EVALUACIÓN DEL ESTADO DE ENVASES PLÁSTICOS PRIMARIOS MEDIANTE REDES NEURONALES CONVOLUCIONALES (CNN)

SEBASTIÁN FRANCO GÓMEZ

JHONATAN OSPINA OSORIO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

2022

EVALUACIÓN DEL ESTADO DE ENVASES PLÁSTICOS PRIMARIOS MEDIANTE REDES NEURONALES CONVOLUCIONALES (CNN)

SEBASTIÁN FRANCO GÓMEZ

JHONATAN OSPINA OSORIO

TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

DIRECTOR

Ph.D GUILLERMO ROBERTO SOLARTE MARTINEZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

2022

AGRADECIMIENTOS

Agradezco a mis chicos que siempre me tuvieron paciencia cuando no rendía como esperaba, agradezco a mis padres que nunca me presionaron por llevar el ritmo con el que me sentía cómodo, agradezco a la universidad por permitirme convertirme en profesional dentro de ella y agradezco a cada persona que de alguna manera llenó de obstáculos mi proceso académico porque en su superación encontré mi realización. Agradezco a mi director de tesis por su paciencia y agradezco a internet, porque en él conecté con una fracción del mundo que exploraré el resto de mi vida. Agradezco a cada persona que ha aportado su conocimiento a la ciencia sin esperar nada a cambio y agradezco a Platzi, que en su cultura encontré el paso final para volverme un profesional. Agradezco a todas las personas que lean nuestro artículo porque nos están permitiendo transmitir nuestros esfuerzos, inmortalizandolos en sus memorias. El mundo es un desierto de conocimiento y podemos sonreír porque ahora hacemos parte de él, aunque solo seamos un grano de arena.

-Sebastián Franco y Jhonatan Ospina, en un consenso de pensamientos.

CONTENIDO

TABLA DE IMÁGENES

RESUMEN

La inteligencia artificial ha permitido la automatización de procesos anteriormente inviables gracias a su flexibilidad en la toma de decisiones y reconocimiento blando de patrones, por lo que tareas de reconocimiento y clasificación ahora son viables de realizar por equipos de cómputo.

En la presente se propone el uso de Redes Neuronales Convolucionales para la clasificación de envases plásticos primarios. Se crea un conjunto de datos de entrenamiento en su totalidad y se modela la arquitectura general del sistema, se configuran parámetros e hiper parámetros y se entrena la red durante 70 épocas usando técnicas de data augmentation y normalización de datos, finalmente se obtienen resultados y se determina la viabilidad del modelo mediante su precisión en datos no entrenados previamente.

ABSTRACT

Artificial intelligence has allowed the automation of previously unfeasible processes thanks to its flexibility in decision making and soft pattern recognition, so that recognition and classification tasks are now feasible to be performed by computer systems.

In the present work, the use of Convolutional Neural Networks for the classification of primary plastic containers is proposed. A training dataset is created in its entirety and the general architecture of the system is modeled, parameters and hyper parameters are configured and the network is trained for 70 epochs using data augmentation and data normalization techniques, finally results are obtained and the viability of the model is determined by its accuracy on previously untrained data.

PALABRAS CLAVE: Clasificación Binaria, Data Augmentation, Dataset, Deep Learning, Redes Neuronales Convolucionales.

KEYWORDS: Binary Classification, Data Augmentation, Dataset, Deep Learning, Convolutional Neural Network.

1. INTRODUCCIÓN

La tecnología se puede considerar como "Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico" según la RAE [1]; soluciona problemas, suple necesidades y crea nuevas oportunidades. Su naturaleza es sencilla y se considera como parte ella cualquier invento hecho por la humanidad, desde prendas primitivas hasta equipos mecánicos de alta complejidad.

La tecnología evoluciona a la par que la humanidad y se transforma para solucionar necesidades emergentes a través del tiempo, donde en el pasado reciente y presente dicha tecnología ha sido orientada a mecanismos automatizados que suplen parcial o totalmente las actividades del humano en diferentes contextos.

Actualmente la tecnología informática se ha convertido en un pilar fundamental de la civilización en diferentes ámbitos [2], donde a nivel empresarial se indaga de manera agresiva la integración de métodos y procesos de computación capaces de optimizar tareas y mejorar resultados; un caso de uso de dicha integración es la automatización de procesos repetitivos que son realizados por trabajadores humanos con rendimiento variable, dichas tareas pueden ser automatizadas para reintegrar a su anterior encargado a procesos blandos y poco automatizables.

Durante la última década se han explorado diferentes tecnologías emergentes que han permitido explotar nuevas perspectivas y automatizar procesos cada vez más complejos, como la inteligencia artificial, la cual genera un patrón de juicio blando según el contexto en el que son aplicadas [3], entonces, ¿Cómo se podría aprovechar la tecnología emergente en la automatización de procesos repetitivos?

2. PLANTEAMIENTO DEL PROBLEMA

La generación en masa de productos tangibles requiere de procesos de aseguramiento de calidad en todas sus etapas, desde la auditoría de materias primas hasta estándares y metodologías propias de cada sector según la etapa del ciclo de vida de fabricación y distribución.

Una vez los diferentes productos son distribuidos a clientes finales estos aplican sus propios procesos de aseguramiento de calidad que varían en su totalidad según el volumen del cliente, si este es un distribuidor o un consumidor final.

Existen limitaciones en el aseguramiento de calidad de los productos, especialmente en su contenido interno en el contexto de empaquetados de diferentes tipos (consumibles, medicina, contenedores, entre otros) del cual se encargan entidades reguladoras, dejando en manos del cliente la verificación de los contenedores de los productos que comercializan.

El proceso de aseguramiento de calidad de contenedores o envases es manual y puede consumir altos volúmenes de tiempo y personal si la cantidad a procesar es densa, por lo que un encargado interno debe llevar a cabo estas tareas rutinariamente.

Dichas tareas son repetitivas, rutinarias y masivas, características óptimas de un proceso automatizable mediante técnicas de computación, pero, ¿Qué tan viable es automatizar la revisión del estado de los envases de productos finales?

3. ESTADO DEL ARTE

Se consideran investigaciones, artículos, y desarrollos relevantes al proyecto:

- "Anomaly detection with convolutional neural networks for industrial surface inspection" [5]

  Autores: Benjamin Staar, Michael Lütjen, Michael Freitag.

  Aportes a la investigación: Se expone una arquitectura especial para el procesamiento interno de imágenes con el fin de encontrar pequeños defectos en diferentes texturas del Dataset DGAM 2007 publicado en Kaggle [6].

- "The Application of One-Class Classifier Based on CNN in Image Defect Detection" [7]

  Autores: Mei Zhang, Jinglan Wu, Huifeng Lin, Peng Yuan, Yanan Song.

  Aportes a la investigación: Se expone el uso de Redes Neuronales Convolucionales en la detección de componentes electrónicos falsos. Se expone la metodología y el proceso interno de convolución para lograr dichas predicciones.

- "Small Defect Detection Using Convolutional Neural Network Features and Random Forests" [8]

  Autores: Xinghui Dong, Chris J. Taylor, and Tim F. Cootes.

  Aportes a la investigación: Se expone el uso de Redes Neuronales Convolucionales y la técnicas de Random Forest para predecir pequeños defectos en un dataset. Se indaga especialmente en la definición del dataset y tratamiento de datos.

- "Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network" [9]

  Autores: Curtis Wigington, Seth Stewart, Brian Davis, Bill Barrett.

  Aportes a la investigación: Se exponen diferentes técnicas de Data Augmentation para el problema de detección de letras en texto escrito a mano. Se indaga especialmente en las diferentes técnicas de obtener nuevos datos a partir de los datos iniciales.

- "Deep Convolutional Neural Networks Design Patterns" [10]

  Autores: Leslie N. Smith, Nicholay Topin.

  Aportes a la investigación: Se exponen diferentes técnicas a aplicar en arquitecturas de Redes Neuronales Convolucionales .Se indaga especialmente en el uso y aplicación de técnicas para mejorar el rendimiento general de la red y mejorar su tiempo de convergencia.

4. OBJETIVOS

4.1 Objetivos general

- Desarrollar un evaluador de envases plásticos primarios mediante modelos de aprendizaje automatizado.

4.2 Objetivos específicos

- Precisar requisitos funcionales y no funcionales del evaluador.
- Delimitar los parámetros del modelo de Deep Learning a elegir.
- Determinar el conjunto de datos de entrenamiento (dataset) del evaluador de envases plásticos primarios.
- Recopilar los datos a entrenar en el modelo de deep learning.
- Entrenar el modelo definido mediante el dataset recopilado.
- Determinar la viabilidad del evaluador.

5. JUSTIFICACIÓN

La presente se propone como una posible solución a los problemas derivados del proceso manual de evaluación del estado de envases plásticos, donde se requiere que un humano supervise individualmente cada envase recibido de diferentes medios, si este es defectuoso debe separarlo e informar sobre dicho descubrimiento.

Se propone la automatización inicial del proceso de evaluación, donde ya no es necesaria la constante supervisión de cada envase sino la clasificación y verificación de las predicciones del sistema, lo que reduce exponencialmente los recursos invertidos en dicha tarea general.

Se aplican tecnologías afines a la inteligencia artificial dada su utilidad en el contexto de problemas de clasificación de alta complejidad y de naturaleza blanda.

Se generan todos los datos de entrenamiento de la red a la medida según un caso de uso controlado para exponer el contexto y ciclo de vida entero del entrenamiento de una Red Neuronal.

Se elige como arquitectura general de entrenamiento una Red Neuronal Convolucional (CNN) dada su afinidad su afinidad en el procesamiento blando de imágenes [4].

Con todo lo anterior, se justifica el desarrollo de un modelo de Deep Learning basado en Redes Neuronales Convolucionales usando como datos de entrenamiento imágenes hechas a la medida en un contexto controlado para optimizar los tiempos de ejecución de tareas de clasificación de envases plásticos.

6. MARCO REFERENCIAL

6.1 Inteligencia Artificial

La inteligencia artificial se remonta a la década de 1950, donde se indaga por primera vez el concepto de autonomía en los equipos de cómputo, buscando que tuviesen la capacidad de tomar decisiones según diferentes entradas al sistema. Los primeros modelos de inteligencia artificial están relacionados a sistemas profundamente controlados y orientados a restricciones (hoy conocidos como sistemas expertos). En la década de 1970 se genera un nuevo auge por los sistemas expertos y mediante una robusta cantidad de restricciones se generaron sistemas que podían realizar acciones de orden lógico como jugar ajedrez o realizar diagnósticos en contextos controlados [11]. La inteligencia artificial busca realizar una mímica de la toma de decisiones humana mediante diferentes ramas segmentadas según caso de estudio, naturaleza de datos o metodología interna.

6.2 Machine Learning

El Machine Learning o aprendizaje de máquina es una rama de la inteligencia artificial que expone a la máquina a una serie de datos para que, mediante patrones internos genere respuestas lógicas a través del tiempo.

El aprendizaje de máquina se divide en 2 escuelas principales: Aprendizaje supervisado y aprendizaje no supervisado, donde la principal diferencia es que, en el aprendizaje supervisado se le entrega al algoritmo su meta o lo que debería calcular, mientras que en contexto no supervisado no se le ofrece un resultado u objetivo, siendo que esta debe encontrar patrones internos mediante correlaciones matemáticas [12]. Los algoritmos de clasificación, agrupamiento y regresión pueden ser parte del aprendizaje de máquina siempre y cuando estos mejoren a través de las iteraciones o épocas.

6.3 Deep Learning

El Deep Learning es una sub rama del Machine Learning que busca generar un mejor procesamiento de los datos mediante el agrupamiento de capas cada vez más complejas, este proceso se obtiene principalmente mediante Redes Neuronales Artificiales (ANN - Artificial Neural Networks en inglés), un proceso matemático que originalmente busca imitar el proceso de aprendizaje del cerebro humano neurona a neurona.

6.4 Redes Neuronales Artificiales

Las Redes Neuronales Artificiales son una abstracción teórica del proceso de aprendizaje del cerebro humano, donde una red de millones de neuronas se encuentran interconectadas y se envían estímulos en masa para generar una reacción a nivel de organismo. La abstracción de Red Neuronal Artificial dicta que entre más capas de neuronas se apilen, más profunda y diversa puede ser la predicción o el estímulo final, como un proceso cerebral real.

Las Redes Neuronales Artificiales se rigen bajo el Teorema de la Aproximación Universal, el cual dispone que cualquier función lineal puede ser aproximada por estas redes si dicha red posee una neurona mientras que cualquier función real integrable puede ser aproximada por una red con 2 o más neuronas [13]. La calidad de la aproximación es directamente proporcional a la cantidad de neuronas, sin embargo, dicha aproximación puede mejorar exponencialmente si se aplica el principio de apilar más y más capas a través de la red.

El ciclo de vida de una red neuronal consta en 3 fases: La fase de entrada de datos, la fase de procesamiento de datos y la fase de salida de datos o predicción/regresión.

En la primer fase se introducen como entrada los datos a ser procesados, estos son abstracciones de datos con sentido, donde una imagen a ojos humanos es una matriz de valores RGB en 3 canales, a su vez que una serie de valores de 0 a 10 pueden ser calificaciones a un servicio domiciliario. La red procesa datos sin contexto y trata de aproximarse a un patrón que corresponda a su predicción esperada.

La segunda fase se especializa en el proceso de aprendizaje y predicción de la red. Cada neurona lleva a cabo una operación matemática que transformará mediante funciones de activación, este proceso es retroalimentativo y cada neurona trabaja con los resultados de neuronas anteriores hasta la predicción final.

La fase de salida consta en la transformación de los cálculos lineales a una función matemática acorde a la predicción esperada; si se espera una regresión no se aplica dicha fase, mientras que si se espera una predicción binaria se aplica una función sigmoide y si se espera una clasificación en un rango de salidas entonces se opta por una función softmax.

Las redes neuronales se dividen en diferentes tipos de redes, cada una con una arquitectura única orientada a procesar tipos específicos de datos, algunas son:

- Las Redes Neuronales Convolucionales (Convolutional Neural Network - CNN en inglés) se especializan en el procesamiento de imágenes de varias dimensiones.

- Las Redes Neuronales Recurrentes (Recurrent Neural Network - RNN en inglés) se especializan en el procesamiento de datos en tiempo real, ideal para audios y videos.

- Las Redes Neuronales de Memoria a Corto Plazo (Long Short Term Memory Networks - LSTM en inglés) se especializan en guardar información de los datos a través de las capas de entrenamiento, ideales para procesos de texto y traducciones que requieren de contexto.

6.5 Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales son un tipo de Red Neuronal Artificial especializadas en el procesamiento de imágenes dada su operación de convolución, además de generar procesos derivados de dicha operación con el fin de potenciar exponencialmente la calidad de predicciones en comparación al sistema base de Redes Neuronales Artificiales.

La convolución es un proceso matemático al que se somete una imagen de entrada para sobresaltar un detalle en específico de la misma. Se genera una matriz de tamaño nxn con valores dependiendo del fin del filtro, posteriormente se somete a la imagen original a una operación de multiplicación de producto punto con el filtro para obtener nuevos valores. Dichos valores son una nueva representación de la imagen original de tamaño reducido, modificada para detectar cambios específicos, como bordes o imperfecciones.

Cada que se aplica una convolución se genera una nueva imagen, por lo que, si en una capa oculta se aplican 16 filtros de convolución, se generarían 16 imágenes nuevas, todas con diferentes filtros que detallarían segmentos diferentes de la misma imagen; esto implica que el crecimiento de imágenes según la original es exponencial, pero la disminución de su tamaño también lo es.

Finalmente, para respetar el proceso original de las Redes Neuronales Artificiales, se transforman todas las convoluciones al cálculo lineal y se someten a una capa final de predicción.

6.6 Dataset

Un Dataset o conjunto de datos es el agrupamiento de un bloque de datos procesados o sin procesar a ser usados en un contexto específico. En el marco de las redes neuronales, son la materia prima con la cual la red entrenará posteriormente.

Para las Redes Neuronales Convolucionales el Dataset es un conjunto robusto de imágenes relacionadas a una predicción esperada. Por ejemplo, se podría tener un Dataset con 1000 imágenes de perros y 1000 imágenes de gatos con sus respectivas etiquetas. Dicho Dataset suele estar estandarizado en formatos, resoluciones, nombres y demás, pero aún no es apto para ser procesado por la red.

6.7 Procesamiento de imágenes

El procesamiento de imágenes es el conjunto de técnicas para contextualizar el Dataset y convertirlo en datos para la red. Una imagen de canales RGB es una matriz de 3 dimensiones con valores entre 0 y 255 que a ojos del usuario final es una representación de una fotografía, además de tener una resolución de (n x m) y un formato (jpg o png).

La conversión de dicha imagen a la red neuronal es una matriz de tamaño (n x m x 3) con valores internos de 0 a 255; de esta manera la red puede leer los datos y aproximarlos con mayor naturaleza a la predicción esperada.

El procesamiento de imágenes limpia y trata las imágenes originales reduciendo su resolución, eliminando ruido, cambiando formatos o modificando levemente la entrada original para crear alta variedad de imágenes nuevas, este último concepto es llamado Data Augmentation y es clave para el aumento artificial del Dataset.

6.8 TensorFlow

TensorFlow es un Framework de Google de código abierto para el desarrollo de inteligencia artificial mediante un modelo de grafos. Google usa esta herramienta para sus desarrollos y aplicaciones tales como el Traductor de Google (Google Translator en inglés) se encuentran creadas en dicho Framework.

6.9 Keras

Keras es una API generada para facilitar la programación de algoritmos de inteligencia artificial desarrollada en el lenguaje de programación Python por François Chollet. Actualmente se encuentra integrada nativamente en TensorFlow y es dicha combinación la usada en el desarrollo de la presente.

7. METODOLOGÍA

7.1 Precisar requisitos funcionales y no funcionales del evaluador

Se determinarán los requisitos funcionales y no funcionales del evaluador según factores como atributos de calidad, estándares y demás reglas de negocio implícitas, además se indagará en el modelo e ingeniería de software del mismo para delimitar funcionalidades externas y restricciones del sistema.

7.2 Delimitar los parámetros del modelo de Deep Learning a elegir

Se delimitaran las variables relacionadas al modelo de de deep learning a elegir, como el tipo de arquitectura, sus parámetros e hiper parámetros. Se determinarán funciones de activación, procesos de optimización y técnicas a aplicar.

7.3 Determinar el conjunto de datos de entrenamiento (Dataset) del evaluador de envases

Se determinará el conjunto de datos (o dataset) a ser entrenado en el modelo de deep learning a elegir. Se generará un dataset propio con envases plásticos adquiridos. Se delimitarán las características del dataset como cantidad de datos, resolución de la imagen, recolección individual y grupal de los datos y agrupamientos de los mismos.

7.4 Recopilar los datos a entrenar en el modelo de deep learning

Se recopilarán datos para entrenar el modelo de deep learning de manera manual, tomando fotografías estandarizadas a cada envase a evaluar; se aplicarán técnicas de procesamiento de imágenes para aumentar artificialmente el tamaño del dataset a entrenar.

7.5 Entrenar el modelo definido mediante el dataset recopilado

Se entrenará el modelo de Deep Learning con los datos previamente recopilados, se probarán diferentes técnicas de entrenamiento con leves cambios.

7.6 Determinar la viabilidad del evaluador

Se determinará la viabilidad del evaluador según los resultados del entrenamiento y pruebas con nuevos envases no usados en anteriores etapas.

8. DESARROLLO

Para desarrollar el proyecto y alcanzar el objetivo general, se explica la resolución de cada objetivo específico durante el transcurso de la investigación.

8.1 Requisitos del evaluador

Se definen los requisitos del evaluador y su flujo de trabajo mediante vistas C4.

Se consideran 2 procesos de uso del software, donde el usuario usa el aplicativo en el entrenamiento de la red en el primer caso y obtiene sus predicciones en el segundo.

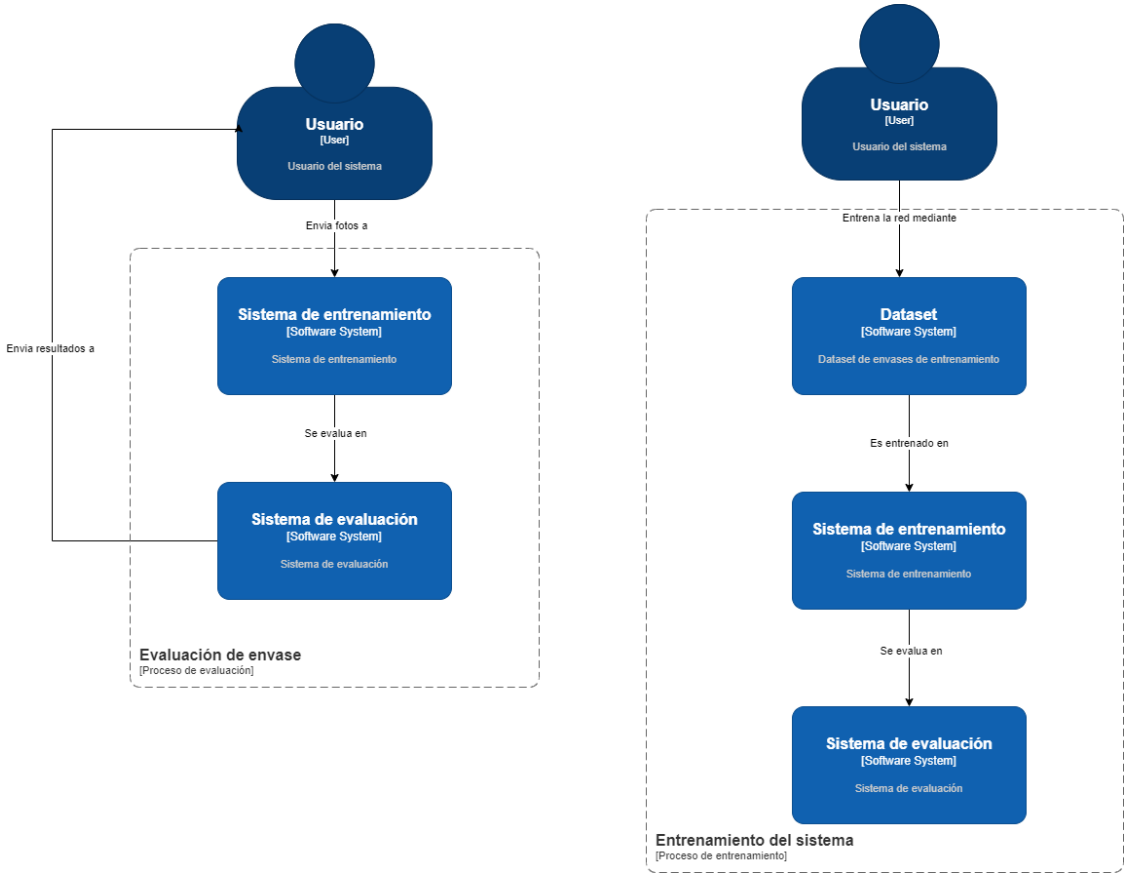

Imagen 1: Vista C1: Contexto del sistema

Se especifica en los contenedores del sistema mediante el vista de contenedores

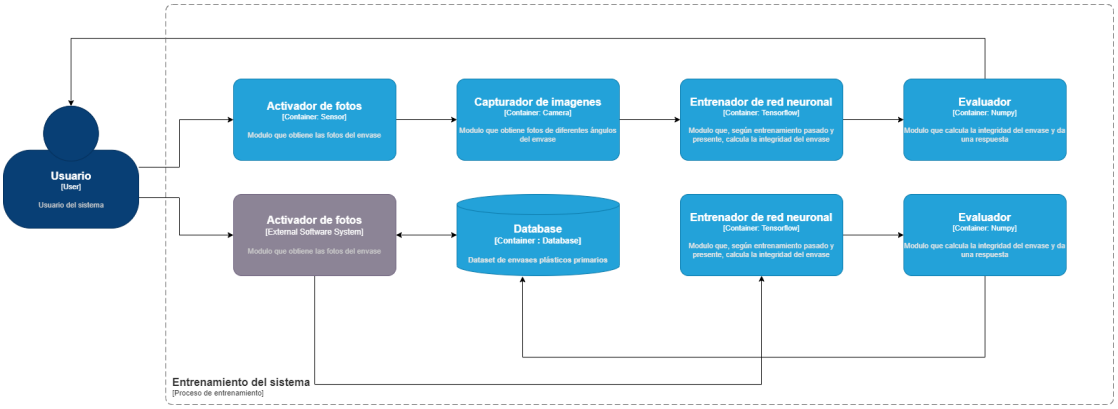

Imagen 2: Vista C2: Contenedores

Se específica cada proceso del ciclo de vida de la red mediante la vista de componentes:

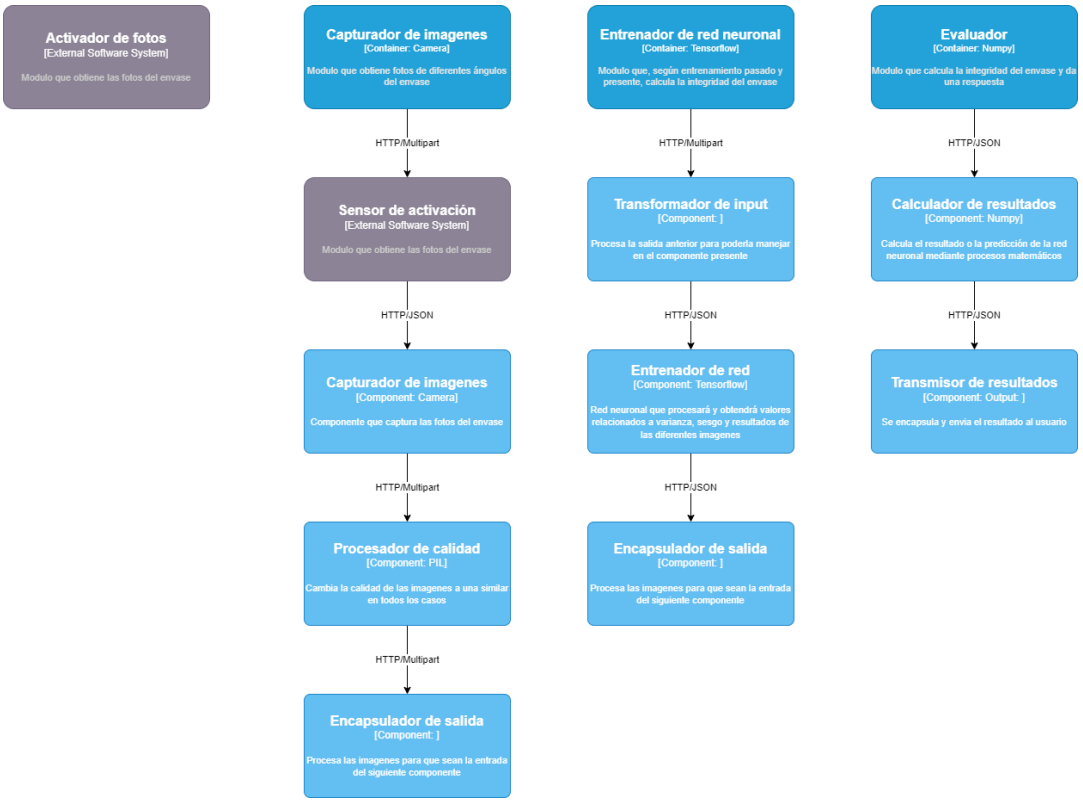

Imagen 3: Vista C3: Vista de componentes

Finalmente, se explora en las clases que tendrá el sistema

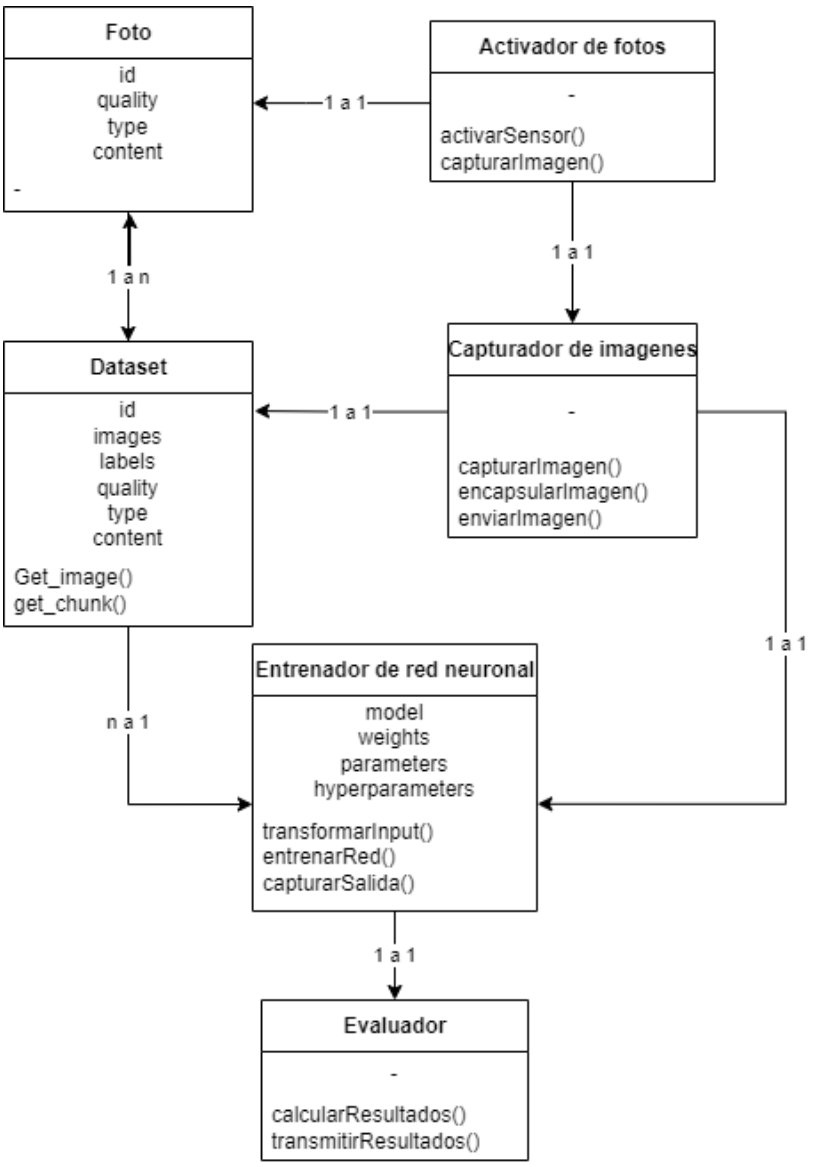

Imagen 4: Vista C4: Vista de clases

8.2 Delimitación de los parámetros del modelo trabajado

Dada la naturaleza del proyecto, se indaga en un modelo de Red Neuronal Convolucional (CNN) como base gracias a que este posee alta afinidad en el procesamiento y detección de patrones en imágenes basado en entrenamiento supervisado; una vez definida la base se determina la arquitectura de la red, donde es importante considerar las 3 bases fundamentales de una red neuronal: La capa de entrada, las capas ocultas y la capa de salida, además de la elección y configuración de los parámetros, hiper parámetros y técnicas a usar.

8.2.1 Capa de entrada

En el proceso de arquitectura de la red, se determina una capa de entrada con input de una imagen de 3 canales (RGB) de resolución 256 x 180 píxeles transformada a un array de numpy internamente; la resolución se elige como un escalado al tamaño

inicial de la imagen (4000 x 3000 píxeles) y se reduce a dicha escala por términos de procesamiento de la imagen y complejidad de entrenamiento del modelo.

8.2.2 Capas ocultas

Se determina un total de 3 capas ocultas de convolución, las cuales llevan a cabo el proceso de aprendizaje y clasificación de cada imagen, cada una posee el mismo procedimiento con una cantidad ascendente de neuronas actuando, añadiendo un nivel mayor de abstracción por capa.

Para cada capa oculta, se maneja una operación de convolución de inicialmente 16 filtros creciente con un kernel de 3x3, sosteniendo el tamaño original de la imagen en cada iteración y retroalimentando la siguiente capa mediante la función de activación ReLU.

8.2.3 Capa de salida

Una vez se procesa la imagen por cada capa oculta, se aplana en una capa especial intermedia denominada “Flatten” y se transforman nuevamente al cálculo lineal para calcular la predicción final mediante 256 neuronas, se evalúa el resultado en una única neurona de salida con la función de activación “Sigmoide”, especial para la clasificación binaria y se obtiene el resultado probabilístico, donde la predicción es positiva si el resultado es superior a 0.5 y negativa en caso contrario.

8.2.4 Parámetros e Hiper Parámetros

Durante el entrenamiento de la red, se configuran una serie de parámetros e hiper parámetros para mejorar el rendimiento de la red y estandarizar los datos.

Dichas configuraciones son:

8.2.4.1 Neuronas por capa

Se inicializa la red con 16 filtros en su primera convolución duplicando sus filtros por cada iteración, siendo respectivamente 16, 32 y 64. La cantidad inicial se elige por facilidad de procesamiento a nivel de hardware, dado que el número 16 es potencia directa de 2 y se incrementa por duplicación para aprovechar al máximo las características de aprendizaje profundo de dichas redes. El valor inicial es mínimo dado que el crecimiento exponencial lleva a la red a buscar patrones más ocultos que únicamente puede hallar con altos volúmenes de datos.

8.2.4.2 Tamaño del kernel

Para la operación de convolución se considera un filtro base o kernel de 3x3 para procesar los patrones más importantes de cada segmento de la imagen a

procesar, además, se mantiene el tamaño real de la imagen después de la operación de convolución para no perder datos entre iteraciones.

8.2.4.3 Funciones de activación

Se elige como función de activación para cada capa oculta la función ReLU, la cual permite un rápido proceso de retroalimentación en el backpropagation y no ralentiza el aprendizaje cuando la predicción falla; por otra parte, se elige como función de activación final (o de predicción) la función sigmoide, la cual, independientemente de su entrada, transforma el resultado a un rango entre 0 y 1, siendo útil en la predicción final del modelo.

8.2.4.4 Regularizadores

Se eligen como regularizadores de pesos regularizadores L1 y L2 en conjunto para evitar que la red sufra de procesos de sobreaprendizaje, se determina un valor lambda mínimo de 1x10^-5 para L1 y 1x10^-4 para L2 respectivamente. Estos valores sólo afectan al aprendizaje de la red si sufre una etapa de sobreaprendizaje y permiten que salgan de dichos mínimos locales de manera óptima.

8.2.4.5 Normalizaciones

Se normalizan los datos cada vez que son procesados por una capa oculta mediante la técnica de BatchNormalization o normalización por lotes para que los datos se aproximen a la media de sus valores y haya convergencia en la menor cantidad de iteraciones posible.

8.2.4.6 Max Pooling

Se aplica la técnica de Max Pooling para resaltar los mejores valores de cada convolución a la vez que reduciendo el tamaño de la imagen, esto se hace para reducir la complejidad de entrenamiento por capa sin necesidad de perder calidad de predicción. Se elige un filtro de 2x2 para reducir la imagen por cada iteración y obtener 1 pixel relevante por cada 4 procesados.

8.2.4.7 Dropout

Se aplica un Dropout de 35% en cada capa oculta para desactivar artificialmente un porcentaje de las neuronas en cada época de aprendizaje, esto se hace para que la red no pueda reforzar patrones por fuerza bruta y detecte correctamente los patrones de predicción.

8.2.5 Arquitectura general de la red

Se divide la arquitectura de la red por para de entrada, capa oculta y capa de salida:

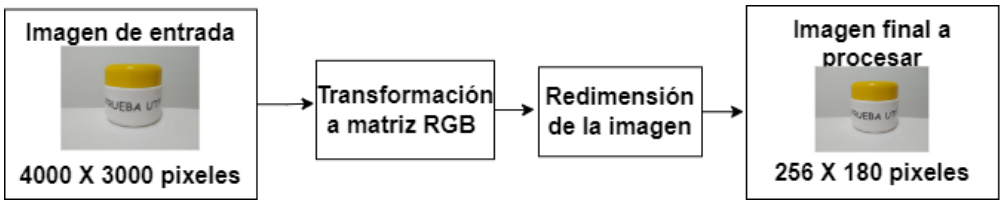

Imagen 5: Arquitectura de la capa de entrada

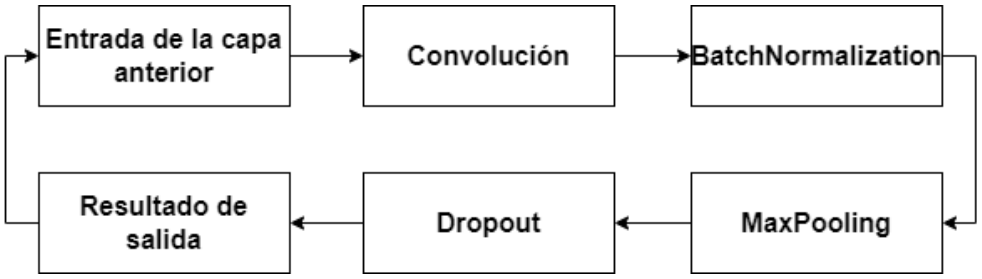

Imagen 6: Arquitectura de las capas ocultas

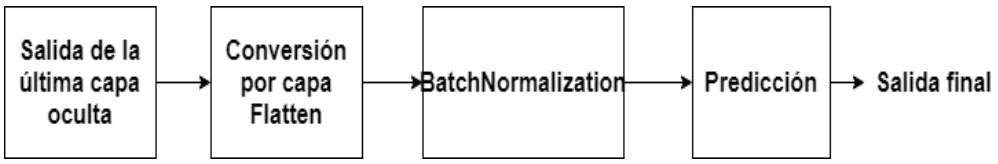

Imagen 7: Arquitectura de la capa de salida

La arquitectura general de la red es el agrupamiento total de las capas anteriores:

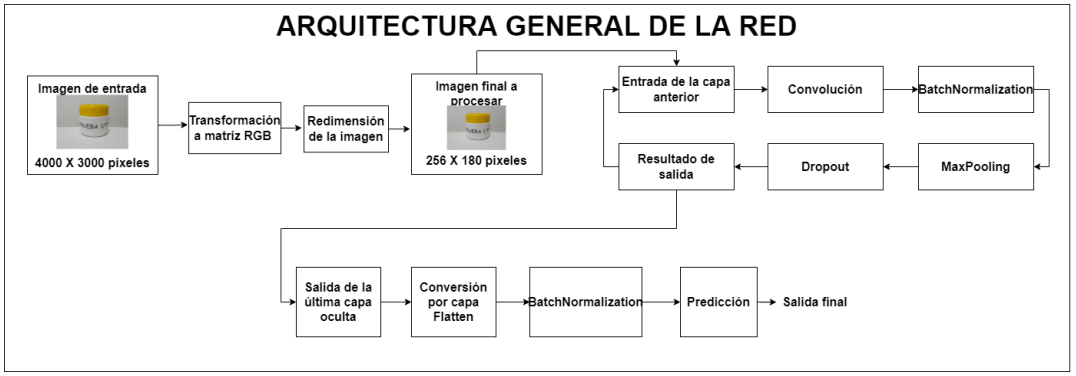

Imagen 8: Arquitectura general de la red

8.3 Delimitación del Dataset

8.3.1 Especificaciones del envase

Para entrenar el modelo de Deep Learning se opta por un prototipo de envase plástico único para probar las características de predicción en un contexto controlado. Se opta por un envase de gel de 250CC en polipropileno de tapa amarilla con los siguientes argumentos:

- Se elige un envase de gel de 250 CC dado que tiene estructura cilíndrica, ideal para fotografías desde cámaras.

- Se opta por el polipropileno como material dado que refleja luz, esto genera mayor variedad de ejemplos a la hora de diversificar el dataset.

- Su tapa es amarilla dado que no se asemeja totalmente a alguna de las escalas RGB, evitando sobrecarga en alguna capa de la convolución.

8.3.2 Estandarización del dataset

Se adquieren un total de 100 envases plásticos con las mismas características, a cada uno se le añade una etiqueta frontal artificial para agregar nuevos patrones a verificar. Un envase promedio del dataset se ve de la siguiente manera:

Imagen 9: Ejemplo frontal de envase

Una vez procesados los envases plásticos, se delimitan las características para convertirlos en un dataset.

Para asegurar la integridad del envase plástico, se deben verificar 3 filtros principales: La parte frontal (donde se encuentra su etiqueta), la parte trasera (donde no hay etiquetas o agregados) y la parte superior (donde se verifica la integridad de la tapa plástica). Los filtros delanteros y traseros además verifican la integridad de la tapa a

nivel lateral, y cada uno verifica aproximadamente 180° de la imagen; las partes traseras y superiores de un envase se ven de la siguiente manera:

Imagen 10: Ejemplo trasero de envase

Imagen 11: Ejemplo superior de envase

8.3.3 Especificaciones de las imágenes

Se recolecta la información mediante una cámara celular de 48 Megapíxeles, resultando en imágenes de 4000x3000 pixeles cada una estandarizadas en 2 planos fotográficos diferentes.

8.3.4 Especificaciones de los ejemplos verdaderos y falsos

Las imágenes anteriores corresponden a ejemplos verdaderos o íntegros de envases plásticos, sin embargo, se debe alimentar a la red con una cantidad similar de ejemplos falsos o en este caso, envases que se encuentren dañados, rotos, alterados o con cualquier estado que ponga en peligro su integridad.

Para los diferentes filtros se crean una serie de casos falsos base que la red aprenderá a detectar para ser interpolados a otras posibilidades por aprendizaje blando. Se manejan los siguientes casos de adulteración de envases:

- Adulteración por edición: Se agrega contenido extra a alguno de los filtros de los envases, puede ser de manera abrupta o sutil, donde se editan con escalas de colores diferentes a la del envase y pueden verse fuerte o levemente afectados. Un ejemplo de esta adulteración es el siguiente:

Imagen 12: Ejemplo de adulteración por edición

- Adulteración por obstáculos: Se agregan obstáculos artificiales a alguno de los filtros de los envases como elementos del entorno o cinta, pueden verse leve o fuertemente editados, Un ejemplo de esta adulteración es el siguiente:

Imagen 13: Ejemplo de adulteración por obstáculos

- Adulteración por destrucción: Se destruye parcial o totalmente alguno de los filtros de los envases mediante la sustracción literal de alguna de sus partes o aplastamiento leve o fuerte. Un ejemplo de esta adulteración es el siguiente:

Imagen 14: Ejemplo de adulteración por destrucción

- Adulteración híbrida: Se combinan 2 o más filtros anteriores y se genera un envase con una fuerte adulteración total. Un ejemplo de esta adulteración es el siguiente:

Imagen 15: Ejemplo de adulteración híbrida

8.4 Recopilación y procesamiento del dataset

Se recopila y procesa el dataset una vez caracterizado el conjunto de entrenamiento; se debe indagar en la creación del mismo teniendo en cuenta los recursos disponibles.

8.4.1 Delimitación de recursos

Se posee un total de 100 ejemplos de envases plásticos íntegros sin modificaciones y se dividen en 2 conjuntos, donde el primero se relaciona con los ejemplos verdaderos de envases y el segundo con ejemplos falsos respectivamente. La división de los envases para cada grupo no es equivalente, dado que, un envase verdadero siempre tendrá las mismas características (a excepción de ciertas adulteraciones tolerables, como inclinación de la etiqueta) mientras que un envase falso puede verse alterado por diferentes situaciones. De esta manera se decide una división del 70% de los recursos a envases falsos y el 30% restante a envases verdaderos, que son balanceados a valores próximos a 50% cada uno con procesos posteriores.

8.4.2 División del dataset

Se divide el total del dataset en 3 sectores diferentes: El dataset de entrenamiento (el cual la red usará para entrenarse) equivalente al 80% total de los datos, el dataset de validación (el cual se usará para validar la predicción real en vivo de la red y nunca será visto por la misma) equivalente al 10% de los datos y el dataset de pruebas (el cual se usará para medir la precisión final de la red) equivalente al último 10% de la red. La cantidad final de imágenes se verá reflejada en esta división.

8.4.3 Recolección del dataset

Para maximizar la precisión del modelo se llevan a cabo procesos para aumentar la cantidad de ejemplos del dataset. Se obtiene un monto total de 100 imágenes (que serán 80 en entrenamiento dada la división expuesta anteriormente) las cuales no son suficientes en un correcto proceso de entrenamiento. Por esta razón, se decide aplicar una serie de técnicas para la obtención de imágenes a convertir en dataset.

Una imagen es una matriz de píxeles con valores RGB en cada coordenada, por lo que un cambio en la posición de la imagen, su contexto (o fondo), su ángulo, su enfoque o su zoom generan una matriz con valores nuevos; no importa que se use un mismo ejemplo, si alguna de las características de su entorno cambia, la red la verá como un ejemplo diferente.

Lo anterior no puede ser abusado dado que puede resultar en un efecto de sobrealimentación en el entrenamiento y aprender el patrón de las imágenes en vez de sus rasgos individuales. Se usan 2 técnicas para aumentar el tamaño original del dataset: Múltiples iteraciones sobre los mismos ejemplos y cambios de escenarios.

8.4.4 Técnica de iteración

Considerando la cantidad máxima de ejemplos disponibles (100), se opta por duplicar la cantidad de imágenes tomadas. Cada envase es capturado en 2 ocasiones diferentes con leves cambios en su entorno. Un ejemplo de un mismo envase en 2 contextos diferentes es el siguiente:

Imagen 16: Primer captura de envase 1

imagen 17: Segunda captura de envase 1

El envase es el mismo pero con leves cambios reflejados en la inclinación de la imagen, su brillo y zoom. Aplicando esta técnica se logra duplicar el total del dataset.

8.4.5 Técnica de cambio de escenario

Esta técnica duplica nuevamente la cantidad total del dataset. Se usan los mismos envases iniciales, con un cambio abrupto de fondo. El primer fondo es blanco en su mayoría y se ha presentado en imágenes anteriores, el segundo fondo es gris para reforzar la percepción del envase como el objeto a evaluar en la red. Se expone el envase anterior con sus respectivas fotografías en el fondo gris:

imagen 18: Tercera captura de envase 1

imagen 19: Cuarta captura de envase 1

De esta manera se cuadruplica el valor inicial de imágenes por cada envase sin poner en riesgo la integridad de la red.

8.4.6 Cantidad total de ejemplos del dataset

Con la técnica de iteración y cambio de escenario se puede duplicar el total de imágenes, resultando en 400 ejemplos a entrenar, sin embargo, la distribución de dichos ejemplos sería de 280 ejemplos falsos (70% del dataset) y 120 ejemplos verdaderos (30% restante) lo cual generará a largo plazo una inconsistencia en el entrenamiento de la red; por esta razón se debe reiterar en los ejemplos verdaderos otra ronda de toma de fotos con sus respectivas técnicas para regular la cantidad total.

Con los datos calibrados, se obtiene un total de 280 ejemplos falsos y 280 ejemplos verdaderos, resultando en 560 imágenes a entrenar.

Una vez aplicada la distribución de datos en los diferentes subconjuntos, se genera el valor final para cada dataset:

- 448 imágenes de entrenamiento (224 verdaderas y 224 falsas) resultado de ser el 80% del valor total de imágenes disponibles.

- 56 imágenes de validación (28 verdaderas y 28 falsas) resultado de ser el 10% del valor total de imágenes disponibles.

- 56 imágenes de prueba (28 verdaderas y 28 falsas) resultado de ser el 10% del valor total de imágenes disponibles.

8.4.7 Cantidad de datasets

El proceso anterior se aplica respectivamente a cada uno de los 3 filtros (frontal, lateral y superior) terminando en 3 datasets con 560 imágenes cada uno, es decir, 1680 imágenes en total a partir de los 100 envases iniciales.

8.4.8 Jerarquía del dataset

Se guardan todas las imágenes separadas en sus respectivos filtros las cuales poseen una estructura requerida por keras para ser leídas por su módulo de "Image Data Generator".

El primer nivel de la estructura se refleja en cada uno de los filtros anteriores (frontal, lateral y superior) de la siguiente manera:

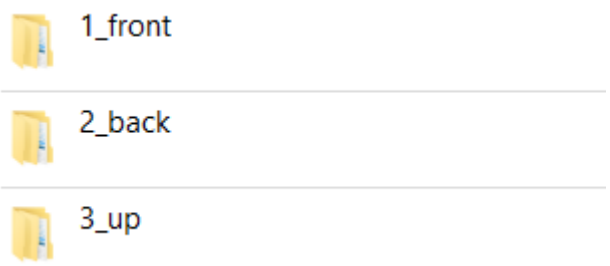

Imagen 20: Primer nivel de la jerarquía del dataset

El contenido interno de cada filtro es el mismo, por lo que se explora arbitrariamente el filtro frontal.

En el segundo nivel de jerarquía se separan los datos en las subdivisiones respectivas de train/validation/test (respectivo a entrenamiento, validación y pruebas) de la siguiente manera:

Imagen 21: Segundo nivel de la jerarquía del dataset

El contenido interno de cada dataset es el mismo, por lo que se explora arbitrariamente el de entrenamiento.

El tercer nivel respecta a los ejemplos verdaderos y falsos, se ven de la siguiente manera:

Imagen 22: Tercer nivel de la jerarquía del dataset

El cuarto y último nivel de la jerarquía se refiere a todos los ejemplos verdaderos y falsos. Se etiquetan con un número como identificador en caso de requerir ser reconocidos, se ve de la siguiente manera:

Imagen 23: Cuarto nivel de la jerarquía del dataset, ejemplos verdaderos

Imagen 24: Cuarto nivel de la jerarquía del dataset, ejemplos falsos

La jerarquía general del dataset tiene la siguiente estructura:

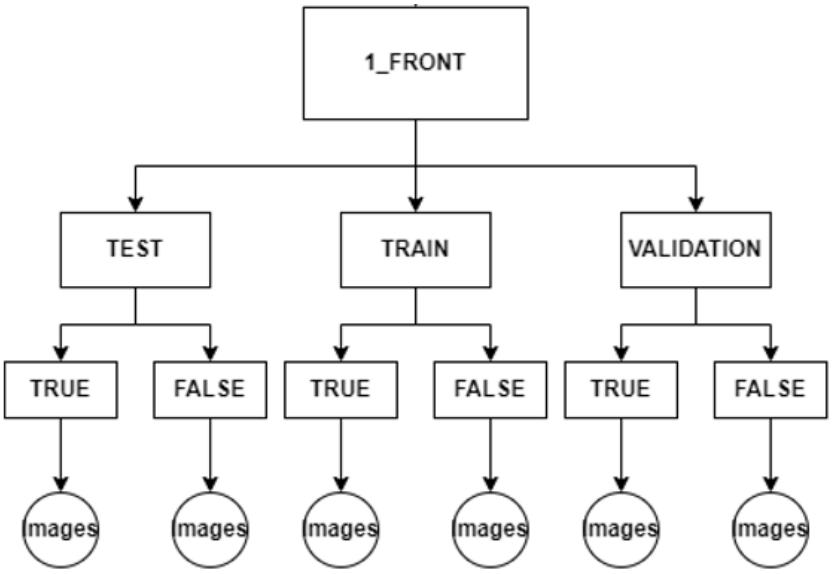

Imagen 25: Jerarquía general del dataset

8.5 Entrenamiento del modelo

Para el proceso de entrenamiento de la red se llevan a cabo procesos de configuración del dataset, creación de nuevas imágenes y control de calidad para maximizar su rendimiento.

8.5.1 Manipulación de imágenes

Las características de cada imagen son de 4000 x 3000 pixeles, sin embargo, dichas dimensiones son robustas y ralentizarán el proceso de convergencia de la red, por lo que una vez dentro de la red, se dimensionan a 256 x 180 píxeles manteniendo la escala rectangular y reduciendo su complejidad de procesamiento.

Una vez reducidas sus dimensiones, se normalizan sus valores RGB los cuales están dados en un rango de 0 a 255. Se dividen las 3 matrices de colores entre 255 para transformar su rango de valores entre 0 y 1, de esta manera la convergencia de la red será más eficiente.

Finalmente, se agrupan en lotes de 4 imágenes para aplicar la técnica de aprendizaje por lotes, la cual reduce el tiempo de entrenamiento en el tamaño del lote pero puede reducir su precisión, esto se compensa mediante data augmentation.

8.5.2 Data Augmentation

Durante la recopilación del dataset se aplicaron varias técnicas para aumentar el tamaño base del dataset de manera orgánica; ahora se aplica la técnica de data augmentation para incrementar exponencialmente el tamaño del dataset de manera artificial durante el entrenamiento.

Se crean una serie de nuevas imágenes en memoria basadas en las originales con una serie de leves cambios:

- Reescalado: Proceso explicado previamente que transforma los valores de la matriz RGB al rango entre 0 y 1. Se aplica en todos los casos.

- Rango de rotación: Rota la imagen arbitrariamente en cualquier dirección, se configura esta variable en un máximo de 20 grados.

- Centro de la imagen: Desplaza horizontal o verticalmente la imagen, se configura esta variable en un máximo de 20% de desplazamiento.

- Rango de inclinación: Inclina la imagen hacia adelante o atrás mediante una operación de ángulos en cada valor de la matriz RGB, se configura esta variable en un máximo de 20% de inclinación.

Se genera una nueva serie de imágenes en cada iteración del entrenamiento diferentes a las originales; nunca serán repetidas en su totalidad y se generan por cada época de entrenamiento, es decir, si se realizan 10 épocas de entrenamiento, la red verá 10 veces sus datos a entrenar.

Lo anterior no implica que la red vería 4480 casos en 10 épocas dado que se agrupan en conjuntos de 4, por lo que el cálculo de imágenes diferentes es el siguiente:

$$Imágenes\ totales\ =\ (dataset\ inicial\ /\ tamaño\ del\ lote)\ *\ épocas$$

$$Imágenes\ totales\ =\ (448\ /\ 4\ )\ *\ 10$$

$$Imágenes\ totales\ =\ 1120$$

De esta forma se incrementan exponencialmente las imágenes de entrenamiento.

8.5.3 Controles de calidad

Se configura un monitor automático durante el entrenamiento que verifica el rendimiento de la red durante cada época de entrenamiento. La red calcula diferentes parámetros de rendimiento, como la pérdida de la función, la precisión durante el entrenamiento respecto a los datos entrenados y datos no ingresados a la red., Los últimos 2 mencionados anteriormente son reconocidos como “accuracy” y “val_accuracy”.

El monitor verifica durante cada época el rendimiento de la red frente al val_accuracy y guarda una instancia de la red en disco si se obtiene un mejor valor al anterior. El primer valor siempre es representativo (menos infinito), por lo que siempre guarda la primera iteración, y a partir de ella, se sobreescribe si se obtiene un pico de aprendizaje, de esta manera se evita sobre enseñar a la red y perder los mejores resultados.

El proceso anterior se conoce como “Checkpoint” y guarda los mejores resultados para ser usados posteriormente durante la etapa de pruebas.

8.5.4 Compilación y entrenamiento de la red

El modelo se compila con la siguiente configuración:

- Función de pérdida - Binary Cross-Entropy: Se elige esta función de pérdida dado que representa correctamente el modelo de clasificación binaria sin mayores complicaciones; permite monitorear el rendimiento de la red en casos de sobrealimentación o falta de datos.

- Optimizador - RMSProp : Se opta por RMSProp como optimizador dado que configura otros hiper parámetros no mencionados anteriormente de manera automática.

- Métricas de éxito - Accuracy: Se mide el éxito de la red mediante su precisión de entrenamiento. Esto es un valor referencial, pero su verdadero éxito se determina en la precisión sobre el dataset de validación y pruebas.

Una vez compilado se inicia el entrenamiento con la totalidad de parámetros e hiper parámetros presentados anteriormente; se elige como la cantidad de épocas a entrenar 70 como un valor arbitrario buscando la mejor época en este rango.

8.6 Evaluación del modelo

La evaluación del modelo se da de manera individual y conjunta, donde se mide el rendimiento individual de la red mediante los resultados obtenidos en los datasets de validación y prueba, mientras que la evaluación conjunta se da por el cálculo estadístico del rendimiento de los 3 filtros a la vez.

Dado que se entrenan una red para cada filtro respectivamente, se obtienen 3 predicciones, donde la red determina la probabilidad de que el envase sea íntegro o no; si la probabilidad es mayor o igual a 50% se considera verdadera la predicción, si es menor, la predicción es negativa.

Al obtener los 3 resultados individuales, se procesan a la vez en una compuerta AND la cual determina la predicción final.

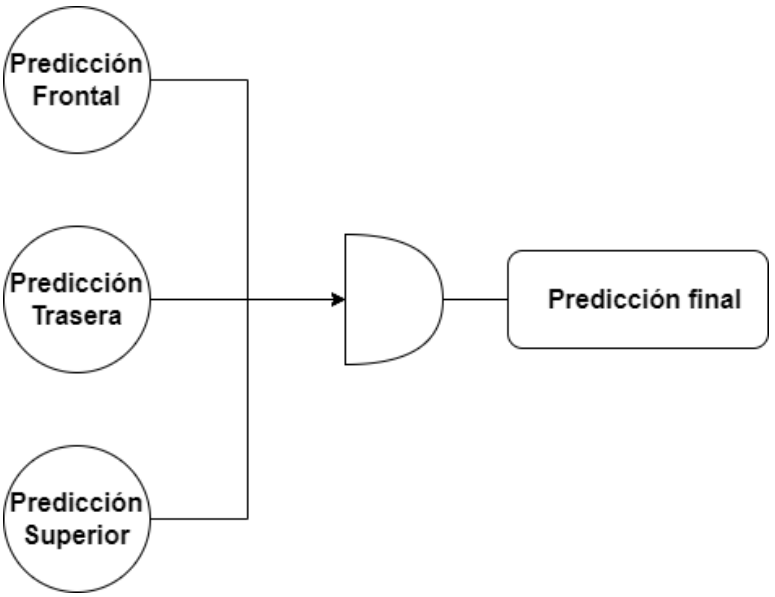

Imagen 26: Predicción final de la red

Lo anterior implica que si alguno de los filtros resulta negativo, se retorna como falsa la predicción.

El desempeño de la predicción final de la red es la multiplicación del rendimiento de sus predicciones individuales.

9. RESULTADOS

Para exponer el rendimiento de la red y su calidad de predicción, se exponen los resultados de entrenamiento individual

9.1 Resultados del entrenamiento

Después de 70 épocas de entrenamiento, se obtiene el resultado de entrenamiento de cada red.

El eje X corresponde a la cantidad de épocas mientras que el eje Y corresponde a la precisión. Se compara el rendimiento de los datos de entrenamiento (representados por la línea azul) con los datos de validación (representados por la línea naranja).

El crecimiento de la línea de entrenamiento es relativamente suave dado que posee suficientes datos para crecer de manera controlada mientras que la línea de validación es abrupta dado que el cambio de pesos por iteración es agresivo y se evalúa sobre datos no entrenados.

9.1.1 Resultados de entrenamiento de filtro frontal

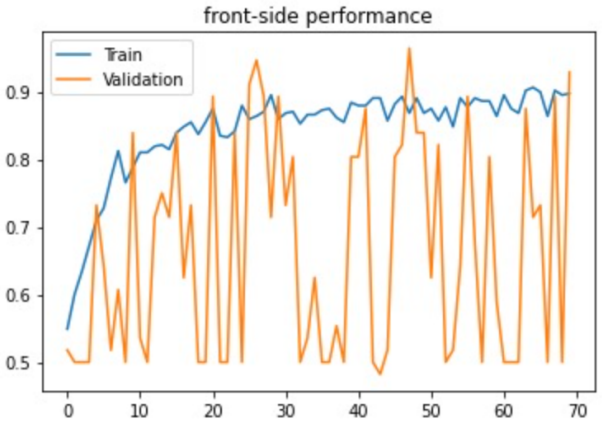

Imagen 27: Rendimiento de entrenamiento del filtro frontal

El filtro frontal se estabiliza entre 80 y 90% de precisión en datos de entrenamiento mientras que los datos de validación iteran de manera agresiva. La mejor época para este filtro es la época 49, donde su predicción en validación es de 96.43% mientras que su predicción en entrenamiento es de 89.06%

```
Epoch 00048: val_accuracy improved from 0.94643 to 0.96429, saving model to tarros_1.hdf5
Epoch 49/70
112/112 [==============================] - 94s 838ms/step - loss: 0.3266 - accuracy: 0.8906
```

Imagen 28: Época óptima del filtro frontal.

9.1.2 Resultados de entrenamiento de filtro trasero

Imagen 29: Rendimiento de entrenamiento del filtro trasero

El filtro trasero se estabiliza entre 80 y 90% de precisión en datos de entrenamiento mientras que los datos de validación iteran de manera agresiva. La mejor época para este filtro es la época 32, donde su predicción en validación es de 96.43% mientras que su predicción en entrenamiento es de 87.28%

```
Epoch 00031: val_accuracy improved from 0.87500 to 0.96429, saving model to tarros_2.hdf5
Epoch 32/70
112/112 [==============================] - 89s 799ms/step - loss: 0.3346 - accuracy: 0.8728
```

Imagen 30: Época óptima del filtro trasero

Se produce el mismo valor de predicción en validación en el filtro frontal y lateral con leves cambios en la predicción de entrenamiento.

9.1.3 Resultados de entrenamiento de filtro superior

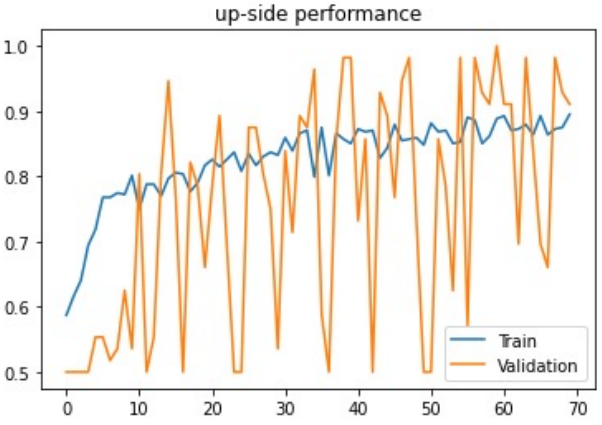

Imagen 31: Rendimiento de entrenamiento del filtro superior

El filtro trasero tarda en estabilizarse hasta la época 40 donde sus valores oscilan entre 80 y 90% en datos de entrenamiento mientras que los datos de validación iteran de manera agresiva. La mejor época para este filtro es la época 61, donde su predicción en validación es absoluta, es decir 100%, mientras que su predicción en entrenamiento es de 89.29%

```
Epoch 00060: val_accuracy improved from 0.98214 to 1.00000, saving model to tarros_3.hdf5
Epoch 61/70
112/112 [==============================] - 108s 964ms/step - loss: 0.3002 - accuracy: 0.8929
```

Imagen 32: Época óptima del filtro superior

9.2 Resultados de pruebas

Se somete a los diferentes filtros en su mejor época a el dataset de pruebas, el cual no ha sido usado anteriormente, de esta manera se puede obtener el rendimiento real de la red al promediar los resultados de validación y prueba.

9.2.1 Pruebas al filtro frontal

Se somete al filtro frontal el dataset de pruebas con los siguientes resultados:

```
test_model()

Found 56 images belonging to 2 classes.
56/56 [==============================] - 9s 153ms/step - loss: 0.2928 - accuracy: 0.8929
```

Imagen 33: Rendimiento del filtro frontal en el dataset de pruebas

El rendimiento frente al dataset de pruebas es de 89.29%

9.2.2 Pruebas al filtro trasero

Se somete al filtro trasero el dataset de pruebas con los siguientes resultados:

```
test_model()

Found 56 images belonging to 2 classes.
56/56 [==============================] - 10s 173ms/step - loss: 0.2462 - accuracy: 0.9464
```

Imagen 34: Rendimiento del filtro trasero en el dataset de pruebas

El rendimiento frente al dataset de pruebas es de 94.64%

9.2.3 Pruebas al filtro superior

Se somete al filtro superior el dataset de pruebas con los siguientes resultados:

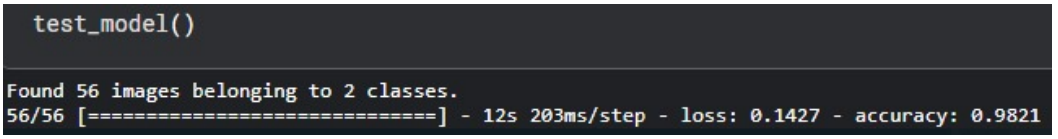

Imagen 35: Rendimiento del filtro superior en el dataset de pruebas

El rendimiento frente al dataset de pruebas es de 98.21%

9.3 Rendimiento total del modelo

Para determinar el rendimiento total del modelo, se obtiene el rendimiento real de cada filtro promediando sus resultados en el dataset de validación y pruebas.

$$Rendimiento\ real\ =\ (rendimiento\ en\ validación\ +\ rendimiento\ en\ pruebas)\ /\ 2$$

Para el filtro frontal:

$$Rendimiento\ de\ validación\ =\ 96.43\%$$

$$Rendimiento\ en\ pruebas\ =\ 89.29\%$$

$$Rendimiento\ total\ =\ (0.9643\ +\ 0.8929)\ /\ 2$$

$$Rendimiento\ total\ =\ 92.86\%$$

Para el filtro trasero:

$$Rendimiento\ de\ validación\ =\ 96.43\%$$

$$Rendimiento\ en\ pruebas\ =\ 94.64\%$$

$$Rendimiento\ total\ =\ (0.9643\ +\ 0.9464)\ /\ 2$$

$$Rendimiento\ total\ =\ 95.53\%$$

Para el filtro superior:

$$Rendimiento\ de\ validación\ =\ 100.0\%$$

$$Rendimiento\ en\ pruebas\ 98.21\%$$

$$Rendimiento\ total\ =\ (1.0\ +\ 0.9821)\ /\ 2$$

$$Rendimiento\ total\ =\ 99.10\%$$

El rendimiento total de la red se mide por la multiplicación de los rendimientos, de esta manera se conoce la predicción conjunta.

$$Rendimiento\ global\ =\ Rendimiento\ frontal\ *\ Rendimiento\ lateral\ *\ Rendimiento\ superior$$

$$Rendimiento\ global\ =\ 0.9286\ *\ 0.9553\ *\ 0.9910$$

$$Rendimiento\ global\ =\ 87.91\%$$

La precisión total de la red es de 87.91%

9.4 Pruebas individuales de la red

Se desarrolla un aplicativo web para probar diferentes imágenes de los 3 filtros principales las cuales retornan la predicción individual y grupal.

El aplicativo permite cargar imágenes desde el dispositivo y las procesa mediante el botón de verificar:

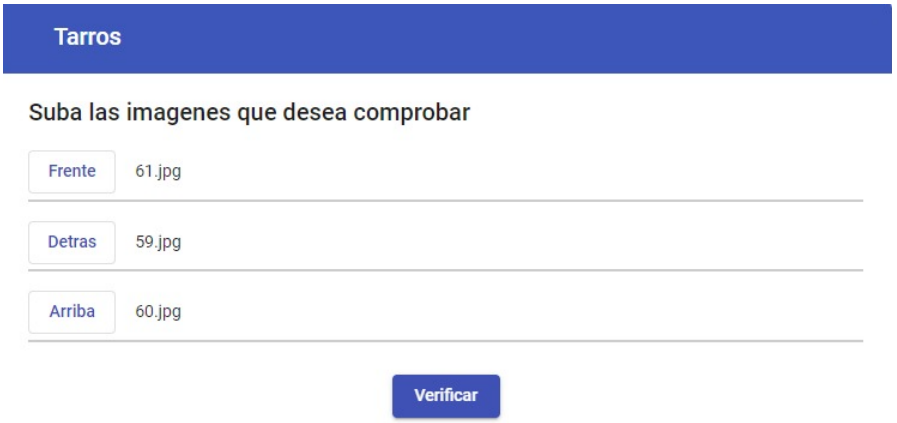

Imagen 36: Cargar y verificar envases

Se procesa cad imagen de manera individual y se obtienen las predicciones individuales y un diagnóstico general:

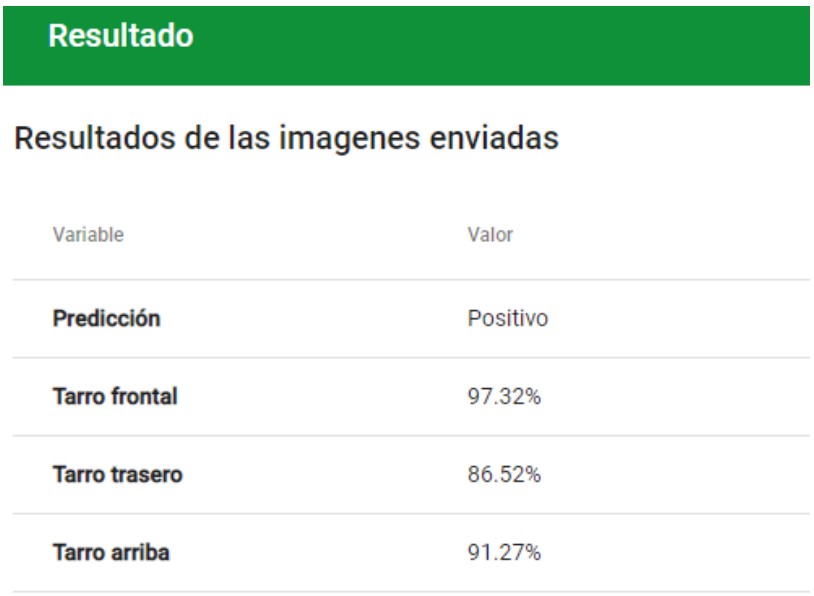

Imagen 37: Primer ejemplo de resultado de la predicción

Si alguna de las imágenes retorna una predicción falsa, el diagnóstico general es falso:

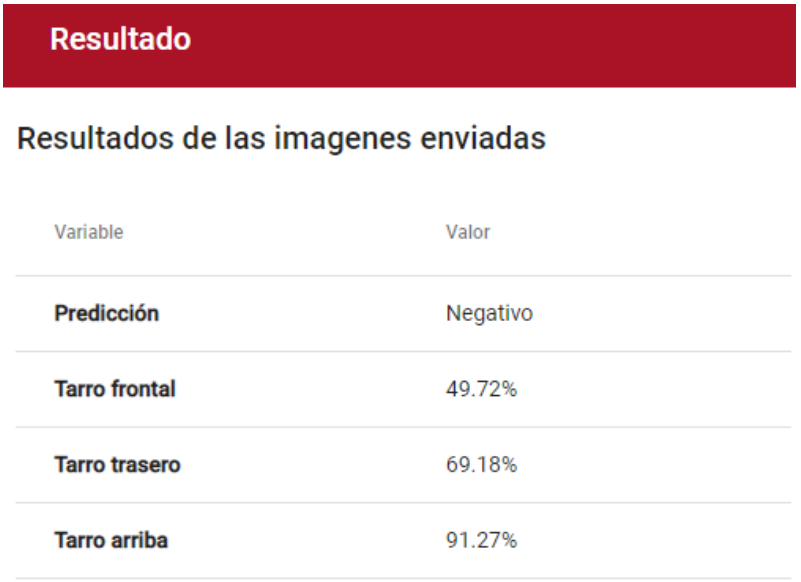

Imagen 38: Segundo ejemplo de resultado de predicción

10. TRABAJO FUTURO

Se contempla la mejora continua del modelo entrenado mediante la iteración de nuevas técnicas referentes al Deep Learning, como uso de nuevas funciones de activación, modificaciones al procesamiento de imágenes y controles de calidad.

Otra posibilidad a indagar en el mediano plazo es el incremento del dataset inicial, dado que su cantidad limitada tiende a ralentizar el aprendizaje de la red, limitando en algunas ocasiones los casos a tener en cuenta. Se propone la obtención de nuevos envases de diferentes características en términos de forma y escala de colores, de esta manera se podría generalizar el sistema.

11. BIBLIOGRAFÍA

[1] *Tecnología | Diccionario de la lengua española.* (s. f.). «Diccionario de la lengua española» - Edición del Tricentenario. https://dle.rae.es/tecnolog%C3%ADa

[2] McCain, A. (2022, 21 abril). *How Fast Is Technology Advancing? [2022]: Growing, Evolving, And Accelerating At Exponential Rates*. Zippia. https://www.zippia.com/advice/how-fast-is-technology-advancing/#:%7E:text=How%20Fast%20Is%20Technology%20Advancing%3A%20Trends%20and%20Predictions,expected%20to%20reach%2050%20billion.

[3] Allen, J. R, West, D. M. (2022, 9 marzo). *How artificial intelligence is transforming the world*. Brookings. https://www.brookings.edu/research/how-artificial-intelligence-is-transforming-the-world/

[4] Chan, M. (2021, 13 diciembre). Convolutional Neural Networks: Why are they so good for image related learning? Medium. https://towardsdatascience.com/convolutional-neural-networks-why-are-they-so-good-for-image-related-learning-2b202a25d757

[5] Staar, B., Lütjen, M., & Freitag, M. (2019). Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP*, *79*, 484–489. https://doi.org/10.1016/j.procir.2019.02.123

[6] Skjelvareid, M. H. (2019, 18 junio). *DAGM 2007 competition dataset*. Kaggle. https://www.kaggle.com/datasets/mhskjelvareid/dagm-2007-competition-dataset-optical-inspection

[7] Zhang, M., Wu, J., Lin, H., Yuan, P., & Song, Y. (2017). The Application of One-Class Classifier Based on CNN in Image Defect Detection. *Procedia Computer Science*, *114*, 341–348. https://doi.org/10.1016/j.procs.2017.09.040

[8] Dong, X., Taylor, C. J., & Cootes, T. F. (2019). Small Defect Detection Using Convolutional Neural Network Features and Random Forests. *Lecture Notes in Computer Science*, 398–412. https://doi.org/10.1007/978-3-030-11018-5_35

[9] Wigington, C., Stewart, S., Davis, B., Barrett, B., Price, B., & Cohen, S. (2017). Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. https://doi.org/10.1109/icdar.2017.110

[10] Smith, L. N., & Topin, N. (2016). Deep convolutional neural network design patterns. *arXiv preprint arXiv:1611.00847*.

[11] Anyoha, R. (2020, 23 abril). *The History of Artificial Intelligence*. Science in the News. https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/

[12] IBM. (2021, 5 noviembre). *Machine Learning*. https://www.ibm.com/cloud/learn/machine-learning#:%7E:text=Machine%20learning%20is%20a%20branch,rich%20history%20with%20machine%20learning.

[13] Sahay, M. (2021, 14 diciembre). *Neural Networks and the Universal Approximation Theorem*. Medium. https://towardsdatascience.com/neural-networks-and-the-universal-approximation-theorem-8a389a33d30a

12. CÓDIGO FUENTE

El código fuente del proyecto se aloja en TheFrancho/projecto-envases:(github.com)

El dataset trabajado se encuentra en Tarros Dataset (kaggle.com)