UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

DESARROLLO DE UN PROTOTIPO DE SISTEMA DE DETECCIÓN DE ETIQUETAS
IMPRESAS EN SACOS DE BALANCEADO MEDIANTE REDES NEURONALES

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTORES: LUIS DAVID LOAIZA JARAMILLO
ERICK XAVIER IBÁÑEZ VACACELA

TUTOR: Ing. Vicente Avelino Peñaranda Idrovo Msc.

Guayaquil – Ecuador
2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Luis David Loaiza Jaramillo** con documento de identificación N° 0706336468 y **Erick Xavier Ibáñez Vacacela** con documento de identificación N° 0706453271; |manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 19 de febrero del 2024

Atentamente,

Luis David Loaiza Jaramillo
C.I: 0706336468

Erick Xavier Ibáñez Vacacela
C.I: 0706453271

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, **Luis David Loaiza Jaramillo** con documento de identificación N° 0706336468 y **Erick Xavier Ibáñez Vacacela** con documento de identificación N° 0706453271, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: **"DESARROLLO DE UN PROTOTIPO DE SISTEMA DE DETECCIÓN DE ETIQUETAS IMPRESAS EN SACOS DE BALANCEADO MEDIANTE REDES NEURONALES"**, el cual ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento cuando entregamos el trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 19 de febrero del 2024

Atentamente,

Luis David Loaiza Jaramillo
C.I: 0706336468

Erick Xavier Ibáñez Vacacela
C.I: 0706453271

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Ing. Vicente Avelino Peñaranda Idrovo,** Msc con documento de identificación N° 0916113426 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **"DESARROLLO DE UN PROTOTIPO DE SISTEMA DE DETECCIÓN DE ETIQUETAS IMPRESAS EN SACOS DE BALANCEADO MEDIANTE REDES NEURONALES"**, realizado por **Luis David Loaiza Jaramillo** con documento de identificación N° 0706336468 y **Erick Xavier Ibáñez Vacacela** con documento de identificación N° 0706453271, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 19 de febrero del 2024

Atentamente,

Ing. Vicente Avelino Peñaranda Idrovo Msc.
C.I: 0916113426

DEDICATORIA

Dedicatoria de Luis Loaiza

A mis amados padres, a mamá, cuya educación y principios han avivado mi deseo constante de conocimiento, aunque su presencia física ya no esté. A papá, por su apoyo inquebrantable, a mi novia y prometida, quien ha sido mi compañera constante en esta jungla urbana. A mis hermanos, por su apoyo incondicional.

Dedicatoria de Erick Ibáñez

Con profunda gratitud y alegría, dedico esta tesis a mis padres, por su amor incondicional, apoyo constante y comprensión, han sido mi mayor motivación durante este arduo proceso, sin su confianza y sacrificio, este logro no hubiera sido posible.

A mi madre, por ser mi fuente de inspiración y por creer en mí incluso cuando yo dudaba de mis propias habilidades. A mi padre, por su constante motivación y por enseñarme la importancia de la perseverancia. A mi Abuelo (+) que fue mi ejemplo a seguir y me brindó su apoyo desde el primer día para cumplir una de mis metas. A mi pareja, su amor incondicional, paciencia y apoyo fueron fundamentales para que pudiera concentrarme en mi trabajo.

AGRADECIMIENTOS

Agradecimiento de Luis Loaiza

Quiero expresar mi sincero agradecimiento a quienes han sido fundamentales durante mi trayecto universitario. A mi hermano mayor y su esposa, por brindarme un apoyo significativo durante toda mi carrera universitaria. A mis padres, por su constante apoyo y por proveerme los recursos necesarios para mi sustento. A mi hermana, por su continua preocupación y ánimo. A mi hermano, por su ayuda discreta pero significativa. Asimismo, agradezco a la familia de mi pareja por su cálida acogida. A mi tutor quien tuvo la paciencia y dedicación en este proyecto en esta última etapa académica.

Su respaldo y amor han sido pilares que han enriquecido mi camino académico.

Agradecimiento de Erick Ibáñez

Agradezco a mis amigos y a todas las personas que, de una u otra manera, formaron parte de este proceso. Sus palabras de ánimo, consejos y críticas constructivas fueron de gran valor para el crecimiento tanto en lo académico como en lo personal, cada palabra escrita en estas páginas lleva un pedacito de su aliento.

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE PROYECTO TÉCNICO	TEMA DE PROYECTO TÉCNICO
2024	Luis David Loaiza Jaramillo Erick Xavier Ibáñez Vacacela	Ing. Vicente Avelino Peñaranda Idrovo Msc.	DESARROLLO DE UN PROTOTIPO DE SISTEMA DE DETECCIÓN DE ETIQUETAS IMPRESAS EN SACOS DE BALANCEADO MEDIANTE REDES NEURONALES

La detección y verificación de las etiquetas en los sacos es un gran desafío para mantener la calidad y la seguridad de los productos en el contexto actual de la industria de producción de balanceados. Históricamente, este proceso se ha llevado a cabo de forma manual, lo que es ineficaz, costoso y susceptible a fallos. El proyecto surge como una solución innovadora para este escenario, automatizando y mejorando este proceso de forma exponencial mediante el uso de redes neuronales y tecnologías avanzadas de visión artificial.

El proyecto se centra en la implementación de un sistema basado en la placa de desarrollo Jetson Nano y el modelo de red neuronal YOLOv8. Estas tecnologías permiten un procesamiento de datos en tiempo real y una detección precisa de etiquetas, lo cual es fundamental en entornos de producción dinámicos. La incorporación de una cámara de alta resolución es esencial para capturar imágenes detalladas, asegurando una detección de etiquetas efectiva y confiable.

Más allá de su aplicación específica en la detección de etiquetas en sacos de balanceado, el proyecto tiene un alcance amplio, con potencial para ser adaptado a otras áreas industriales donde la precisión en la identificación de productos es clave. La utilización de estas tecnologías avanzadas no solo mejora la eficiencia operativa, sino que también abre puertas a la innovación en la automatización y control de calidad en diferentes sectores.

La contribución del proyecto a la sostenibilidad de los procesos industriales es un aspecto importante. La precisión mejorada en la detección de etiquetas ayuda a reducir el desperdicio causado por errores en el empaquetado y el etiquetado, lo que alinea esta innovación tecnológica con las demandas actuales del medio ambiente.

Este método demuestra que la sostenibilidad y la eficiencia industrial pueden avanzar juntas, ofreciendo soluciones tanto económicas como ecológicamente responsables.

Palabras claves: Detección de Etiquetas, Automatización, Sostenibilidad, Inteligencia Artificial, YoloV8, Label, Red Neuronal, Skretting.

ABSTRACT

YEAR	STUDENTS	DIRECTOR OF TECHNICAL PROJECT	TECHNICAL PROJECT THEME
2024	Luis David Loaiza Jaramillo Erick Xavier Ibáñez Vacacela	Ing. Vicente Avelino Peñaranda Idrovo Msc.	DEVELOPMENT OF A PROTOTYPE DETECTION SYSTEM FOR PRINTED LABELS ON BALANCED SACKS USING NEURAL NETWORKS

The detection and verification of labels on sacks is a significant challenge in maintaining the quality and safety of products in the current context of the balanced feed production industry. Historically, this process has been carried out manually, which is inefficient, costly, and susceptible to failures. The project emerges as an innovative solution to this scenario, automating and exponentially improving this process using neural networks and advanced artificial vision technologies.

The project focuses on the implementation of a system based on the Jetson Nano development board and the YOLOv8 neural network model. These technologies enable real-time data processing and precise label detection, which is crucial in dynamic production environments. Incorporating a high-resolution camera is essential to capture detailed images, ensuring effective and reliable label detection.

Beyond its specific application in detecting labels on balanced feed sacks, the project has a broad scope, with potential for adaptation to other industrial areas where precise product identification is key. The use of these advanced technologies not only improves operational efficiency but also opens doors to innovation in automation and quality control in various sectors.

The project's contribution to the sustainability of industrial processes is an important aspect. Improved accuracy in label detection helps reduce waste caused by errors in packaging and labeling, aligning this technological innovation with current environmental demands.

This method demonstrates that sustainability and industrial efficiency can advance together, offering solutions that are both economically and environmentally responsible.

Keywords: Label Detection, Automation, Sustainability, Artificial Intelligence, YOLOv8, Neural Network, Skretting.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE GRAFICAS

INTRODUCCIÓN

En un mundo donde la tecnología avanza a pasos agigantados, el desafío de fusionar la innovación con las prácticas cotidianas de la industria nunca ha sido tan crucial. Este proyecto representa una encrucijada donde la tradición se encuentra con la modernidad, transformando uno de los procesos más fundamentales en la detección y verificación de etiquetas en sacos.

Durante décadas, la tarea de garantizar la precisión de las etiquetas ha sido una labor intensiva y propensa a errores, realizada manualmente. Ahora, se enfrenta a la oportunidad de cambiar este panorama por completo, integrando la visión artificial en el corazón del proceso de control de calidad. Este enfoque no solo promete una revolución en la eficiencia y precisión, sino que también es un testimonio de cómo la tecnología puede ser una aliada poderosa en la mejora continua de los estándares de calidad y seguridad de los productos.

Lo que hace a este proyecto particularmente atractivo es su capacidad para ir más allá de las mejoras operativas. Se trata de un camino hacia la sostenibilidad y la responsabilidad ambiental en la producción industrial. Al reducir los errores y el desperdicio, esta iniciativa no solo se traduce en beneficios económicos, sino que también refleja un compromiso con prácticas más ecológicas y sustentable.

1. EL PROBLEMA

1.1.Descripción general del problema

La verificación de las etiquetas impresas en los sacos en la industria de producción de balanceados es un proceso complicado y plagado de dificultades. Esta tarea se realiza manualmente, lo que conlleva una serie de inconvenientes importantes. Debido a factores como la fatiga del personal, la probabilidad de errores en la interpretación humana y las limitaciones en la rapidez y precisión de la detección de errores. Estos pueden provocar costos adicionales, pérdida de tiempo, problemas de calidad que afectan la percepción del cliente y la confiabilidad del producto.

La necesidad específica que aborda este proyecto es múltiple: mejorar la eficiencia y la precisión en la detección de etiquetas impresas en los sacos de balanceado. Esto implica no solo identificar etiquetas incorrectas o mal impresas sino también asegurar que todas las etiquetas correctas sean reconocidas de manera rápida. El desafío radica en reemplazar el proceso manual, lento y propenso a errores con una solución tecnológica avanzada que pueda manejar estas tareas con mayor confiabilidad y eficiencia.

El problema se basa en crear e implementar un prototipo de un sistema de detección de etiquetas basado en redes neuronales. Se espera que esta tecnología automatice el proceso de verificación de etiquetas, lo que reduce de forma exponencial la probabilidad de errores humanos y optimiza el uso de recursos. El sistema utiliza algoritmos sofisticados para analizar y verificar la precisión de las etiquetas en términos de contenido, diseño y calidad de impresión, asegurando que solo los productos etiquetados de manera correcta llegan al mercado.

1.2.Importancia y alcance

El enfoque del proyecto en la detección efectiva de etiquetas en sacos de balanceado es básico para la industria acuícola. Esta tarea, que normalmente se realiza manualmente, es propensa a fallos, lo que puede afectar la calidad y la seguridad del producto. El objetivo de automatizar y modernizar este proceso con tecnologías de visión artificial es aumentar la precisión y eficiencia en la verificación de las etiquetas, lo que aumenta la confiabilidad del producto y optimiza los recursos de la empresa.

La incorporación del algoritmo de detección de objetos en imágenes y videos YOLOv8 y la placa de desarrollo Jetson Nano es un paso importante en este sentido. Estas herramientas permiten un procesamiento de datos más rápido y preciso, lo que es vital en entornos de producción donde los errores pueden ser costosos. La detección de etiquetas efectiva y confiable requiere una cámara de alta resolución para capturar imágenes detalladas.

Por otro lado, YOLOv8, una red neuronal avanzada en detección de objetos facilita la identificación de etiquetas en tiempo real, permitiendo una respuesta inmediata y precisa ante variaciones en el entorno de producción.

Más allá de mejorar el proceso de etiquetado en sacos de balanceado, el proyecto tiene el potencial de ser aplicado en otras áreas industriales donde la precisión en la identificación de productos es esencial. Asimismo, este enfoque innovador en la detección de objetos puede servir como base para futuras mejoras en el campo de la visión artificial.

Además, este proyecto enfatiza la sostenibilidad en los procesos industriales. El desperdicio relacionado con errores de empaquetado y etiquetado se reduce de forma significativa al aumentar la precisión de la detección de etiquetas. Esto tiene consecuencias tanto económicas como ambientales, ya que la reducción de desechos promueve métodos de producción más ecológicos.

1.3.Justificación

La justificación de este proyecto radica en la necesidad de desarrollar una solución tecnológica que permita la detección automatizada de etiquetas en los sacos de balanceado. La implementación de una red neuronal especializada en el reconocimiento de patrones visuales tiene como objetivo mejorar la eficiencia del proceso de verificación y garantizar la precisión en la identificación de etiquetas correctas y bien impresas.

La automatización de procesos en la industria es cada vez más necesaria para mejorar la eficiencia y reducir los costos operativos. En el caso específico la verificación manual de etiquetas impresas en sacos de balanceado u otro tipo de producto puede ser un proceso lento y propenso a errores, lo que afecta de forma perjudicial la eficiencia y la calidad del producto

Al lograr una inspección automatizada y confiable, se espera optimizar los recursos y reducir los costos asociados con la verificación manual. Además, se mejora la calidad de los productos al minimizar los errores humanos y se fortalece la confianza de los clientes en los productos.

1.4.Delimitación

1.4.1. Delimitación Temporal

El proyecto abarca el período comprendido desde mayo 2023 hasta agosto 2023. Durante este tiempo, se lleva a cabo el desarrollo, entrenamiento e implementación de la red neuronal para la detección de etiquetas impresas en sacos de balanceado. Además, se permite realizar las pruebas y evaluaciones necesarias para validar la efectividad y el desempeño del sistema propuesto

1.4.2. Delimitación Espacial

El proyecto técnico se centra en la empresa Skretting y su ubicación Km 6.5 & 4.5 Vía Durán-Tambo, Durán, donde se llevará a cabo el desarrollo e implementación de la red neuronal para la detección de etiquetas impresas en los sacos. La investigación se enfoca de forma específica en los sacos de la empresa Skretting presentes en sus instalaciones de producción industrial de balanceado.

1.4.3. Delimitación Académica

El proyecto permite explorar técnicas de aprendizaje profundo, visión por computadora y procesamiento de imágenes para abordar el problema de verificación de etiquetas en el contexto industrial. Se tiene en cuenta los aspectos teóricos y prácticos relacionados con estas disciplinas en el desarrollo de la solución propuesta.

- Se enfoca en la investigación y desarrollo de un prototipo de sistema de detección de etiquetas impresas en sacos de balanceado utilizando redes neuronales.
- Se utilizan recursos académicos y tecnológicos disponibles en el medio y la empresa mencionada.

1.5.Objetivos

Se plantean los objetivos del proyecto tanto general como específicos detallados a continuación:

1.5.1. Objetivo General

- Desarrollar un prototipo de sistema de detección de etiquetas impresas en sacos de balanceado mediante redes neuronales.

1.5.2. Objetivos Específicos

- Recopilar una base de datos de imágenes de etiquetas impresas en sacos de balanceado para el entrenamiento y validación de la red neuronal.
- Entrenar una red neuronal utilizando Python para la verificación de la calidad del etiquetado en los sacos de balanceado.
- Evaluar la eficiencia del proceso de verificación de etiquetas utilizando la red neuronal diseñada.

2. FUNDAMENTACIÓN TEÓRICA

2.1.Antecedentes

En el marco de la producción de balanceado, la validación manual de etiquetas impresas en los sacos de piensos ha sido una operación de gran impacto, pero que enfrenta desafíos susceptibles a errores, lo que puede generar ineficiencias en términos de tiempo y recursos. La necesidad de garantizar la exactitud y el cumplimiento del correcto etiquetado es de suma importancia para asegurar la calidad del producto y la adhesión a los estándares regulatorios y de la industria.

El proceso manual es propenso a errores debido a factores como la fatiga y la subjetividad. La identificación de etiquetas espurias en conjuntos de datos de aprendizaje automático, como se ve en la industria automotriz, resalta la importancia de contar con métodos robustos y automatizados para garantizar la precisión de las etiquetas. (M. Mues, 2020)

Dado que este proceso manual está sujeto a errores humanos y es ineficiente en términos de tiempo y recursos. Un estudio sobre la verificación y reconocimiento de etiquetas de envío mediante redes neuronales profundas demostró una precisión del 96% en condiciones invariantes a la rotación, lo que sugiere un potencial significativo para la automatización en la producción de balanceado. (S. Suh, 2019)

En respuesta a estas limitaciones, el presente proyecto se fundamenta en la aplicación de tecnologías avanzadas de visión artificial y aprendizaje profundo. El uso de redes neuronales, en particular las convolucionales, ha demostrado ser eficaz en el tratamiento de etiquetas ruidosas. El purificador de etiquetas Decoupled Meta Label Purifier (DMLP) ha mostrado resultados prometedores en la reutilización de etiquetas purificadas para el reentrenamiento de redes y en otros métodos de aprendizaje robustos. (al, 2023)

La implementación de redes neuronales convolucionales, como YOLOv8, ha demostrado ser un avance significativo en la detección de objetos en tiempo real, ofreciendo un rendimiento de vanguardia en términos de precisión y velocidad. Lanzado en enero de 2023, YOLOv8 se construye sobre los logros de versiones anteriores de YOLO, introduciendo nuevas características y optimizaciones que lo convierten en una opción ideal para diversas tareas de detección de objetos (Hussain, 2023)

2.2.Visión Artificial

La Visión Artificial es un ámbito de estudio Pluridisciplinario en el que intervienen conceptos de procesamiento de imágenes a través de la inteligencia artificial y la psicología visual. Un caso de estudio destacado donde se aplica la Visión Artificial es, en los sistemas de reconocimiento facial, en los cuales se utilizan algoritmos de aprendizaje no supervisado para identificar y autenticar rostros en tiempo real. Según un estudio realizado (Trujillo & Rodríguez, 2023), se demostró que los sistemas de Visión Artificial tienen un grado de precisión elevado en la detección de gestos, el software de reconocimiento de gestos ha demostrado una fiabilidad mayor al 96% en el reconocimiento de los gestos, con una duración de menos de 2 segundos en el 90% de los casos. Estos resultados indican que el sistema es preciso y eficiente incluso en condiciones de iluminación y ángulos desfavorables.

En la figura 1 se puede observar el conjunto de pasos esenciales para el procesamiento de datos. (González & Woods, 2002)

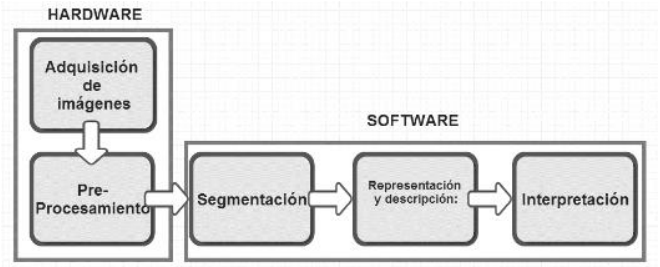


Figura 1. Diagrama de procesamiento de imágenes (Digital image processing).

Además, la Visión Artificial también se aplica en el campo de la robótica, donde se utiliza para el desarrollo de prototipos de seguimiento de objetos. En su proyecto de titulación, Pérez (2018) implementó un sistema de detección imperfecciones en telas planas de lienzo de la industria textil mediante Visión Artificial para diferenciar tela buena defectuosa. En los resultados se puede apreciar una mejora considerable en la precisión en la identificación del color del objeto que se desea seguir.

2.3.Aprendizaje Profundo

El aprendizaje profundo es una técnica de aprendizaje automático que se enfoca en aprender características de alto nivel en conjuntos de datos a través de redes neuronales profundas. A diferencia de los algoritmos de aprendizaje clásicos que necesitan que las variables sean seleccionadas por humanos para que el sistema pueda aprender, el aprendizaje profundo usa técnicas de extracción de características basadas en la representación de los datos. Según (Uriarte de la Cruz, 2022), la representación de la característica es un paso esencial para aprender de manera eficiente conceptos complejos y abstracciones en grandes conjuntos de datos.

Hay varias categorías de datos que se pueden emplear en el aprendizaje automático. Entre ellas, los datos categóricos, numéricos, de series temporales y de texto emergen como los pilares fundamentales utilizados por estos algoritmos para realizar predicciones.

El aprendizaje profundo consta de cuatro fases principales: la creación de un conjunto de datos para caracterizar el problema a resolver, el entrenamiento del modelo utilizando los datos de entrada, la evaluación del modelo entrenado y el despliegue y ejecución del modelo para una aplicación específica. Estas fases se describen en detalle en la Figura 2. (Azurmendi Marquinez, 2022)

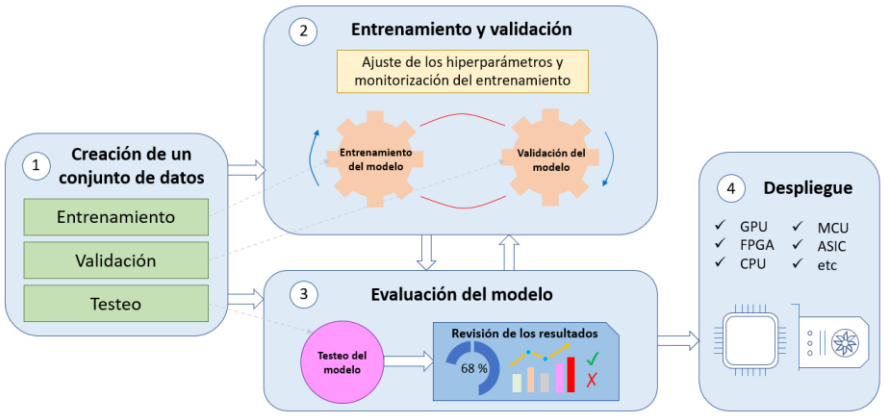


Figura 2. Diagrama de flujo del aprendizaje profundo (Uriarte de la Cruz, 2022).

El aprendizaje profundo se basa en modelos computacionales de varias capas para imitar la forma en que el cerebro procesa lo datos. Un caso de estudio de los resultados logrados gracias a esta técnica es el empleo de la red neuronal convolucional ResNet para la identificación de características y reconocimiento de imágenes. Tal y como lo revela la investigación llevada a cabo por (K. He, 2016), ResNet logró una mejora significativa en la precisión del reconocimiento de imágenes a gran escala.

En el marco de esta investigación, se explora el poder y la versatilidad de las redes convolucionales, una innovadora técnica de procesamiento de información visual.

2.4.Red neuronal convolucional

Se basa en un método de aprendizaje supervisado y una forma de red neuronal simulada para procesar imágenes de entrada y obtener la representación de la imagen, similar a cómo ve el ojo humano. (Anasi Nasimba & Martínez Arellano, 2023)Se puede definir de forma sencilla como una red de neuronas, que permite manejar patrones complejos con pocos parámetros computacionales.

Una red neuronal convolucional se destaca por su arquitectura especializada, donde cada neurona procesa datos solo del área designada, en la cual se organiza los datos para brindar una visión completa de la imagen. Esta capacidad de procesamiento localizado se asemeja al funcionamiento del sistema de visión humano. De forma similar al proceso visual, las

redes neuronales convolucionales operan con una jerarquía de capas. Con el flujo de información a través de estas capas, se revelan características cada vez más complejas y abstractas, permitiendo que la red aprenda y distinga diferentes elementos en las imágenes. Las redes neuronales convolucionales (CNN) constituyen una categoría avanzada de algoritmos de aprendizaje profundo, diseñada de forma especial para el análisis y procesamiento de imágenes. Al utilizar un aprendizaje controlado y una variante del perceptrón multicapa, estas redes se despliegan de forma eficaz para interpretar imágenes de entrada y recuperar la percepción visual, generando una representación computacional similar al campo de visión del ojo humano. (Villagomez Salazar & Coronel Alarcón, 2022) A continuación, en la figura 3 se muestra la arquitectura básica de una red convolucional.

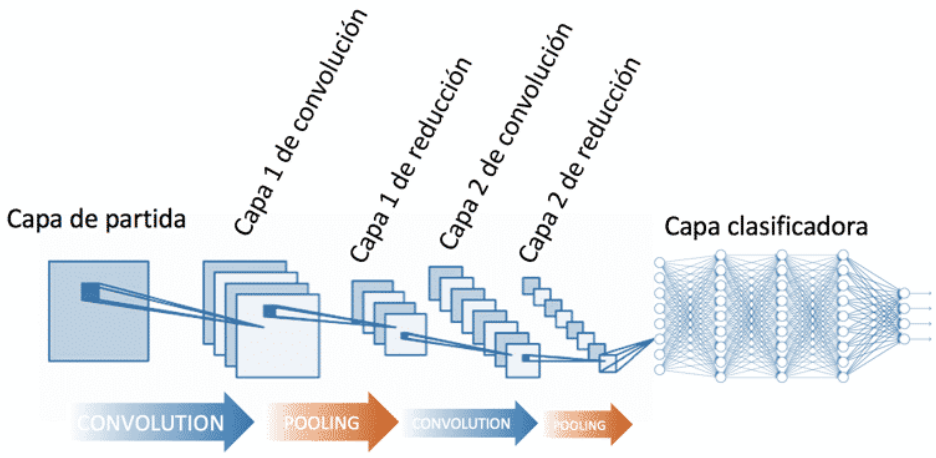


Figura 3. Arquitectura de una red convolucional (diegocalvo.es).

Pese a su apariencia simple como una red neuronal estándar, las CNN pueden incorporar múltiples neuronas, lo que les permite expresar patrones complejos con un conjunto pequeño de parámetros.

La arquitectura de una red neuronal convolucional (CNN) se distingue por su diseño bien estructurado, que se asemeja a la operación del sistema visual humano. La Figura 3 ilustra esta arquitectura en detalle. (Calvo, 2017)

En la parte inicial de la CNN, se ven capas de convolución, que son fundamentales en la extracción de características. Cada capa de convolución utiliza filtros o núcleos para

escanear la imagen de entrada y detectar patrones específicos, como bordes y texturas. A medida que la información fluye a través de estas capas, se generan mapas de características que capturan información relevante a diferentes niveles de abstracción.

Después, se presentan las capas de agrupación (o pooling), que tienen la función de reducir la dimensionalidad del mapa de características. Estas capas seleccionan las características más relevantes y disminuyen la cantidad de información, lo que conduce a una representación más compacta y manejable.

La sección subsiguiente corresponde a capas conectadas. En estas capas, las características extraídas antes se utilizan para realizar tareas específicas, como la clasificación de objetos en la imagen. Aquí, las neuronas se conectan a todas las neuronas de la capa anterior, permitiendo la integración y combinación de características para realizar decisiones finales.

Finalmente, es importante notar que la arquitectura de una CNN puede variar según la aplicación y los objetivos del proyecto. La disposición de capas, el número de neuronas y las funciones de activación pueden ajustarse para maximizar el rendimiento en tareas específicas, como la detección de objetos, el reconocimiento facial o la segmentación semántica.

2.4.1. La Neurona

Las redes neuronales, conforme su denominación implica, están constituidas por una plétora de unidades neuronales artificiales. Estas unidades neuronales, a su vez, desempeñan la función de realizar operaciones computacionales y ejecutar funciones que confieren la esencia a la red, generando una salida pertinente en correspondencia con las exigencias formuladas.

Un ejemplo elemental de una configuración de red neuronal es el perceptrón de una única capa, caracterizado por su constancia en poseer una unidad neuronal. Concebido por Frank Rosenblatt en 1957 para calibrar parámetros libres en un proceso de aprendizaje, dicho perceptrón instrumentaliza una función lineal para amalgamar ponderaciones almacenadas

con datos de entrada. Esta función lineal faculta la realización de clasificaciones binarias, limitadas a dos categorías.

La generación de una salida binaria se logra mediante la implementación de una función de activación de tipo escalón unitario. Al aumentar la dimensión de la salida del perceptrón mediante la adición de más unidades neuronales, se amplía la capacidad de clasificación para abarcar más de dos categorías tal como se observa en la figura 4. No obstante, es esencial que las categorías sean linealmente discernibles para garantizar la consecución de resultados deseados, lo que implica que deben posicionarse en ambos lados de un hiperplano. (Mendoza Barrionuevo, 2021)

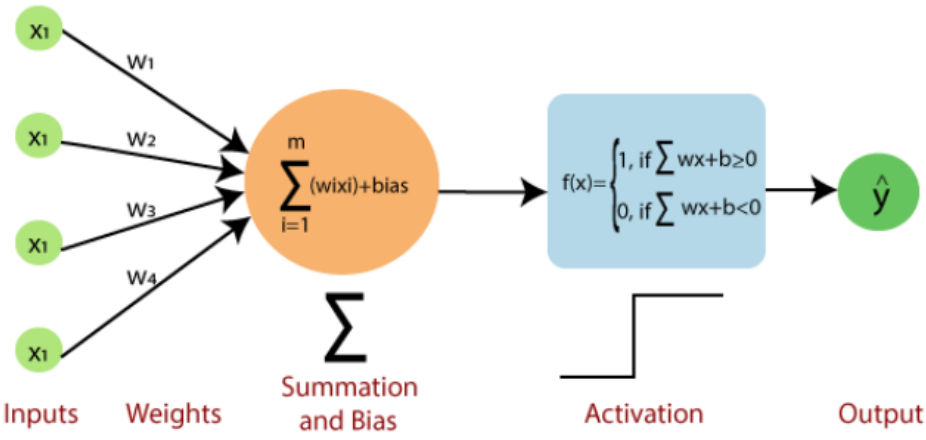


Figura 4. Rep. gráfica de un perceptrón de una neurona (Mendoza Barrionuevo, 2021).

Variables de Entrada y Pesos Sinápticos: La representación gráfica de una neurona muestra las variables de entrada, presentadas como un vector (x = [x1, x2, x3,...,xn]). A cada variable de entrada se le asigna un peso sináptico, representado como otro vector (w = [w1, w2, w3,,wn]), que refleja la importancia relativa de cada conexión.

Función de Propagación: Una parte fundamental de la representación es la función de propagación. Esta función, que en perceptrones es lineal, lleva a cabo operaciones con las entradas y los pesos sinápticos. Su resultado contribuye a generar la salida deseada, siendo esencial en el procesamiento y transmisión de señales dentro de la neurona.

Procesamiento de Información: En esencia, la representación gráfica de una neurona visualiza cómo se procesa la información. Mediante la interacción de las variables de entrada, los pesos y la función de propagación, la neurona integra y evalúa datos, lo que a su vez tiene un impacto en la generación de una respuesta apropiada.

2.4.2. Transferencia del Aprendizaje

La transferencia de aprendizaje es una estrategia que capitaliza el conocimiento adquirido por un modelo en un conjunto de datos específicos, para elevar su rendimiento en otros datos relacionados. La técnica es relevante en situaciones donde la recalibración de un modelo en un nuevo conjunto de datos se traduce en costos considerables en tiempo y recursos computacionales. Mediante la aplicación de la transferencia de aprendizaje, se logra extraer ventajas del conocimiento preexistente del modelo, adaptándolo a nuevas tareas o conjuntos de datos. Este enfoque agiliza de manera significativa el proceso de entrenamiento y, a la vez, optimiza el desempeño global del modelo resultante.

Aspecto	*Descripción*
Definición	Técnica que aprovecha conocimiento previo adquirido por un modelo en un conjunto de datos A.
Ventajas	- Mejora del rendimiento en conjunto de datos B relacionado.
	- Reducción del tiempo y recursos en reentrenamiento.
Aplicaciones	- Clasificación de imágenes.
	- Procesamiento de lenguaje natural.
	- Detección de objetos.
Estrategias	- Aprovechamiento de características aprendidas en A para B.
	- Congelamiento de capas preexistentes.
	- Ajuste fino en capas específicas.
Desafíos	- Generalización en dominios distintos.
	- Sobreajuste del modelo transferido.
Ejemplos de Implementación	- Utilizar una red neuronal preen trenada para identificar objetos en un conjunto de imágenes nuevo.
	- Adaptar un modelo de procesamiento de lenguaje natural a un nuevo idioma.
Herramientas	- Bibliotecas de aprendizaje profundo como TensorFlow, PyTorch.
	- Modelos preentrenados como VGG, ResNet.

Tabla 1. Aspectos clave en la transferencia de aprendizaje.

2.5.Nvidia Jetson nano

La placa NVIDIA Jetson Nano es una placa de desarrollo compacto diseñada para brindar un alto rendimiento en tareas de inteligencia artificial en un formato eficiente. Su tamaño, potencia y costo lo hacen una opción excepcional para proyectos que involucran marcos y modelos de IA (Inteligencia Artificial), como clasificación de imágenes, detección de objetos y procesamiento de voz. El kit puede alimentarse con un puerto micro-USB y ofrece opciones de entrada/salida (I/O), incluyendo GPIO y CSI, lo que facilita la conexión

de sensores para soportar aplicaciones de IA. Con un consumo de energía mínimo de 5 vatios, esta placa es muy eficiente.

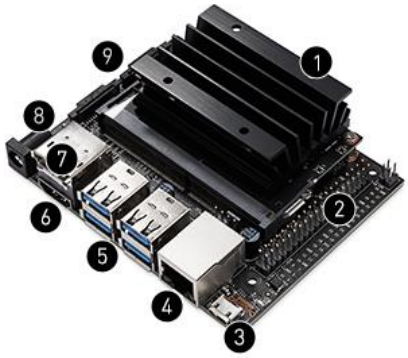


Figura 5**.** Jetson nano 4GB (Amazon.com).

La placa cuenta con cuatro puertos USB 3.0, un conector de cámara MIPI CSI-2, HDMI 2.0 y DisplayPort 1.3, Ethernet Gigabit, un módulo M.2 Key-E, una ranura para tarjeta MicroSD y un encabezado GPIO de 40 pines. A continuación, se muestra una

La tabla 2 con cada característica señalada en la figura 5.

Núm.	Característica	Núm.	Característica
1	Ranura para tarjeta microSD para almacenamiento principal	6	Puerto de salida HDMI
2	Encabezado de expansión de 40 pines	7	Conector DisplayPort
3	Puerto Micro-USB para entrada de energía de 5V, o para Modo de Dispositivo	8	Conector de barril DC para entrada de energía de 5V
4	Puerto Ethernet Gigabit	9	Conectores de cámara MIPI CSI-2
5	Puertos USB 3.0 (x4)		

Tabla 2. Características MicroSD.

2.6.Cámara de 8MP IMX219-77

La cámara IMX219-77 de 8 megapíxeles es una opción compatible con el Jetson Nano Developer Kit y el Módulo de cálculo 3/3+. En las figuras un componente básico para proyectos de inteligencia artificial (IA) y visión por computadora. Con un sensor IMX219, ofrece una alta resolución de 3280 x 2464 píxeles, lo que la hace ideal para aplicaciones que requieren detalles precisos en la captura de imágenes.

Figura 6**.** Cámara IMX219-77 (Amazon.com).

La cámara proporciona una salida de potencia de 3.3V y presenta un tamaño compacto de 0.984 x 0.945 pulgadas. Su versatilidad se refleja en aplicaciones de IA, como reconocimiento facial, detección de marcas de carretera y reconocimiento de matrículas. La cámara viene acompañada de un cable FFC de 15 pines para su conexión.

A continuación, se presenta una tabla con las especificaciones de la cámara IMX219-77:

Característica	Detalles
Resolución	3280 x 2464 píxeles
Tamaño del Sensor	1/4 pulgadas
Apertura (F)	2.0
Longitud Focal	0.117 pulgadas
Ángulo de Visión (Diagonal)	77 grados
Distorsión	Menos del 1%
Dimensiones de la Lente	0.256 x 0.256 pulgadas
Agujeros para Tornillos	4
Tamaño Total	0.984 x 0.945 pulgadas
Aplicaciones	- Reconocimiento facial
	- Detección de marcas de carretera
	- Reconocimiento de matrículas
Contenido del Paquete	- 1 cámara IMX219-77
	- Cable FFC de 15 pines
Observaciones	- Servicio técnico disponible
	- Opción de cable FFC más largo

Tabla 3. Especificaciones de la cámara.

Su alta resolución de 8 megapíxeles y capacidad para capturar imágenes detalladas y nítidas a una resolución de 3280 x 2464 píxeles la convierten en un componente valioso para nuestro objetivo. La cámara permite la captura precisa de las etiquetas impresas en los sacos de balanceado, Según (Santos, 2020) en su estudio sobre el sistema de detección y reconocimiento en tiempo real para la conducción asistida la cámara es necesaria en sistemas embebidos eficientes y en tiempo real para lograr una detección y reconocimiento precisos en entornos cambiantes.

2.7.YOLOv8

You Only Look Once (Yolo) es una arquitectura avanzada de detección de objetos basada en redes neuronales convolucionales (CNN) que ha revolucionado la detección en imágenes. A diferencia de las redes convolucionales tradicionales, que involucran múltiples etapas de procesamiento, YoloV8 adopta un enfoque de detección end-to-end, permitiendo una detección rápida y precisa en una sola pasada de la imagen.

La arquitectura de YoloV8 se basa en una red neuronal profunda con múltiples capas convolucionales. La característica distintiva de YoloV8 es su capacidad para predecir de forma simultánea múltiples cajas delimitadoras y las probabilidades de clases correspondientes en una cuadrícula predefinida de la imagen. Esto se logra a través de la división de la imagen en celdas, donde cada celda es responsable de predecir las cajas y las clases de los objetos presentes en su región. (Gama García, 2020)

En la detección de etiquetas para balanceado, YoloV8 ayuda a capturar detalles, reconocer etiquetas y hacer detecciones precisas en tiempo real, lo que mejora la eficiencia y calidad del proceso de detección en entornos industriales.

La efectividad de YoloV8 radica en su capacidad para capturar características en diferentes escalas y niveles de abstracción. A través de capas convolucionales con filtros de diferentes tamaños, la red puede detectar objetos de variados tamaños y formas, lo que lo hace apto para aplicaciones de detección en contextos diversos.

La arquitectura de YOLOv8 tal como se muestra en la figura 7 presenta innovaciones cruciales que mejoran su capacidad de detección de objetos.

Una característica fundamental es la incorporación de conexiones de salto (skip connections) y convoluciones residuales, lo que permite una mejor representación de características a diferentes escalas y niveles de abstracción. La novedad distintiva de YOLOv8 es su enfoque en realizar detecciones en tres escalas distintas tal como se observa en la figura 7, lo que lo hace eficiente y preciso para identificar objetos en una variedad de tamaños y contextos.

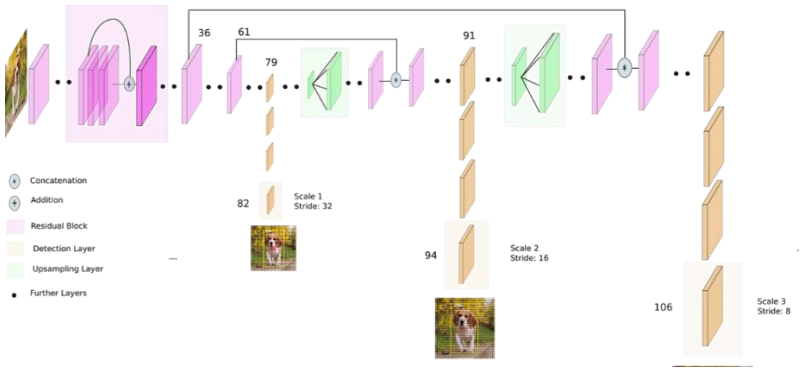


Figura 7**.** Arquitectura de Yolov8 (Kathuria, 2018).

En la implementación de YOLOv8, la detección se logra mediante el uso de kernels de detección de tamaño 1 x 1 x (B x (5 + C)), donde B representa las cajas delimitadoras, "5" denota los atributos de la caja y la confianza del objeto, y C es el número de clases. En el caso de YOLOv8 entrenado en COCO, se emplea B = 3 y C = 80, resultando en un tamaño de kernel de 1 x 1 x 255. La aplicación de estos kernels en mapas de características de tres tamaños diferentes en tres puntos específicos de la red permite una detección precisa y exhaustiva. (Kathuria, 2018)

2.8.Roboflow: Herramienta para la anotación de imágenes

Roboflow surge como una herramienta fundamental en la construcción de aplicaciones de Visión Artificial. (Mateos Dominguez, 2023) en primer lugar, fue pensada para facilitar la gestión de extensos conjuntos de imágenes y agilizar tareas de etiquetado y agrupamiento, su evolución la ha llevado a abarcar una gama completa de procesos en sistemas de Visión Artificial. Desde la adquisición de datos hasta el preprocesamiento, Roboflow opera como una plataforma integral, presentando los resultados en una interfaz amigable.

Entre sus capacidades está cargar anotaciones preexistentes en formatos como XML, JSON, COCO y YAML, entre otros. La interfaz de Roboflow, representada en la Figura, se caracteriza por su simplicidad y accesibilidad, brindando una comprensión intuitiva a los usuarios.

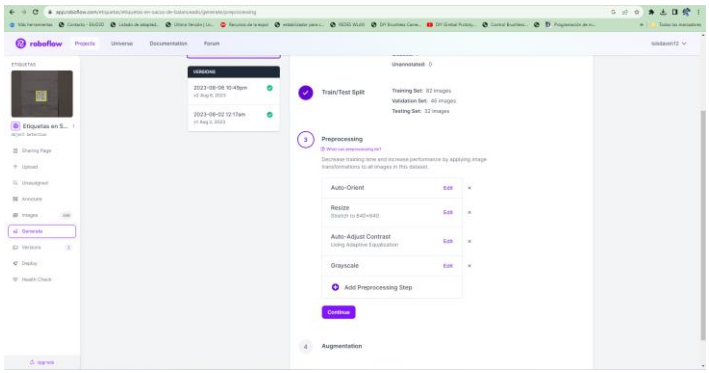


Figura 8**.** Interfaz de Roboflow (Roboflow.com).

La Figura 8 muestra la interfaz de Roboflow, que se caracteriza por su diseño funcional y accesible. Esta interfaz complementa las diversas capacidades de la herramienta, proporcionando a los usuarios una plataforma práctica y eficiente para la creación y mejora de aplicaciones de Visión Artificial.

2.9.Herramientas de desarrollo

En el desarrollo de firmware es necesario contar con las herramientas adecuadas para optimizar la calidad de los proyectos. Dos de las herramientas más destacadas en este contexto son Anaconda Navigator y JupyterLab. Estas plataformas proporcionan un entorno amigable y robusto para desarrolladores y científicos de datos, permitiendo una gestión integral de proyectos, la creación de entornos virtuales y la exploración interactiva de código y datos.

2.9.1. Anaconda Navigator

Es una plataforma integral para la administración y desarrollo eficiente de proyectos en lenguaje Python. Diseñado de forma específica para la ciencia de datos y la programación, Anaconda Navigator proporciona un ambiente de trabajo unido que incluye múltiples bibliotecas y paquetes preinstalados, facilitando la creación y gestión de proyectos complejos. Su interfaz gráfica intuitiva permite la gestión de entornos virtuales, la instalación de paquetes y la exploración de recursos educativos, lo que lo convierte en una opción potente para desarrollar aplicaciones con visión artificial. (Alfonso López, 2021)

2.9.2. JupyterLab

JupyterLab, una plataforma interactiva y de código abierto, mejora significativamente la experiencia de desarrollo para Python y otros lenguajes de programación. La interfaz es bastante personalizable, esta herramienta ayuda a los programadores y científicos de datos a trabajar con cuadernos interactivos. JupyterLab no solo facilita la exploración y experimentación con datos y algoritmos, sino que también promueve un enfoque de análisis de datos más dinámico y visual.

La capacidad de administrar y visualizar varias consolas simultáneamente es una de las características más notables de JupyterLab, lo que permite a los usuarios comparar, contrastar y analizar conjuntos de datos de manera eficiente. Su editor de código también ofrece funcionalidades de edición avanzadas que agilizan la escritura y depuración de código. (Cleofe Huamani Huancara, 2022)

3. MARCO METODOLÓGICO

3.1.Diseño del Estudio y Enfoque Metodológico

La presente investigación se centra en el desarrollo y la implementación de una red neuronal especializada con el propósito de detectar etiquetas en sacos de balanceado de camarón, evaluando la calidad de impresión y adherencia de dichas etiquetas. Este proyecto tiene un enfoque definido y un objetivo fundamental: mejorar la eficiencia y la calidad del proceso de etiquetado en la industria de la acuicultura.

El marco metodológico a continuación es esencial para alcanzar los objetivos de esta investigación. En él, se detalla la estrategia que se sigue para diseñar, entrenar y evaluar la red neuronal. La elección de una metodología sólida es crucial, ya que garantiza que la red sea capaz de identificar con precisión las etiquetas de los sacos de balanceado de camarón y verificar si están impresas y adheridas de forma correcta, se describe el enfoque de investigación, que combina elementos de visión por computadora y aprendizaje profundo para crear un sistema de detección eficiente y preciso. Además, se expone las técnicas de

recopilación de datos, que pueden incluir la captura de imágenes de los sacos de balanceado de camarón en condiciones diversas.

3.2.Declaración de los Objetivos de Investigación

El presente estudio tiene como objetivo principal diseñar, desarrollar e implementar una red neuronal capaz de detectar etiquetas en los sacos de balanceado de camarón, con la capacidad de evaluar si las etiquetas están impresas y adheridas de manera adecuada. Para alcanzar este objetivo general, se propone:

Diseño de la red neuronal: Recopilación de datos, es la primera pauta para iniciar con el proyecto el cual se enfoca en tomar fotografías de los sacos de balanceado que se tienen como ejemplo.

3.3.Proceso de Recolección y Análisis de Datos

Cualquier sistema basado en aprendizaje profundo requiere el proceso de recolección y análisis de datos, especialmente en proyectos que requieren reconocimiento y validación de imágenes, como es el caso del sistema de detección de etiquetas. La eficacia del modelo de aprendizaje automático depende de la calidad y diversidad de los datos recopilados. El objetivo de reunir un conjunto de datos suficientemente diverso y representativo fue un desafío en este proyecto.

Para superar este desafío, se parte de un enfoque basado en la recopilación de imágenes. Se inicio por tomar fotografías desde múltiples ángulos y condiciones de iluminación. Además, se cambió las configuraciones de la cámara para obtener imágenes con diferentes calidades y perspectivas. Se utilizo una cámara NIKON D5200 para este propósito, y logramos recopilar un total de 180 fotografías para nuestra base de datos. La cantidad de las imágenes debe ser alta para entrenar un modelo de red neuronal robusto y adaptable a variadas condiciones de visualización.

Figura 9**.** Captura de fotos para base de datos (autores).

Se usa Roboflow, una herramienta que permite aumentar un conjunto de imágenes, para mejorar la variedad y calidad de un conjunto de datos. Roboflow es una plataforma que facilita el procesamiento y la manipulación de imágenes para la creación de conjuntos de datos para el entrenamiento de modelos de visión por computadora. Se utilizo sus características para aplicar técnicas de aumento de datos, como el ajuste de brillo y contraste, la rotación y el cambio de escala, lo que permite simular diferentes condiciones ambientales y de captura. Esto mejora el conjunto de datos y la capacidad del modelo para generalizar a partir de nuevas perspectivas.

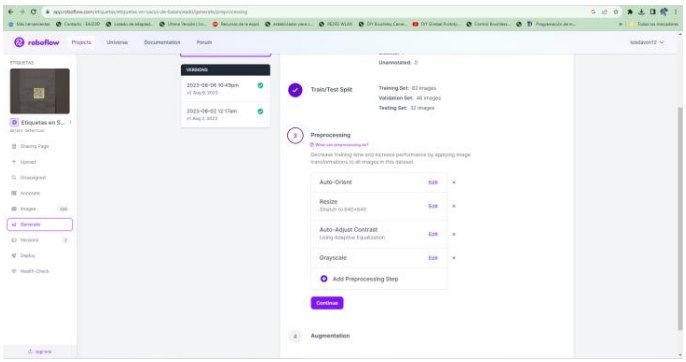


Figura 10**.** Proceso de multiplicación de imágenes (autores).

El proceso de análisis de los datos se llevó a cabo de manera minuciosa después de la recopilación de los datos. Para asegurarse que todas las imágenes fueran pertinentes y de alta calidad, se inspeccionaron y limpiaron los datos.

Las imágenes se etiquetaron cuidadosamente para garantizar que la información en las etiquetas fuera precisa y consistente. Este paso es importante porque las etiquetas sirven como base para el entrenamiento del modelo.

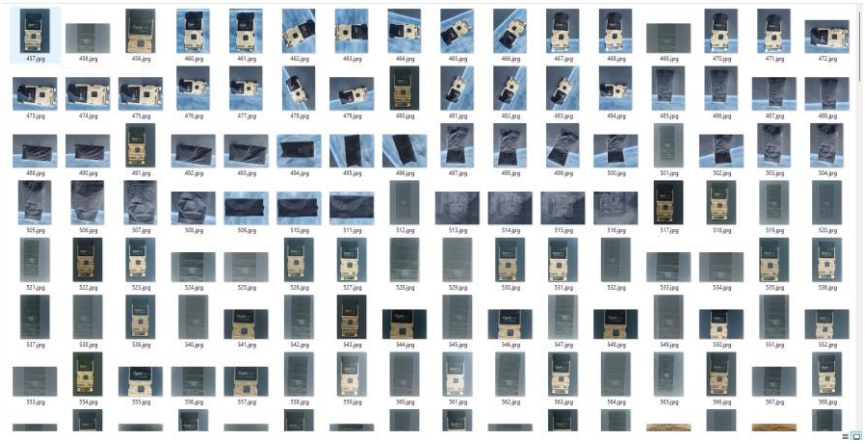


Figura 11**.** Base de datos de imágenes capturadas (autores).

La modificación del contraste y la iluminación de las imágenes se enfocaron en esta etapa para fortalecer el modelo frente a las variaciones en las condiciones de visualización.

Para lograr el objetivo, Se emplea Adobe Photoshop, un programa de edición de imágenes de gran utilidad. Se ajusta minuciosamente el contraste y la iluminación de cada fotografía con Photoshop. Se hace pensando en las diferentes condiciones de luz y ambiente que se pueden encontrar en la línea de producción en la vida real. Se escoge con mucho cuidado cada ajuste para que las imágenes reflejaran el mundo en el que funcionará el sistema tal como se ve en la figura 12.

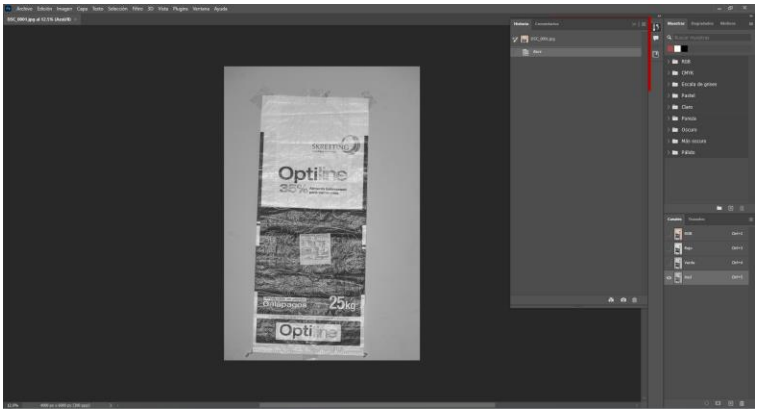


Figura 12**.** Proceso de ajuste de imágenes (Photoshop).

Se aplicaron los mismos ajustes de contraste e iluminación a todas las fotografías de manera uniforme y efectiva utilizando la función de procesamiento por lotes de Photoshop. Este método garantizó que la calidad y las características de las imágenes fueran coherentes en todo el conjunto de datos. Además, permitió el procesamiento de una gran cantidad de imágenes en un tiempo significativamente más corto, lo que optimizó el flujo de trabajo.

3.4. Etiquetado

Para la puesta en marcha de la red neuronal, es necesario generar etiquetas, denominadas 'labels'. Se requiere la creación de dos modelos distintos: uno para la detección del saco de balanceado y otro con dos etiquetas, una para etiquetas bien impresas, así como para etiquetas mal impresas.

El programa 'LabelMe' se utiliza para este proceso de etiquetado. Su instalación se realiza en el entorno de PyCharm mediante el comando en la consola:

```
pip install labelme
```

Luego de instalar, se ingresa el comando “labelme” por consola para iniciar el programa, así como se ve en la figura 13.

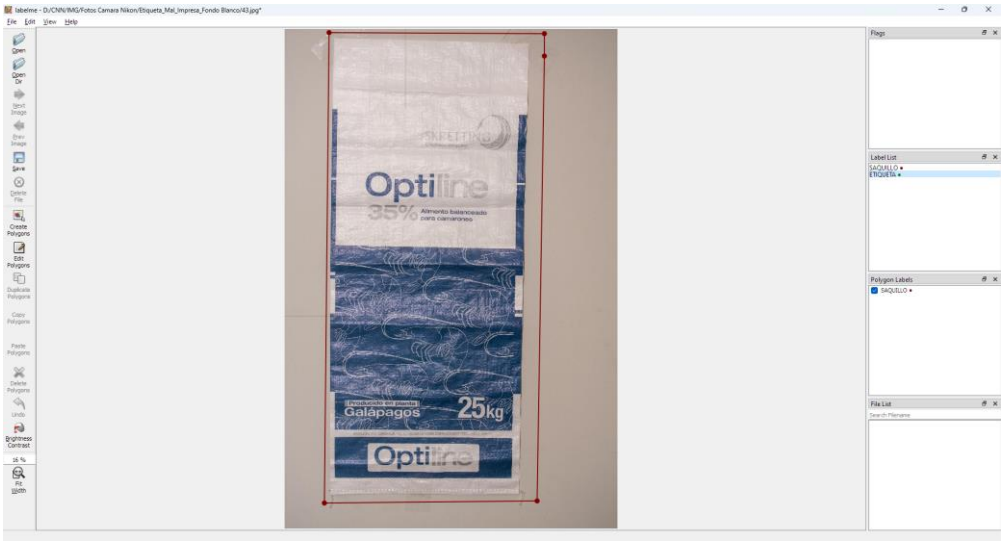


Figura 13. IDE del programa labelme.

En el entorno configurado, se procede a la generación de polígonos de etiquetas para diferenciar los modelos. Para el Modelo 1, la etiqueta se diseña específicamente para identificar el saquillo.

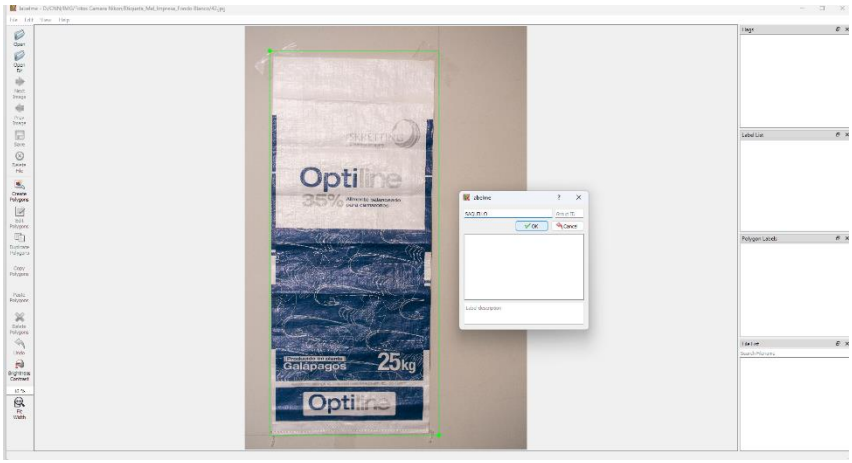
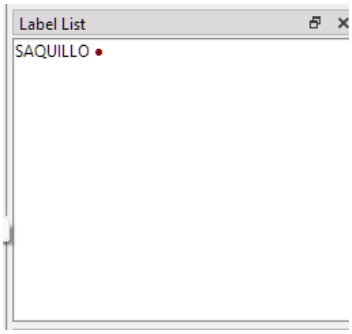


Figura 14a. IDE para saquillo.



14b. Label list.

Para el Modelo 2, se establece la identificación de la calidad de impresión de la etiqueta. Esta clasificación se divide en dos etiquetas: una para las etiquetas bien impresas como se ve en la figura 15a y otra para las mal impresas (figura 15b).

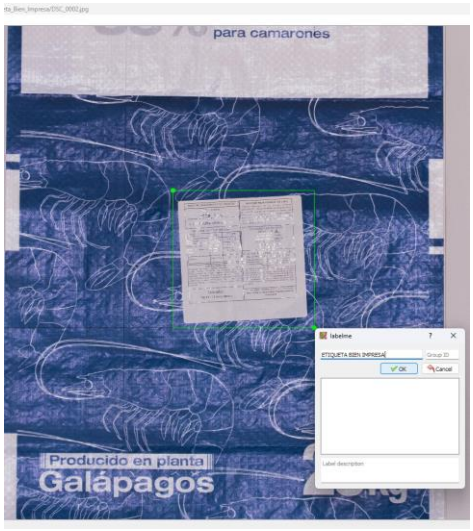


Figura 15a Etiqueta bien impresa

15b Etiqueta mal impresa

A continuación, Se registra en el software una lista de etiquetas con los nombres antes mencionados tal como se ve en la figura 16.

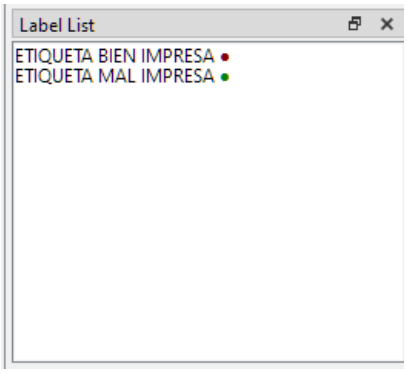


Figura 16: Lista de etiquetas.

Este proceso se repite con todas las fotografías disponibles en la base de datos, seguido por el entrenamiento de la red neuronal.

3.4.1. Entrenamiento de la Red Neuronal

Se utiliza el modelo YOLOv8, como se indicó anteriormente. Para comenzar a usarlo, primero debe instalar sus paquetes y luego ejecutar el código correspondiente en el terminal de PyCharm.

```
pip install ultralytics
```

Posteriormente, se debe verificar la tarjeta gráfica porque las redes neuronales utilizan más la GPU que la CPU del computador. Para optimizar el uso de la tarjeta gráfica, se instala CUDA utilizando el siguiente código:

```
python
import     torch
torch.cuda.is_available()
```

Para ello, se ingresa al entorno de Pyrhon y se importa la librería torch así como se ve en la figura 17.

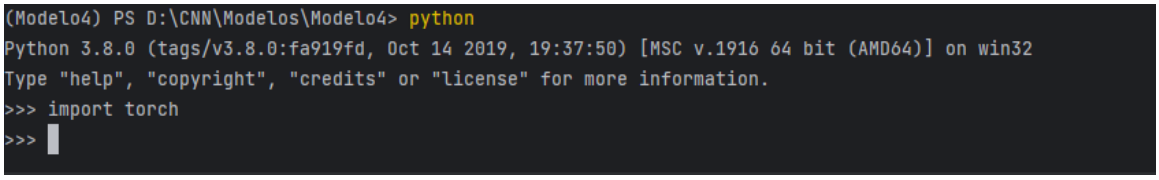


Figura 17. Entorno de desarrollo de Python.

A continuación, se verifica que CUDA esté habilitado y sea compatible con la versión instalada tal como se ve en la figura 18.

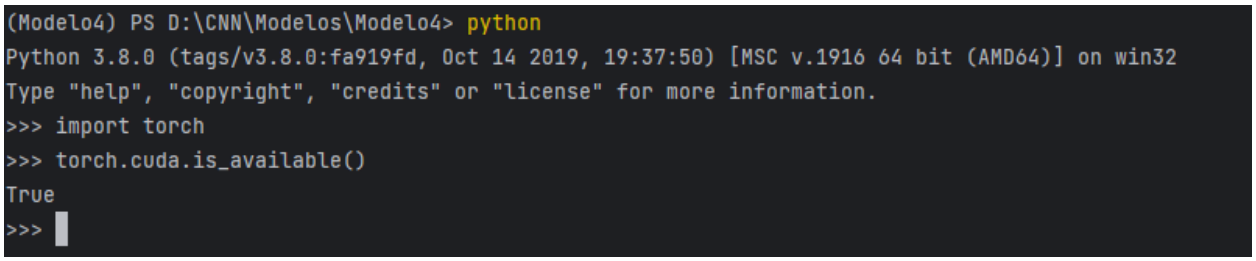


Figura 18. Verificación CUDA.

En este caso, se puede continuar con el proceso porque la verificación de CUDA ha resultado en "True". Las instrucciones sobre cómo instalar la versión de CUDA adecuada para cada sistema operativo se muestran en anexos.

Convertir todos los archivos al formato YOLO es esencial para avanzar en el entrenamiento. Se crea un archivo JSON después de aplicar etiquetas a cada imagen. Se utiliza el código siguiente para convertir estos archivos JSON en un formato YOLO.

```
pip install labelme2yolo
labelme2yolo --json_dir tu/direccion/sin/espacios
```

Disponer de una cantidad adecuada de imágenes para el entrenamiento de la red es esencial. Se crean dos carpetas para este propósito: una llamada 'train' (entrenamiento) y otra llamada 'val' (validación). Las imágenes y sus etiquetas se almacenan dentro de estas carpetas, estructuradas de la siguiente manera en la figura 19:

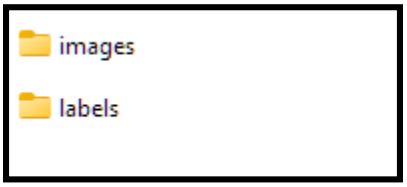


Figura 19. Estructura carpetas.

Es necesario contar con un conjunto de datos que proporcione las rutas a la información relevante para el entrenamiento de redes neuronales. Durante la conversión de los archivos JSON al formato YOLO, se crea automáticamente este archivo de colección. Es fundamental organizar las carpetas como se explicó anteriormente y luego extraer el archivo "dataset.yaml" de la carpeta de datos.

Como se muestra en la figura 20, se deben especificar las rutas donde se encuentran las carpetas que se han convertido a la versión YOLO.

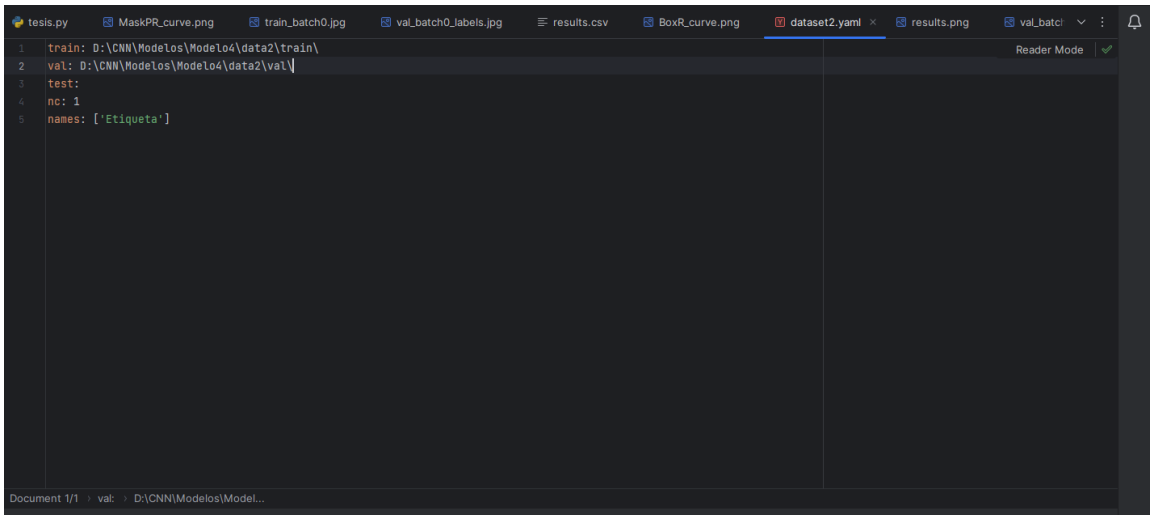


Figura 20: Archivo dataset.

El siguiente desafío consistió en realizar pruebas y errores para elegir la mejor versión para nuestras necesidades. Para este propósito se toma como referencia una tabla disponible en el sitio web de Ultralytics que detalla las diferentes versiones de YOLOv8 para la detección y segmentación de objetos.

Modelo	Tamaño (píxeles)	mAP caja 50-95	mAP máscara 50-95	Velocidad CPU ONNX (ms)	Velocidad A100 TensorRT (ms)	Parámetros (M)	FLOPs (B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

Tabla 4: Versiones de YOLO disponibles.

Después de un proceso de prueba y error, el modelo YOLOv8m-seg fue elegido por su eficiencia y velocidad y se consideró adecuado para su uso en entornos de fábrica. El siguiente código es necesario para entrenar la red:

```
yolo task=segment mode=train epochs=30 data=dataset.yaml
model=yolov8m-seg.pt imgsz=640 batch=2
```

YOLO: Hace referencia al framework o sistema basado en YOLO (You Only Look Once).

task=segment: Indica que la tarea que realiza en este caso es de segmentación. Esto implica que el modelo entrenado se utiliza para segmentar objetos en imágenes, es decir, identificar y delimitar las áreas de interés.

mode=train: Define que el modo en el que se utiliza este comando es para entrenar un modelo.

epochs=130: Especifica la cantidad de épocas o iteraciones completas que se llevan a cabo durante el entrenamiento del modelo. En este caso, se realiza en 130 épocas de entrenamiento.

data=dataset.yaml: Hace referencia al archivo YAML (dataset.yaml) que contiene la configuración de los datos utilizados para el entrenamiento. En este archivo se suelen especificar las rutas a las imágenes, las etiquetas, entre otros datos necesarios para el proceso de entrenamiento.

model=yolov8m-seg.pt: Señala el modelo base que se utiliza para iniciar el entrenamiento. En este caso, se emplea un modelo llamado 'yolov8m-seg.pt'. Los archivos con extensión '.pt' suelen contener los pesos preentrenados del modelo.

imgsz=640: Define el tamaño de las imágenes de entrada durante el entrenamiento. En este caso, se utilizan imágenes de tamaño 640x640 píxeles.

batch=4: Indica el tamaño del lote (batch size) que se utiliza durante el entrenamiento. El tamaño del lote determina la cantidad de muestras de datos que se ejecuta en una iteración del entrenamiento.

3.5.Datos de entrenamiento

3.5.1. Implementación del Sistema de Detección de Etiquetas

Ahora se procede a la implementación del sistema en la fábrica. Para ello, se requiere:

- la placa Jetson Nano
- cámara web
- fuente de alimentación eléctrica.

Se adquiere un Tablero Plástico Gris 28X21X13CM C/PLAFON-FAMATEL para organizar todos los componentes, incluyendo una pantalla para visualizar los resultados. Ahora se instala la pantalla y se ensambla la Jetson Nano. Estos elementos se colocan dentro del tablero, utilizando pegatanke para asegurarlos firmemente en su posición. Además, se organiza cuidadosamente el cableado para garantizar un montaje ordenado y funcional tal como se ve en la figura 21.

Figura 21:Implementación del sistema

Se realizan perforaciones en el tablero para instalar los botones de Power y Reset, facilitando así el inicio y control del sistema, así como se ve en la figura 22.

Figura 22:Tablero del sistema.

Se instala un Toma Empotrable 2P+T 16A 240V IP44-FAMATEL para proporcionar energía al sistema. Se conecta a una Clavija 2P+T 16A 240V IP44-FAMATEL. Esta configuración permite alimentar todo el sistema internamente, incluido un tomacorriente adicional donde se conectará la fuente de poder específica de Jetson Nano.

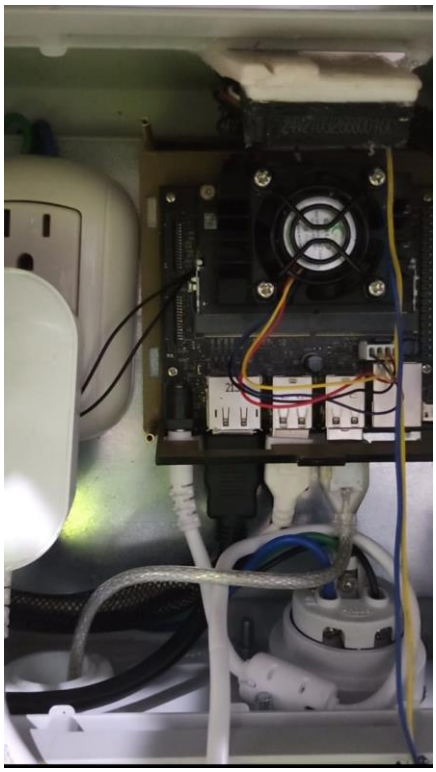


Figura 23: Enchufes y Conexiones.

Se deben sacar las antenas laterales que permiten la conexión a la placa wifi.

Figura 24: Salida de antenas WiFi.

El cable que conecta la cámara web para leer los sacos debe salir del lado opuesto. En la parte superior se encuentra un extractor para extraer el calor que se almacena en la caja, así como un ventilador para introducir aire.

Posteriormente, el cable que se conecta a la cámara web se encamina utilizando un tubo galvanizado de 3⁄4. Esta cámara se encuentra en una caja con luces LED, que están diseñadas para iluminar adecuadamente los sacos de balanceado durante la inspección.

Figura 25: Salida para la cámara web.

3.5.2. Configuración de la Placa Jetson Nano

El primer paso para configurar la placa Jetson Nano es instalar el sistema operativo recomendado por el fabricante, JetPack 4.4.1, que es adecuado para el Kit de desarrollo Jetson Nano 4GB. El JetPack SDK 4.4.1 Archive se puede descargar desde la página oficial de Nvidia.

Se han instalado paquetes adicionales de acuerdo con las recomendaciones de los especialistas que han trabajado con esta placa. La expansión de la memoria virtual y la configuración del control del ventilador en función de la temperatura de la placa son ejemplos de estos. Las líneas de código específicas se utilizan para resumir y ejecutar estas configuraciones.

Primero el código que sirve para aumentar la memoria virtual

Setup Swap File (installSwapFile.sh): Este script crea un archivo de intercambio (swap file), útil para usar en medios externos como unidades USB o SSD.

```
installSwapFile.sh [[[-d directory] [-s size] -a] | [-h]]
```

Opciones:

- -d | --dir [directoryname]: Directorio donde se coloca el archivo de intercambio (por defecto es /mnt).
- -s | --size [gigabytes]: Tamaño del archivo de intercambio, por defecto es 6 GB.
- -a | --auto: Habilita el archivo de intercambio en el arranque en /etc/fstab (por defecto: "Y").
- -h | --help: Muestra información de ayuda.

Nota: Si se habilita el archivo de intercambio en el arranque, también se recomienda automatizar el montaje del medio donde se está utilizando.

Automount (autoMount.sh): Este script permite el montaje automático de un dispositivo basándose en su etiqueta, útil para medios externos como unidades USB.

```
usage: autoMount.sh [ [-l label] | [-h]]
```

Opciones:

- -l | --label [labelname]: Etiqueta para buscar y montar el dispositivo.
- -h | --help: Muestra información de ayuda.

```
Example usage:
$ ./shellScript.sh -l RaceUSB
```

En este ejemplo, "RaceUSB" es la etiqueta del dispositivo que se monta en **/media/jetsonhacks/RaceUSB**.

Función: El script busca el dispositivo, punto de montaje y UUID basándose en la etiqueta dada. Opcionalmente, se puede añadir al **/etc/fstab** para el montaje automático.

El segundo caso para el ventilador:

La configuración del hardware y software necesarios para controlar un ventilador en un Jetson Nano:

Hardware necesario

- Se requiere un ventilador de 5V PWM. Se utiliza el modelo Noctua nf-a4x20 5V PWM.
- Se recomienda usar el conector de barril con una fuente de alimentación de 4A.

Software necesario

- Se asume el uso de la imagen estándar en el Jetson Nano.
- Python 3 debería estar preinstalado. Se puede verificar con ***python3 --version***. La versión 3.5 o superior es adecuada.
- Si no está instalado, se puede instalar con ***sudo apt install python3-dev***.

Instalación y Configuración

- Ejecutar sudo **./install.sh** para instalar.
- El script se ejecuta automáticamente en el arranque.
- Es un sistema de configuración única, a menos que se desee ajustar las velocidades del ventilador.

Personalización

- Para personalizar, **editar /etc/automagic-fan/config.json** con un editor de texto (por ejemplo, nano): **sudo nano /etc/automagic-fan/config.json**.
- Configuraciones en config.json:
    - "**FAN_OFF_TEMP**": Temperatura (°C) por debajo de la cual el ventilador se apaga.
    - "**FAN_MAX_TEMP**": Temperatura (°C) por encima de la cual el ventilador funciona al 100%.

- "**UPDATE_INTERVAL**": Frecuencia de actualización de la velocidad del ventilador (en segundos).
- "**MAX_PERF**": Valores mayores a 0 maximizan el rendimiento del sistema aumentando las velocidades de reloj de CPU y GPU.

- Se pueden utilizar números enteros o flotantes en estos campos.
- La precisión de los sensores térmicos es de 0.5 °C.
- Los cambios se aplicarán después del próximo reinicio.
- Para aplicar cambios inmediatamente: ***sudo service automagic-fan restart***.
- Para verificar el estado del servicio: ***sudo service automagic-fan status***.

Se procede a pasar el modelo a formato ONNIX el cual tiene un mejor procesamiento de CPU para la Jetson Nano.

```
# Evaluate the model's performance on the validation set
results = model.val()

# Perform object detection on an image using the model
results = model('https://ultralytics.com/images/bus.jpg')

# Export the model to ONNX format
success = model.export(format='onnx')
```

3.5.3. Desarrollo del Modelo de Red Neuronal YOLOv8

El desarrollo del modelo de red neuronal YOLOv8 es un paso importante hacia la creación de un sistema de detección de etiquetas efectivo. La precisión y eficacia del modelo en la identificación de etiquetas en sacos de balanceado bajo diferentes condiciones, implica una serie de pasos:

Entrenar la red neuronal con un conjunto de datos representativo

El primer paso en el desarrollo del modelo es la creación de un conjunto de datos representativos. Este conjunto debe incluir una variedad de imágenes de sacos de balanceado que muestren las etiquetas en diferentes estados de impresión y adherencia. Este conjunto de datos debe ser diverso para que el modelo pueda identificar etiquetas en una variedad de situaciones reales. El modelo debe ser sólido y adaptable, y las imágenes deben incluir variaciones en la iluminación, el ángulo de visión y posibles obstrucciones.

Para lograr esto, es necesario crear la etiqueta, la cual debe colocarse en cada imagen que contiene una etiqueta ilustrativa. La forma de seleccionar una etiqueta, que se debe reconocer e identificar, se muestra en la Figura 26.

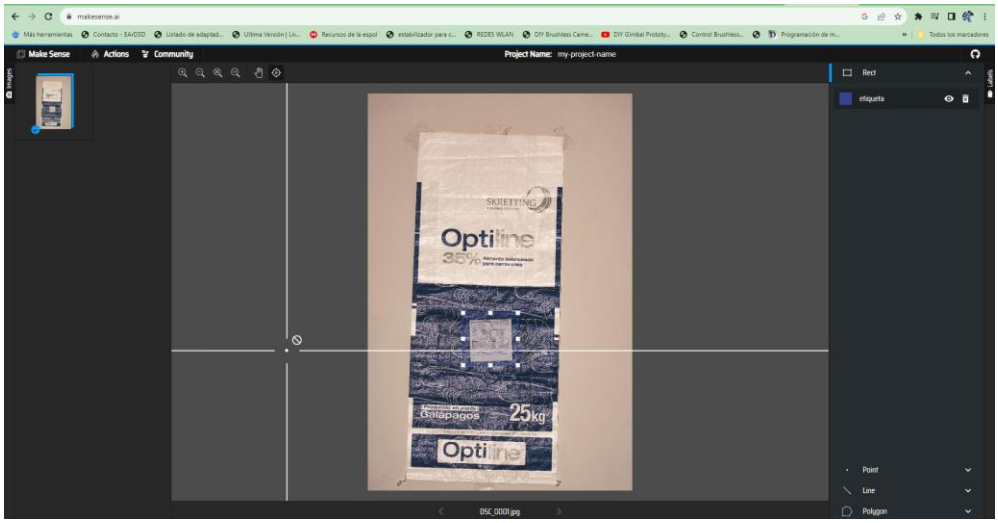


Figura 26: Selección de etiqueta.

"LabelMe" es una herramienta de etiquetado más efectiva que se puede instalar desde el entorno de programación Python. Para la programación en este caso, se está utilizando PyCharm. La siguiente es la forma de instalar LabelMe en PyCharm(Figura 27):

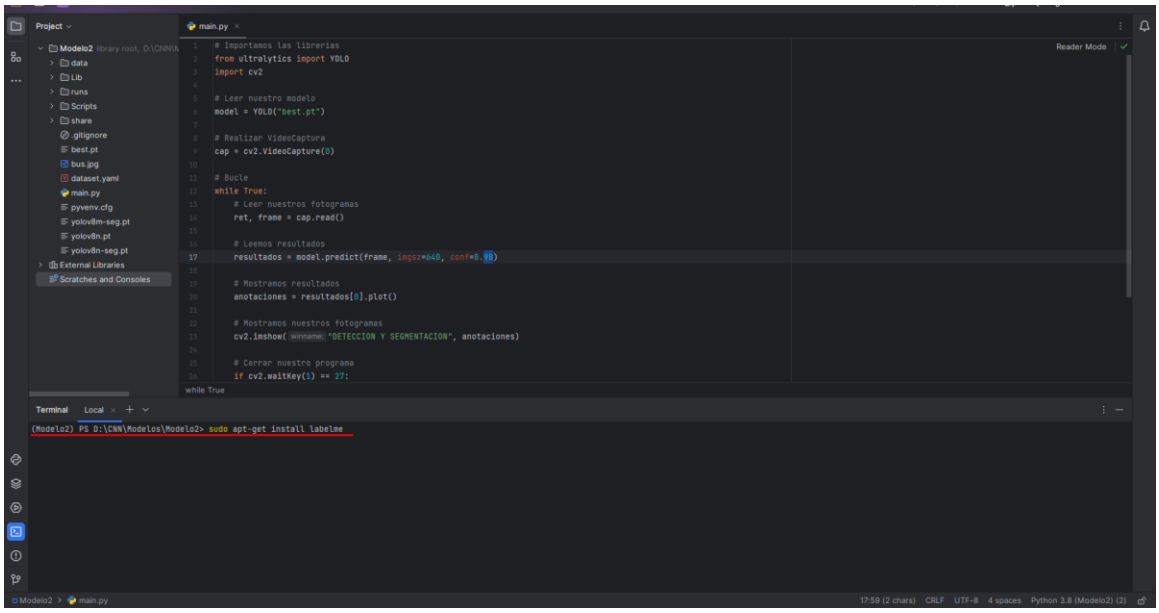


Figura 27: Instalación labelMe.

Dado que se está trabajando con Python, la instalación del paquete LabelMe se puede realizar de manera directa utilizando pip3. Este programa, similar en funcionalidad a MakeSense AI, se destaca por su interfaz de usuario más intuitiva y fácil de manejar, lo que lo hace preferible por su simplicidad. La plataforma de LabelMe tal como se ve en la figura 28 ofrece una experiencia de usuario más accesible, facilitando el proceso de etiquetado de imágenes para el entrenamiento de modelos de aprendizaje automático.

Figura 28: Paquete LabelMe

Para este proyecto, LabelMe es la mejor opción de etiquetado porque puede crear anotaciones en formas poligonales y figuras predefinidas. Esta característica es fundamental para garantizar la precisión en el proceso de etiquetado.

Ambos programas se aseguran de que las etiquetas se guarden en JSON. Instalar la herramienta "labelme2yolo" para convertir estas etiquetas al formato YOLO. Debido a que 'labelme2yolo' es un paquete de Python, su instalación se realiza con el comando siguiente:

```
pip install labelme2yolo.
```

Para realizar la conversión, se debe tener todos los archivos JSON en una carpeta y usar el siguiente comando para convertir los archivos JSON a Yolo:

```
labelme2yolo --json_dir tu/direccion/sin/espacios
```

Por lo tanto, se puede cambiar todo al formato YOLO.

3.5.4. Escoger el mejor modelo entrenado.

En este apartado se declara el por qué el modelo 1 mejor llamado Skretting-1, es la mejor opción. Se maniobro con varios modelos de los cuales los resultados no eran los mejores, motivos: como al no detectar las clases creadas en tiempo real (en planta) no fueron escogidos como el modelo definitivo de nuestra red, luego se hablarán sus resultados en el capitulo 4.

El modelo 1 fue entrenado con fotografías obtenidas de videos grabados en la línea de producción donde se instaló el prototipo, del video se procede a sacar los fotogramas utilizando el siguiente código ejecutado en un entorno de Google Colab:

```
VIDEO_DIR_PATH = f"{HOME}/videos"
IMAGE_DIR_PATH = f"{HOME}/images"
```

```
FRAME_STRIDE = 10 #Aqui se define cada cierto cantidad de fotogramas
Obtiene la imagen
```

```
import supervision as sv
from tqdm.notebook import tqdm

video_paths = sv.list_files_with_extensions(
    directory=VIDEO_DIR_PATH,
    extensions=["mov", "mp4"])

TEST_VIDEO_PATHS, TRAIN_VIDEO_PATHS = video_paths[:2],
video_paths[2:]

for video_path in tqdm(TRAIN_VIDEO_PATHS):
    video_name = video_path.stem
    image_name_pattern = video_name + "-{:05d}.png"
    with sv.ImageSink(target_dir_path=IMAGE_DIR_PATH,
image_name_pattern=image_name_pattern) as sink:
        for image in
sv.get_video_frames_generator(source_path=str(video_path),
stride=FRAME_STRIDE):
            sink.save_image(image=image)
```

Al obtener una cantidad de fotografías y filtrado aquellas que no se usara se procede a hacer su etiquetado en la Roboflow. Esto ayudo a que la eficiencia sea del 100% en resultados se dará todos estos datos.

3.5.5. Implementación en la línea de producción

En la planta de producción se realiza la instalación de la caja previamente ya instalado la JetsonNano y la alimentación de corriente, el montaje debe ser después del proceso de etiquetado por lo que se selecciona una malla como se observa en la figura 29.

Figura 29: Figura de la malla.

Se coloca la cámara en posición hacia abajo apuntando la cinta transportadora donde pasaran los sacos tal como se observa en la figura 30, la colación tiene que estar enfocada a la banda.

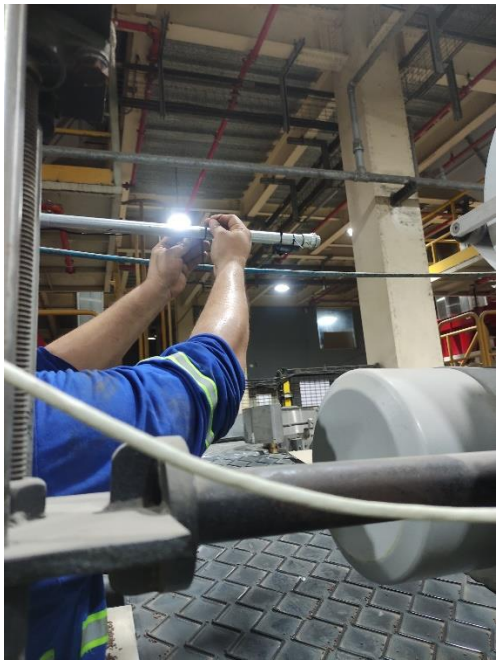


Figura 30: Pasarela sacos.

Centrada y a una altura que pueda obtener la totalidad del saco y la distancia correcta para ver la etiqueta como se observa en la figura 31.

Figura 31: Imagen de referencia para cámara.

4. RESULTADOS

4.1.Análisis y Discusión de los Resultados Obtenidos

Los sistemas de inspección visual automatizados son un avance significativo para garantizar la eficiencia operativa y la calidad en las instalaciones de producción. Su uso en el control de la aplicación de etiquetas se ha convertido en una práctica estándar para proteger la reputación de una organización al evitar y corregir los errores antes de que los productos lleguen al consumidor final. Estos sistemas no solo detectan errores en las etiquetas, sino que también verifican su colocación correcta, lo que evita cualquier error que pueda dañar la percepción de la marca.

Este capítulo presenta el análisis obtenido de la implementación de redes neuronales en un sistema de detección de etiquetas impresas en sacos de balanceado. En un esfuerzo por automatizar la inspección de etiquetas y superar los desafíos inherentes a los métodos de control de calidad manual, Se han evaluado cuatro modelos de redes neuronales, Skretting-1 a Skretting-4.

La precisión, el mAP50 y otras métricas importantes que determinan la viabilidad de estos modelos en un entorno de producción real se incluyen en este análisis. La capacidad de realizar verificaciones de etiquetado precisas mientras los productos pasan por la línea de producción fortalece los protocolos de control de calidad y aumenta la productividad y reduce los costos asociados con los errores de etiquetado.

A continuación, se explica los resultados de cada modelo y se ofrece conocimientos y sugerencias basados en los datos recopilados, con el objetivo final de elegir el sistema más eficaz para nuestra aplicación.

4.1.1. Resultados de las pruebas en planta

Se puede ver en la tabla 5 que Skretting-1 mantiene un rendimiento alto y consistente en ambos entornos, comparando los resultados de los modelos en un entorno controlado (según el F1 Score/mAP50 de la primera imagen) con los resultados de la planta. Skretting-2 sigue siendo fuerte, aunque la precisión en el entorno real ha disminuido un poco.

Modelo	Fecha	Hora	Tiempo	Total de Sacos	Correctos	Incorrectos	Precision	%
Skretting-1	24/01/2024	10:00	1 Hora	86	85	1	0,98837209	98,8372093
Skretting-1	24/01/2024	12:00	30 min	32	32	0	1	100
Skretting-1	24/01/2024	14:00	1 min	13	13	0	1	100
Skretting-2	21/01/2024	15:00	1 Hora	56	54	2	0,96428571	96,42857143
Skretting-2	21/01/2024	17:00	30 min	36	35	1	0,97222222	97,22222222
Skretting-2	21/01/2024	18:00	1 min	12	12	0	1	100
Skretting-3	17/01/2024	14:00	1 Hora	136	114	22	0,83823529	83,82352941
Skretting-3	18/01/2024	16:00	30 min	70	60	10	0,85714286	85,71428571
Skretting-3	19/01/2024	12:00	1 min	12	10	2	0,83333333	83,33333333
Skretting-4	20/01/2024	10:00	1 Hora	100	95	5	0,95	95
Skretting-4	20/01/2024	12:00	30 min	40	30	10	0,75	75
Skretting-4	20/01/2024	14:00	1 min	10	9	1	0,9	90

Tabla 5: Pruebas de testeo en planta.

4.1.2. Resultados de las pruebas de interacción en CNN

Skretting-2

La métrica del Modelo 2 mAP50 (Mejor Precisión Media en Intersección al 50% sobre la Unión) se muestra en la gráfica de la figura 32. Se muestra una curva de precisión que asciende rápidamente hasta alcanzar y permanece cerca del valor 1, lo que indica que el modelo tiene una precisión casi perfecta en la detección de etiquetas a lo largo de varias pruebas. La línea naranja parece representar el umbral sobre el cual se calcula la precisión, mientras que la línea sólida azul indica la precisión del modelo en esos umbrales específicos. La estabilización de la curva superior indica que el modelo puede identificar con precisión la mayoría de las etiquetas con un alto grado de confianza.

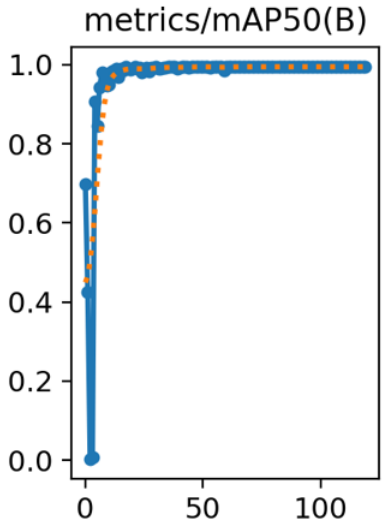


Figura 32: Gráfica de la métrica de mAP50 para el modelo 2.

Análisis de los Resultados de la Planta del Modelo

El Modelo 2 demostró una alta efectividad con porcentajes de precisión del 96,43%, 97,22% y 100% en tres fases de testeo diferentes. Esto demuestra un desempeño consistente y fiable en la detección de etiquetas correctas en un entorno de producción real, lo que indica que el modelo es resistente a las variaciones que pueden ocurrir en situaciones reales.

Comparativa de las Métricas del Modelo

Existe una relación evidente con la precisión del entorno controlado de pruebas y los altos porcentajes de precisión de la planta al comparar el mAP50 de la gráfica con los resultados de la planta. El mAP50 de 0,963 indica que el modelo tiene una precisión del 96,3% en detectar etiquetas con al menos el 50% de IoU en promedio.

Esto se alinea con los datos de precisión observados en la planta, lo que valida que el Modelo 2 funciona bien para detectar las etiquetas impresas en sacos de balanceado.

Skretting-3

La métrica mAP50 del Modelo 3 se muestra en la gráfica de la figura 33. La precisión del modelo a diferentes umbrales de detección se muestra en la línea azul, mientras que los puntos naranjas pueden representar el umbral específico de IoU (Intersection over Union) en el que se calcula esta precisión. La precisión presenta una gran variabilidad a lo largo de los diversos umbrales, con valores que oscilan entre aproximadamente 0.5 y 0.8. Este patrón de altibajos indica que el modelo puede tener un rendimiento inconsistente en la detección de etiquetas. Esto puede ser el resultado de variaciones en la calidad de las imágenes, cambios en las condiciones de iluminación o características de etiquetas que dificultan el reconocimiento del modelo.

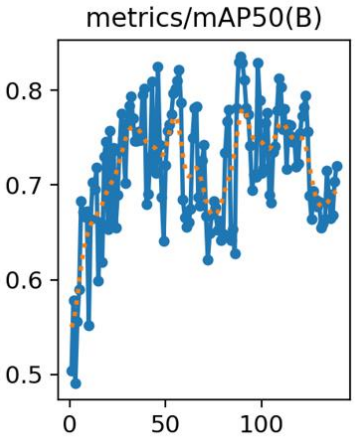


Figura 33: Métricas del modelo 3.

Análisis de los Resultados de la Planta del Modelo

El Modelo 3, evaluado en un entorno de producción real, obtuvo un mAP50 de 0.86, lo que refleja una precisión promedio del 86% en la detección correcta de etiquetas con al menos un 50% de IoU, un equilibrio entre detección y localización precisa.

Comparativa de las Métricas del Modelo

El Modelo 3 tiene un 86% de precisión en la detección de etiquetas con al menos el 50% de superposición con la verdad de referencia, según el mAP50 de 0.86. Sin embargo, al comparar este valor con los resultados de la planta, se ve una precisión que oscila entre el 83,82% y el 85,71%, lo que indica que el modelo puede no ser tan consistente en las condiciones variables de una línea de producción real.

Skretting-4

La métrica mAP50 (Precisión Mediana en 50% de IoU) se muestra en la gráfica de la figura 34. La precisión del modelo a diferentes umbrales de confianza se muestra rápidamente por la línea azul, que asciende rápidamente y alcanza un pico cerca del valor de 1, lo que indica una alta precisión en la detección de etiquetas. Luego, la línea experimenta una pequeña caída antes de estabilizarse y permanecer cerca de 1. Los umbrales específicos de IoU en los que se calcula la precisión pueden ser representados por puntos naranjas. La estabilización de la precisión en un nivel alto indica que el modelo es capaz de mantener una detección consistente y precisa en la mayoría de las condiciones probadas.

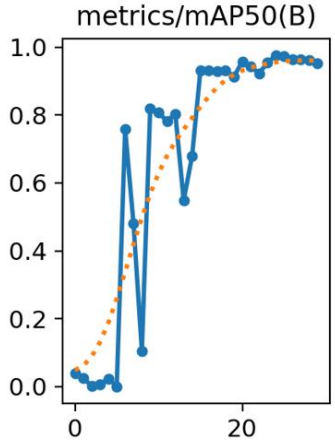


Figura 34:Métrica modelo 4.

Análisis de los Resultados en Planta del Modelo

El Modelo 4 demostró resultados en la planta con una precisión del 95 %, 75 % y 90 % en las pruebas correspondientes. Aunque estos resultados son menores que los del Modelo 2, demuestran que el Modelo 4 es capaz de reconocer la mayoría de las etiquetas en condiciones de producción. La variación en la precisión sugiere que el modelo puede ser susceptible a variaciones en el entorno, como variaciones en la iluminación o la calidad de las etiquetas.

Comparativa de las Métricas del Modelo

Se puede ver que el Modelo 4 tiene un rendimiento general bueno, pero con más variabilidad que el Modelo 2, comparando el mAP50 de la gráfica con los resultados en planta. El modelo tiene una precisión promedio del 95.4% en detectar etiquetas con al menos el 50% de IoU, según un mAP50 de 0.954.

Skretting-1

La métrica del Modelo 1 mAP50 (Mejor Precisión Media en Intersección al 50% sobre la Unión) que se muestra en la gráfica de figura 34. Se muestra una curva de precisión que asciende rápidamente hasta alcanzar prácticamente valor 1, lo que indica que el modelo tiene una precisión casi perfecta en la detección de etiquetas a lo largo de varias pruebas alcanzó precisiones de 98.84%, 100%, y 100% en las pruebas en planta.

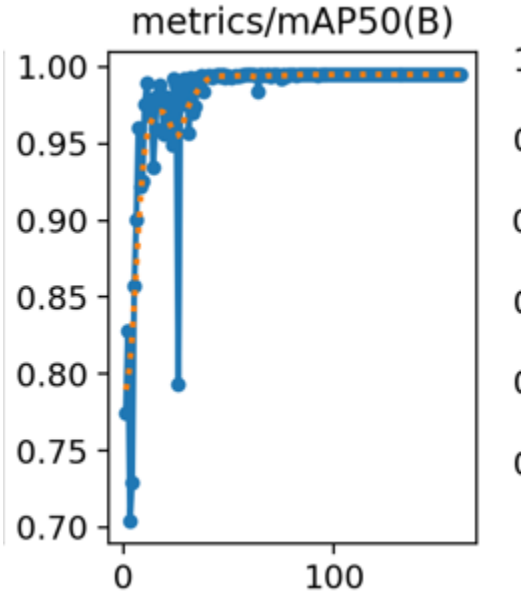


Figura 35: Métrica modelo 1

Análisis de los Resultados en Planta del Modelo

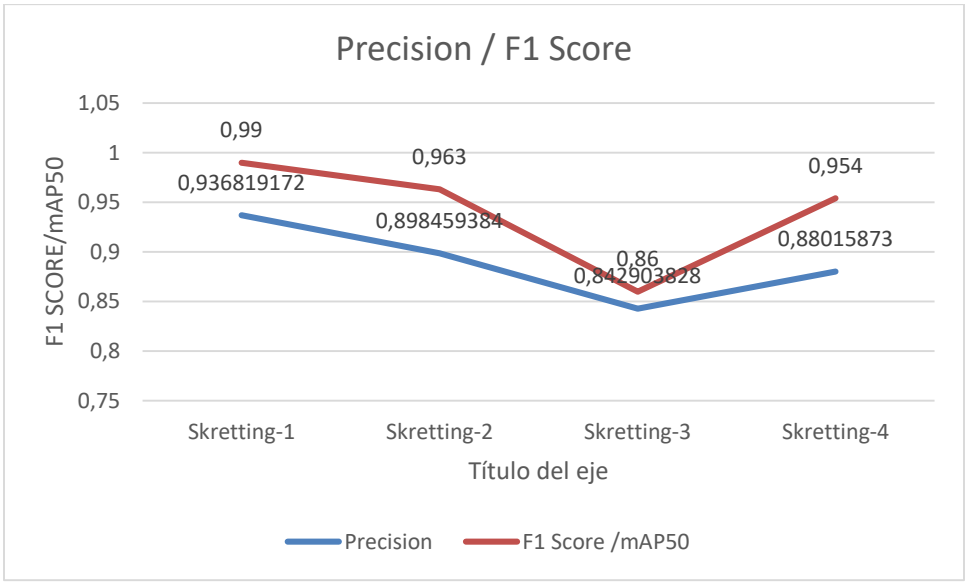
Los resultados de las pruebas del Modelo 1 en la planta sugieren que este modelo es extremadamente confiable y consistente en su capacidad para identificar correctamente las etiquetas impresas en los sacos de balanceado. Las altas tasas de precisión indican que el modelo es robusto en distintas condiciones de producción.

Comparativa de las Métricas del Modelo

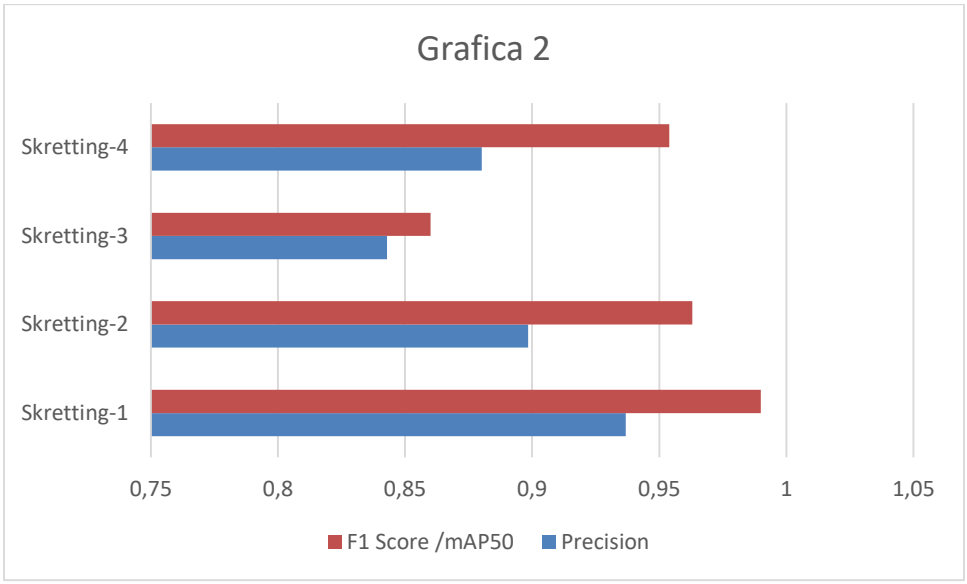
Al comparar el mAP50 de la gráfica con los resultados de la planta, se puede ver claramente una relación entre la precisión del entorno controlado de pruebas y los altos porcentajes de precisión de la planta. Se concluye que el Modelo 1 sería el mejor para el proyecto de detección de etiquetas impresas en sacos de balanceado si se comparan estos datos con las métricas teóricas de mAP50, F1 Score y precisión.

El modelo tiene una precisión tan alta que tiene un alto F1 Score, lo que demuestra un equilibrio ideal entre precisión y recuerdo, lo que resulta en un rendimiento excepcional en

la detección de etiquetas precisas y fiables. Esto también lo reflejamos en el grafico 1 y el grafico 2 donde comparamos a los demás modelos según su precisión y el F1 Score que es el valor mAP50 de nuestra red neuronal.



Gráfica 1: Comparativa linean de los modelos.



Gráfica 2: Comparativa barras lineales de los modelos.

4.2.Evaluación de la Eficiencia y Precisión del Sistema

El segundo punto importante del análisis del proyecto es resolver los problemas específicos de la planta, según el jefe de Planta, Msc. Andrés Vallejo. Él resalta que hay casos relacionados con la política de devoluciones, con lotes que son devueltos debido a la falta de etiquetado adecuado. Además, la falta de un control de calidad estricto en esta área da lugar que los clientes devuelvan sacos dañados o próximos a su fecha de caducidad, aprovechando la situación.

Con la implementación del sistema, la planta no solo logra mitigar el riesgo de errores en el etiquetado, sino que también se establece una base más sólida para una política de devoluciones más estricta. Este avance asegura que solo los sacos de balanceado que cumplan con los estándares establecidos sean despachados, y proporciona una evidencia documentada que respalda la calidad y el estado del producto en el momento de la venta, lo que resulta transacciones más eficientes si se toma como base un volumen mínimo de compra de 400,000 $ de balanceado.

4.2.1. Análisis de Costos en la Compra de Etiquetas

En base a los resultados obtenidos se puede realizar el siguiente análisis con el

- **Costo de las Etiquetas**: El costo de cada etiqueta varía entre 2 centavos (ctvs.) y 5 centavos.
- **Gasto Mensual en Etiquetas**: El gasto mensual actual oscila entre 800$ y 1400$ dólares.
- **Ahorro Estimado con el Sistema**: Se estima un ahorro de al menos el 5% del gasto actual en etiquetas gracias al nuevo sistema.
- El gasto actual se utiliza para calcular la cantidad mínima y máxima de etiquetas que se pueden comprar cada mes.

Cálculo del Número de Etiquetas:

- **Caso de Gasto Mínimo (800 dólares) con Costo Máximo por Etiqueta (5 centavos)**:
    - Se calcula dividiendo el gasto total por el costo de una etiqueta.

$$Número\ de\ etiquetas\ = \frac{800\$}{0.05\$/etiqueta} = 16{,}000\ etiquetas$$

Caso de Gasto Máximo (1400 dólares) con Costo Mínimo por Etiqueta (2 centavos):

- De igual manera, se divide el gasto total por el costo de una etiqueta.

$$Número\ de\ etiquetas\ = \frac{1400\$}{0.02\$/etiqueta} = 70{,}000\ etiquetas$$

Cálculo del Ahorro Estimado con el Sistema:

- **Ahorro con un Gasto de 800 dólares**:

Se calcula como el 5% del gasto total: Ahorro = 800$ ×5% = 40 $

- **Ahorro con un Gasto de 1400 dólares:**

Igualmente, se calcula el 5% del gasto total: Ahorro = 1400$ ×5% = 70 $

CONCLUSIONES

El Modelo 1 se destaca significativamente en términos de precisión y confiabilidad, lo que se debe a la calidad y diversidad de la base de datos recopilada para su entrenamiento. Este modelo demuestra la capacidad de la red neuronal para identificar etiquetas de manera precisa y consistente bajo diversas condiciones, con un mAP50 cercano a 1 y una precisión casi perfecta en las pruebas en planta.

Se ha demostrado que la automatización de la inspección de etiquetas con redes neuronales es una estrategia rentable que puede reducir significativamente los costos operativos. La implementación del sistema reduce los costos de etiquetado hasta un 5 %. La disminución de este porcentaje en los costos operativos no solo aumenta la eficiencia y la productividad, sino que también fortalece la viabilidad económica de utilizar tecnologías avanzadas en los procesos de control de calidad.

La implementación de la red neuronal diseñada ha resultado en un proceso de verificación de etiquetas más eficiente. La automatización de esta tarea reduce los errores humanos y acelera el proceso de inspección. Además, el sistema permite una política de devoluciones más estricta y justa, como se refleja en la retroalimentación del jefe de Planta, Msc. Andrés Vallejo. La eficiencia mejorada, junto con la capacidad de monitorear y validar la calidad del etiquetado en tiempo real, sugiere un retorno significativo de la inversión y fortalece la operación y la reputación de la empresa en el mercado competitivo de alimentos para camarón.

RECOMENDACIONES

- Se recomienda buscar un equilibrio entre la complejidad del modelo y su capacidad de procesamiento, ya que modelos más pesados pueden dificultar el procesamiento. Para garantizar que el modelo sea lo suficientemente liviano para funcionar en la línea de producción sin comprometer su precisión, es esencial optimizarlo.
- Se recomienda entrenar a la red neuronal con imágenes capturadas por la misma cámara que se utiliza en la línea de producción para mejorar la eficiencia del modelo. Esto garantiza que el modelo se adapte a las condiciones y características de las imágenes en un entorno real.
- Se recomienda entrenar la red neuronal durante más de 130 épocas para alcanzar una precisión alta, como un mAP50 de 0.99. Un entrenamiento prolongado ayuda al modelo a aprender de los datos de manera más efectiva, lo que mejora la precisión de la detección de etiquetas.
- Es recomendable realizar pruebas en una PC antes de implementar el modelo en la Jetson Nano. Esto mejora la comprensión del comportamiento del modelo en entornos con más recursos y facilita la optimización antes de que el modelo comience a trabajar en la línea de producción.
- Se recomienda mantener varios conjuntos de datos entrenados para garantizar una amplia cobertura y precisión. Esto permite un mapeo de datos más completo y mejorar la precisión del sistema en varios escenarios de producción.

BIBLIOGRAFÍA

al, Y. T. (2023). Learning from Noisy Labels with Decoupled Meta Label Purifier,. 19934-19943. doi:doi: 10.1109/CVPR52729.2023.01909.

Alfonso López, B. (jun de 2021). *Detección precoz de covid-19 a partir de imágenes de radiografías de tórax mediante redes neuronales convolucionales*. Obtenido de http://hdl.handle.net/10609/132968

Alvarez Acosta, O. E. (2021). *Desarrollo de un sistema de visión artificial para la detección de hongos en plantas de cannabis medicinal mediante redes neuronales en ambientes controlados.* Obtenido de http://repositoriodspace.unipamplona.edu.co/jspui/handle/20.500.12744/5316

Anasi Nasimba, R. A., & Martínez Arellano, G. A. (03 de 2023). *Sistema de control para distribución de combustibles mediante la identificación de placas en vehículos utilizando visión artificial.* Obtenido de http://dspace.ups.edu.ec/handle/123456789/24505

Azurmendi Marquinez, I. (20 de 06 de 2022). *DESARROLLO DE ALGORITMOS DE APRENDIZAJE PROFUNDO BASADOS EN VISIÓN ARTIFICIAL PARA VEHÍCULOS AUTÓNOMOS DE INTERIORES.* Obtenido de http://hdl.handle.net/10810/57117

Calvo, D. (20 de 07 de 2017). *Red Neuronal Convolucional CNN*. Obtenido de https://www.diegocalvo.es/red-neuronal-convolucional/

Cleofe Huamani Huancara, Y. A. (30 de 03 de 2022). *Sistema automático para calificación de vino mediante Redes Neuronales*. doi:https://doi.org/10.48168/innosoft.s8.a51

Gama García, Á. M. (23 de 06 de 2020). *Aplicación de percepción mejorada para personas con problemas visuales usando deep learning y dispositivos vestibles.* Obtenido de http://hdl.handle.net/10045/107576

González, R., & Woods, R. (2002). *Digital image processing.* Prentice Hall. Retrieved from Prentice Hall.

Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection*, 25. doi:https://doi.org/10.3390/machines11070677

K. He, X. Z. (2016). *Deep Residual Learning for Image Recognition.* Obtenido de IEEE Conference on Computer Vision and Pattern Recognition (CVPR): doi: 10.1109/CVPR.2016.90.

Kathuria, A. (23 de 04 de 2018). Obtenido de https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

M. Mues, S. G. (2020). Identification of Spurious Labels in Machine Learning Data Sets using N-Version Validation,. *Identification of Spurious Labels in Machine Learning Data Sets using N-Version Validation,,* 1-7. doi:doi: 10.1109/ITSC45102.2020.9294223

Mateos Dominguez, A. (12 de 07 de 2023). *DETECCIÓN Y POSPROCESO DE INSCRIPCIONES Y ESCUDOS EN IMÁGENES DE FACHADAS USANDO REDES NEURONALES Y VISIÓN ARTIFICIAL.* Obtenido de https://hdl.handle.net/10115/23730

Mendoza Barrionuevo, A. (2021). *Sistema de detección y clasificación de obstáculos para JetBot basado en redes neuronales.* Obtenido de https://idus.us.es/handle/11441/126521

S. Suh, H. L. (2019). Robust Shipping Label Recognition and Validation for Logistics by Using Deep Neural Networks,. *Robust Shipping Label Recognition and Validation for Logistics by Using Deep Neural Networks,,* 4509-4513. doi:doi: 10.1109/ICIP.2019.8803412.

Santos, A. A. (2020). *Real-time traffic sign detection and recognition system for assistive driving.* Obtenido de https://www.astesj.com/publications/ASTESJ_050471.pdf

Trujillo, H. B., & Rodríguez, E. Y. (31 de 07 de 2023). *Programación de Movimiento de un Robot Paralelo Delta Mediante un Sistema de Visión Artificial con*

Reconocimiento de Gestos Manuales . Obtenido de https://doi.org/10.15665/rp.v21i1.2979

Uriarte de la Cruz, P. J. (11 de 2022). *Sistema de detección y clasificación de defectos superficiales en materiales metálicos para aplicaciones de ingeniería mediante visión por computadora.* Obtenido de https://hdl.handle.net/20.500.12930/10240

Villagomez Salazar, F. E., & Coronel Alarcón, J. R. (2022). *Sistema de monitoreo de medidas de bioseguridad con contador para permitir acceso a personas.* Obtenido de http://dspace.ups.edu.ec/handle/123456789/23847

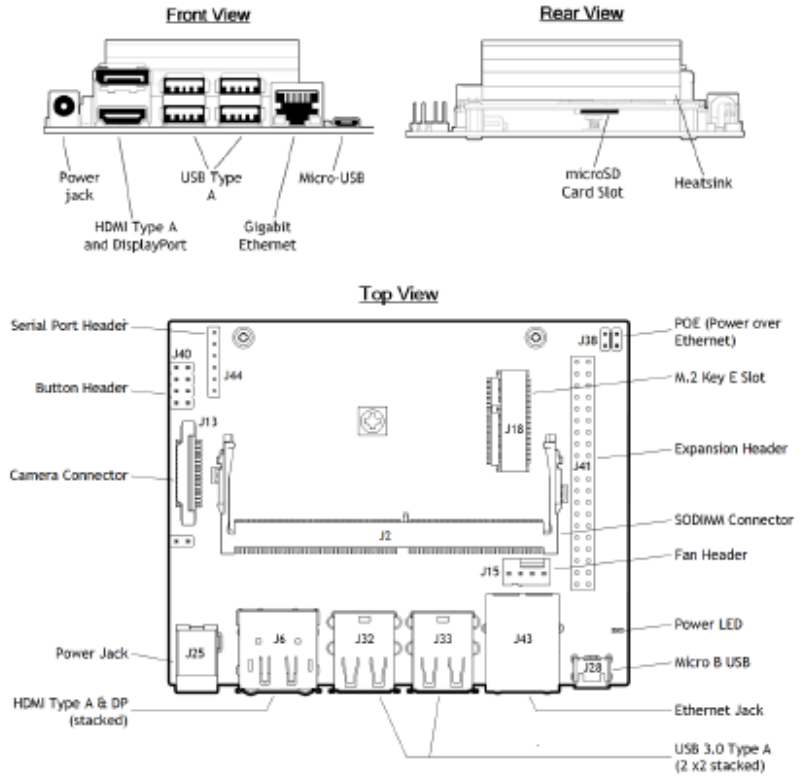
ANEXOS

INCLUDED IN THE BOX

- Jetson Nano module (P3448-0000)
- Reference carrier board (P3449)
- Small paper card with quick start and support information
- Folded paper stand for the developer kit

DEVELOPER KIT INTERFACES

Developer kit module and carrier board



CRONOGRAMA

CRONOGRAMA							
N.	**Actividades**	**Mes 1**	**Mes 2**	**Mes 3**	**Mes 4**	**Mes 5**	**Mes 6**
1	**Definición de objetivos**	x					
2	**Revisión de literatura**	x					
3	**Planificación del proyecto**	x					
4	**Recopilación de datos**		x				
5	**Preparación de imágenes**		x				
6	**Inicio del etiquetado**		x				
7	**Finalización del etiquetado**			x			
8	**Entrenamiento del modelo**			x			
9	**Pruebas iniciales**			x			
	Optimización del modelo				x		
10	**Pruebas con datos de validación**				x		
11	**Evaluación del modelo**					x	
12	**Ajustes finales del modelo**					x	
13	**Preparación e implementación**					x	
14	**Monitoreo y feedback**						x
15	**Ajustes post-implementación**						x

PRESUPUESTO

PRESUPUESTO						
No.	CODIGO	DENOMINACION	CANT	U	COSTO UNITARIO($)	COSTO TOTAL ($)
1		Kit de desarrollador para Jetson Nano -7 pulgadas Touch	1	U	$199,50	$199,50
2		SMAKN DC 5V/4A 20W Adaptador de fuente de alimentación conmutada 100-240 Ac	1	U	$10,00	$10,00
3		NVIDIA Jetson Nano Kit de desarrollador (945-13450-0000-100)	1	U	149,99	$149,99
4		TUBO ELECTRICO EMT DE ¾ (cantidad 3)	3	U	$3,73	$11,19
5		CABLE CONCENTRICO No 3x18 AWG FLEXIBLE	1	U	$3,73	$3,73
6		TABLERO PLASTICO GRIS 28X21X13CM C/PLAFON-FAMATEL	1	U	$29,05	$29,05
7		CAJA RECTANGULAR TIPO FS-3/4"	1	U	$3,49	$3,49
8		PLACA CIEGA METALICA P/CAJA RECTANGULAR FS	1	U	$0,66	$0,66
9		CONECTOR EMT 3/4"	1	U	$0,32	$0,32
10		RIEL DIN 35MM ACERO (1MT)	1	U	$2,16	$2,16
11		CLAVIJA 2P+T 16A 240V IP44-FAMATEL	1	U	$5,89	$5,89
12		TOMA EMPOTRABLE 2P+T 16A 240V 1P44-FAMATEL	1	U	$5,44	$5,44
13		TOMACORRIENTE DOBLE POLARIZADO 15A 127V MODUS STYLE-BTICINO	1	U	$2,19	$2,19
14		Suscripcion Google Colab Pro	1	U	$9,99	$9,99
15		Transporte y movilización	1	U	$87,66	$87,66
16		Articulos varios	1	U	$70,50	$70,50
					Sub-Total	$591,76
					IVA 12%	$71,01
					TOTAL	$662,77

INSTALACIÓN DE CUDA

Para instalar y tener listo CUDA, accedemos a la página web de Pytorch https://pytorch.org/ quien dará según la versión que estemos trabajando en nuestro entorno o seleccionamos la que se desee, según su estructura, versión de sistema operativo, el paquete, el lenguaje a trabajar, y la plataforma según su versión. Por último, se copia el comando y se le ejecuta según la necesidad.