Redes neuronales convolucionales para clasificación de componentes electrónicos

Convolutional neural networks for electronic component classification

César Armando Morales Molina
Universidad de Sonsonate
cmorales@usonsonate.edu.sv
https://orcid.org/0009-0000-6716-1392

Colaboradores
Ernesto Antonio Morales Rodríguez
Marvin Adonay Alarcón Segura
Rodrigo Alejandro Centeno Flores

Resumen

Este artículo presenta un enfoque basado en redes neuronales convolucionales (CNN) para la clasificación de componentes electrónicos. El objetivo es desarrollar un sistema inteligente que sea capaz de identificar componentes electrónicos en entornos diversos, lo cual es una tarea que, en aplicaciones de educación y control de inventario en la industria electrónica, puede beneficiar a la eficiencia de los procesos. El conjunto de datos es creado con imágenes de internet y propias, para buscar un mejor resultado al entrenar y evaluar el modelo propuesto, comparando también con la red preentrenada MobileNetV2. Los resultados muestran que las redes neuronales artificiales lograron una alta precisión en la detección de componentes electrónicos utilizando la técnica de transferencia de aprendizaje.

Palabras clave: Componentes electrónicos, redes neuronales convolucionales, redes neuronales artificiales, visión por computadora, MobileNetV2

Abstract

This article presents an approach based on convolutional neural networks (CNN) for classifying electronic components. The objective is to develop an intelligent system capable of identif-

ying electronic components in diverse environments. This task can benefit process efficiency in education and inventory control applications in the electronics industry. The dataset was created using both internet and proper images to improve the training and evaluation of the proposed model. The results were then compared to those of the pre-trained network MobileNetV2. The findings indicate that the artificial neural networks achieved high accuracy in detecting electronic components by using the transfer learning technique.

Keywords: Electronic components, convolutional neural networks, artificial neural networks, computer vision, MobileNetV2

Introducción

La electrónica se ha convertido en una disciplina fundamental en la sociedad, desempeñando un papel vital en, prácticamente, todos los aspectos de la vida cotidiana. Esta rama de la ciencia no solo se centra en productos comerciales, también es referente para impulsar el progreso científico, mejorar la calidad de vida y abre un abanico de posibilidades en diversos campos de la industria e investigación. Según Coluccio Leskow (2021), la importancia del estudio de la electrónica ya ha sido planteada, por eso se ha convertido en una parte integral de la humanidad.

En el ámbito industrial, la clasificación y reconocimiento de imágenes se ha vuelto esencial para optimizar la gestión y el control de los recursos disponibles. Se ha explorado el uso de técnicas de aprendizaje automático, como las redes neuronales convolucionales o Convolutional Neural Networks (CNN), que han demostrado un gran éxito en la clasificación de imágenes en una amplia gama de aplicaciones.

Las redes neuronales artificiales imitan la estructura del sistema nervioso. Al comprender que estos sistemas pueden llegar a tener la capacidad humana para aprender y tomar decisiones, su uso, en diversos ámbitos, continúa experimentando un crecimiento significativo, por lo que se llevan a cabo investigaciones constantes con el objetivo de optimizar su aprovechamiento (Pajares y de la Cruz García, 2007).

Para los procesos más especializados, como el de clasificación de imágenes, se desarrollaron las Redes Neuronales Convolucionales. Estas son un tipo de red neuronal artificial inspirada en la organización del sistema visual de los humanos. Su capacidad para aprender patrones y características relevantes en imágenes, ha revolucionado el campo del reconocimiento de imágenes y la visión por computadora. El funcionamiento de estas redes depende de las capas convolucionales, ya que son las encargadas de detectar y extraer, mediante filtros matriciales, características primarias de una imagen; luego de su extracción, almacenan cada una de las propiedades detectadas para realizar un entrenamiento de clasificación mediante las capas completamente conectadas; estas se encargan de realizar la clasificación final (Izquierdo Dussan y Puerto Rodríguez, s.f.).

El objetivo principal de este artículo es presentar un enfoque innovador basado en la aplicación de redes neuronales convolucionales,

en la clasificación de imágenes de equipos electrónicos. Esto, pensado para automatizar el proceso de clasificación y, en un futuro, servir de apoyo para cualquier actividad relacionada con la electrónica. El proceso de creación de la red y del dataset se hará en el lenguaje de programación Python, en conjunto con la librería TensorFlow y Keras.

Fundamentación

Dataset

El conjunto de datos, también conocido como dataset, se compone de dos grupos principales que corresponde a entrenamiento y validación, necesarios para realizar el entrenamiento de la red neuronal convolucional, así como también uno opcional, como es el de prueba, encargado de realizar la confirmación con la red entrenada. Para conseguir un resultado óptimo, se mantiene un estándar mínimo en la proporcionalidad que debe poseer cada subgrupo que es entre 70 % a entrenamiento y 30 % a validación.

En la clasificación de componentes electrónicos, se tiene un problema de aprendizaje supervisado con 26 clases. En ellas, no solo se incluyen elementos básicos como resistencias, leds, capacitores o inductores, también se desea ampliar más la aplicación del proyecto, por lo que se decide incluir microcontroladores, sensores y herramientas de trabajo.

Alegre et al. (2016) explica una etapa fundamental para el desarrollo de un clasificador. El conjunto de datos con el que se va a realizar el aprendizaje debe ser representativo de las condiciones en las que va a operar el clasificador. Teniendo eso en cuenta, se decide crear el dataset utilizando imágenes de internet y también fotografías propias en el entorno donde se pondrá en ejecución.

Se designa la metodología de trabajo anterior para la creación del dataset, sabiendo que, en un problema de reconocimiento de patrones, cada muestra se identifica por un conjunto de características representadas por un vector n-dimensional (x= [x1, x1, x3,…,xn]); estas características pueden ser cuantitativas o cualitativas. La red podrá analizar características también del entorno donde se pondrá en ejecución el sistema y así poder omitirlas al compararlas con las imágenes de internet con fondo liso.

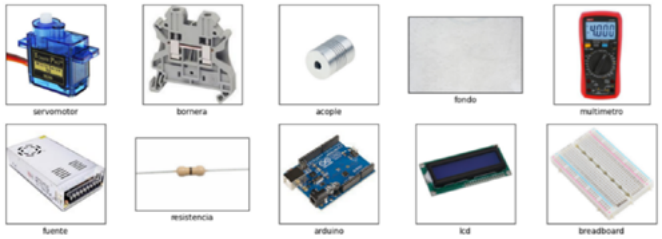
Al total de clases a categorizar se sumó una, llamada fondo, la cual consta de diferentes entornos ya sean de un solo color o con objetos varios que no correspondan a ningún objeto que se desea analizar; esta se usa por si la imagen a clasificar no se detecta en ninguna de las categorías, así no habrá errores de confusión para usuarios. La red, al no encontrar la categoría de un objeto, intenta conectar sus similitudes con la clase de objeto que comparta más atributos, por lo que se recomienda el uso de una categoría extra para mitigar el error.

En la Figura 1 se observan imágenes de 10 componentes respectivos a una categoría que se utilizará en el modelo. Para el proyecto se determinó que cada categoría tuviera un total de 70 imágenes, esto para tener un rango

amplio de trabajo de entrenamiento y validación, dando un total de 1,820 elementos, divididos en entrenamiento y validación a un 75 % y 25 %, respectivamente.

Figura 1

Categorías de objetos a clasificar



Incremento de datos

Las redes usan una gran cantidad de datos para aprender; para tener mejores resultados se realiza un incremento de datos con la ayuda de las librerías TensorFlow y Keras, implementadas en Python. Estas permiten utilizar diferentes opciones como rotar, estirar, recortar, acercar y más herramientas, para manipular las imágenes del dataset original y obtener nuevas versiones de las imágenes.

Figura 2

Incremento de datos de imágenes

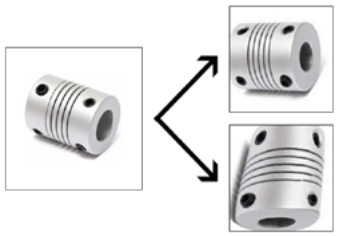


Imagen original | Aumento de datos

El aumento de datos es utilizado para la reducción de sobreajuste y cuando no se dispone con los datos suficientes para un buen entrenamiento. En la Figura 2 se observa cómo se producen alteraciones de la imagen original, modificando su rotación, tamaño y estiramiento para conseguir nuevas formas; este tratamiento de imágenes da la posibilidad de aumentar la cantidad de datos con una imagen repetida para mejorar el entrenamiento de la red.

Se realizó un aumento de datos con las imágenes recopiladas de los componentes, ya que muchas son muy similares y eso provoca un sobreajuste y el sistema se quedaría con una cantidad reducida de características obtenidas.

Clasificación de imágenes

Análisis de imágenes

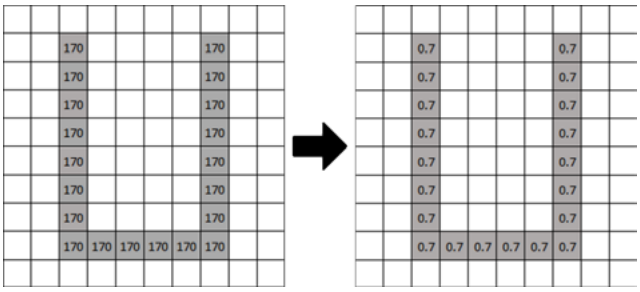
Para comenzar con el análisis de redes neuronales convolucionales se debe saber qué es una imagen y cómo es que se obtienen características. Las imágenes son matrices de píxeles con rangos entre 0 a 255, donde 0 representa el color negro y el 255 es el color blanco. Cuando decimos que las imágenes solo tienen un canal, se considera que son en escala de grises.

Las fotografías trabajan bajo el concepto de la síntesis aditiva del color, la cual muestra los colores primarios basados en la luz y su espectro de onda. Estos colores ideales son el rojo, verde y azul, conocidos también como RGB. Teniendo en cuenta el concepto de color y los rangos en los que pueden ser representados

en un ordenador, la fusión de los 3 colores, o también llamados canales, en diferentes escalas y combinaciones, pueden dar todos los colores existentes.

Figura 3

Normalización de píxeles de una imagen



Cuando se trabaja con redes neuronales convolucionales, lo más recomendado es realizar una normalización de los valores de una imagen para obtener un nuevo rango de 0 a 1, esto se observa en la Figura 3; para realizar la transformación de cada pixel se utiliza

$$x_{normalizada} = \frac{x}{x_{\max} - x_{\min}} \quad 2),$$

resultando un valor con la condición buscada.

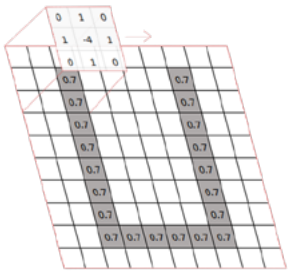
Convolución

La red neuronal necesita datos para poder realizar cálculos y dar una predicción conforme a su etiqueta de salida, por lo que la red toma como entrada los píxeles de una imagen para realizar un análisis y formar un diccionario de características. Si se tiene una imagen de 128x128 píxeles, en escala de grises se tiene una cantidad de neuronas de entrada de 16,384; si se quisiera analizar otra imagen con la misma cantidad de píxeles de alto y ancho, pero con los 3 canales, se tendrían una cantidad de 49,152 neuronas de entrada. El factor de los canales es muy importante, ya que al tener más neuronas de entrada conlleva a un mayor trabajo de procesamiento.

Las capas convolucionales son exclusivas para el análisis de imágenes. La convolución consiste en analizar grupos de píxeles cercanos delimitados a una dimensión dada por una matriz llamada kernel, la dimensión de esta herramienta debe ser inferior a la imagen original, por lo general no excede un tamaño de 5x5. El kernel, también llamado filtro, es el encargado de recorrer toda la imagen e ir realizando una multiplicación escalar con los datos normalizados de la imagen como se muestra en la Figura 4.

Figura 4

Kernel 3x3 recorriendo imagen



Cuando se obtiene la matriz resultante se realiza una sumatoria de todos los elementos dado por el tamaño del kernel mxn en la función

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}$$

Estas operaciones se repiten la cantidad de veces,

$$I = (w+1-n)(h+1-m)$$

w = ancho de la imagen original

h = alto de la imagen original

donde también se representan las nuevas dimensiones de la imagen de salida que serán analizadas en la siguiente capa de la red, eso sucede por las alteraciones realizadas al ancho y alto de la imagen original por el tamaño del kernel.

En la Figura 5 se muestra el proceso de convolución realizando una limpieza de datos, ya que los kernels pueden contener números negativos para extraer diferentes características de la imagen, se deben obtener resultados que se puedan volver a analizar, datos mayores o iguales a 0; por eso se emplea la función de activación

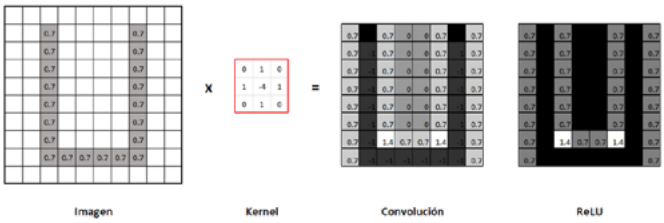
$$f(x)\begin{cases} 0, x < 0 \\ x, x > 0 \end{cases}$$

ReLU (Rectifier Linear Unit). Se pueden utilizar otras funciones, pero ReLU es la más común para las CNN.

Este proceso de convolución se repite dependiendo de la cantidad de canales de la imagen y también de la cantidad de filtros; generalmente en una capa de convolución se utilizan una cantidad considerable para extraer un mayor conjunto de características.

Figura 5

Proceso de convolución



Las convoluciones exigen un gran procesamiento computacional, por lo que una buena práctica es comenzar con una cantidad menor a 64 filtros e ir aumentando mientras más pequeña sea la imagen, ya que se realizarán un menor número de operaciones, pero se obtendrán más características.

Agrupamiento

Al terminar la extracción de características, se necesita realizar un pooling o agrupamiento; esta tarea tiene el fin de resumir los resultados obtenidos en el anterior proceso. El pooling hace una reducción de la imagen de una manera similar a la convolución; se crea una matriz cuadrada de orden n, que se encargará de recorrer la imagen y dependiendo el tipo de agrupamiento seleccionar el valor deseado.

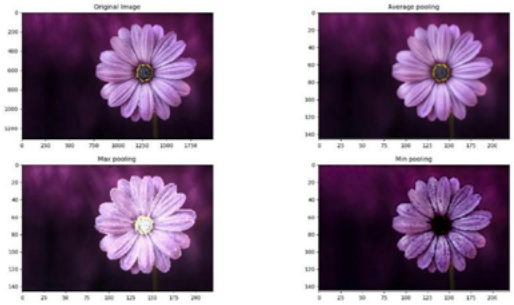
La tarea de agrupamiento es fundamental para un desarrollo óptimo de los análisis computacionales que realiza la red, reduciendo la cantidad de neuronas obtenidas en las convoluciones. Se toma como ejemplo una imagen de 128x128 a la que se realiza el tratamiento de convolución con 16 filtros de 3x3, obteniendo un total de 254,016 neuronas de salida a la siguiente convolución. Esa cifra tan elevada no

resulta factible, ya que el procesamiento será mayor y puede llegar a no permitir el trabajo normal de una computadora.

Para reducir el número de neuronas se debe reducir el tamaño de las imágenes filtradas, tomando en cuenta que las características más importantes, generadas por cada filtro, deben mantenerse. En la Figura 6 se muestran los 3 principales tipos de pooling: max pooling, min pooling y average pooling.

Figura 6

Tipos de pooling



Nota. Fuente: Basavarajaiah (2019).

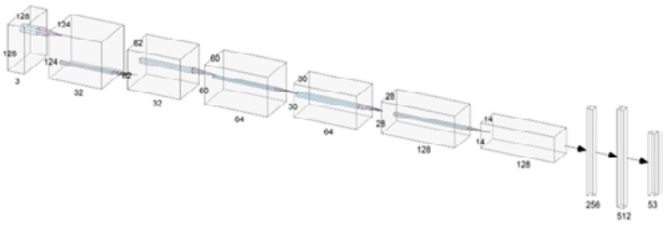
Modelo y arquitectura de la red neuronal convolucional

Se diseñó una arquitectura de red neuronal convolucional, como se muestra en la Figura 7, la cual está compuesta por 4 capas convolucionales con filtros que varían desde dimensiones 3x3 hasta 5x5 con función de activación ReLU; 3 capas de agrupamiento de tamaño 2x2, y 2 capas densas. La entrada a la red consiste en una imagen de tamaño 128x128x3; se utilizan estas dimensiones por el hecho que las imágenes a clasificar tienen muchos detalles para poder ser diferenciados de otros componentes, ya que muchos tienen bastantes similitudes.

En la primera etapa la imagen se somete a una convolución utilizando un conjunto de 32 filtros de tamaño 5x5; esta primera capa desempeña el papel de aprender las características de nivel más básico. Después de la capa convolucional se aplica la función ReLU, seguida de una capa de agrupamiento de tipo max pooling de 2x2, esto ayuda a reducir la dimensionalidad de las características a la mitad. A estas capas les siguen una serie de capas convolucionales de 3x3 y de agrupamiento de dimensiones 2x2, que continúan con la extracción de características más detalladas. Cuando se termina la extracción de características se pasan por una capa flatten, la cual reduce los datos de entrada a una sola dimensión en lugar de 2 dimensiones para posteriormente ser analizados por las capas densas.

Figura 7

Arquitectura de red neuronal convolucional propuesta



Haciendo diferentes pruebas con cinco objetos, se llega a buenos resultados con la red, pero cuando se empieza a aumentar las clasificaciones ya no resulta eficiente el modelo

propuesto, ya que con tan pocas capas no es capaz de extraer todas las características necesarias para hacer una buena predicción. Al observar la gráfica de precisión de la red, se observa que se consigue un sobreajuste, por el hecho de que la red no es capaz de extraer las suficientes características de las imágenes.

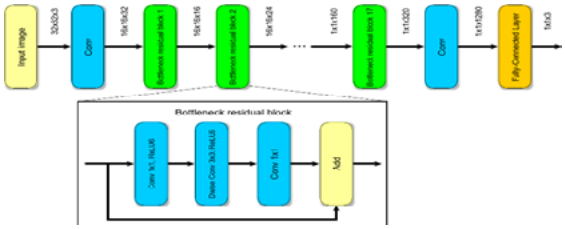
Transferencia de aprendizaje

Al incrementar el número de capas convolucionales, es notoria la necesidad de un sistema de procesamiento computacional con altas prestaciones. Para mejorar los resultados, se utilizó la técnica de transferencia de aprendizaje; esta consiste en reutilizar modelos previamente entrenados para resolver los desafíos actuales. Buscando modelos para realizar la técnica, se selecciona MobileNetV2, una red creada por Google y entrenada para realizar la clasificación de 1,000 categorías.

La arquitectura de MobileNetV2, ejemplificada en la Figura 8, muestra cómo tiene un bloque compuesto por tres convoluciones en serie, donde la convolución puntual al final del bloque, hace justo lo contrario, reduce el número de canales. La primera capa del bloque será ahora una capa de expansión. El nombre con el que se la conoce es capa de cuello de botella (bottleneck layer), ya que hace más pequeñas las imágenes que recorren la red.

Figura 8

Arquitectura de MobileNetV2. Tomado de (Seidaliyeva et al. 2020)



Estas características hacen que MobileNetV2 sea muy eficiente, y también interesante, para conocer su rendimiento con el sistema de adaptación propuesto.

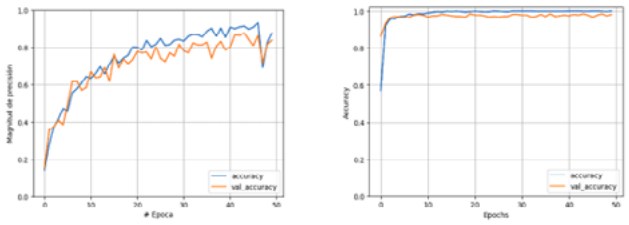
La transferencia de aprendizaje se creó para solucionar los problemas del entrenamiento y extracción de características; se sabe que las convoluciones requieren de un gran poder de cálculo computacional y, entre mayor calidad de información se desea obtener de una imagen, será más complicado para la máquina. Estos modelos entrenados son creados por grandes compañías que cuentan con equipos de alta eficiencia para realizar grandes entrenamientos con suficientes datos.

Resultados

Utilizando el modelo de MobileNetV2, se obtuvo una gran mejora con respecto a la red que se había planteado anteriormente. En las gráficas de precisión, se observa como el sobreajuste se corrige en la Figura 9, disminuyendo la oscilación de los valores.

Figura 9

Comparación de precisión de modelos



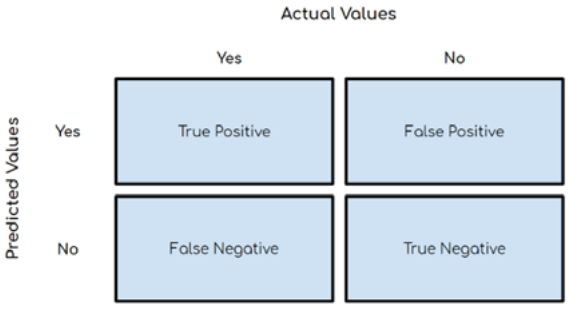
(a) CNN Propia *(b) MobileNetV2*

Matriz de confusión

La matriz de confusión se utiliza para evaluar el rendimiento de una CNN al analizar las predicciones en función de las clases reales. Permite identificar patrones de errores comunes, como falsos positivos o falsos negativos, y determinar en qué clases el modelo tiene un mejor o peor desempeño. Esto ayuda a ajustar y mejorar la arquitectura de la CNN, así como los parámetros de entrenamiento, para obtener resultados más precisos y confiables.

Figura 10

Análisis de matriz de confusión



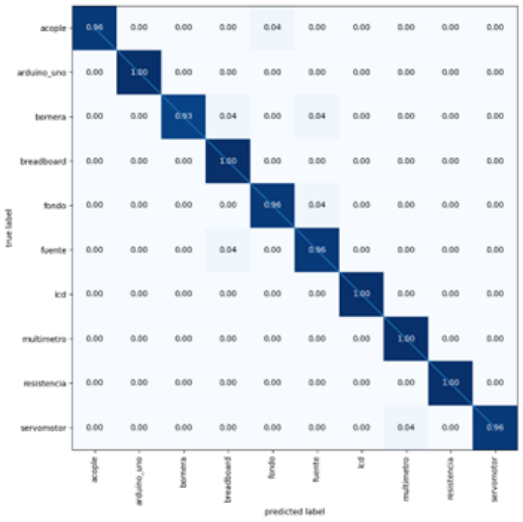
Casos posibles de una matriz de confusión (Basavarajaiah, 2019):

- Verdadero Positivo (TP): resultado en el que el modelo predice correctamente la clase positiva.
- Verdadero Negativo (TN): resultado donde el modelo predice correctamente la clase negativa.
- Falso Positivo (FP): también llamado error de tipo 1, resultado donde el modelo predice incorrectamente la clase positiva cuando en realidad es negativa.
- Falso Negativo (FN): También llamado error de tipo 2, un resultado en el que el modelo predice incorrectamente la clase negativa cuando en realidad es positiva.

Para aumentar la precisión de la red, se implementaron nuevas capas densas, de 128 neuronas cada una, a la salida del modelo MobileNetV2, para hacer un afinamiento. Este procedimiento se utiliza para que, al obtener los valores, las capas agregadas puedan mejorar las predicciones. Realizando la matriz de confusión se observa que las coincidencias son casi perfectas, como se muestra en la Figura 11.

Figura 11

Matriz de confusión utilizando modelo MobileNetV2



Conclusiones

Para diseñar una red neuronal convolucional desde cero se debe tomar en cuenta factores de hardware que se tienen para realizar el entrenamiento, ya que es uno de los mayores problemas a los que nos enfrentamos con estas tecnologías. Realizar una buena cantidad de entrenamientos y tener una red compleja que permita extraer la mayor cantidad de características de una imagen, conlleva una gran complejidad computacional; sin una tarjeta gráfica dedicada, podría llevar mucho tiempo para realizar los procesos.

Se determinó que el aumento de datos es una herramienta esencial para mejorar la calidad del dataset, ya que la complejidad de obtención de una gran cantidad de imágenes puede ser compleja.

Al no contar con un hardware adecuado, la técnica de transferencia de conocimiento es la opción más adecuada, dado que los valores para obtener características básicas de los objetos ya están inicializados y dan buenos resultados; solo es necesario realizar un pequeño ajuste para obtener resultados sobresalientes.

Futuras investigaciones

Un trabajo futuro es mejorar la red para clasificar más de 100 componentes electrónicos. Teniendo un modelo lo suficiente robusto, se pretende realizar su implementación con el desarrollo de una página web que pueda realizar la visión por computadora para mostrar las predicciones.

Referencias

Alegre, E., Pajares, G., y de la Escalera, A. (Eds.) (2016). *Conceptos y Métodos en Visión por Computador*. Grupo de Visión del Comité Español de Automática. España.

Basavarajaiah, M. (8 de febrero de 2019). Maxpooling vs minpooling vs average pooling. *Medium*. Recuperado el junio de 2023, de https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9

Bradski, G., y Kaehler, A. (2008). *Learning OpenCV*: Computer Vision with the OpenCV Library. O'Reilly.

Bermejo Peláez, D., San José Estépar, R., y Ledesma-Carbayo, M. (2016). Detección y clasificación de enfisema pulmonar en imágenes de TAC mediante Redes Neuronales Convolucionales Multiescala. En: *XXXIV Congreso Anual de la Sociedad Española de Ingeniería Biomédica, 23-25 de noviembre de 2016*. Valencia, España.

Caparrini, F. S. (2020). Cursos: Inteligencia Artificial. Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Sevilla. Recuperado el 2023, de https://www.cs.us.es/~fsancho/?e=165

Coluccio Leskow, E. (15 de julio de 2021). *Concepto.de*. https://concepto.de/electronica/

Cruz, P. P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Alfaomega Grupo Editor.

García Crespo, N. (2019). *Adaptación de un sistema de detección de personas en cámaras omnidireccionales a descriptores Deep Learning*. Biblos-e Archivo Repositorio. Universidad Autónoma de Madrid. https://repositorio.uam.es/handle/10486/688925

Hagan, M. T., Demuth, H. B., Beale, M. H., y De Jesús, O. (2014). *Neural Network Design*. PWS publishing.

Izquierdo Dussan, D., y Puerto Rodríguez, M. (2020). *Diseño de una arquitectura de redes neuronales convolucionales distribuidas*. Repositorio Institucional. Universidad Distrital Francisco José de Caldas. https://repository.udistrital.edu.co/handle/11349/27971

Lee, A. (20 de mayo de 2020). Comprensión de la Matriz de Confusión y Cómo Implementarla en Python. DataSource.ai/. Recuperado el junio de 2023. https://www.datasource.ai/es/data-science-articles/comprension-de-la-matriz-de-confusion-y-como-implementarla-en-python

Pajares, G., y de la Cruz García, J. (2010). *Aprendizaje automático. Un enfoque práctico*. Ra-Ma Editorial y Publicaciones.

Pajares, G., y de la Cruz García, J. (2007). *Visión por computador. Imágenes Digitales y Aplicaciones*. Ra-Ma Editorial y Publicaciones.

Santillana Quesada, S. (2022). *Introducción a las redes neuronales para el tratamiento de imágenes*. [Trabajo para optar al Grado en Matemática]. Facultad de Ciencias, Universidad de Granada.

Seidaliyeva, U., Akhmetov, D., Ilipbayeva, L., y Matson, E. (2020). Real-Time and Accurate Drone Detection in a Video with a Static Background. *Sensors*. https://www.mdpi.com/1424-8220/20/14/3856

Soto, O., Corral, A., Ramírez, J. y Rojo, C. (2019). Análisis del desempeño de redes neuronales profundas para segmentación semántica en hardware limitado. *ReCIBE, Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 8(2), 1-21. http://recibe.cucei.udg.mx/index.php/ReCIBE/article/view/142

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., y Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning, 15, 1929-1958. https://jmlr.org/papers/v15/srivastava14a.html

Vorobioff, J., Cerrotta, S., Morel, E., y Amadio, A. (2022). *Inteligencia Artificial y Redes Neuronales: Fundamentos, Ejercicios y Aplicaciones con Python y Matlab*. EdUTecNe, Editorial de la Universidad Tecnológica Nacional. https://www.researchgate.net/publication/359716455_Inteligencia_Artificial_y_Redes_Neuronales_Fundamentos_Ejercicios_y_Aplicaciones