

Sistema de visión artificial basado en redes neuronales convolucionales para la selección de arándanos según estándares de exportación

Artificial vision system based on convolutional neural networks for the selection of blueberries according to export standards

Recibido: mayo 14 de 2021 | Revisado: julio 21 de 2021 | Aceptado: agosto 20 de 2021

WILY ALDO NARCISO HORNA¹
EDGAR ANDRE MANZANO RAMOS²

RESUMEN

La exportación de arándano es una de las actividades que mayormente ha crecido durante los últimos años en el sector agroindustrial del Perú. Sin embargo, las empresas locales aún presentan dificultades con la automatización de sus procesos, destacando entre ellos la selección de las frutas, la cual de realizarse inadecuadamente puede provocar su rápido deterioro. Es así que, la presente investigación propone un sistema de visión artificial para la detección de los defectos más comunes en los arándanos, de tal forma que puedan ser clasificados para su potencial exportación. Para este cometido, se ha utilizado el modelo de detección Faster R-CNN que se basa en la aplicación de redes neuronales convolucionales y propuesta de regiones. El trabajo ha involucrado la creación de una base de datos propia, el pre-procesamiento de las imágenes adquiridas para su posterior entrenamiento con las redes neuronales, y finalmente la validación de los resultados obtenidos en base a la opinión de un grupo de expertos, cuya información se registró a partir de encuestas. Finalmente, se utilizó el criterio IoU (intersección sobre unión) para poder determinar la precisión de nuestro sistema.

Palabras clave: arándano; redes neuronales convolucionales; visión artificial; Faster-RCNN

ABSTRACT

The export of blueberries is one of the activities that has grown the most in recent years in the agro-industrial sector of Peru. However, local companies still have difficulties with the automation of their processes, highlighting among them the selection of fruits, which if done improperly can cause rapid deterioration. Thus, the present research proposes an artificial vision system for the detection of the most common defects in blueberries, in such a way that they can be classified for their potential export. For this purpose, the Faster R-CNN detection model has been used, which is based on the application of convolutional neural networks and the proposal of regions. The work has involved the creation of its

¹ Departamento de Ingeniería Mecatrónica, Universidad Nacional de Trujillo. Trujillo – Perú. wnarciso@unitru.edu.pe

² Departamento de Ingeniería Mecatrónica, Universidad Nacional de Trujillo. Trujillo – Perú. emanzano@unitru.edu.pe

Autor para correspondencia
E-mail: narciso@unitru.edu.pe

© Los autores. Este artículo es publicado por la Revista Campus de la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres. Este artículo se distribuye en los términos de la Licencia Creative Commons Atribución No-comercial – Compartir-Igual 4.0 Internacional (<https://creativecommons.org/licenses/by/4.0/>), que permite el uso no comercial, distribución y reproducción en cualquier medio siempre que la obra original sea debidamente citada. Para uso comercial contactar a: revistacampus@usmp.pe.

<https://doi.org/10.24265/campus.2021.v26n32.01>

own database, the pre-processing of the acquired images for their subsequent training with neural networks, and finally the validation of the results obtained based on the opinion of a group of experts, whose information was recorded from surveys. Finally, the IoU (intersection over union) criterion was used to determine the accuracy of our system.

Key words: blueberry; artificial neural network; artificial vision; Faster-RCNN

Introducción

Una de las frutas que es, principalmente, exportada por el Perú hacia países como Estados Unidos, China y los Países Bajos es el arándano. Al 2019, el Perú posee más de 12000 hectáreas ya cultivadas, las cuales se distribuyen en seis de sus regiones: Lambayeque, La Libertad, Arequipa, Cajamarca, Ancash y Lima. La Libertad es la región que abarca más del 60% de la producción nacional.

El mismo año, el Perú se posicionó por primera vez como el principal exportador de arándanos a nivel mundial, superando a otros países exportadores como Chile y México, generando ganancias de hasta 810 millones de dólares. Así también, la creciente importación de arándanos por parte de los países inicialmente mencionados, especialmente, China, ha elevado los estándares de calidad y velocidad de producción requeridos por parte de los países productores (Rivas, 2018; Rojas, 2020).

Es así que empresas locales, especialmente, de la región La Libertad han visto la necesidad de optimizar sus procesos, incluyendo sistemas automatizados que trabajan en base a sistemas de visión artificial para el proceso de selección de arándanos. Esto también se debe a que una selección manual conlleva a la pérdida de Bloom, cubierta del fruto que previene su descomposición. Procesos de gran volumen

(mayores a 1 T/h), generalmente, requieren la utilización de máquinas de las marcas TOMRA y WECO, mientras que otros de menor escala tienen como alternativa marcas como CNTAIZI o UNITEC. Sin embargo, la utilización de las máquinas mencionadas conlleva una inversión muy elevada tanto en la adquisición como en los gastos de importación (Rudas, 2015; Rojas, 2020).

Debido a lo anteriormente expuesto, se encuentra la necesidad del desarrollo de visión artificial que pueda cumplir con los estándares de selección de arándanos, evitando de esta manera su manipulación excesiva; y así también, ser potencialmente utilizado en un sistema automatizado de bajo costo.

La utilización de sistemas de visión artificial en la selección y clasificación de frutas se da desde hace varios años, incluso la utilización de la inteligencia artificial. Por ejemplo, Zhang, Wang, Ji y Phillips (2014) proponen en su trabajo un método de clasificación híbrido basado en el algoritmo de colonia de abejas caóticas artificiales a escala de aptitud (FSABC) para el entrenamiento de su red neuronal de alimentación (FNN), con el propósito de clasificar frutas en base a sus características de textura.

Un par de años más tarde, ya se trataban de investigaciones en esta área utilizando las redes neuronales convolucionales

(CNN). Yalcin y Razavi (2016) utilizan esta arquitectura para la clasificación de diferentes especies vegetales. Así también, Yu, Mao y Wang (2016) proponen un método para el reconocimiento de categorías de alimentos, utilizan las CNN, además de otras técnicas del Deep Learning.

Un trabajo que podríamos considerar más cercano es el desarrollado por Reátegui y Velazco (2018) que desarrollan una aplicación informática para el reconocimiento de camu camu, utilizando también redes neuronales convolucionales para este cometido.

No obstante, uno de los modelos de detección de objetos más contemporáneo basados en las CNN es el Faster R-CNN. Mai, Zhang, Jia y Meng (2018) evalúan las primeras versiones de esta técnica con el objetivo de detectar frutas de pequeño tamaño en diferentes fotografías.

Por su parte, Basri, Syarif y Sukaridhoto (2018) en su investigación proponen

un método de aprendizaje profundo mediante el uso del modelo Faster R-CNN para clasificar varias frutas, enfocándose en el mango y la pitaya, utilizando la plataforma Tensorflow para esta aplicación, llegando a la conclusión que se trata de una técnica apropiada para sistemas en tiempo real. Dentro de sus parámetros de precisión, haciendo uso de su aplicación, obtienen un rango del 64.4% al 70.6%, el cual varía de acuerdo a los parámetros considerados.

De la misma manera, Wan y Goudos (2020) destacan la eficacia de este método, especialmente, para el desarrollo de sistemas de detección de frutas en tiempo real, puesto que ellos utilizan la técnica mencionada como parte del desarrollo de un sistema robótico de visión de detección de diferentes tipos de fruta.

Es así que se concluye que, el modelo Faster R-CNN es idóneo para la detección y clasificación de frutas en sistemas en tiempo real.

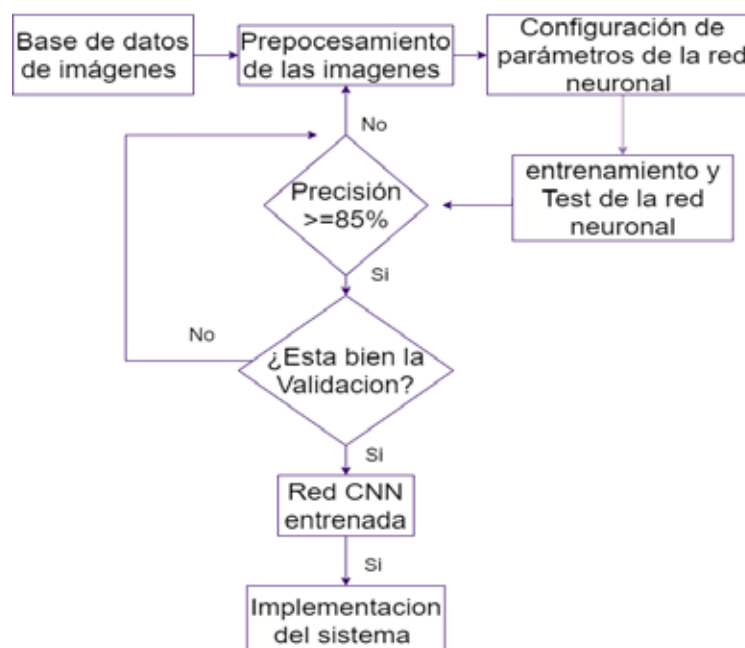


Figura 1. Metodología de trabajo

Método

Para el desarrollo de nuestro sistema de visión artificial se ha considerado la estructura de trabajo indicada por la Figura 1 que señala cada una de las etapas incluidas, las cuáles se detallan a continuación.

El sistema de visión artificial tiene la siguiente estructura de desarrollo como se aprecia en la Figura 1.

A. Base de datos de imágenes

Para el presente trabajo se crea una base de datos propia y se realiza la toma de fotos de diferentes calidades de arándanos que fueron obtenidas de un mercado local con la finalidad de tener disponibilidad de los diversos potenciales defectos que pudiese tener esta fruta que es nuestro objeto principal de estudio.

Es así que se logró recaudar un total de 820 imágenes, de las cuales se utilizaron 500 que corresponden a los cinco defectos más comunes que se presentan en las empresas agroindustriales de la región La Libertad como son: corola, desgarrado, picado de pájaro, pedicelo y rojo. La Figura 2 da un esbozo de la clasificación considerada.

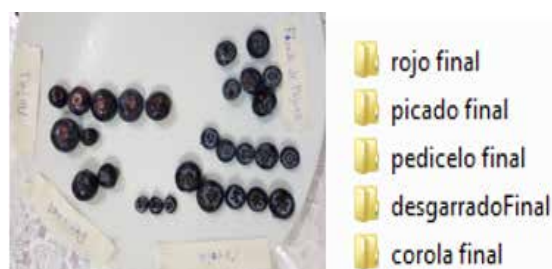


Figura 2. Base de datos construida

Claro está, también se realizó el descarte de diferentes fotografías que no eran adecuadas para el entrenamiento.

B. Preprocesamiento de las imágenes

Después de haber recaudado cientos de imágenes de arándano, se encontró que estas tenían tamaños y pesos por encima de 4032 x 2268 p y 6.4 MB. Estas imágenes necesitan de un preprocesamiento con el objetivo de disminuir su peso y tamaño para que el entrenamiento no sea tan complejo. Lo recomendado era tener imágenes con las siguientes condiciones: 1) resolución menor a 1280x 720 p y 2) peso menor o igual a 150kb; con estos valores establecidos se procedió con la etapa de preprocesamiento haciendo uso de la librería Image Processing Toolbox del software MatLab, con la cual también se pudo recortar algunas imágenes ya que fue mucho el espacio dejado alrededor del arándano como se muestra en la Figura 3.

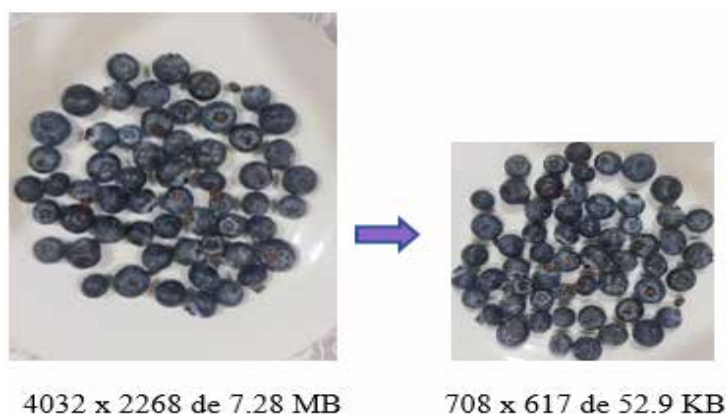


Figura 3 . Reducción del peso y tamaño de las imágenes.



Figura 4. Etiquetado de los defectos en las imágenes.

C. Preparación de los datos de entrada

Para el entrenamiento de nuestro sistema, se necesita que las imágenes se encuentren etiquetadas. Por tanto, con las imágenes ya preprocesadas y clasificadas para el entrenamiento y test, nos disponemos a etiquetar las imágenes con la ayuda del software LabeledImg. Para ser más exactos, procedemos a etiquetar los defectos que presenta cada arándano, defectos mencionados anteriormente y que fueron indicados por expertos a partir de las encuestas realizadas.

Se crea un archivo *.xml con las coordenadas correspondientes a cada defecto localizado en cada imagen. La Figura 4 muestra una imagen ya etiquetada.

Una vez realizado el etiquetado para todas las imágenes y generado los archivos *.xml, se necesitan que estos estén contenidos en un solo archivo para lo cual lo convertimos a uno en formato *.csv el cual viene a ser una hoja de cálculo en la que se encuentran listadas todas las imágenes y coordenadas, tanto para las imágenes del entrenamiento como las de test.

Por último, convertimos estos archivos *.csv de entrenamiento y test a uno en formato TFrecord (formato simple para almacenar una secuencia de registros binarios.). El proceso de conversión de las imágenes a archivos *.csv y TFrecord, lo podemos resumir en la Figura 5.



Figura 5. Proceso de conversión de las imágenes a un registro de archivo

D. Configuración de los parámetros de la red neuronal

Para la configuración de la red hemos considerado un modelo pre-entrenado de detección Faster RCNN con 204 neuronas con 13 capas convoluciones, 13

capas rectificadas (ReLU) y cuatro (04) capas de agrupación.

Para comenzar con el procedimiento mencionado se ha de crear un mapa de etiquetas en un archivo de extensión *.pbtxt, el cuál luego vendrá a ser un archivo gráfico para la plataforma TensorFlow.

La Figura 6 detalla las cinco clases de defectos considerados por el mapa de etiquetas creado.

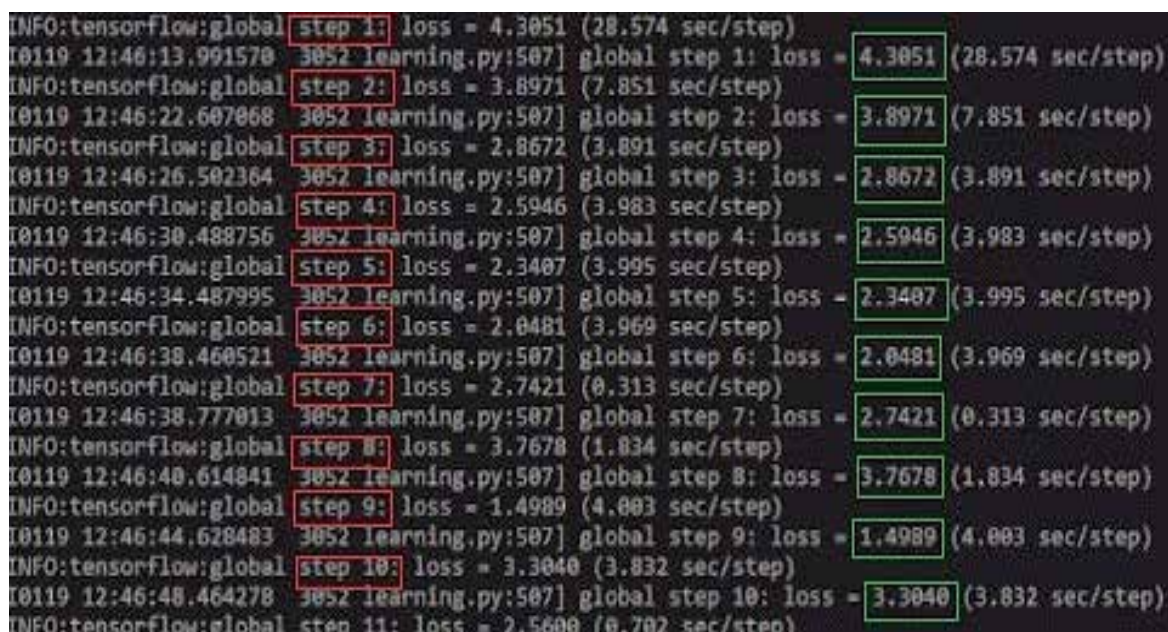
```
item {
  id: 1
  name: 'rojo'
  id: 2
  name: 'pedicelo'
  id: 3
  name: 'picado de pajaro'
  id: 4
  name: 'corola'
  id: 5
  name: 'desgarrado'
}
```

Figura 6. Creación del mapa de etiquetas para la clase de arándanos

A continuación, se procede con los siguientes pasos:

1. Especificación del modelo de detección.
2. Obtención de parámetros del modelo a partir del archivo *.ckpt generado.
3. Ingreso del número de imágenes a considerar para el Test (20% de la cantidad total de imágenes de la base de datos).
4. Direccionamiento de los archivos record del test y mapa de etiquetas para que se utilicen como entradas de la evaluación del modelo.

De la misma manera, se configuran los números de pasos, el tamaño de matriz de maxPooling, el máximo número de clases, el número máximo de detecciones por imagen, número de propuestas de región, umbrales de puntaje e intersección de unión (IoU), entre otros.



Step	Loss	Time (sec/step)
1	4.3051	28.574
2	3.8971	7.851
3	2.8672	3.891
4	2.5946	3.983
5	2.3407	3.995
6	2.0481	3.969
7	2.7421	0.313
8	3.7678	1.834
9	1.4989	4.003
10	3.3040	3.832
11	2.5600	0.702

Figura 7. Pérdidas en el inicio del entrenamiento

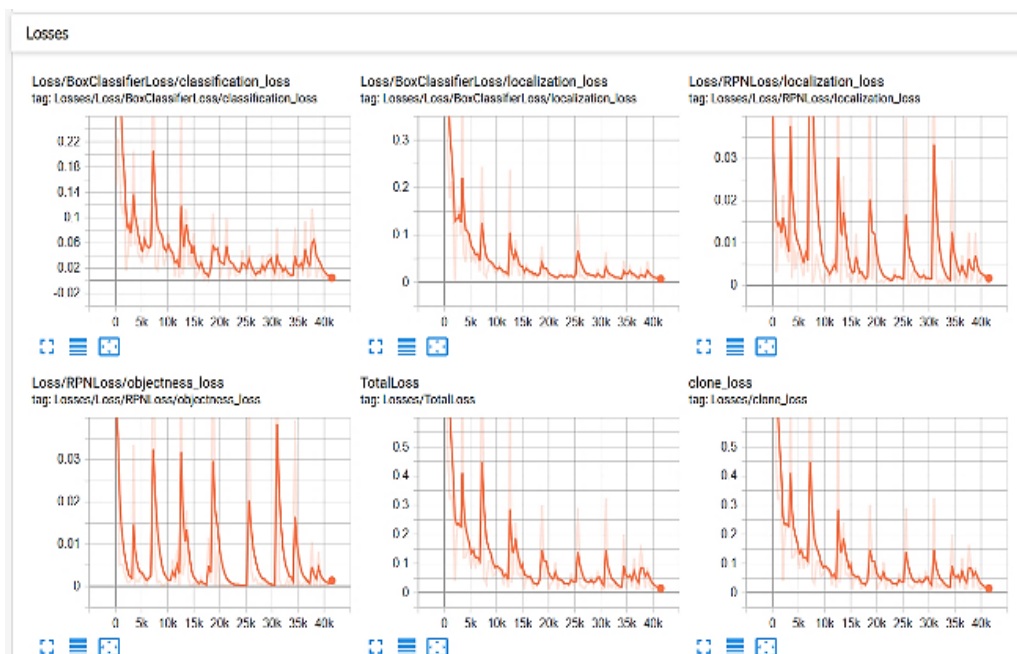


Figura 8. Gráfica de pérdidas durante el entrenamiento

```

INFO:tensorflow:global step 41486: loss = 0.0231 (0.381 sec/step)
I0119 16:23:48.386552 3052 learning.py:507] global step 41486: loss = 0.0231 (0.381 sec/step)
INFO:tensorflow:global step 41487: loss = 0.0409 (0.286 sec/step)
I0119 16:23:48.674583 3052 learning.py:507] global step 41487: loss = 0.0409 (0.286 sec/step)
INFO:tensorflow:global step 41488: loss = 0.0421 (0.277 sec/step)
I0119 16:23:48.954834 3052 learning.py:507] global step 41488: loss = 0.0421 (0.277 sec/step)
INFO:tensorflow:global step 41489: loss = 0.0365 (0.281 sec/step)
I0119 16:23:49.238272 3052 learning.py:507] global step 41489: loss = 0.0365 (0.281 sec/step)
INFO:tensorflow:global step 41490: loss = 0.0562 (0.282 sec/step)
I0119 16:23:49.522480 3052 learning.py:507] global step 41490: loss = 0.0562 (0.282 sec/step)
INFO:tensorflow:global step 41491: loss = 0.1941 (0.286 sec/step)
I0119 16:23:49.810740 3052 learning.py:507] global step 41491: loss = 0.1941 (0.286 sec/step)
INFO:tensorflow:global step 41492: loss = 0.0234 (0.287 sec/step)
I0119 16:23:50.100955 3052 learning.py:507] global step 41492: loss = 0.0234 (0.287 sec/step)
INFO:tensorflow:global step 41493: loss = 0.0171 (0.275 sec/step)
I0119 16:23:50.379191 3052 learning.py:507] global step 41493: loss = 0.0171 (0.275 sec/step)
INFO:tensorflow:global step 41494: loss = 0.0124 (0.279 sec/step)
I0119 16:23:50.661435 3052 learning.py:507] global step 41494: loss = 0.0124 (0.279 sec/step)
INFO:tensorflow:global step 41495: loss = 0.0143 (0.267 sec/step)
I0119 16:23:50.929717 3052 learning.py:507] global step 41495: loss = 0.0143 (0.267 sec/step)
INFO:tensorflow:global step 41496: loss = 0.0318 (0.278 sec/step)
I0119 16:23:51.210964 3052 learning.py:507] global step 41496: loss = 0.0318 (0.278 sec/step)
Traceback (most recent call last):

```

Figura 9. Valores de pérdidas para 41400 pasos

E. Entrenamiento

Una vez configurados los parámetros del modelo, se procede con el entrenamiento del sistema. La Figura 7 muestra como en el inicio del proceso se tienen pérdidas, desviaciones entre las predicciones realizadas y los datos reales, con valores relativamente altos.

Así también, la herramienta Tensorboard nos permite monitorear en tiempo real las pérdidas de manera clasificada. Las gráficas correspondientes a este monitoreo se detallan en la Figura 8.

Finalmente, tal como se puede verificar en las gráficas de la Figura 8, así como en los valores encontrados para la etapa final de

entrenamiento (Figura 9), las pérdidas llegan a valores aceptables. Para ello, se realizaron 41 400 pasos en un entrenamiento que duró aproximadamente cinco horas.

Resultados y Discusión

A. Test individuales

Primero, a partir del archivo de imágenes, se realizan pruebas para fotografías que contienen claramente

los Pdefectos de los arándanos que se desean detectar (corola, rojo, desgarrado, cicatriz y pedicelo), esto con la finalidad de verificar a primera mano la eficacia del sistema desarrollado.

Cabe señalar, que las imágenes consideradas para todo el proceso de evaluación (Test) no son las mismas que las utilizadas para el proceso de entrenamiento y validación de nuestra red neuronal.



Figura 10 . Prueba para arándanos con corola

a.1. Corola

De 24 arándanos con el defecto de corola visible, el sistema logró detectar

11 correctamente, tal como se muestra en la Figura 10. Concluyendo que el porcentaje de detección para este defecto está por encima del 45 %.



Figura 11. Prueba para arándanos rojo

a.2. Rojo

Así también, de 32 arándanos rojos o inmaduros, el sistema logró detectar ocho

correctamente. Esto se puede detallar en la Figura 11. Se concluye una detección de arándanos rojos del 25 %.

Tabla 1

Porcentaje de detección

Característica	Rojo	Cicatriz	Pedicelo	Desgarrado	Corola
Porcentaje	25 %	48%	42%	64.%	45%



Figura 12. Prueba para arándanos desgarrados

a.3. Desgarrado

La imagen considerada para la evaluación de arándanos desgarrados se muestra en la Figura 12. Se tiene una detección mayor al 64 % para este defecto en particular.

a.4. Cicatriz (picado de pájaro)

Por su parte, la detección para el defecto de cicatriz o picado de pájaro logró alcanzar una detección mayor al 48%. La Figura 13 muestra un ejemplo de la detección obtenida.



Figura 13. Prueba para arándanos con cicatriz



Figura 14. Prueba para arándanos con pedicelo

a.5. Pedicelo

La Figura 14 muestra cómo se realizó la detección del defecto pedicelo. La evaluación de éxito del sistema obtuvo un porcentaje superior al 42 %.

Los resultados obtenidos para cada uno de los defectos considerados se resumen en la Tabla 1.

B. Test grupales:

Así también, se procedió a realizar pruebas de precisión del sistema de visión artificial para la detección de todos los defectos de arándanos considerados, los cuales fueron agrupados de forma indistinta en diferentes imágenes.



Figura 15. Ejemplo de prueba grupal con detecciones correcta



Figura 16. Ejemplo de prueba grupal con falsas detecciones

Luego de realizar las pruebas grupales correspondientes, se obtuvieron detecciones como la que se muestra en la Figura 15, donde aparecen las etiquetas para cada defecto encontrado.

Por otro lado, también se encontraron falsas detecciones por parte del sistema. Un ejemplo claro se muestra en la Figura 16, donde se han enmarcado en rojo en la imagen de la izquierda aquellos defectos que fueron etiquetados con un defecto que no corresponde.

Tabla 2

Precisión en cada defecto

Característica	Rojo	Cicatriz	Pedicelo	Desgarrado	Corola
Precisión	36%	24%	26%	40.5%	39.8%

```

Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.360
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.268
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.240
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.398
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.405
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.447
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.505
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.512
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.400
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.525
  
```

Figura 17. Porcentaje de precisión del sistema

C. Precisión

Para la evaluación de la precisión de nuestro sistema se utilizó la medida de evaluación IoU (intersección sobre la unión), la cual es, generalmente, utilizada para evaluar este parámetro (precisión) en sistemas de detección de objetos. Esto se puede evidenciar en la Figura 17, donde cada uno de los cinco defectos considerados se evaluó en una fila, teniendo como resultado final de precisión el valor que se encuentra a la derecha. Estos datos se resumieron para una mejor visualización en la Tabla 2.

Tomando en consideración el trabajo realizado por Basri et al. (2018), donde se alcanzaron precisiones del 64.4% al 70.6%, nuestro sistema muestra un intervalo con valores más pequeños. Este vendría ser del 24% al 40.5%, tal como se puede concluir a partir de la Tabla 2.

Realizando un análisis de todo el proceso realizado, se podría concluir que la principal causa de los resultados obtenidos es la cantidad de imágenes utilizadas en el proceso de entrenamiento. Así también, se vienen realizando las pruebas

experimentales de nuestro sistema para nuevos valores de los hiperparámetros considerados inicialmente.

Un dato adicional es que el tiempo de procesamiento y detección para cada imagen es de aproximadamente cinco segundos.

Conclusiones

En el presente trabajo se diseñó un sistema de visión artificial por medio de redes neuronales convolucionales (CNN) aplicando el modelo Faster R-CNN con el propósito de desarrollar una alternativa para la automatización del proceso de selección de arándano para su correspondiente exportación.

Se ha realizado una evaluación de expertos por medio de encuestas, en la cual se ha determinado los defectos más comunes en arándanos además de las variedades más cultivadas en las empresas agroindustriales liberteñas de este sector.

El entrenamiento demandó un tiempo aproximado de cinco horas que en total involucró 41 400 pasos. Se obtuvo un

rango de precisiones del 24% al 40.5% para los cinco defectos de arándano considerados.

Se concluye que los resultados

pueden ser mejorados mediante el enriquecimiento de nuestra base de datos que será utilizada por nuestro sistema tanto para el entrenamiento como la validación.

Referencias

- Basri, H., Syarif, I., & Sukaridhoto, S. (2018, October). Faster R-CNN implementation method for multi-fruit detection using tensorflow platform. In 2018 international electronics symposium on knowledge creation and intelligent computing (IES-KCIC) (pp. 337-340). IEEE.
- Mai, X., Zhang, H., Jia, X., & Meng, M. Q. H. (2020). Faster R-CNN with classifier fusion for automatic detection of small fruits. *IEEE Transactions on Automation Science and Engineering*, 17(3), 1555-1569
- Reátegui Pezo, A., & Velasco Meléndez, M. A. (2018). Aplicación informática para reconocimiento de la especie camu camu (*Myrciaria Dubia*) a través de redes neuronales convolucionales, en Iquitos Perú, durante el año 2017. [Tesis de Maestría]. Universidad Nacional de la Amazonía Peruana.
- Rivas, P. (2018). Blueberries Consulting. <https://blueberriesconsulting.com/la-industria-del-arandano-en-sudafrica-busca-competir-con-productores-latinoamericanos/>
- Rojas, N. (2020). Arándanos Perú. <https://arandanosperu.pe/2020/05/05/superficie-destinada-al-cultivo-de-arandanos-podria-llegar-a-14000-hectareas-en-peru/#:~:text=Actualmente%20las%20%C3%A1reas%20de%20ar%C3%A1ndano,ejecutivo%20de%20Cultivada%20Carlos%20Rodr%C3%ADguez>
- Rudas Ramírez, C. A. (2016). Diseño de un sistema mecatrónico para la selección de arándanos. [Tesis de Maestría]. Pontificia Universidad Católica del Perú.
- Wan, S., & Goudos, S. (2020). Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks*, 168, 107036.
- Yalcin, H., & Razavi, S. (2016, July). Plant classification using convolutional neural networks. In 2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics) (pp. 1-5). IEEE.
- Yu, Q., Mao, D., & Wang, J. (2016). Deep learning-based food recognition. Technical report, Stanford University.
- Zhang, Y., Wang, S., Ji, G., & Phillips, P. (2014). Fruit classification using computer vision and feedforward neural network. *Journal of Food Engineering*, 143, 167-177.