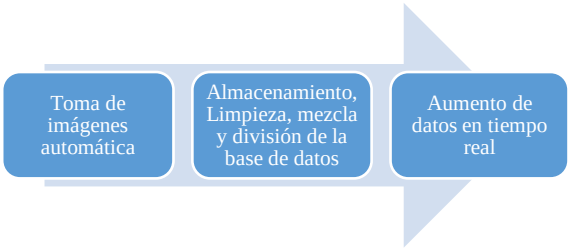


Fig. 1. Flujograma de proceso para la construcción de la base de datos

2.1. Toma de imágenes automática

El procedimiento de la toma automática se muestra en la figura 2:

Fig. 2. Ciclo continuo de la toma de imágenes automática

La base de datos es uno de los rasgos más importantes para lograr buenos resultados en los modelos de aprendizaje automático mediante aprendizaje supervisado, se debe tener en cuenta el tipo de problema para saber de qué forma es más sencillo y rápido construir una base de datos concisa.

El dispositivo móvil que se usó para la toma de imágenes fue un *Motorola G8 Plus, con* cámara principal trasera de 48MP. Se tomaron 20738 imágenes en formato *JPG* a una resolución de 4000x3000 utilizando el modo ráfaga en secuencias de 200 imágenes por cítrico con diferentes condiciones de exposición como la rotación automática, iluminación y la pose del objeto, ver figura 3.

(a) (b) (c)

Fig. 3. Cambio de pose de la fruta. (a) Posición boca arriba. (b) Posición boca abajo. (c) Posición lateral (Fuente: Autor)

El mecanismo que permite la rotación del objeto fue construido a partir de una base de madera de 50x30x4.5 cm en posición lateral que soporta una bandeja en acero inoxidable de 36x24 cm acoplada a un motor que gira a una rapidez de más o menos 3 RPM, similar al mecanismo que implementaron (Dmitriev, 2019; Muresan y Oltean, 2018), ver figura 4:

Fig. 4. Mecanismo de rotación. (Fuente: Autor)

El giro de la bandeja hace que la fruta vaya cambiando de orientación automáticamente, mientras que la cámara actúa en modo ráfaga y extrae diferentes imágenes de la fruta durante su rotación, la luz que recibe la fruta es variable por medio de una lámpara led de luz blanca, modificando así su brillo (ver figura 5). Con esto ya se pueden obtener grandes cantidades imágenes con pocos objetos o frutas, en vez de realizar un arduo trabajo de campo en tomar manualmente cientos de imágenes.

Fig. 5. Secuencia de rotación mientras se varia la luminosidad del objeto. (Fuente: Autor)

2.2. Almacenamiento, limpieza, mezcla y división de la de datos

La figura 6 muestra el proceso de almacenamiento, limpieza, mezcla y división de la base de datos:

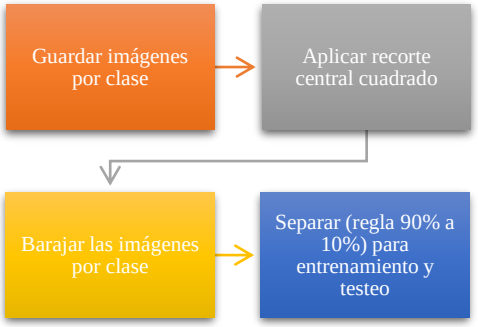


Fig. 6. Proceso de limpieza, mezcla y división de la base de datos

La estructura que se determinó para el almacenamiento de las imágenes de acuerdo con las clases se muestra en la figura 7:

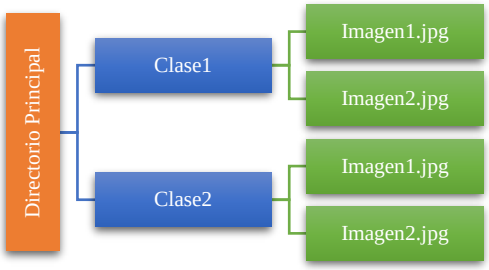
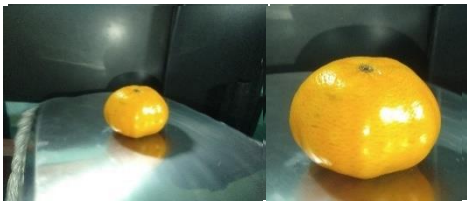


Fig. 7. Estructura de almacenamiento

Como se pudo apreciar en la subsección 2.1, las imágenes capturadas contienen mucho fondo y debido a que los modelos de clasificación con aprendizaje por transferencia se entrenan en su mayoría con imágenes de relación dimensional cuadrada (Keras, 2016), es necesario poder realizar un redimensionamiento de las imágenes eliminando parte del fondo a partir de un recorte central cuadrado. El resultado final será una imagen con dimensiones cuadradas, teniendo en cuenta que la relación de aspecto no se altere ya que, la relación de aspecto original de 4:3 (ver figura 8a) difiere con la de destino de 1:1, respectivamente (ver figura 8b).



(a) (b)

Fig. 8. (a) Imagen original-4:3. (b) Recorte central cuadrado-1:1. (Fuente: Autor)

Por defecto las imágenes capturadas se guardan con el nombre que proporciona la aplicación de la cámara, es decir, se tiene en cuenta el formato de fecha, hora y el tipo de captura que para este caso fue el modo ráfaga y después son almacenadas manualmente por clase siguiendo la estructura de la figura 7 que por defecto se ordenan alfanuméricamente ver figura 9, lo cual no es adecuado para dividir una base de datos:

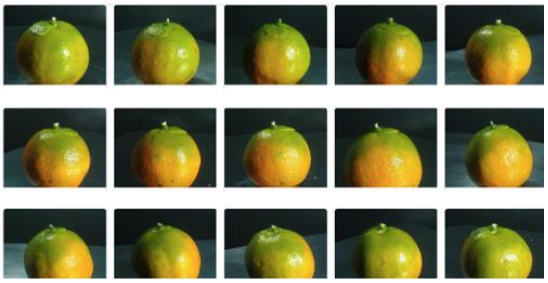


Fig. 9. Orden secuencial alfanumérico de las imágenes para la clase limón mandarino. (Fuente: Autor)

Para barajar las imágenes se implementa un algoritmo que recorre cada una de las clases y cambia el nombre de las imágenes ordenadas alfanuméricamente por un valor numérico que se genera a partir de un vector que contiene números aleatorios sin repetirse dentro del rango de cero a la longitud de muestras por clase. La figura 10 representa las imágenes barajadas.

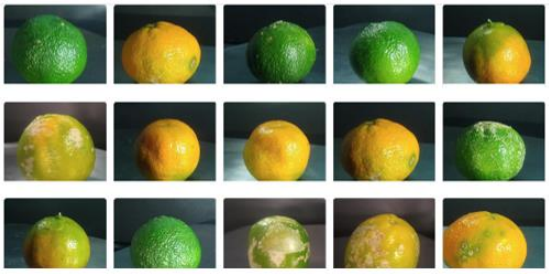


Fig. 10. Imágenes barajadas pertenecientes a la clase limón mandarino. (Fuente: Autor).

En la figura 11 se muestra la estructura final de la base de datos, dividiendo el 90% de las imágenes por clase dentro del conjunto de entrenamiento y el 10% restante para el conjunto de prueba.

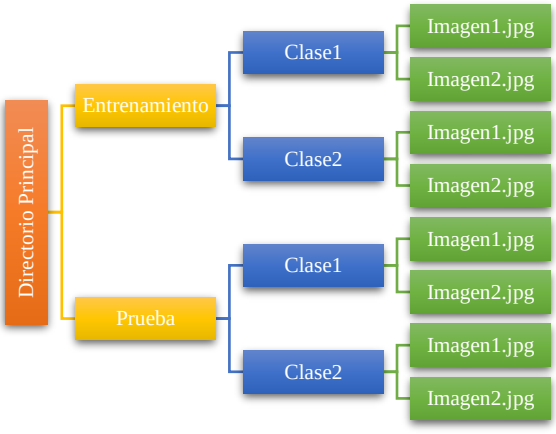


Fig. 11. Estructura Final de la base de datos

2.3. Aumento de datos en tiempo real

Para hacer más dispersa la base de datos y mitigar un poco el problema del sobreajuste en un modelo de aprendizaje profundo se utiliza la técnica del aumento de datos (Geron, 2019; Sarmiento y Villa 2008) la cual hace uso de las transformaciones lineales y cambio de brillo generando lotes de imágenes con transformaciones aleatorias durante el entrenamiento. Las transformaciones seleccionadas fueron las siguientes (ver figura 12):

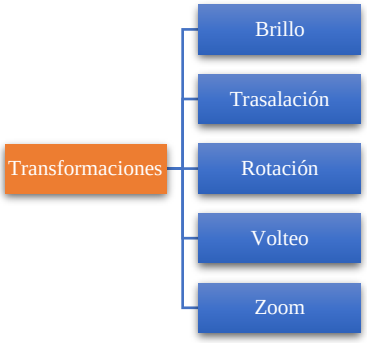


Fig. 12. Transformaciones para aumento de datos

2.3.1 Brillo

En primera instancia el brillo se configura mediante el argumento *brightness_range* al constructor de la clase *ImageDataGenerator*. Esta clase utiliza el método *flow_from_directory* que toma una ruta de un directorio y genera lotes de imágenes aumentadas en forma de tensores de precisión simple (Keras, 2015). El aumento de color por brillo altera las intensidades de los canales RGB, por lo tanto, cada píxel de una imagen en RGB original, según la Ecuación (1):

$$I_{xy} = [I_{xy}^R, I_{xy}^G, I_{xy}^B] \quad (1)$$

Se multiplica por un factor constante de escala que aumenta o disminuye los valores de intensidad, según la Ecuación (2):

$$I_{xy}^T = K \cdot [I_{xy}^R, I_{xy}^G, I_{xy}^B] \quad (2)$$

Donde I_{xy}^T representa el vector del píxel transformado y los valores de $I_{xy}^R, I_{xy}^G, I_{xy}^B$ representa las componentes del píxel en la imagen original mientras que el valor de K es un factor de escala constante que cambia aleatoriamente con cada imagen dentro del lote. Los valores de K inferiores a 1.0 oscurecen la imagen. Por ejemplo, $[0.5, 1.0]$, mientras que los valores superiores a 1.0, la iluminan. Por ejemplo, $[1.0, 1.5]$, donde 1.0 no tiene ningún efecto sobre el brillo (Brownlee, 2019). La figura 13 muestra el flujograma.

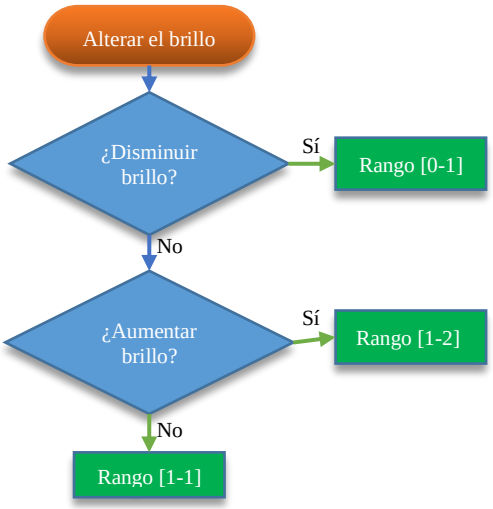
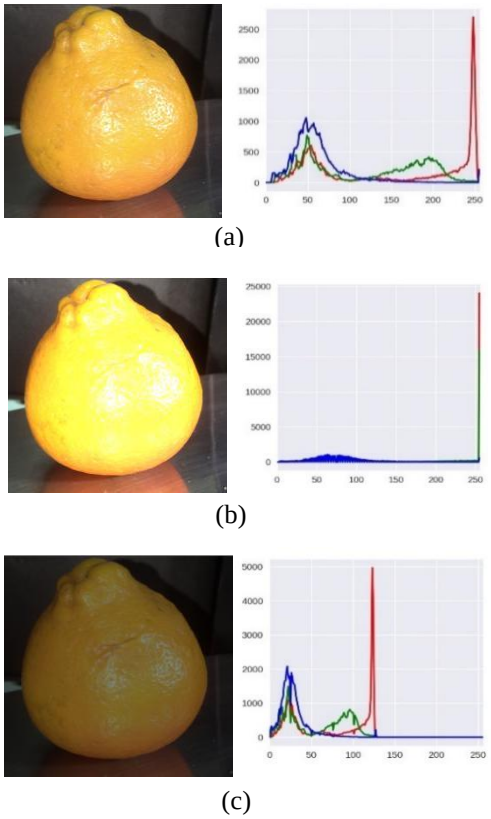


Fig. 13. flujograma para configurar el brillo. (Fuente: Autor)

El resultado de variar el brillo en una imagen con resolución de $214x214$ se muestra en la figura 14, con el respectivo histograma que representa el comportamiento de los canales RGB.



(a)

(b)

(c)

Fig. 14. Variación del brillo. (a) Imagen original, $K = 1$ (b) Imagen clara, $K > 1$ (c) Imagen oscura, $K < 1$. (Fuente: Autor)

2.3.2 Translación

La translación espacial aleatoria se hace por medio de una capa de procesamiento en *Keras*, llamada *RandomTranslation* (Keras, 2020a). Dicha transformación mueve todos los pixeles de la imagen en una dirección, tanto horizontal como verticalmente, manteniendo las mismas dimensiones de la imagen. Esto significa que algunos de los píxeles se recortarán en la imagen (figura 15b) y habrá una región donde se deberán especificar nuevos valores de píxeles (figura 15c).

(a) (b) (c)

Fig. 15. Translación espacial. (a) Imagen original. (b)Efecto de translación, modo de relleno constante. (c) Modo de relleno: vecinos cercanos. (Fuente: Autor)

La ecuación 3 es tomada de (Chollet, 2020), representa la translación fija de los pixeles originales a partir de unos factores que controlan la cantidad de desplazamiento horizontal y vertical respectivamente.

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}factor\ horizontal\\ factor\ vertical\end{bmatrix} + \begin{bmatrix}x\\ y\end{bmatrix} \qquad (3)$$

2.3.3 Rotación

La rotación aleatoria gira los pixeles en una imagen en sentido horario y antihorario en una cantidad determinada de grados de 0 a 360 a través de la capa *RandomRotation* (Keras, 2020b). Es probable que la rotación gire los píxeles fuera del marco de la imagen y deje áreas del marco sin datos de pixeles (figura 16b) que deben completarse (figura16c).

(a) (b) (c)

Fig. 16. Rotación. (a) Imagen original. (b)Efecto de rotación, modo de relleno constante. (c) Modo de relleno: vecinos cercanos. (Fuente: Autor)

La rotación de un vector 2D por un ángulo θ alrededor del origen se puede lograr mediante un producto escalar con una matriz de transformación de $2x2$ también llamada matriz de rotación en $\boldsymbol{R^2}$ definida por la ecuación 4, tomada de (Chollet, 2020):

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix} \cdot \begin{bmatrix}x\\ y\end{bmatrix} \qquad (4)$$

2.3.4 Volteo

Un volteo de imagen significa invertir las filas o columnas de píxeles en el caso de un giro vertical u horizontal, respectivamente. El aumento de volteo se especifica mediante un argumento booleano *horizontal_flip* o *vertical_flip* para la capa *RandomFlip* (Keras, 2020c), para fotografías con fondo como la imagen de la fruta utilizada en este ejemplo, los volteos horizontales pueden tener sentido (figura 17b), pero los volteos verticales no.

(a) (b)

Fig. 17. Reflexión sobre el eje Y. (a) Imagen original (b) Volteo horizontal. (Fuente: Autor)

Matemáticamente el volteo horizontal se le conoce como una reflexión de un vector 2D con respecto al eje Y que viene dada por la matriz de transformación de $2x2$, como se representa en la ecuación 5, tomada de (Redondas, 2016):

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}x\\ y\end{bmatrix} \qquad (5)$$

2.3.5 Zoom

Por último, un aumento de *zoom* expande o comprime aleatoriamente las dimensiones de una imagen y agrega nuevos valores de píxeles alrededor de la imagen o interpola los valores de píxeles, respectivamente. El *zoom* de la imagen se puede configurar mediante la capa *RandomZoom* a través de un rango entre cero y dos tanto para X y Y (Keras, 2020d). Teniendo en cuenta que los valores de *zoom* inferiores a 1.0 acercan la imagen, y los valores superiores a 1.0 alejan la imagen. Un *zoom* de 1.0 no tiene ningún efecto.

La figura 18b muestra el aumento de *zoom* en dirección Y de la imagen original mostrada en la figura 18a.

(a) (b) (c)

Fig. 18. Zoom. (a) Imagen Original. (b) Aumento de tamaño en el eje Y. (c) Modo de relleno: vecinos cercanos. (Fuente: Autor).

La ecuación 6 es tomada de (Redondas, 2016), muestra la variación de la escala vertical y horizontal en una imagen mediante un producto escalar con una matriz diagonal de $2x2$, en donde las constantes K_x y K_y representan los factores de escala.

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}K_x & 0\\ 0 & K_y\end{bmatrix} \cdot \begin{bmatrix}x\\ y\end{bmatrix} \quad (6)$$

3. RESULTADOS Y DISCUSIONES

En la tabla 1, se muestra la cantidad de imágenes recopiladas por clase.

Tabla 1: Base de datos

Etiqueta	Clase	Datos
1	Limón Criollo	2506
2	Limón Mandarino	2709
3	Limón Tahití	3415
4	Mandarina Israelí	2475
5	Mandarina Rugosa	2579
6	Naranja Valencia	2638
7	Tangelo	2085
8	Toronja	2331
total		20738

Los datos obtenidos se representan mediante un histograma en la figura 19. En primera instancia, se observa que la cantidad de datos por clase cumplen con un mínimo de 2000 imágenes, siendo una cantidad equilibrada para entrenar modelos con aprendizaje por transferencia también se observa que la base de datos es desbalanceada por una razón y tiene que ver con la cantidad de muestras representativas que conforman las clases ya que, algunas clases presentan más variedad de características que otras y, por lo tanto, se requieren de más muestras. Por ejemplo, la clase Limón Mandarino y Tahití son consideradas las clases más diversas.

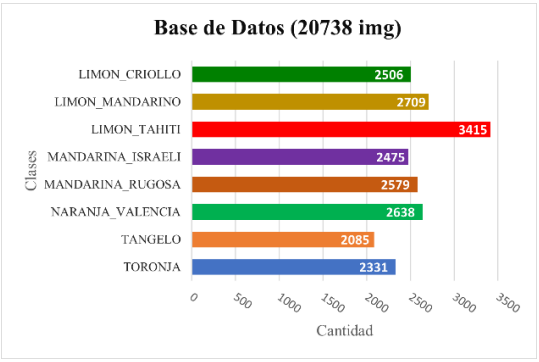


Fig. 19. Distribución de la base de datos

Es importante separa el conjunto de datos para entrenamiento y prueba, normalmente se suele repartir en un 80% de datos de entrenamiento y un 20% de datos de prueba, pero se puede variar la proporción según el caso como se muestra en la figura 20.

Fig. 20. Visualización de la separación

La tabla 2, muestra las proporciones de los datos divididos en los conjuntos de entrenamiento y prueba.

Tabla 2: división de los datos

Clase	Prueba	Entrenamiento
Limón Criollo	250	2256
Limón Mandarino	270	2439
Limón Tahití	341	3074
Mandarina Israelí	247	2228
Mandarina Rugosa	257	2322
Naranja Valencia	263	2375
Tangelo	208	1877
Toronja	233	2098
total	2069	18669

Los resultados encontrados en este estudio en cuanto a la cantidad de datos de entrenamiento por clase con una media de 2333 y una desviación estándar de 324 pueden indicar una base de datos

única y representativa para la clasificación de cítricos colombianos. Estos resultados por una parte coinciden con la literatura (Dimitrie, 2021; Olsson y Femling, 2018; Beleño *et al.,* 2020), los cuales señalan la construcción de una base de datos para modelos de aprendizaje profundo con un promedio mínimo de 400 a 1000 imágenes por clase para entrenamiento.

La investigación también concluyo que eliminar el fondo en las imágenes del conjunto de entrenamiento no conduce a una mejor clasificación precisamente porque los modelos modernos de aprendizaje profundo excluyen ese trabajo pesado y son capaces de aprender características que son invariables a su ubicación en la imagen, algo que no coincide con los trabajos de (Muresan y Oltean, 2018), quienes consideraron reemplazar el fondo de las imágenes implementando un algoritmo de tipo *flood fill.*

5. CONCLUSIONES

El desarrollo de la base de datos presentada en este documento tiene como característica principal un etiquetado de forma automática, de esta manera se elimina el error producido por la intervención humana, así como el tiempo requerido para desarrollar dicho proceso de forma manual, por otro lado, dicha base de datos es escalable para problemas de detección que conllevan un nivel de complejidad mayor, siendo los datos obtenidos de uso multipropósito permitiendo análisis más profundos bajo modelos de alta complejidad.

Los resultados muestran un conjunto de datos equilibrado para entrenar modelos con aprendizaje por transferencia bajo la implementación sutil de las técnicas artificiales del aumento de datos que evitan tener grandes cantidades de datos para entrenamiento. Además, la metodología se puede trabajar de forma detallada en futuras propuestas, donde se puede incluir más información para generalizar y mitigar el desbalance de las clases Tangelo y Toronja.

RECONOCIMIENTO

Los autores agradecen a la Universidad Santo Tomás seccional Bucaramanga por el apoyo en este trabajo. Además, un agradecimiento especial al profesor Carlos Andrés Madrigal del Grupodot por su apoyo en la creación de la base de datos para modelos de clasificación empleada para desarrollar la presente propuesta.

REFERENCIAS

Dandavate R. y Patodkar V. (2020). “CNN and Data Augmentation Based Fruit Classification Model”. IEEE Xplore. DOI: 10.1109/I-SMAC49090.2020.9243440

Roldan D. y Salazar M. (2002). *La cadena de cítricos en Colombia*, Ministerio de Agricultura y Desarrollo Rural , Observatorio Agrocadenas Colombia. Recuperado de http://repiica.iica.int/docs/B0148e/B0148e.pdf

Arrieta P., E. y Rico H., H. (2005). *Producción de plántulas de limón criollo (citrus aurantifolia) mediante la técnica de micropropagación in vitro*. Universidad de Sucre, Colombia

Chollet, F. (2020). *Deep Learning with Python*, Editorial Manning ediciones MEAP, Segunda edición. Recuperado de https://livebook.manning.com/book/deep-learning-with-python-second-edition/chapter-2/v-7/209

González P., M. A y Ojeda R., J. G. (2003). *Sistema de visión artificial para reconocimiento de objetos*, Revista Colombia de Tecnologías Avanzada RCTA , ISSN: 1692-7257

Muresan H. y Oltean M. (2018). *Fruit recognition from images using Deep learnig*, Recuperado de https://doi.org/10.2478/ausi-2018-0002 Revista: Sciendo

Dimitrie M. (2021). *Fruit Dataset. Fruit Classification.* Universidad Alexandru Ioan Cuza de Iași. Recuperado de https://www.kaggle.com/aelchimminut/fruits262

Olsson A y Femling F. (2018). *Fruit and Vegetable Identification Using Machine Learning for Retail Applications*, *Halmstad University*. DOI: 10.1109/SITIS.2018.00013

Geron A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Sarmiento M., H.O y Villa A., W. M. (2008). *Inteligencia artificial en pronóstico de demanda de energía eléctrica: una aplicación en optimización de recursos energéticos*, Revista Colombia de Tecnologías de Avanzada RCTA, ISSN: 1692-7257

Redondas. D. (2016). “Transformaciones Lineales En R^2” Universidad Politécnica de Madrid. Recuperado de https://www.edificacion.upm.es/personales/re

dondas/docencia/fundamentos/objetos/tema8b.pdf

Beleño D., S. L., Martinez L., C. V., Mondragón B., I. F., Parra R., C. A., (2020). *Inspección de Aisladores en Líneas de Transmisión Eléctrica usando Inteligencia Artificial*, Revista Colombia de Tecnologías de Avanzada RCTA, ISSN: 1692-7257

SITIOS WEB

Santomera. L. (2017). *El limonar de Santomera*, Santomera, Murcia, España. Recuperado de https://www.el-limonar.com/productor citricos/blog/54/tipos-de-limones-y-como diferenciarlos.html (Consultado: 3 de abril 2021)

Sabelotodo. (2010). *Limón Criollo*, Recuperado de http://www.sabelotodo.org/agricultura/frutales/limon.html (Consultado: 4 de abril 2021)

Ecured. (2020). *Citrus × limonia (Limón mandarina)*, Recuperado de https://www.ecured.cu/Citrus_%C3%97_limonia_(Lim%C3%B3n_mandarina) (Consultado: 10 de abril 2021)

Persian (2019). *El limón Tahití*. Recuperado de https://persianlime.co/Persian/el-limon-tahiti/ (Consultado: 15 de abril 2021)

Xeridia (2020). *La importancia del preprocesamiento de datos en Inteligencia Artificial: Limpieza de datos*. Recuperado de https://www.xeridia.com/blog/la-importancia-del preprocesamiento-de-datos-en-inteligencia-artificial-limpieza-de-datos (Consultado: 16 de abril 2021)

Dmitriev, D. (2019). *VGG16 Neural Network Visualization*. Recuperado de https://www.youtube.com/watch?v=RNnKtNrsrm&t=273s (Consultado: 2 de mayo 2021)

Keras. (2016). *Keras Applications*, Recuperado de https://keras.io/api/applications/ (Consultado: 20 de mayo 2021)

Keras. (2015). *Image data preprocessing*, Recuperado de https://keras.io/api/preprocessing/image/. (Consultado: 26 de mayo 2021)

Brownlee, J. (2019). *How to configure Image Data Augmentation in Keras*. Recuperado de https://machinelearningmastery.com/how-to configure-image-data-augmentation-when-training-deep-learning-neural-networks/. (Consultado: 5 de junio 2021)

Keras. (2020a). *RandomTranslation layer*. Recuperado de https://keras.io/api/layers/preprocessing_layers/image_augmentation/random_translation/. (Consultado: 28 de junio 2021)

Keras. (2020b). *RandomRotation layer*. Recuperado de https://keras.io/api/layers/preprocessing_layers/image_augmentation/random_flip/. (Consultado: 28 de junio 2021)

Keras. (2020c). *RandomFlip layer*. Recuperado de https://keras.io/api/layers/preprocessing_layers/image_augmentation/random_rotation/. (Consultado: 28 de junio 2021)

Keras. (2020d). *RandomZoom layer*. Recuperado de https://keras.io/api/layers/preprocessing_layers/image_augmentation/random_zoom/. (Consultado: 28 de junio 2021)