

DeepSeed: aplicación multiplataforma para estimar la calidad de granos de maíz

Máximo Librandi¹, Joshua Corino¹, Paula Tristan¹, Laura Felice¹,

¹Instituto de Investigación en Tecnología Avanzada – INTIA - Facultad de Ciencias Exactas.

Universidad Nacional del Centro de la Pcia. de Buenos Aires.

{maximolibrandi, joshuc98}@gmail.com, {ptristan,
lfelice}@exa.unicen.edu.ar

Abstract. En los últimos años, la implementación de nuevas tecnologías en la producción agropecuaria ha permitido un importante salto en las magnitudes producidas. Sin embargo, la determinación de calidad de los granos como requisito para establecer su precio aún no ha generado ningún cambio o innovación durante su proceso. Luego de décadas, la tarea de clasificación y estimación del grado de calidad comercial continúa realizándose de forma manual por los peritos clasificadores de granos. En este artículo se presenta DeepSeed, una aplicación multiplataforma que, utilizando el modelo de Deep Learning FasterRCNN Resnet152 COCO provisto por TensorFlow, determina el grado de comercialización del maíz, a través del procesamiento de la imagen de una pequeña muestra.

Keywords: Clasificación, Calidad de granos, Inteligencia Artificial, Deep Learning, Faster RCNN, Progressive Web App.

1 Introducción

En las últimas dos décadas el sector agrícola argentino registró grandes transformaciones; cambios en las formas organizacionales, en las técnicas productivas y en la tecnología aplicada que dieron lugar a un salto de gran magnitud en las cantidades producidas.

El contexto de trabajo de DeepSeed está centrado en una etapa de la cadena agroexportadora donde la evolución tecnológica aún no ha introducido cambios: la determinación de la calidad de los granos. Se entiende por calidad de un cereal al conjunto de defectos que desmejoran una partida. Estos defectos son los factores que se tienen en cuenta para determinar la calidad de una partida en función de la cantidad o la intensidad que los mismos se encuentran presentes en una muestra.

Las metodologías de clasificación y medición de la calidad de los granos no han cambiado desde sus orígenes, ya que aún continúan realizándose de forma manual por los Peritos Clasificadores de Granos. En este contexto, la efectividad depende en gran medida de la capacidad y experiencia que tenga el perito, así como de otros factores externos como estrés o cansancio. De este modo, la automatización de este proceso

impondría numerosos beneficios a lo largo de la cadena de comercialización, permitiendo ofrecer una herramienta de soporte y asistencia a la tarea de los peritos. Además, brindaría asistencia a los productores agropecuarios durante el proceso de recolección, permitiendo aplicar ajustes en la maquinaria en caso de ser necesario, o realizar una adecuada segmentación de su cosecha previa a su almacenamiento.

La Inteligencia Artificial (IA) [15], se ha afianzado y constituye hoy la gran tendencia en desarrollo de software y tecnología. El Deep Learning (DL) [3], [5], [8] es sin duda el área de investigación más popular dentro del campo de la IA. El área de Computer Vision [1] se ha vuelto algo mucho más fácil e intuitivo gracias a los recientes avances en DL, el cual ha revolucionado el reconocimiento de patrones. Este reconocimiento se hace a través de las redes neuronales convolucionales (CNNs) [11], una técnica de DL que utiliza filtros con el fin de extraer características de las imágenes y aprenderlas, para luego poder realizar tareas como detección de objetos o clasificación de imágenes. En el uso de nuevas tecnologías en el desarrollo de aplicaciones multiplataforma, cabe mencionar las Progressive Web Apps (PWA) [23], aplicaciones web que proporcionan un producto instalable y una experiencia de aplicación en computadoras de escritorio y dispositivos móviles, que se crean y entregan directamente a través de la web.

Se presenta aquí un resumen del desarrollo de una aplicación multiplataforma (DeepSeed) que, utilizando técnicas de DL, permite detectar y clasificar los objetos presentes en una foto de una muestra de granos de maíz y así determinar su calidad de acuerdo a las normas vigentes.

Este documento está organizado de la siguiente forma: en la sección 2 se introducen los trabajos más importantes relacionados con la determinación de calidad de diversos cultivos. En la sección 3 se detalla cada etapa de la aplicación propuesta. La sección 4 presenta las métricas de performance analizadas sobre el modelo de DL y analiza los resultados obtenidos por DeepSeed en cuatro casos de estudio. En la sección 5 se describen las conclusiones arribadas y los trabajos futuros que surgen del análisis del trabajo finalizado. Finalmente, en la sección 6, se encuentran las referencias bibliográficas.

2 Estado del arte

En esta sección se resume el trabajo de algunos proyectos de investigación académica y a nivel empresarial de la clasificación de maíz y de otros cultivos, permitiendo entender la evolución de las diferentes investigaciones y desarrollos.

El trabajo de Saleres (2018) [2], considerado antecesor principal de DeepSeed, se basa en una aplicación web que permite determinar el grado de calidad de una muestra de granos de maíz a partir de una imagen de la misma, utilizando técnicas de procesamiento de imágenes.

En cuanto a los trabajos relacionados con el maíz, se destacan el trabajo de Bhurtel et al. (2019) [10] donde se implementa un sistema de clasificación de calidad de un lote de semillas de maíz utilizando técnicas de DL. A partir de imágenes que incluyen semillas en buen estado, semillas dañadas y materia extraña, el sistema categoriza la

calidad del lote como excelente, buena, promedio, mala y pésima. En la propuesta de Huang et al. (2019) [13] se comparan las técnicas de Convolutional Neural Networks (CNNs) y Transfer Learning con los algoritmos tradicionales de Machine Learning en la clasificación de semillas de maíz.

En el tratamiento de otras semillas, se puede mencionar a Riat et al. (2018) [19] creadores de CamWheat: una tecnología que permite conocer la calidad del trigo en 8 segundos, también basada en IA. Suseendran et al. (2020) [7] aplican las técnicas Multi-Layer Perceptron (MLP) Neural Network y Neuro-Fuzzy Classifier para la clasificación entre trigo, arroz y maíz.

Por otro lado, ZoomAgri [24] es una empresa nacional que desarrolla tecnología de determinación de calidad de commodities agrícolas, por medio de Procesamiento de Imágenes, Inteligencia Artificial e Internet de las Cosas (IoT). Su primera gran disrupción es ZoomBarley, un escáner que permite determinar la variedad de cebada en menos de 5 minutos y a una fracción del costo de los métodos actuales.

3 Metodología aplicada

La Figura 1 resume la metodología empleada para obtener la aplicación multiplataforma objetivo. Básicamente, a partir de una imagen de una muestra de maíz, se determina el grado de calidad comercial de la misma, utilizando un modelo de DL entrenado y técnicas de procesamiento de imágenes. Seguidamente se describen las partes correspondientes al modelo diseñado.

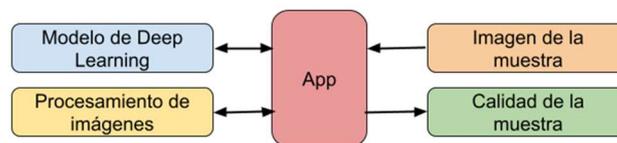


Fig. 1. Metodología de trabajo propuesta.

3.1 Modelo de Deep Learning

La tarea de construcción del modelo de DL, sin dudas la más importante de esta propuesta, consta de la ejecución de varias tareas que se detallan a continuación.

3.1.1 Generación del dataset

En primer lugar, y con el fin de reentrenar la red neuronal convolucional, es necesario crear el dataset. Para esto, basados en el caso de aplicación, los peritos de la Cámara Arbitral de la Bolsa de Cereales de Buenos Aires separaron y etiquetaron objetos dentro de las distintas clases a tener en cuenta. Entre estas clases se encuentran: granos de maíz (semillas en buen estado), granos dañados, granos quebrados, materia extraña, semillas de chamico y granos picados.

Posteriormente se capturaron numerosas imágenes variando las clases y cantidades de objetos, concluyendo en un total de 909 imágenes de 4096 x 3024 píxeles. Debido a que se trabaja con un algoritmo de detección de objetos supervisado, resulta necesario etiquetar los objetos presentes en cada imagen, entregando al modelo no solo la imagen, sino también los cuadros delimitadores y la clase de cada objeto que figura en la misma. Para cumplir con esta tarea se utilizó LabelImg [22], que contabilizó un total de 4948 etiquetas en el dataset.

Finalmente, el dataset fue dividido aleatoriamente, tomando el 80% de las imágenes para entrenamiento, y el 20% restante para evaluación. Como resultado de esta división, 3898 objetos fueron utilizados para entrenamiento, mientras que los otros 1050 para evaluación.

3.1.2 Entrenamiento del modelo

Una vez generado el dataset adecuado, el siguiente paso es el entrenamiento del modelo. Esta compleja tarea requiere de la ejecución iterativa de varias acciones. En particular, la evaluación y validación resultan necesarias para asegurar la correctitud del mismo. En la Figura 2 se puede visualizar el pipeline propuesto para obtener un modelo de DL capaz de detectar y clasificar los objetos presentes en una imagen de una muestra de maíz. A continuación, se describe detalladamente cada paso del entrenamiento y las decisiones tomadas en cada uno.

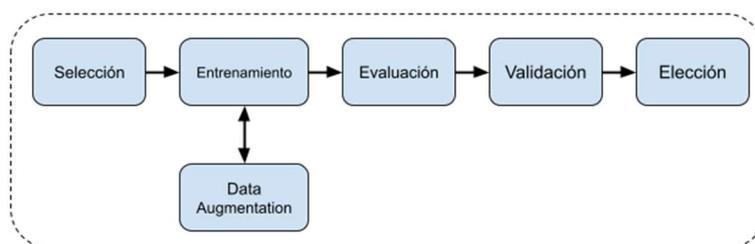


Fig. 2. Pasos para el entrenamiento del modelo de Deep Learning.

Selección del modelo

En principio, es necesario seleccionar el modelo de DL al que se le aplicará Transfer Learning entre los modelos que provee la API de detección de objetos de TensorFlow [18].

En primer lugar, se estudió la posibilidad de emplear un modelo basado en el algoritmo Single Shot Detector (SSD) [17]. Sin embargo, luego de tiempo de investigación y debido a las restricciones de tamaño de imágenes del SSD, lo cual lo desfavorecen en la detección de pequeños objetos, la siguiente alternativa fue el algoritmo Faster RCNN [14], el cual ha probado ser uno de los más eficientes en la familia de los RCNN para la detección de objetos en la actualidad [12].

Se optó por el modelo Faster RCNN ResNet 152 800x1333 COCO. Dicho modelo posee como extractor de características una ResNet 152 [9], una red residual de 152 capas. Otra característica principal, tal como su nombre lo especifica, es que el

modelo define 800 píxeles como valor mínimo y 1333 como máximo para la redimensión de las imágenes durante su aprendizaje, manteniendo la relación de aspecto. TensorFlow ofrece este modelo pre-entrenado con el dataset de COCO [16], sobre el cual se aplicará la técnica de Transfer Learning para ser utilizado en un set de datos más específico: el de granos de maíz.

Si bien el modelo elegido no puede ser deployado en una aplicación móvil, requiere una conexión a Internet mínima para su funcionamiento a través de una API y demanda mayor tiempo y recursos para su entrenamiento, brinda mayor precisión en la detección de objetos pequeños al aceptar imágenes de mayor tamaño.

Entrenamiento del modelo

Con el modelo adecuado elegido, la siguiente tarea es el reentrenamiento del mismo para adaptarlo al caso de aplicación. El reentrenamiento se realizó utilizando Google Colab [21], para lo cual fue necesario la configuración de parámetros, como por ejemplo, el número de clases a detectar, el tamaño del dataset, entre otros.

Durante este proceso fue necesario aplicar técnicas de Data Augmentation, para ampliar el tamaño del dataset realizando una serie de cambios aleatorios en las imágenes. En este trabajo se aplicaron operaciones de flip y rotación, sin aplicar operaciones de transformación del color, brillo u otros factores que reducen la sensibilidad del modelo al color, ya que fueron probadas causando que el modelo confunda las principales clases asociadas al maíz.

Para poder evaluar el entrenamiento del modelo, se generan archivos de control o checkpoint cada 5 mil steps, a partir de los cuales se construye el grafo de inferencia utilizado luego para la validación del modelo.

Evaluación del modelo

La evaluación aporta una noción de cómo el modelo está aumentando su capacidad de generalización. Esta tarea fue automatizada modificando la versión original provista por TensorFlow para que se ejecute cada vez que el entrenamiento genere un nuevo checkpoint. Como resultado de este proceso se obtuvieron las métricas que se listan a continuación:

- RPN Localization Loss: significa la pérdida de localización de los cuadros delimitadores generados por la red de propuesta de región (RPN o Region Proposal Network) que incluye la arquitectura Faster RCNN.
- RPN Objectness Loss: es la pérdida del clasificador de la RPN que determina si un cuadro propuesto es un objeto de interés o forma parte del fondo de la imagen.
- Box Localization Loss: representa el error de los recuadros de los objetos identificados, es decir, de las coordenadas sugeridas para cada objeto.
- Box Classification Loss: determina el error existente en la clasificación de los objetos detectados, donde se define a qué clase pertenece cada objeto.
- Total Loss: es la suma total de las cuatro pérdidas definidas anteriormente.

En la sección 4 -Resultados- se muestran los gráficos de la evolución de dichas métricas a lo largo del entrenamiento.

Validación del modelo

Durante el proceso de validación del modelo se procesan imágenes no utilizadas para el entrenamiento y para la evaluación, para así poder observar su comportamiento a través de métricas de performance y en base a ello tomar decisiones de implementación. Si se utilizan métricas más significativas sobre un conjunto de datos acotado, es posible seleccionar el modelo que brinda los mejores resultados en la generalización. Este proceso también se realizó automáticamente cada vez que se generaba un nuevo checkpoint durante el entrenamiento, pero esta vez sobre imágenes similares a las que la aplicación procesará una vez deployada para su uso (es decir, imágenes con más de 100 granos). De este modo, fueron tomadas 27 nuevas imágenes que contenían un total de 3119 objetos.

Cada nuevo checkpoint significa un nuevo posible modelo que debía ser validado, lo que requiere que se realice la detección de objetos en cada una de las 27 imágenes destinadas a la validación. Se utilizaron tres alternativas para llevar a cabo la detección de objetos: una con la imagen en tamaño original; la segunda, basada en [6], en donde la imagen es dividida en cuatro subimágenes a las cuales se le aplica la detección, y los resultados son unidos eliminando los que poseen un alto grado de superposición (IoU); y la última, una combinación de las dos versiones anteriores que obtiene los resultados con mayor confiabilidad. Por último, se compararon los resultados obtenidos utilizando una matriz de confusión para cada alternativa de detección, en donde se tienen filas y columnas por cada clase que detecta el modelo.

Elección del modelo

Se seleccionó el modelo a incorporar en la versión final de la aplicación utilizando la métrica conocida como Macro F1 Score [4], la cual permite evaluar la performance de un algoritmo de clasificación multiclase utilizando la aproximación “one vs all”. En esta aproximación, los valores de una clase particular son seleccionados y convertidos en ejemplos positivos, mientras que el resto de las clases, en negativos. Este proceso se repite para cada clase detectada por el modelo y se obtienen las métricas correspondientes a cada una de ellas.

De este modo, utilizando los conceptos de verdaderos positivos, falsos positivos y falsos negativos, se calcularon las métricas Precision, Recall y F1 Score para cada clase. Posteriormente, con el objetivo de comparar la performance de los modelos entre sí, se combinaron los F1 Scores en un solo número: el F1 Score general del modelo. La denominada Macro F1 Score se calcula como la media aritmética simple de los F1 Scores por clase. En la sección 4 -Resultados- se ilustra la evolución de dicha métrica durante el entrenamiento.

El modelo correspondiente al checkpoint 315.000 fue el seleccionado para incluir en la aplicación resultante. A pesar de que por una mínima diferencia no fue el que logró el mayor valor para la métrica Macro F1 Score, se eligió ya que consiguió el valor más alto de F1 Score para la clase granos dañados.

3.2 La Aplicación

Se desarrolló una Progressive Web App (PWA) que, a partir de la imagen de la muestra de maíz que se desea analizar, determina su grado de calidad. Esta app se implementó como una aplicación web, utilizando HTML, CSS y JavaScript, y se le añadieron los componentes necesarios para convertirla en una PWA.

La API Rest desarrollada es una aplicación Python en donde se emplea Flask [20] para definir los endpoints necesarios. La API, la cual ya tiene el modelo de DL precargado con el objetivo de optimizar el tiempo de respuesta, una vez que recibe la imagen, realiza la detección de los objetos, obteniendo las coordenadas de los cuadros delimitadores, las clases a las que pertenecen y las confianzas con las que fueron detectados. Luego, utilizando procesamiento de imágenes, más precisamente la técnica de binarización, se aproxima el valor del peso hectolítrico (PH) y se calculan los porcentajes de superficie que ocupa cada rubro de calidad, utilizando la hoja A4 como objeto de referencia. Con estos datos, la API determina el grado de calidad de la muestra y retorna los resultados a la PWA en formato JSON.

La PWA está hosteada en los servidores de Firebase de Google, mientras que la API Rest desarrollada corre en sistemas locales. Si se deseara escalar la aplicación, solo sería necesario subir y correr la API en un servidor o servicio en la nube.

4 Resultados

En primer lugar, en esta sección se muestran los análisis de evaluación y validación realizados sobre el modelo de DL que permitieron concretar la selección de la mejor opción del modelo. Posteriormente, se presentan los resultados obtenidos con la aplicación en el análisis de cuatro muestras de maíz.

4.1 Precisión del Modelo

Como resultado del proceso de evaluación del modelo de DL se obtuvieron las métricas RPN Localization Loss, RPN Objectness Loss, Box Localization Loss, Box Classification Loss y Total Loss. La evolución de la pérdida total en los 500.000 pasos del entrenamiento puede visualizarse en la Figura 3. Se observa cómo el valor de las pérdidas tiende a converger en un valor mínimo.

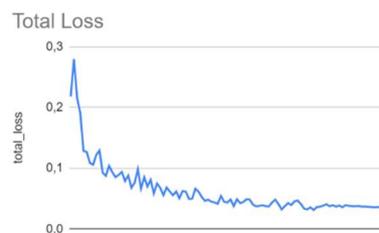


Fig. 3. Evolución de la métrica de evaluación Total Loss a lo largo del entrenamiento.

Para la validación del modelo se procesaron otras 27 imágenes que contenían 3119 objetos. Luego, utilizando la matriz de confusión, se calcularon las métricas definidas a continuación para cada clase detectada, en las distintas alternativas de detección.

- Precisión (P): la precisión es intuitivamente la capacidad del modelo de no etiquetar como positiva una muestra que es negativa.
- Recall (R): responde a la pregunta ¿qué proporción de los positivos reales se clasifica correctamente?
- F1 Score: es un número real que combina la precisión y el recall, otorgando un mayor peso a los números más bajos. Se calcula utilizando una media armónica.

El siguiente paso consiste en combinar los F1 Scores por clase en el F1 Score general del modelo, para lo cual se utilizó la métrica Macro-F1 Score, calculada como la media aritmética simple de los F1 Scores por clase.

$$\text{Macro F1 Score} = \frac{\sum_{i=1}^K \text{F1 Score}(K)}{K}$$

En la Figura 4 se puede observar el comportamiento de la métrica Macro F1-Score para las tres alternativas de detección durante el proceso de validación. Se deduce que la opción que combina las dos alternativas de detección obtiene mejores resultados en la mayoría de los casos. Se utilizó el modelo correspondiente al checkpoint 315.000 para la versión final de la aplicación, el cual obtuvo un valor de Macro F1 Score de 0.89928, logrando el segundo mayor valor. El modelo 305.000 lo superó con un valor de 0.90735. Sin embargo, el modelo 315.000 consiguió un F1 Score para la clase grano dañado de 0.7535, siendo el mejor para este rubro.

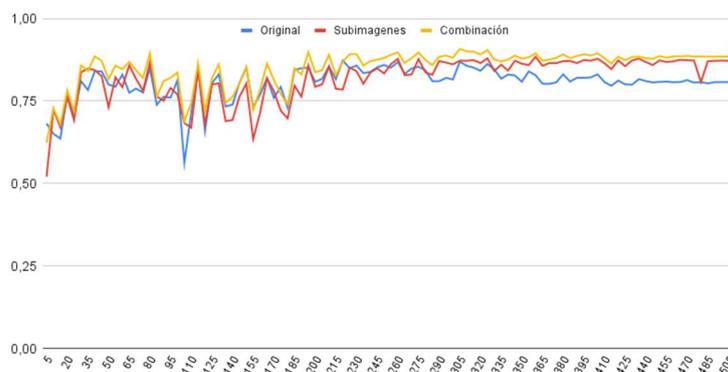


Fig. 4. Macro F1 Score para las tres alternativas de detección propuestas.

4.2 Calidad del maíz

En esta sección se detallan los resultados obtenidos por DeepSeed con el objetivo de evaluar su desempeño. Para esto, se adquirieron cuatro submuestras de maíz de 50 gramos a partir de cuatro muestras clasificadas por peritos clasificadores de granos de la Cooperativa Agropecuaria Gral. Necochea. Las submuestras fueron dispersadas individualmente sobre una hoja A4 blanca y fotografiadas. Estas cuatro pruebas,

llamadas casos de estudio, fueron evaluadas por DeepSeed y sus resultados se comparan con el análisis realizado por los peritos clasificadores en la Tabla 1.

Tabla 1. Comparación de resultados obtenidos por DeepSeed y peritos.

Rubro	Caso estudio 1		Caso estudio 2		Caso estudio 3		Caso estudio 4		Error
	Deep Seed	Peritos							
G. dañados	1.40	0.80	0.00	0.00	0.00	0.00	0.71	0.40	0.2275
G. quebrados	0.00	0.20	0.83	0.10	0.16	0.30	0.00	0.20	0.3175
Mat. extraña	1.81	1.00	0.46	0.05	0.00	0.10	0.65	1.00	0.4175
P.Hectolítrico	71.0	71.0	73.0	73.0	77.72	77.8	75.1	75.0	0.045
Grado	3	3	2	2	1	1	1	1	0

Se concluye que la aplicación desarrollada logra un buen comportamiento al momento de determinar el grado de calidad de cada submuestra analizada, ya que logra enmarcar los cuatro casos de estudio en el mismo grado de calidad resultantes de la tarea manual. Se puede deducir que DeepSeed obtuvo un error absoluto promedio de 0.2015. Cabe destacar que las submuestras analizadas por DeepSeed no son las mismas que las analizadas manualmente por los expertos, por lo que los errores obtenidos pueden deberse a esta diferencia. Por su parte, los altos valores alcanzados por la aplicación para ciertos rubros de calidad son causados por la no homogeneidad de las submuestras seleccionadas.

5 Conclusiones y trabajos futuros

El presente trabajo intenta cubrir la vacancia en la automatización de la clasificación de granos, introduciendo al sector agropecuario en el uso de la IA. Estas nuevas tecnologías aplicadas a la agricultura son sumamente necesarias para el incremento de la productividad.

La aplicación desarrollada, además de incorporar el uso de DL para clasificar los granos de maíz, posee el agregado de ser una PWA, lo que posibilita el acceso desde cualquier dispositivo. Para su uso solo se requiere depositar la muestra sobre una hoja blanca tamaño A4 y tomar una fotografía.

Esta propuesta permite conocer la calidad de los granos durante la cosecha y en sus posteriores etapas de almacenamiento y comercialización, elevando así los estándares de exportación y suministrando trazabilidad a la cadena agroexportadora. Además, constituye una importante herramienta de soporte para los peritos clasificadores.

Con el objetivo de perfeccionar la herramienta, será necesario aumentar el volumen de los datos para entrenar el modelo de DL, ya sea incorporando nuevas muestras al dataset o tomando nuevas capturas bajo diferentes condiciones y entornos. Otra variante sería utilizar un modelo de DL que esté pre entrenado con un set de datos más relacionado a los objetos que posteriormente reconocerá.

Finalmente, una vez optimizado el modelo para la clasificación de granos de maíz, resulta interesante extender la herramienta a otros cultivos como trigo, soja, girasol y

cebada. Esta extensión implica nuevos entrenamientos del modelo para detectar los rubros determinantes de calidad en cada cultivo.

6 Referencias

1. A. Jain, Deep Learning for Computer Vision – Introduction to Convolution Neural Networks. 2016. Disponible en: <https://www.analyticsvidhya.com/blog/2016/04/deep-learning-computer-vision-introduction-convolution-neural-networks/> (accedido Ago. 2021)
2. A. S. Saleres, Aplicación web para la clasificación de granos de maíz. Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires. 2018.
3. A. Zhang, Z. C. Lipton, M. Li y A. J. Smola, Dive into Deep Learning. 2020.
4. B. Shmueli, Multi-Class Metrics Made Simple, Part II: the F1-score. 2019. Disponible en: <https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1> (accedido Ago. 2021)
5. C. Aggarwal, Neural Networks and Deep Learning: A Textbook. Springer. 2018.
6. F. Ozge Unel, B. O. Ozkalayci y C. Cigla, The Power of Tiling for Small Object Detection. 2019.
7. G. Suseendran, E. Chandrasekaran, D. Akila y D. Balaganesh, Automatic Seed Classification by Multi-Layer Neural Network with Spatial-Feature Extraction. 2020.
8. I. Goodfellow, Y. Bengio y A. Courville, Deep Learning. 2016.
9. K. He, X. Zhang, S. Ren y J. Sun, Deep Residual Learning for Image Recognition. 2015.
10. M. Bhurtel, J. Shrestha, N. Lama, S. Bhattarai, A. Uprety y M. Kumar Guragain. Deep Learning based Seed Quality Tester. 2019.
11. M. Manav, Introduction to Convolutional Neural Networks (CNN). 2021. Disponible en: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/> (accedido Ago. 2021)
12. P. Sharma, A Step-by-Step Introduction to the Basic Object Detection Algorithms (Part 1). 2018. Disponible en: <https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/> (accedido Ago. 2021)
13. S. Huang, X. Fan, L. Sun , Y. Shen y X. Suo, Research on Classification Method of Maize Seed Defect Based on Machine Vision. 2019.
14. S. Ren, K. He, R. Girshick y J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015.
15. S. Rusell y P. Norvig, Artificial Intelligence: A modern approach (Fourth Edition). 2020.
16. T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick y P. Dollár, Microsoft COCO: Common Objects in Context. 2014.
17. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu y A. C. Berg, SSD: Single Shot MultiBox Detector. 2015.
18. API de detección de objetos de TensorFlow. GitHub. Disponible en: https://github.com/tensorflow/models/tree/master/research/object_detection
19. Crearon una tecnología que permite conocer la calidad del trigo en 8 segundos. Puntobiz.com. Disponible en: <https://puntobiz.com.ar/lo-nuevo/2018-8-9-6-1-0-crearon-una-tecnologia-que-permite-conocer-la-calidad-del-trigo-en-8-segundos> (accedido Ago. 2021)
20. Flask. Disponible en: <https://flask.palletsprojects.com/en/2.0.x/>
21. Google Colaboratory. Disponible en: <https://colab.research.google.com/>
22. LabelImg. Disponible en: <https://github.com/tzutalin/labelImg>
23. Progressive Web Apps. Disponible en: <https://web.dev/progressive-web-apps>
24. ZoomAgri. Disponible en: <https://zoomagri.com/>