ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS ADMINISTRATIVAS

DISEÑO Y SIMULACIÓN DE UN SISTEMA DE CONTROL DE CALIDAD BASADO EN REDES NEURONALES PARA LA CLASIFICACIÓN DE PANELES DECORATIVOS. CASO EMPRESA DEL SECTOR MADERERO.

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE

MAGISTER EN INGENIERÍA INDUSTRIAL Y PRODUCTIVIDAD

LUIS ANDRES CUMBAJIN SUAREZ

luis.cumbajin@epn.edu.ec

Director: Ing. DÁVILA FRÍAS ALEX VICENTE, Ph.D.

alex.davila@epn.edu.ec

QUITO, diciembre 2023

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación "DISEÑO Y SIMULACIÓN DE UN SISTEMA DE CONTROL DE CALIDAD BASADO EN REDES NEURONALES PARA LA CLASIFICACIÓN DE PANELES DECORATIVOS. CASO EMPRESA DEL SECTOR MADERERO." Desarrollado por Luis Andres Cumbajin Suárez, estudiante de la Maestría en Ingeniería Industrial y Productividad, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobado la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la defensa oral.

Ing. DÁVILA FRÍAS ALEX VICENTE, Ph.D.

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

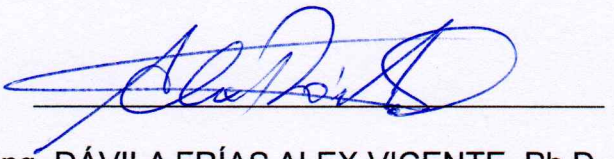
Yo, Luis Andres Cumbajin Suárez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de la Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

LUIS ANDRES CUMBAJIN SUÁREZ

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Luis Andres Cumbajin Suárez, bajo mi supervisión.



Ing. DÁVILA FRÍAS ALEX VICENTE, Ph.D.

AGRADECIMIENTO

A Dios por permitirme llegar a esta etapa de mi vida, mis profesores por compartir sus enseñanzas, fortaleciendo mis conocimientos, por brindarme una perspectiva de la vida mucho más amplia y así poder finalizar mis estudios con éxito.

A José Luis Ramírez quien me supo brindar su respaldo, apertura y predisposición durante este trabajo.

Finalmente, a mi tutor Ing. Alex Dávila, por confiar en mí, inculcarme de sus conocimientos y ser un gran guía a lo largo de este trabajo.

DEDICATORIA

A mi madre por ser mi pilar, quien, con su sudor, esfuerzo, dedicación y su lucha día a día, se convirtió en mi ejemplo de superación.

A mi esposa, por creer en mí, desde el inicio de este largo camino, siendo mi apoyo incondicional para mi superación profesional.

Wendy y Alexandra mis preciadas hermanas, que vieron el largo sacrificio y sirva de ejemplo para que logren sus objetivos y metas de vida.

Finalmente, a mi familia, abuelos y amigos que compartieron buenos y malos momentos, gracias por su apoyo y consejos de vida para mi superación personal y profesional.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

La gran cantidad de tableros decorativos laminados que se elabora en empresas que se dedican a producir paneles de madera, en conjunto con cumplir las expectativas de los clientes ha generado la obligación de mantener el control de calidad durante su elaboración con el objetivo de garantizar un producto fiable. En la etapa final del proceso de laminado de tableros, se lleva a cabo una verificación de calidad, en ocasiones no se detectan errores muy comunes, debido a la rapidez de la línea de producción y la verificación al ser visual es propensa a errores. La mala clasificación genera reclamos tanto del mercado nacional como del mercado de exportación, estos reclamos son causantes de reprocesos y retrasos prolongados no previstos a los clientes.

En la primera parte de la investigación se establece la pregunta de investigación, los objetivos y la hipótesis, seguido de un referencial teórico donde se establece los fundamentos conceptuales ligados al estudio. Posterior se indica como se diseñó y simulo el sistema de control de calidad mediante red neuronal, para identificar los tableros clase B, los cuales presentan fallas de calidad ya sea de papel roto o papel pegado, y poder comparar los beneficios que agregaría al proceso de producción, con los tableros identificados y no identificados durante su producción, cuya información se ha recolectado desde el 2021 hasta el 2022 mediante históricos de producción.

Para culminar la investigación se presenta una propuesta conformada por un presupuesto para una posible implementación del sistema de clasificación de tableros.

Palabras clave: Control de Calidad, Proceso, Producción, Simulación, Red neuronal.

ABSTRACT

The large amount of decorative laminated panels that are manufactured in companies dedicated to the production of wood panels, in conjunction with meeting the expectations of customers, has generated the obligation to maintain quality control during their production in order to ensure a reliable product. In the final stage of the board lamination process, a quality check is carried out, sometimes very common errors are not detected, due to the speed of the production line and the visual verification is prone to errors. Misclassification generates complaints from both the domestic and export markets, causing reprocessing and unanticipated long delays to customers.

The first part of the research establishes the research question, the objectives and the hypothesis, followed by a theoretical referential where the conceptual foundations linked to the study are established. Subsequently, it is shown how the quality control system was designed and simulated by means of a neural network, to identify the class B boards, which present quality failures either of torn paper or glued paper, and to compare the benefits that it would add to the production process, with the boards identified and not identified during its production, whose information has been collected from 2021 to 2022 by means of production records.

To conclude the research, a proposal is presented, consisting of a budget for a possible implementation of the board classification system.

Keywords: Quality control, Process, Production, Simulation, Neural Network.

DISEÑO Y SIMULACIÓN DE UN SISTEMA DE CONTROL DE CALIDAD BASADO EN REDES NEURONALES PARA LA CLASIFICACIÓN DE PANELES DECORATIVOS. CASO EMPRESA DEL SECTOR MADERERO.

1. INTRODUCCION

El control de calidad es un componente esencial en cualquier proceso de producción industrial, ya que garantiza que los productos o servicios cumplan normas y estándares de calidad previamente establecidas. Históricamente, las técnicas de control de calidad se basan en métodos estadísticos y sistemas de inspección manual. Sin embargo, con el desarrollo de la inteligencia artificial y el aprendizaje automático han surgido nuevas opciones para mejorar y optimizar el control de calidad.

En este contexto, se encuentran las redes neuronales, que son un tipo de algoritmo de aprendizaje automático basado en el funcionamiento del cerebro humano, han ganado prominencia. Las redes neuronales son capaces de aprender patrones y relaciones complejas en datos, lo que las convierte en una herramienta poderosa para el control de calidad en procesos de producción, donde la rapidez de la línea dificulta la inspección visual por parte de los colaboradores de la organización.

El uso de redes neuronales en el control de calidad ofrece varias ventajas. En primer lugar, permiten detectar defectos y anomalías en los productos de manera más eficiente y precisa que los métodos tradicionales. Las redes neuronales pueden analizar grandes cantidades de datos en tiempo real, identificando sutiles

variaciones y patrones irregulares que podrían ser pasados por alto por un inspector humano. Además, a medida que la red neuronal se expone a más datos, su capacidad de detección mejora, lo que significa que puede adaptarse a diferentes tipos de productos y situaciones de producción.

Esta investigación pretende demostrar cómo las redes neuronales pueden ser implementadas en un proceso de producción para el control de calidad. Se estudiará la arquitectura de rede neuronal convolucionales (CNN) para el análisis de imágenes.

1.1. Pregunta de investigación

¿Como influye el diseño y simulación del sistema de control de calidad basado en redes neuronales para mejorar la clasificación de paneles decorativos, en una empresa del sector maderero?

1.2. Objetivo General

Diseñar y simular un sistema de control de calidad basado en redes neuronales para la clasificación paneles decorativos.

1.3. Objetivos Específicos

- Realizar el análisis del proceso, para identificar causas principales de los problemas de clasificación de tableros.
- Diseñar un algoritmo basado en redes neuronales mediante clasificación de imágenes para la identificación de tableros con fallas de calidad.
- Segmentar imágenes para aislar la o las zonas de interés donde se encuentre el papel pegado y roto.

- Simular el sistema de clasificación de tableros mediante el entrenamiento de la red neuronal.
- Comparar los resultados obtenidos mediante métricas de evaluación de modelos de red neuronal para establecer el modelo con mejor desempeño.

1.4. Hipótesis

El sistema de control de calidad mediante redes neuronales disminuye los errores de clasificación de paneles decorativos.

2. MARCO TEÓRICO

2.1. Concepto de calidad

No es posible definir un concepto único de calidad, ya que depende fundamentalmente del enfoque que se trate, (Chávez, 2018) indica “La calidad de un producto se determina por la capacidad de satisfacer necesidades de los clientes, logrando mantener el menor coste posible”. En resumen, se tiene tres puntos sobresalientes, que van en conjunto con la calidad moderna; Clientes, expectativa del cliente o consumidor y el menor costo.

2.2. Características de calidad

Las cualidades y características que los consumidores buscan en los productos que adquieren se denominan atributos de calidad de tal modo, cuando se requiere un tablero decorativo, no se adquiere cualquier tipo o modelo de tablero, si no aquel que tenga todas las particularidades que responden a los clientes. Como indica (Maldonado, 2015), “Para que un producto sea de alta calidad, debe poseer las cualidades y características que satisfagan al comprador”. En el caso del tablero, son características de calidad, el espesor requerido, tonalidad del color, uniformidad

en la superficie, cero contaminaciones en el papel, entre otros. Por tanto, se evalúa la calidad de un producto en la medida que cumple con las características que el cliente busca, en otras palabras, en la medida que es “funcional”.

2.3. Control de calidad

Se define como el conjunto de herramientas, procedimientos, técnicas o equipos empleados en un proceso de producción para identificar errores en los productos.

La inspección es la fase inicial, durante la cual se confirma que el producto cumple los requisitos de diseño, el segundo paso es el control de calidad, “momento en el cual se comprueba que el producto cumpla con los requisitos de fabricación, la última etapa es el aseguramiento de la calidad, consiste en asegurar un nivel constante de calidad del producto”. (Juran, Dr. Frank M, & Bingham, 2021)

2.4. Mentalidad cero defectos

(Olabe, 2008)Su objetivo es la de eliminar los residuos en todas sus formas, en otras palabras, reducir las actividades que no agregan valor al proceso productivo, esta filosofía “cero defectos” requiere tener una mentalidad hacia el no-error.

Según (Maldonado, 2015), “Se debe evitar el uso común de la frase “errar es de humanos”, no se trata de castigar a los colaboradores cuando cometen errores, ya que hacerlo podría limitar su iniciativa. En cambio, se debe incentivar una conciencia de no equivocarse, bajo el concepto cero defectos.

2.5. Estadística descriptiva (obtención de datos)

Según (Gutiérrez Pulido, 2014) “A la hora de tomar una decisión crítica como determinar la causa de un problema, hay que disponer de información que permita

determinar las condiciones en que surge el problema, es decir, identificar su regularidad estadística y fuentes de variabilidad".

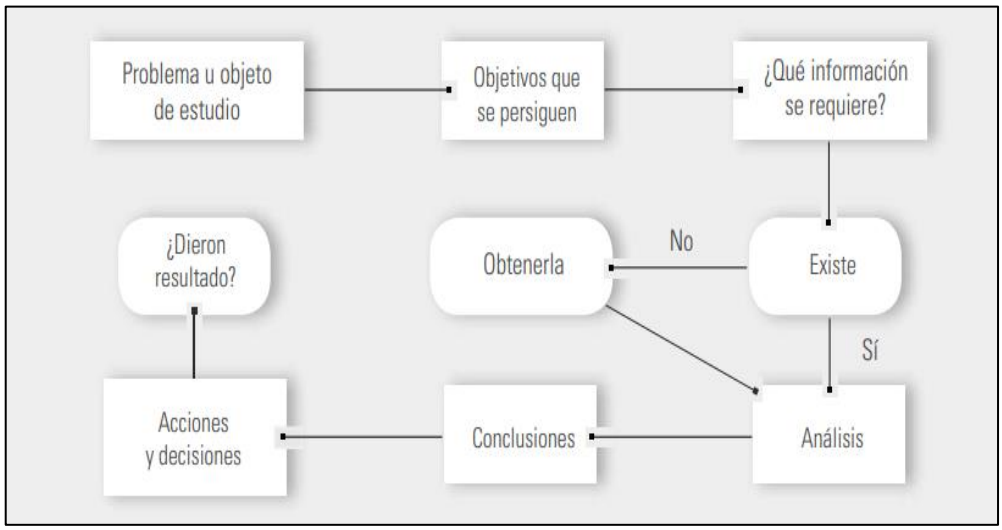
Es fundamental tener claro el objetivo que se busca y los recursos junto con el tiempo disponible antes de obtener información sobre un problema o situación específica. De esta manera, se podrá abordar el problema de manera más efectiva y eficiente.

Una vez localizado el problema, definido los objetivos e identificado el tipo de información necesaria, es común que surjan problemas como la forma de obtener dicha información, la cantidad necesaria y cómo analizarla adecuadamente.

Figura 1

La toma de decisiones y la estadística descriptiva

Fuente: (Gutierrez Pulido & De la Vara Salazar, 2013)



La figura 1, muestra el papel que realiza la estadística descriptiva, se observa que su principal objetivo es apoyar en la toma de decisiones o satisfacer necesidades de información sobre un proceso, indicando los pasos a seguir para una correcta planeación de solución a problemas.

2.6. Tipos de variables

Su clasificación viene dada por variables cuantitativas y cualitativas. Las variables cualitativas son aquellas características no numéricas, por ejemplo, tipo de producto, marca de producto entre otros, mientras que las variables cuantitativas pueden registrarse numéricamente sus características, por ejemplo, peso de un lote, o en este caso número de tableros defectuosos junto con los tipos de errores de calidad en su superficie.

2.7. Medidas de tendencia central

Con las medidas tomadas de fallas de calidad del tablero de tipo cuantitativo, el primer paso es encontrar la tendencia central de los datos, identificando que datos tienden a agruparse. “Esto permite conocer el valor y evaluar si el proceso se encuentra centrado, es decir conocer si la tendencia central de la variable de salida es igual o está próxima a los valores nominales deseados” (Gutierrez Pulido & De la Vara Salazar, 2013)

Se tiene X1, X2, X3…, Xn las cuales son observaciones numéricas de una muestra, siendo la tendencia central la media o promedio muestral, que es igual a la siguiente formula:

$$\bar{X} = \frac{x1 + x2 + x3 \dots \dots \dots \dots + xn}{n} = \frac{\sum_{i=1}^{n} xi}{n}$$

Es decir, la media muestral se obtiene sumando todos los datos y el resultado se divide entre el número de datos (n).

2.8. Medidas de dispersión o variabilidad

Dentro de un conjunto de datos es necesario conocer el nivel de diferencia entre si, en otras palabras, conocer su variabilidad o dispersión, siendo un análisis vital en un estadístico de datos.

“Para ello se emplea la desviación estándar muestral, indicando la medida más usual de la variabilidad, muestra que tan disperso se encuentran los datos con respecto a la media muestral” (Gutierrez Pulido & De la Vara Salazar, 2013). Su símbolo es S, su fórmula es la siguiente:

$$S = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2}{n - 1}}$$

Donde $x_1, x_2, \ldots, x_n$, son las observaciones de la muestra, mientras que $\bar{x}$ es la medida muestral. Mientras más grande sea S, mayor dispersión o variabilidad existirá entre los datos.

2.9. Cartas de control

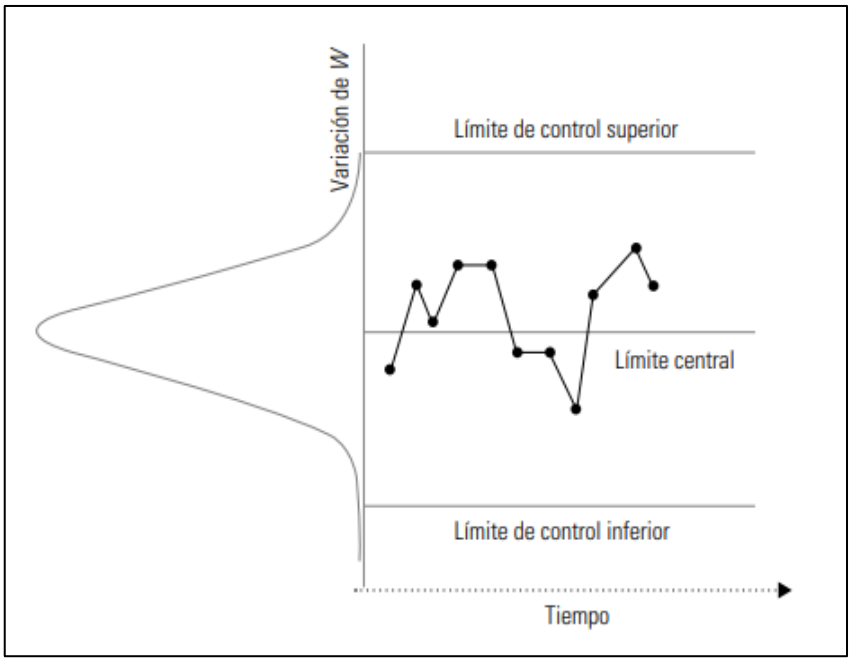
Según (Gutiérrez Pulido, 2014) “Las cartas de control grafican lo que sirve, para observar y analizar la variabilidad y el comportamiento de un proceso a través del tiempo”.

Un componente de la carta de control es los límites de control, estos indican la variabilidad esperada para un estadístico, como la medio o el rango de una variable de proceso, estos son calculados a partir de los datos de la muestra y son diferentes a las especificaciones para la variable.

Figura 2

Límites de una carta de control

Fuente: (Gutierrez Pulido & De la Vara Salazar, 2013)



La detección de un punto fuera de los límites de control en una carta de control no es el único indicador de un problema. También es importante considerar cualquier patrón o formación de puntos que tenga una probabilidad muy baja de ocurrir en condiciones normales, lo cual puede indicar la presencia de causas especiales y es una señal de alerta para posibles cambios en el proceso.

2.10. Carta U (número promedio de defectos por unidad)

Este tipo de carta de control es utilizada cuando el tamaño de subgrupo no es constante, se caracteriza por analizar la variación del número promedio de defectos por unidad. Se calcula de la siguiente manera:

$$u_i = \frac{c_i}{n_i}$$

Donde c_i es la cantidad de defectos en el subgrupo i y n_i es el tamaño del subgrupo i. Para calcular los límites es necesario estimar la media y la desviación estándar del estadístico u_i que, en el supuesto de que c_i sigue una distribución Poisson, resultan ser:

$$u_{ui} = \bar{u} = \frac{Total\ de\ defectos}{Total\ de\ articulos\ inspeccionados}$$

$$\sigma_{ui} = \sqrt{\frac{\bar{u}}{n}}$$

Donde n es el tamaño del subgrupo. De tal forma que los límites de control están dados por las siguientes ecuaciones:

$$Limite\ de\ control\ superior = LCS = \bar{u} + 3\sqrt{\frac{\bar{u}}{n}}$$

$$Línea\ central = \bar{u}$$

$$Limite\ de\ control\ inferior = LCI = \bar{u} - 3\sqrt{\frac{\bar{u}}{n}}$$

2.11. Implementación y operación de una carta de control

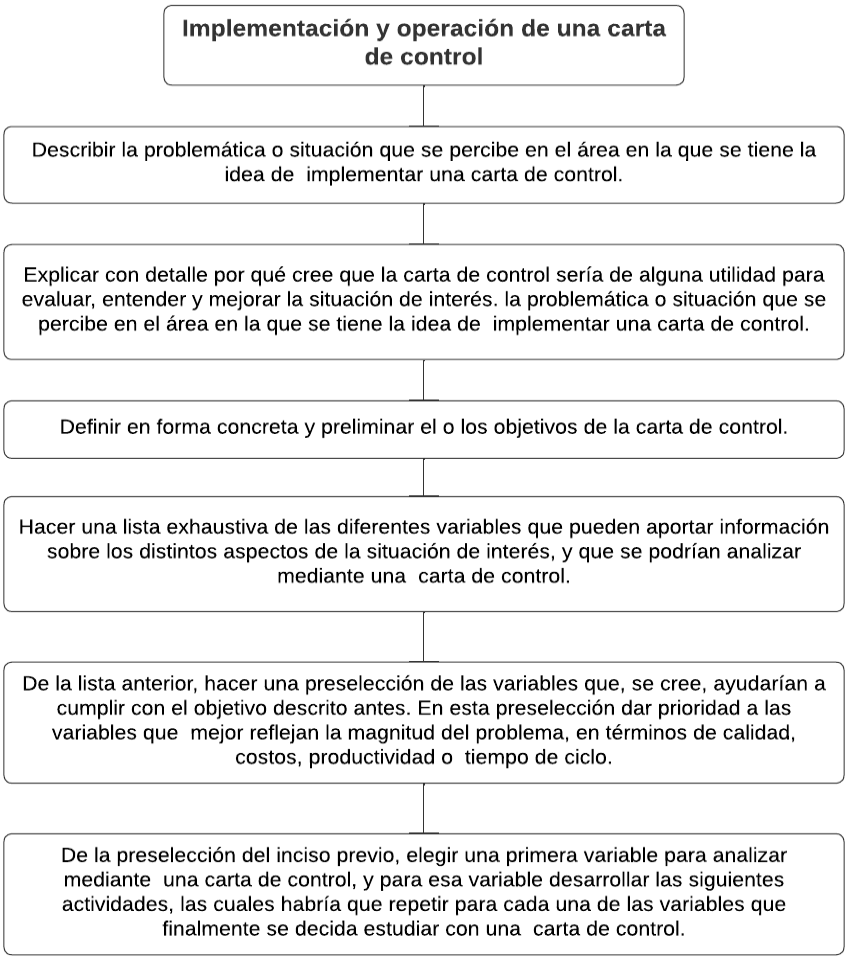
Para la implementación (Gutiérrez Pulido, 2014) indica que “Una carta de control es beneficioso en la medida en que satisface un requisito visto por los responsables del proceso y dependerá, por supuesto, de la eficacia con que se utilice y ejecute”.

Los pasos para una correcta implementación son indicados en la figura 3:

Figura 3

Implementación de una carta de control

Fuente: Elaboración propia



2.12. Inteligencia Artificial

(Badaro, Ibañez, & Aguero, 2013) indica, “el termino inteligencia artificial, (IA), se refiere a la capacidad de emular las funciones inteligentes del cerebro humano”.

Mediante técnicas de predicción y aprendizaje basadas en intrincados algoritmos, su objetivo es poder crear comportamientos capaces de imitar acciones, pensamientos e incluso la toma de decisiones humanas.

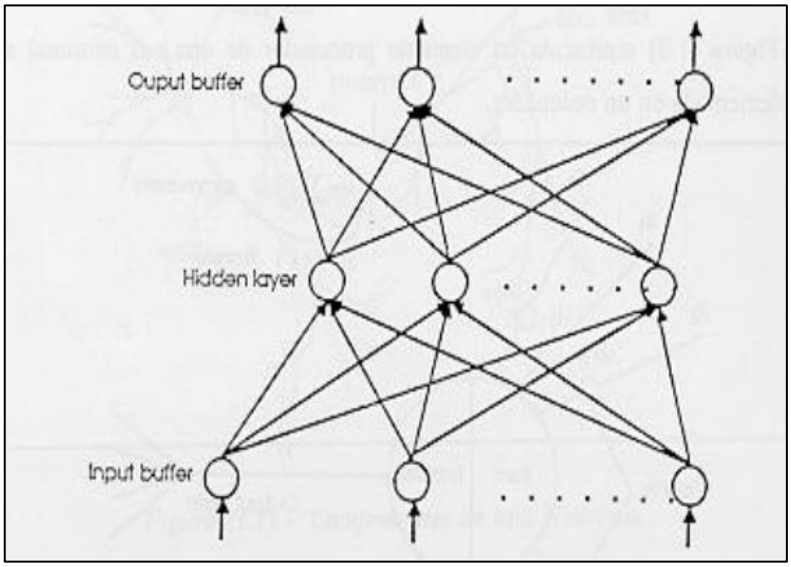
2.13. Red Neuronal

Se entiende por red neuronal artificial (ANN), o redes neuronales simuladas (SNN), a un subconjunto de machine learning, (Olabe, 2008) indica que “se encuentra dentro de los algoritmos de Deep learning. Cuyo nombre y estructura se encuentran inspirados en el cerebro humano, imitando la forma en que las neuronas biológicas se transmiten entre sí”.

Figura 4

Red neuronal simple

Fuente: (Olabe, Redes Neuronales Artificiales y sus Aplicaciones)



Según (Costa Albuquerque, Auzuir, & Cortez, 2009). “Las redes neuronales artificiales buscan imitar el comportamiento del cerebro, el cual está compuesto por neuronas que almacenan y procesan grandes cantidades de información, generando sistemas complejos, creando una estructura donde se utilizan elementos simples”.

Los modelos más utilizados son la multicapa y las redes neuronales convolucional, esta última es la especializada para el trabajo con imágenes.

En la práctica las redes neuronales artificiales aprenden del mismo modo que las neuronas del ser humano, exponiéndose a ejemplos, las entradas externas se reciben, procesan y activan de forma similar al cerebro humano.

2.14. Machine Learning o aprendizaje automático

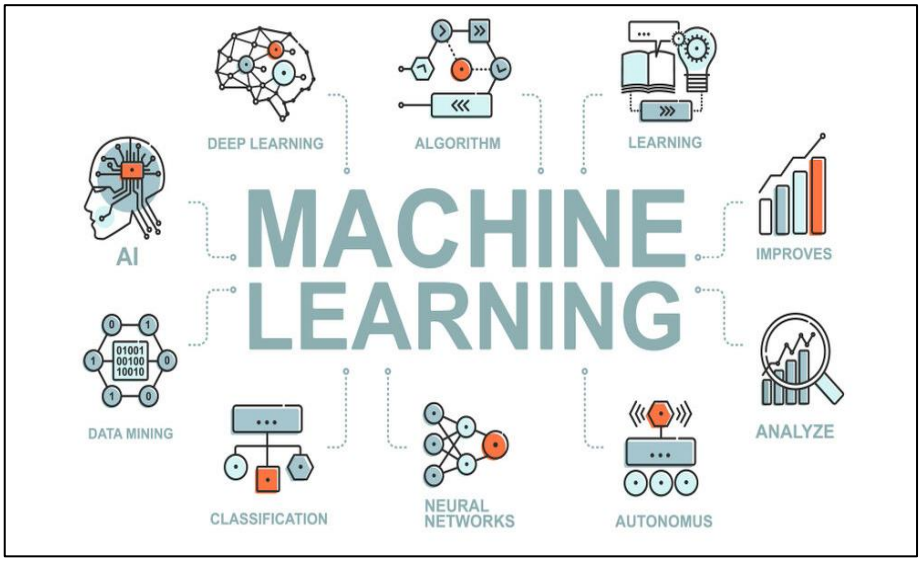
"El Machine Learning, permite identificar patrones aplicando algoritmos que clasifican cada factor en función de su grado de influencia, y esto se logra mediante el aprendizaje continuo y la mejora constante del proceso". (Solutions, 2018)

En términos sencillos, el Machine Learning consiste en cargar datos a un computador y utilizar técnicas estadísticas para ayudarle a "aprender" mejorando progresivamente en una tarea, sin tener la necesidad de ser programado específicamente para una tarea, logrando así eliminar una cantidad elevada de líneas de código.

Figura 5

Estructura del machine Learning

Fuente: (LIDERLOGO (SEO y Machine Learning: Cómo la Inteligencia Artificial Está Cambiando la Optimización de Búsqueda), 2023)



2.15. Sistemas Difusos

Se basa en la observación de un escenario y su relatividad, tomando valores al azar, relacionándolos entre sí, (Helena, Cezar, & Miranda, 2015) argumentan que “Los conjuntos difusos se han utilizado en el trabajo con imágenes, ya que no se limita numéricamente el desarrollo, siendo útil con imágenes, debido a que sus características no se pueden delimitar de una forma exacta”, Funciona a partir de reglas heurísticas inicialmente de Si (antecedente), Entonces (consecuente), siendo conjuntos difusos puros o ya sea del resultado de alguna operación.

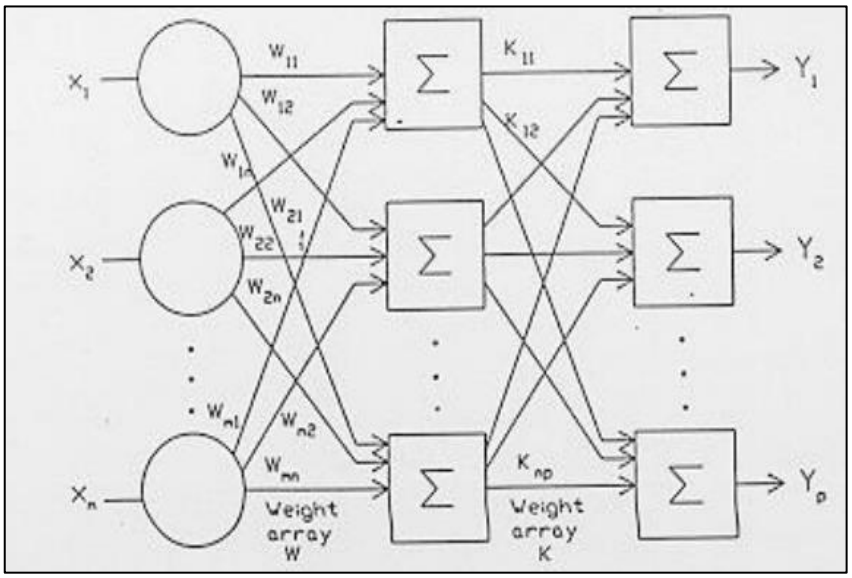
2.16. Red neuronal multicapa

“Conocida como MultiLayer Perceptron o MLP, Se basa en la unión de neuronas organizadas y distribuidas en diferentes capas” (Islam , Hannan , Basri , & Hussain , 2014). Funciona imitando el comportamiento del sistema utilizando entradas y salidas previamente conocidas, teniendo la capacidad de clasificar elementos.

Figura 6

Red neuronal multicapa

Fuente: (Olabe, Redes Neuronales Artificiales y sus Aplicaciones)



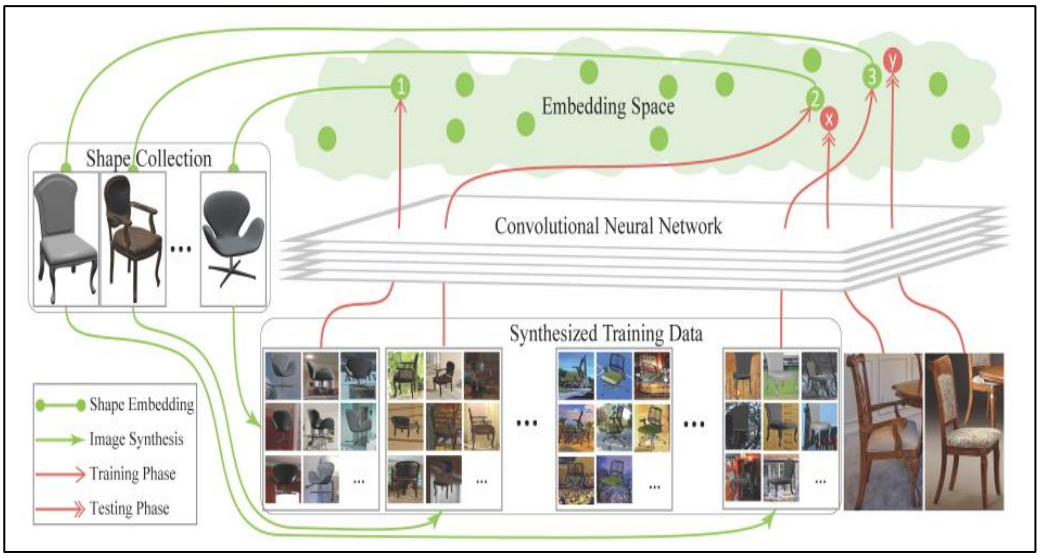
2.17. Red neuronal convolucional

"Esta red convolucional (CNNs) principalmente es utilizado en el análisis de imágenes, destacando su uso en los procedimientos de clasificación y reconocimiento". (Zhang, Zeng, Zhang, Zhang, & Wu, 2017).

Figura 7

Funcionamiento red neuronal convolucional

Fuente: (García Villanueva & Romero Muñoz , 2020)



La figura 7 muestra, la estructura que presenta la relación entre un objeto 3D y varias imágenes patrón, se utilizan estas imágenes para ser comparadas con fotografías de muestra y así poder detectar el objeto en uno de los grupos creados, con un porcentaje de precisión determinado.

2.18. Arquitectura de red neuronal convolucional

"Se caracteriza por extraer las características propias de un elemento que lo diferencian de otros, proceso que habitualmente realizan las personas con la vista y que las CNNs tratan de emular, cuando se trata de imágenes se utiliza este tipo

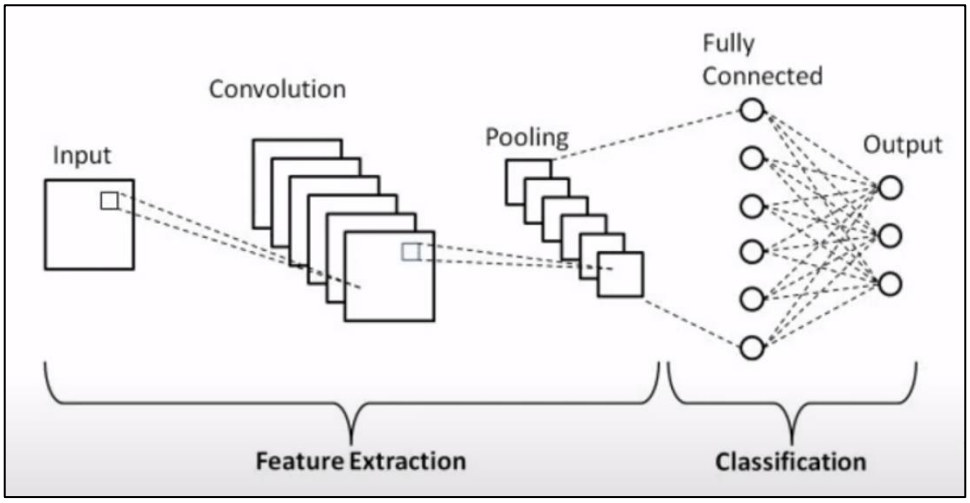
de redes porque permite el manejo de grandes estructuras de datos". (Ferrari, Lombardi, & Signoroni, 2017).

La figura 8 muestra la arquitectura de una red neuronal convolucional, donde cada uno de los bloques representa una capa diferente de la red de neuronas convolucionales, conformado por una entrada, convolución, pooling, fully connected y una o varias salidas.

Figura 8

Arquitectura red neuronal CNN

Fuente: (Carrión, 2020)



2.19. Operación de convolución

(Ian , Yoshua , & Aaron , 2016) indica que "la convolución es una operación sobre dos funciones de un argumento de valor real"

Para indicar esta operación se tiene de ejemplo lo siguiente; suponiendo que se rastrea una nave espacial con un sensor laser, que proporciona una salida x(t), la posición de la nave en el tiempo t. Tanto x como t tienen valores reales, en otras palabras, se obtiene lecturas diferentes del sensor en cualquier instante.

Ahora suponiendo que el sensor laser es ruidoso, para obtener la estimación menos ruidosa de la posición de la nave, lo ideal sería promediar las mediciones, siendo las mediciones más recientes las más relevantes, entonces se debe tener un promedio ponderado que de más peso a las mediciones recientes, para lo cual se hace una función de ponderación w(a), donde a es la antigüedad de una medición, si se aplica la operación promedio ponderada en cada momento, se obtiene una nueva función (s) que proporciona una estimación suavizada de la posición de la nave espacial.

$$s(t) = \int x(a)w(t-a)da$$

Esta operación se llama convolución, la operación de convolución típicamente es denotada con un asterisco.

$$s(t) = (x * w)(t)$$

Para el ejemplo, w debe ser una función de densidad de probabilidad valida o el resultado no será un promedio ponderado, además w debe ser 0 para todos los argumentos negativos, o mirará hacia el futuro. En general “la convolución se define para cualquier función para la que se define la integral anterior y puede usarse para otros fines además de tomar promedios ponderados” (Goodfellow , Bengio, & Courville, 2016).

En la terminología de redes convolucionales, el primer argumento de este ejemplo, la función (x) de la convolución a menudo se denomina entrada y el segundo argumento la función (w) como núcleo o (kernel).

Para ser más realista en el ejemplo se supone que el láser proporcionara una medición una vez por segundo, el índice de tiempo (t) solo puede adoptar

valores enteros. Si ahora asumimos que (x) y (w), están definidos solo en el número entero (t), podemos definir la convolución discreta:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

En las aplicaciones de aprendizaje automático, la entrada suele ser una matriz multidimensional de datos y el kernel suele ser una matriz multidimensional de parámetros que son adaptados por el algoritmo de aprendizaje, estas matrices multidimensionales son denominadas como tensores, debido a que cada elemento de la entrada y del kernel debe almacenarse explícitamente por separado, generalmente se asume que las funciones son 0 en todas partes excepto en el conjunto finito de puntos para los cuales se almacenan los valores. Esto significa que en la práctica podemos implementar la suma infinita como una suma sobre un numero finito de elementos de una matriz.

Finalmente se utiliza convoluciones en más de un eje a la vez, por ejemplo, si se usa una imagen bidimensional (I) como entrada, probablemente también se use un kernel bidimensional K:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I\,(m,n)K(i-m, j-n)$$

La convolución es conmutativa, lo que significa que podemos escribir de manera equivalente:

$$S(i,j) = (K * I)(i,j) = \sum_{m}\sum_{n} I\,(i-m, j-n)K(m,n)$$

Esta última formula es más sencilla de implementar en una biblioteca de aprendizaje automático, debido a que hay menos variación en el rango de valores validos de (m) y (n).

(Goodfellow , Bengio, & Courville, 2016), indica que la propiedad conmutativa de la convolución “surge porque se invierte el kernel en relación con la entrada, en sentido de que (m) aumenta, el índice en la entrada aumenta, pero el índice del kernel disminuye, la única razón para invertir el kernel es para obtener la propiedad conmutativa”.

En cambio, en muchas bibliotecas de redes neuronales se implementa una función relacionada llamada correlación cruzada, que es lo mismo que la convolución, pero sin invertir el kernel:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I\,(i+m, j+n)K(m,n)$$

“Muchas bibliotecas de aprendizaje automático implementan la correlación cruzada, pero la llaman convolución”. (Ian , Yoshua , & Aaron , 2016).

2.20. Red YOLOv5

“Este tipo de arquitectura de red neuronal convolucional se caracteriza por la detección de objetos de forma precisa y rápida, el cual fue desarrollada por Redmond Joseph” (Hui, 2018), debido a que se compone de una arquitectura muy rápida, es ideal para la detección de fallas de calidad en la línea de producción de laminado debido a la rapidez de la misma. YOLO predice de forma simultánea múltiples cuadros delimitadores previamente colocados en las imágenes de

entrenamiento y las probabilidades de la clase de falla de calidad que delimitan los cuadros limitadores respectivamente.

2.21. Funcionamiento Red YOLOv5

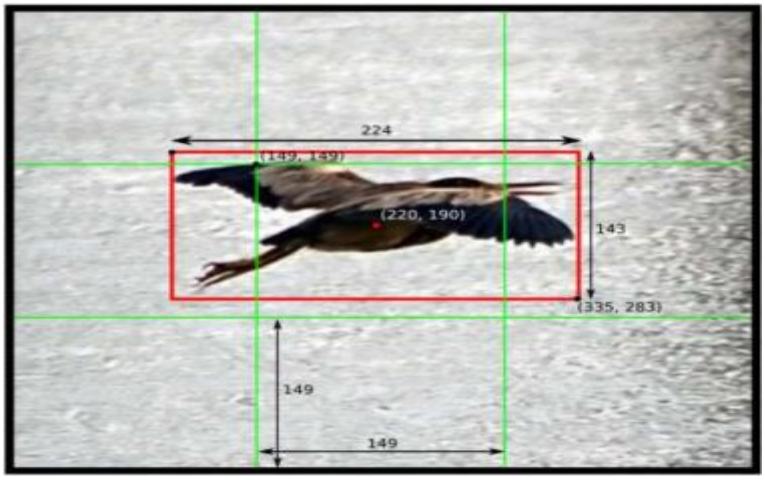
YOLO se destaca por su enfoque de detección de objetos en una sola pasada ("You Only Look Once"). A diferencia de los modelos de dos etapas que primero proponen regiones de interés y luego clasifican esas regiones, YOLO realiza ambas tareas simultáneamente en una sola inferencia.

(Raneros, 2021) indica que la imagen se divide en una rejilla de tamaño S x S, de tal manera que si un objeto cae en una celda de la cuadricula, esa celda es responsable de detectar el objeto, como se aprecia en la figura 9.

Figura 9

Ejemplo cálculo de coordenadas del cuadro de una imagen de tamaño 448x448 pixeles y S=3

Fuente: (Raneros, 2021)

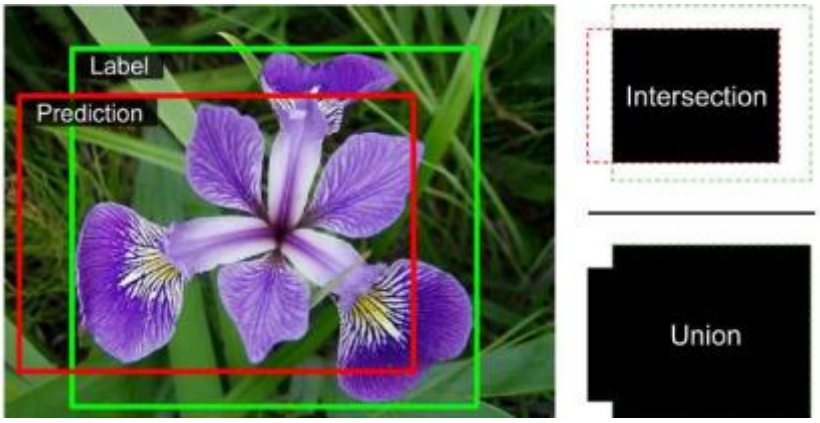


Posterior se realiza una intersección del cuadro delimitador y la anotación realizada sobre la imagen en la etapa de etiquetado, como se indica en la figura 10.

Figura 10

Descripción de la intersección sobre la unión

Fuente: (Raneros, 2021)



2.22. Arquitectura de red YOLOv5

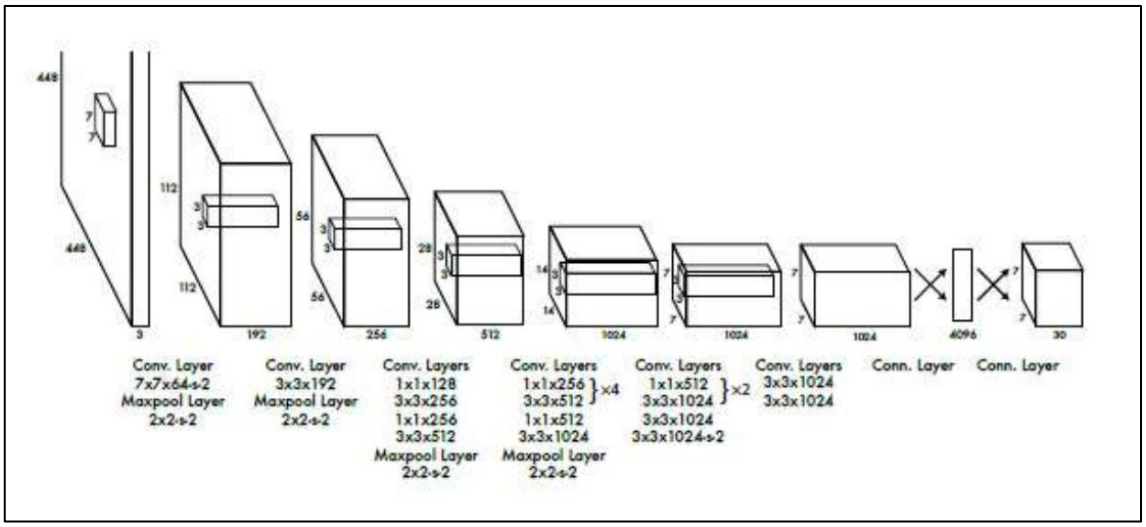
Su modelo se basa en red neuronal convolucional, debido que solo cuenta con capas convolucionales denominadas FCN (Fully convolutional network, (Raneros, 2021), nos indica que “las capas iniciales extraen características de la imagen y las capas completamente conectadas predicen las probabilidades de salida y las coordenadas. Esta red se inspira en el modelo GoogleNet usado en la clasificación de imágenes”.

YOLOv5 se compone de 24 capas convolucionales y 2 capas completamente conectadas, con una convolución de 1x1 para reducir la profundidad de los mapas de características. Continua con una capa de convolución de 3x3. Dichas capas se van alternando como se puede observar en la figura 11.

Figura 11

Esquema de la arquitectura de la red neuronal YOLOv5

Fuente: (Redmon, Divvala, Girshick, & Farhadi, 2015)



“La última capa de convolución genera un tensor de (7,7,1024), finalmente el tensor se reduce aplicando dos capas conectadas, obteniendo en la salida un tensor de tamaño 7x7x30” (Redmon, Divvala, Girshick, & Farhadi, 2015), como se aprecia en la figura 11.

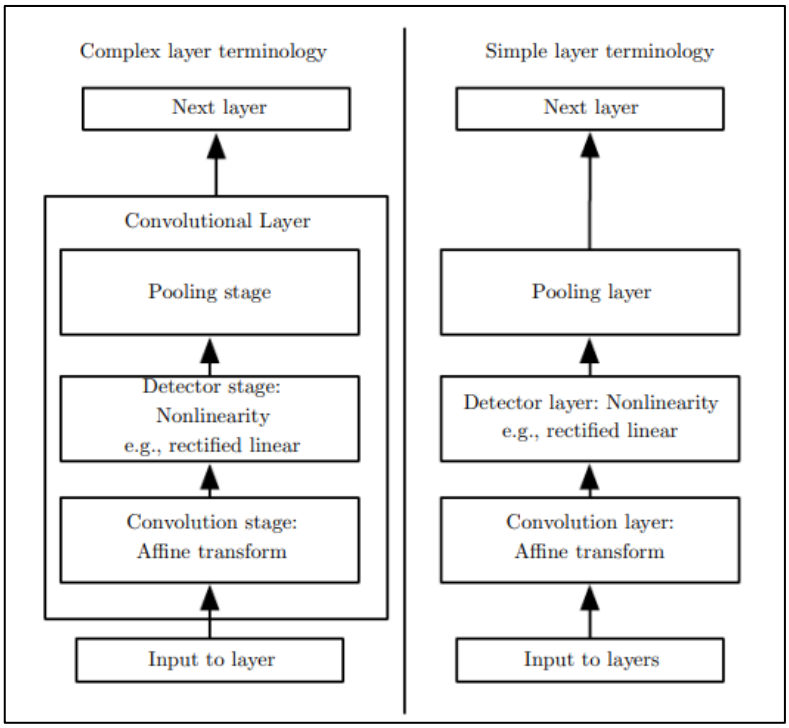
2.23. Agrupación

En una capa típica de red convolucional consta de tres etapas como muestra la figura 12.

Figura 12

Componentes de una capa de red neuronal convolucional típica

Fuente: (Ian , Yoshua , & Aaron , 2016)



(Ian , Yoshua , & Aaron , 2016), indica que “en la primera etapa, la capa realiza varias convoluciones en paralelo para producir un conjunto de activaciones lineales, en la segunda etapa, cada activación lineal se ejecuta a través de una función de activación no lineal, como la función de activación lineal rectificada, en la tercera etapa se utiliza una función de agrupación para modificar aún más la salida de la capa”.

2.24. Herramienta CLAHE

El método CLAHE (Contrast Limited Adaptive Histogram Equalization) es una técnica de procesamiento de imágenes utilizada para mejorar el contraste local en una imagen, que ajusta el contraste de toda la imagen por igual, el CLAHE ajusta

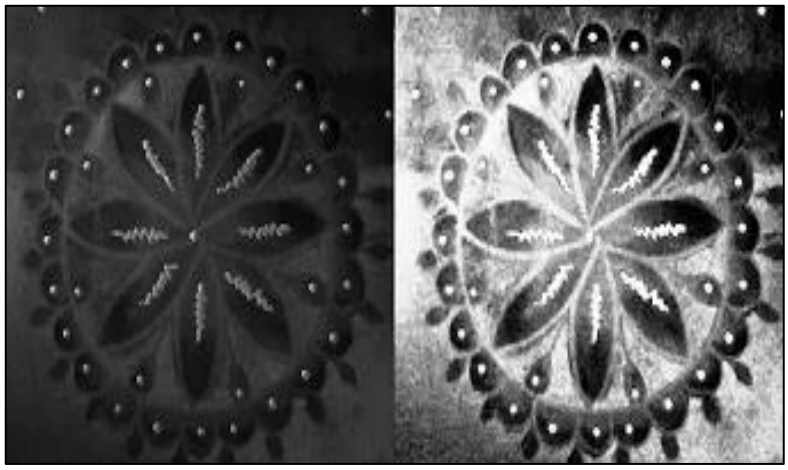
el contraste en regiones locales de la imagen de manera adaptativa. (Unipython, Ecualización de histogramas, 2018).

Consiste en dividir la imagen en regiones que no se superponen sobre otras que tengan tamaños casi iguales, mejorando los detalles como se muestra en la figura 13.

Figura 13

Ejemplo aplicando método CLAHE

Fuente: Unipython.com



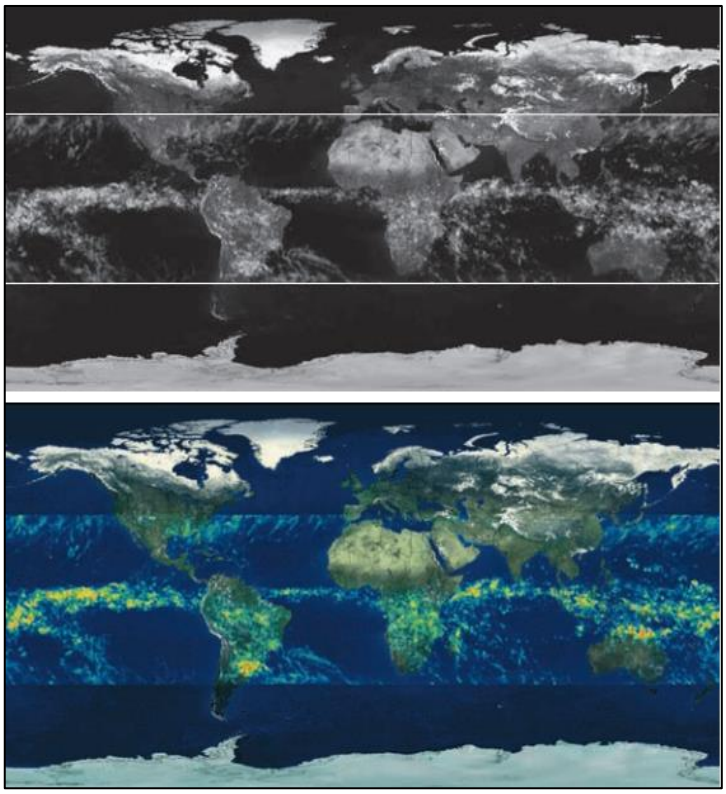
2.25. Escala de grises

En la mejora de imágenes, “la escala de grises se refiere a la representación de una imagen utilizando sólo tonos de gris en lugar de colores. Cada píxel de una imagen en escala de grises está representado por un único valor de intensidad que va del negro (el valor más bajo) al blanco (el valor más alto)”. (Gonzales & Richard E, 2018), la escala de grises es una forma de simplificar la información de color de una imagen, lo que puede ser útil en diversas aplicaciones de procesamiento y mejora de imágenes, como indica la figura 14.

Figura 14

Ejemplo de escala de grises

Fuente: (Gonzales & Richard E, 2018, pág. 424)



2.26. Métricas para evaluar el modelo

2.27. Precisión y Recall

(Garcia , Fernandez, Luengo, & Herrera, 2009) indica que “La precisión es la métrica más utilizada para estimar el rendimiento de los sistemas de aprendizaje en los procesos de clasificación”. En otras palabras, describe lo cerca que está una medición de ser exacta.

En la siguiente ecuación se puede apreciar cómo se puede calcular:

$$Precisión = \frac{TP}{TP + FP} = \frac{TP}{all\ detection}$$

Mientras que la sensibilidad (Recall), indica la fracción de objetos, de los que en realidad existen, localiza el detector, la siguiente ecuación indica como calcularla:

$$Recall = \frac{TP}{TP + FP} = \frac{TP}{all\ ground\ truths}$$

Donde:

TP = True positive

TN = True negative

FP = False positive

FN = False negative

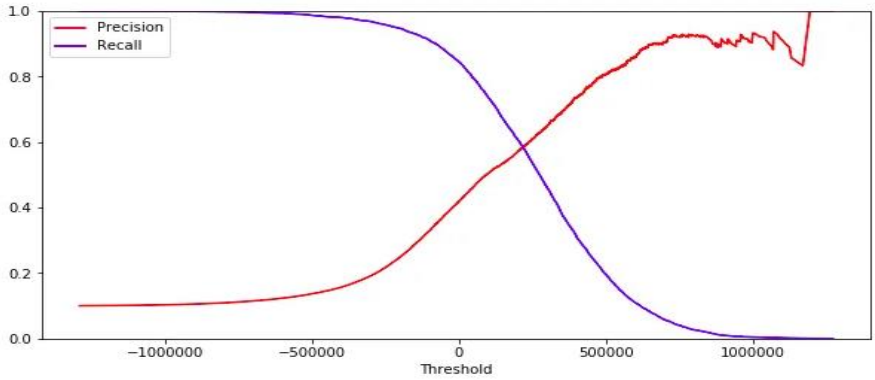
2.28. Curva de precisión Recall

Esta curva va a depender en general del tipo de configuración que se adopte, es decir mientras se tenga “una configuración de mayor precisión, se tendrá una disminución del recall es decir el sistema será menos sensible y detectara menos objetos, al contrario, si la configuración es para tener más sensibilidad, el sistema será menos preciso en otras palabras al momento de la clasificación será menos estricto” (Ramírez, 2018).

Figura 15

Equilibrio precisión-recall en función del umbral de decisión.

Fuente: (Ramírez, 2018)



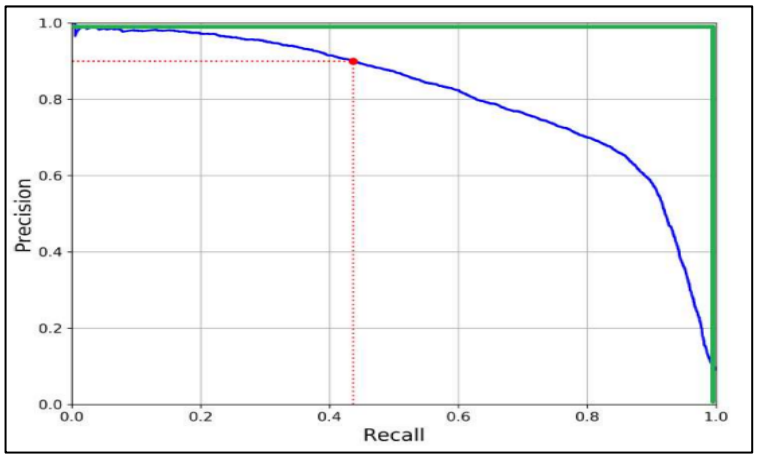
En la figura 15 se puede apreciar como el umbral de decisión afecta en base a lo que se desea tener en el sistema puede cambiar y variar en los resultados de clasificación final, una correcta decisión al momento de elegir el umbral nos ayudara a tener un equilibrio entre precisión y sensibilidad.

El sistema se considera bueno, si es directamente proporcional, en otras palabras, si se aumenta el recall la precisión se va manteniendo alta, como indica la figura 16.

Figura 16

Ejemplo de una curva de un sistema de detección (azul) y la curva ideal (verde)

Fuente: (Raneros, 2021)



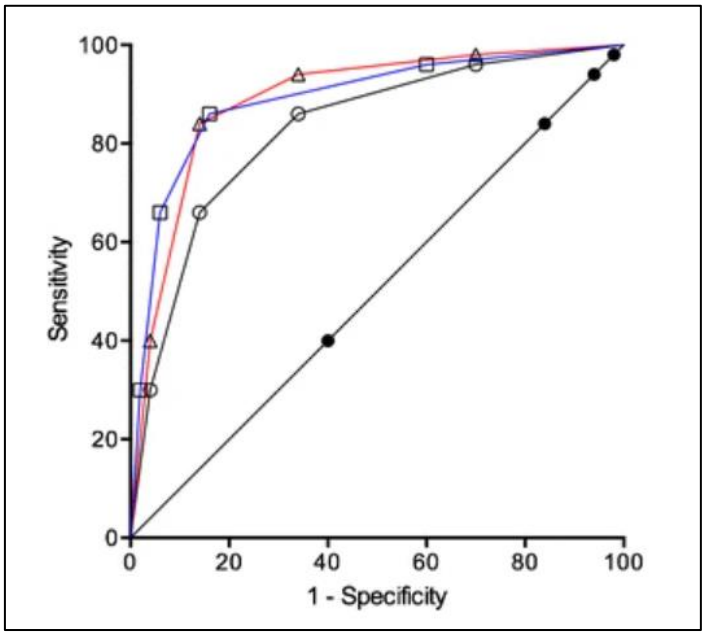
2.29. Curva ROC/AUC

Según (Steve, Douglas G, & Susan, 2015) “La curva ROC, es un tipo de grafico el cual comprende la tasa de verdaderos positivos (eje y), frente a la tasa de falsos positivos correspondiente (eje x), es decir sensibilidad versus especificidad como muestra la figura 17.

Figura 17

Ejemplo de curva ROC/AUC

Fuente: (Steve, Douglas G, & Susan, 2015)



(Benavides, 2017), indica que “la sensibilidad cuantifica la proporción de individuos que presenta el evento de interés y son clasificados como portadores de dicho evento, mientras que la especificidad cuantifica la proporción de individuos que no lo presentan y son clasificados por la prueba como tal”.

De manera practica se puede definir de la siguiente manera:

Sensibilidad. - Es la probabilidad de, dado un tablero con falla de calidad, la prueba lo clasifique como roto o papel pegado.

Especificidad. - Es la probabilidad de, dado un tablero sin falla de calidad, la prueba no lo clasifique como roto o pegado, en otras palabras, la calidad es aceptable y no presenta fallas.

2.30. Precisión media (Map)

Esta grafica es otro de las formas de evaluar si un detector de objetos es ideal o deficiente, (Raneros, 2021) indica en su trabajo que “se trata de encontrar el área bajo la curva de precisión-recall, de tal forma que se llega a una medida numérica, donde facilita el trabajo de conocer si el detector es bueno o no”.

2.31. Matriz de confusión

Según (López, 2018) “La matriz de confusión es el método más popular y aceptado para informar sobre la corrección temática de los productos obtenidos a partir de la categorización precedente de imágenes”.

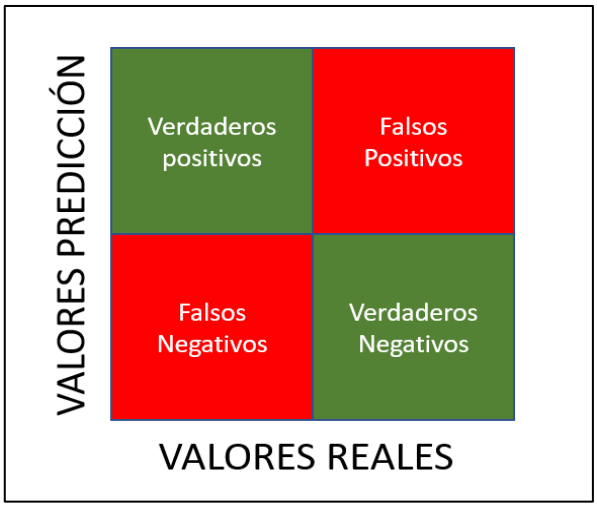
2.32. Construcción de la matriz

Para entrenar una red neuronal, y validar la veracidad de los aciertos es necesario aplicar esta herramienta para validar el desempeño del modelo utilizado. La matriz de confusión es simplemente una tabla donde indica que tan confundido se encuentra el modelo al momento de la clasificación, indicando los aciertos y desaciertos para cada una de las categorías, su estructura se indica en la figura 18.

Figura 18

Matriz de confusión

Fuente: (Arce, 2019)



Un tablero “normal” es clasificado correctamente como “normal”. A esto lo llamaremos verdadero positivo.

Un tablero “normal” es clasificado incorrectamente como “anormal”. A esto lo llamaremos un falso negativo.

Un tablero “anormal” es clasificado correctamente como “anormal”. A esto lo llamaremos un verdadero negativo.

Y un tablero “anormal” es clasificado incorrectamente como “normal”. A esto lo llamaremos un falso positivo (porque realmente no es un caso positivo).

En resumen, los verdaderos positivos y los verdaderos negativos serán los aciertos, mientras que los falsos positivos y los falsos negativos serán los desaciertos. (Sotaquirá, 2022)

3. METODOLOGÍA

La presente investigación “Diseño y simulación de un sistema de control de calidad basado en redes neuronales para la clasificación de paneles decorativos. caso empresa del sector maderero”, es de naturaleza cuantitativa, (Hernández Sampieri, Fernández Collado, & Pilar Baptista, 2014), indica que “El conjunto de procedimientos secuenciales y probatorios, donde cada etapa precede a la siguiente sin omitir etapas, partiendo de una idea delimitada se establecen objetivos, preguntas de investigación, hipótesis y variables desarrollando un plan (diseño), realizando análisis mediante métodos estadísticos”, para así obtener conclusiones con respecto a la hipótesis planteada inicialmente.

3.1. Alcance

La investigación tiene un alcance descriptivo considerando que “Selecciona características, fundamentales del objeto de estudio, junto con la descripción detallada del objeto. Se muestran e identifican hechos, se diseñan productos, modelos entre otros” (Bernal Torres, 2010), para esto se va a trabajar sobre dos características de calidad, en la superficie del tablero, donde no debe existir imperfecciones, se centra en buscar papel roto y papel pegado, diseñando un algoritmo capaz de identificar las inconsistencias en la superficie del tablero mediante comparación fotográfica, siendo capaz de identificar la existencia de fallas de calidad sea papel roto o papel pegado sobre la superficie del tablero.

Partiendo de la base de datos provisto por el departamento de calidad, se extraen imágenes referentes a reclamos y tableros en B, detectados en la planta de producción. La base de datos está conformada por 131 imágenes, segmentadas

en 2 tipos de defectos de calidad, para desarrollar esta investigación, se utilizará imágenes en textura fantasía, como muestra la tabla 1.

Tabla 1

Base de datos

Fuente: Elaboración propia

Tipo de falla	Acabado	Imágenes Totales
Papel pegado	Fantasía	65
Papel roto	Fantasía	66
Total		**131**

3.2. Diseño de la investigación

Se trata de una investigación con diseño experimental ya que se va a manipular variables intencionalmente, en este caso la calidad de la superficie, para generar así tableros clase B, donde la red neuronal pueda aprender y sea capaz de identificar la calidad, llegando a generar líneas de código que cumplan con los requisitos del tablero expuestos anteriormente. (Hernández Sampieri, Fernández Collado, & Pilar Baptista, 2014) indica que “Un diseño experimental es usado cuando el investigador, pretende establecer efectos al manipular intencionalmente una variable independiente (supuestas causas), para analizar consecuencias que la manipulación tiene sobre la variable dependiente (supuestos efectos), dentro de una situación de control para el investigador”.

3.3. Selección de la muestra

En la investigación se encuentra limitada la muestra de la población de tableros laminados dentro de la empresa en las distintas laminadoras que son línea 1, línea 2, línea 3, línea 4 y línea 5, en texturas fantasía, mate, poro, madereado, roble y nogal, para el estudio se toma la muestra de dos periodos anuales de producción desde el año 2020 a 2022 de tableros laminados blancos, de la línea 4 de laminado ya que es la línea que produce principalmente tableros blancos de consumo local y exportación.

3.4. Recolección de datos

3.4.1. Diagnóstico de la situación actual del proceso.

Es necesario diagnosticar cómo se encuentra el proceso de clasificación, debido a que se detecta que existen fallas al momento visualizar las fallas de calidad, el colaborador realiza la observación y clasificación muy rápida, generando una clasificación propensa a errores. Los procesos son claves y fundamentales para las organizaciones, que busca innovación y mejora en la calidad.

Según (Robalino Lopez, Ramos, Franco , & Undoa , 2017) “La innovación de procesos permite a las empresas aprovechar las oportunidades que presentan los cambios, fomentando una cultura innovadora y posicionando a la empresa para adaptarse a las nuevas demandas y situaciones del mercado”.

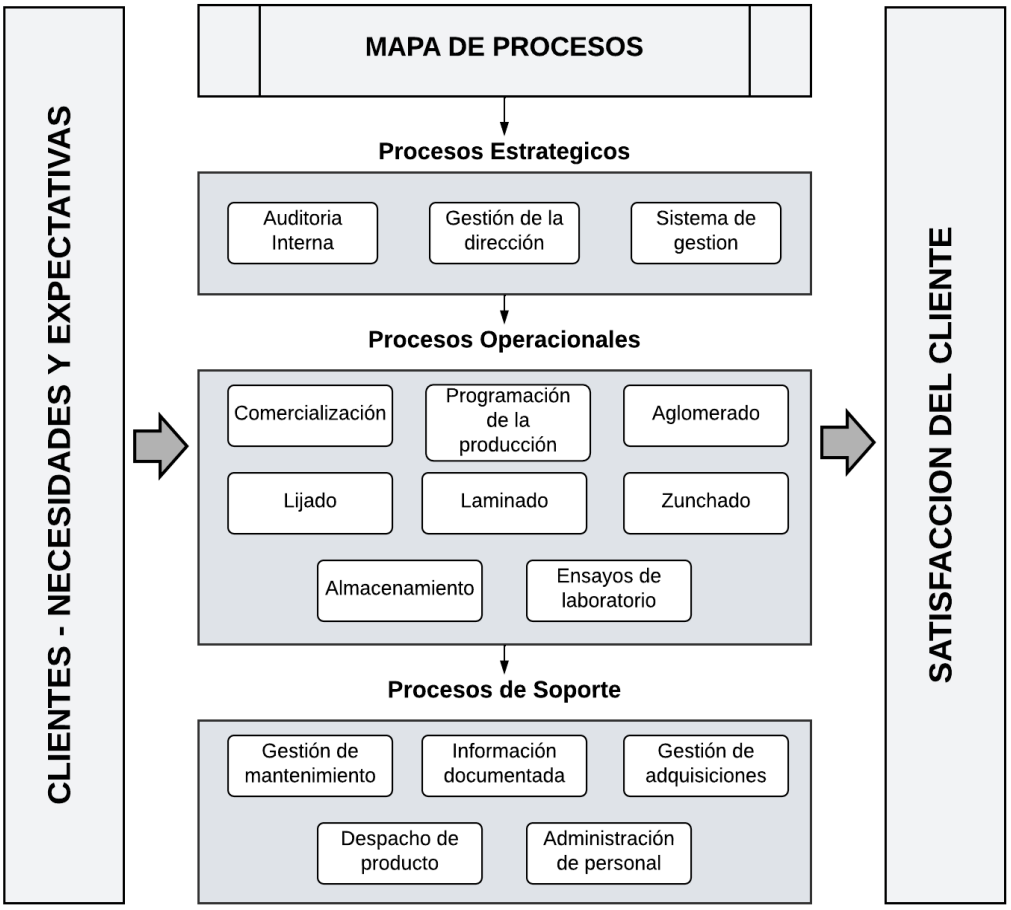
Cada actividad del proceso debe tener como objetivo añadir valor agregado al producto. Por ello, es esencial que los colaboradores se comprometan con los procesos, que vayan acompañados de responsables y controles que garanticen su eficiencia y eficacia.

La figura 19 muestra el mapa de procesos de la organización el cual este compuesto por los tres procesos: Estratégicos, Operativos, y de Apoyo.

Figura 19

Diagrama de proceso

Fuente: Elaboración propia

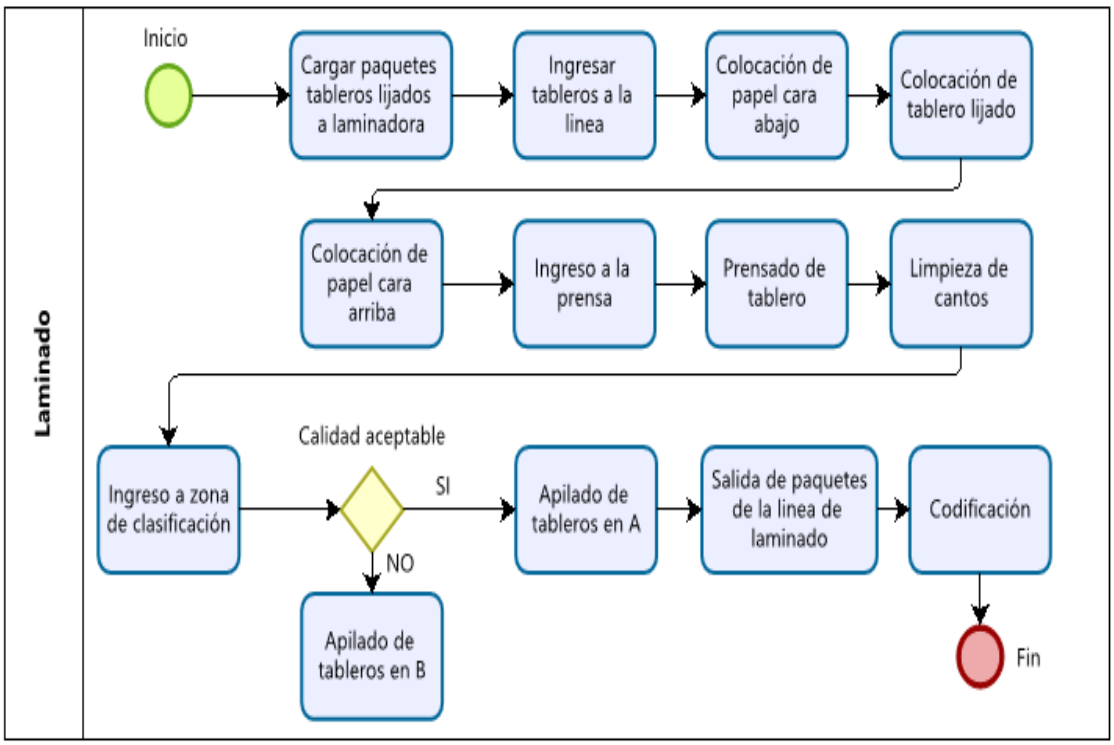


Esta investigación se centra en el proceso operativo (laminado), ya que es donde se realiza la actividad de clasificación de tableros. La figura 20 indica el diagrama de flujo del proceso de laminado donde se realiza la clasificación de tableros calidad A y calidad B, se entiende por calidad B, a todos los tableros que presenten una falla de calidad ya sea papel roto o papel pegado.

Figura 20

Diagrama de flujo proceso de laminado

Fuente: Elaboración propia



3.4.2. Diagrama Causa-Efecto (Ishikawa)

Para analizar el proceso de clasificación de tableros se utiliza el diagrama causa-efecto, una herramienta grafica que relaciona un problema con las causas que lo generan y los efectos que se producirán. (Ocaña Raza, Lara Calle , Mayorga Paredes , & Saá Tapia , 2017) indica que “Este tipo de herramientas de calidad facilitan la recopilación de datos, su cuantificación y el uso de los resultantes como indicaciones para diagnosticar el proceso proponiendo alternativas de mejora”.

La herramienta causa - efecto se aplica a un proceso ineficiente, bajo el método de las 6 M, estos seis aspectos se definen de forma global, aportando de forma individual una variabilidad al proceso estudiado. “Por lo que es natural

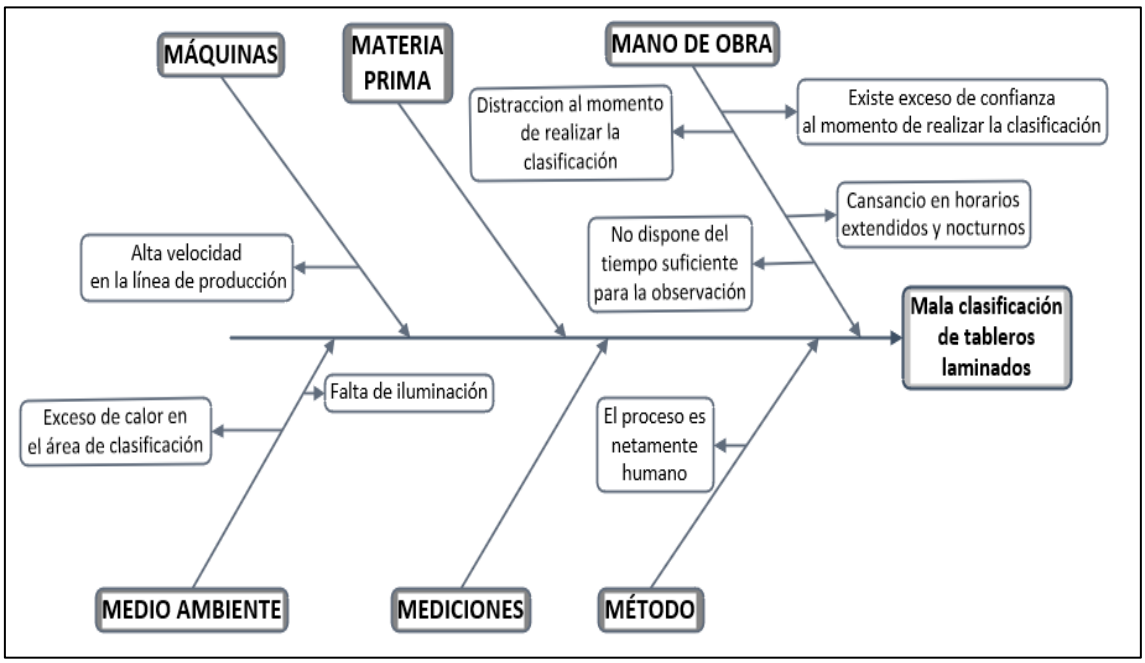
esperar que alguna de las causas de un problema esté relacionado con alguna de las 6M". (Gutierrez Pulido & De la Vara Salazar, 2013)

La figura 21 indica las causas y efectos encontrados durante el proceso de clasificación de tableros, en los apartados de materia prima y mediciones no se encuentran problemas sustanciales que involucren el proceso de clasificación de tableros.

Figura 21

Diagrama Causa – Efecto

Fuente: Elaboración propia



Terminado el diagrama causa – efecto, se establece criterios en una escala de 1 a 3, con la finalidad de evaluar las causas, donde 1 representa menos factible, 2 factible y 3 significativamente factible, donde indica que "de esta manera se logra transformar una medición subjetiva que es la que corresponde a los diagramas causa-efecto, a una valoración objetiva".

La puntuación, la estableció personas expertas en el proceso, de la clasificación y calidad de tableros laminados los cuales son:

- Jefe de producto terminado (Ingeniero electrónico)
- Responsable de producción (Ingeniero Industrial)
- Responsable de calidad (Ingeniero Químico)

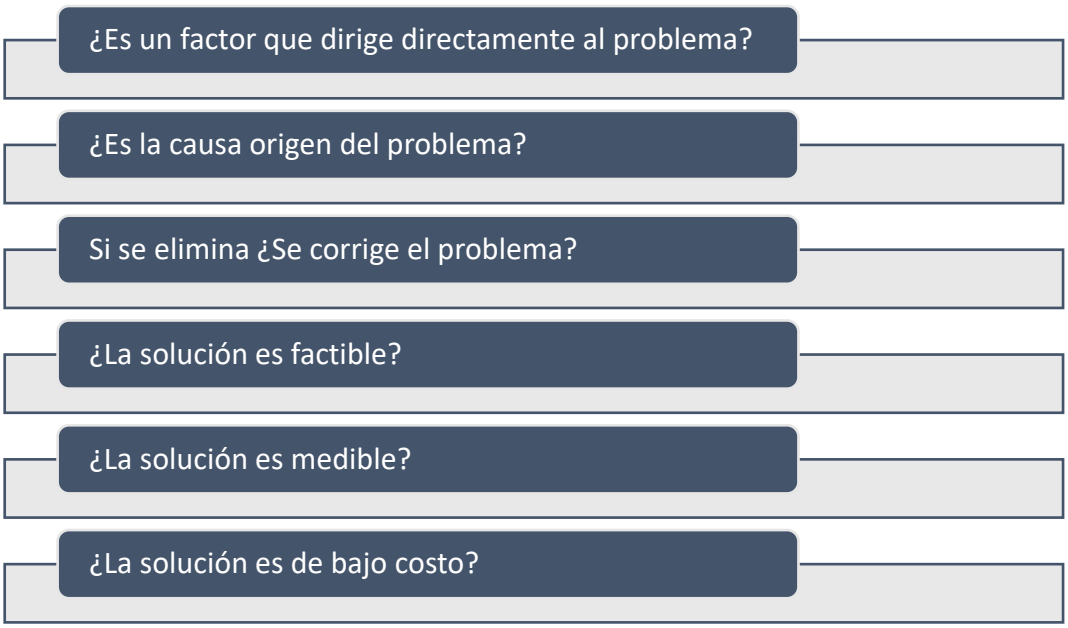
3.4.3. Matriz de criterios ponderados

Esta herramienta de calidad consiste en una “matriz de doble entrada, con el objetivo de obtener la solución más idónea y factible, al problema que se pretende resolver”. (Betancourt, 2018), la figura 22 indica los criterios empleados en la matriz.

Figura 22

Criterios empleados en la matriz de ponderación

Fuente: Elaboración propia



La tabla 2, muestra la construcción de la matriz de criterios ponderados, para seleccionar de entre todas las opciones, la que tenga el peso más representativo, en función de los criterios previamente definidos y ponderados.

Tabla 2

Matriz de criterios ponderados

Fuente: Equipo de planta experto en proceso de laminado.

Causa	Soluciones	Criterios						TOTAL
Maquinas	**Solución**	**Factor**	**Causa Directa**	**Solución**	**Factible**	**Medible**	**Bajo costo**	
Velocidad de avance de los tableros en la línea de laminado demasiado alta	Automatizar clasificación con respuesta automática y criterio del clasificador	3	3	3	3	3	1	16
Mano de obra	**Solución**	**Factor**	**Causa Directa**	**Solución**	**Factible**	**Medible**	**Bajo costo**	
Distracción de los clasificadores Cansancio por horarios extendidos y nocturnos	Rotar personal para evitar trabajo monótono	2	2	1	1	2	3	11
Tiempo de respuesta rápida sin una correcta observación	Aumentar tiempo de revisión	3	3	1	1	1	1	10
Exceso de confianza en los clasificadores	Capacitación constante	2	3	3	2	1	3	14
Medio ambiente	**Solución**	**Factor**	**Causa Directa**	**Solución**	**Factible**	**Medible**	**Bajo costo**	
Exceso de calor en el área de clasificación	Instalar ventiladores y ductos de sección	2	1	1	1	1	2	8
Iluminación inadecuada	Instalar luminaria en puntos estratégicos	3	3	2	3	1	2	14
Método	**Solución**	**Factor**	**Causa Directa**	**Solución**	**Factible**	**Medible**	**Bajo costo**	
Proceso de clasificación netamente humano	Automatizar clasificación con respuesta automática y criterio del clasificador	3	3	3	3	3	1	16

3.4.4. Valoración de soluciones

En este apartado se resume las soluciones con sus respectivas ponderaciones, indicando el resultado más representativo que es “Automatizar clasificación con respuesta automática y criterio del clasificador”, ver tabla 3, con una ponderación de 16 el más alto sobre las otras posibles soluciones, para disminuir en lo posible la mala clasificación de tableros laminados, (Maya Carillo, Vallejo Villarreal, Ramos, & Borsic Laborde, 2018) nos indica que “Así también, se promueve asumir grandes retos y resultados por parte de los colaboradores, en la organización la eficiencia, control de procesos y correcta realización de trabajo son factores claves, definiendo al éxito en términos de reconocimiento del recurso humano”, ya que con esto no se pretende eliminar la mano de obra humano, ya que nunca una maquina podrá razonar a detalle agentes externos al momento de clasificar un tablero laminado.

Tabla 3

Valoración de soluciones

Fuente: Elaboración propia

N°	SOLUCIONES	VALORACIÓN
1	Automatizar clasificación con respuesta automática y criterio del clasificador	16
2	Capacitación constante	14
3	Instalar luminaria en puntos estratégicos	14
4	Rotar personal para evitar trabajo monótono	11
5	Aumentar tiempo de revisión	10
6	Instalar ventiladores y ductos de sección	8

3.4.5. Históricos de producción

La recolección de los datos iniciales se realizará mediante registros históricos de producción y calidad, garantizando así la validez, objetividad y confiabilidad de la información, mientras que para la segunda etapa de simulación se realizará mediante observación ya que se pretende manipular la calidad de tableros para que la red neuronal sea capaz de identificar la calidad del tablero, mediante prueba y error.

Tabla 4

Histórico de producción 2021 línea 4 de laminado

Fuente: Elaboración propia

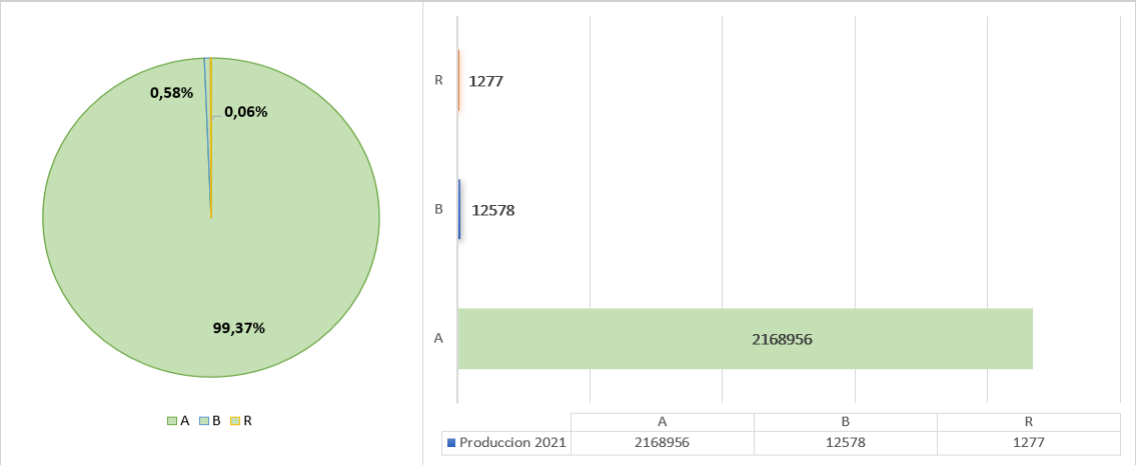
	A	B	R
Enero	1597	3	0
Febrero	191426	1387	114
Marzo	188027	1337	93
Abril	186320	1063	84
Mayo	166591	1205	109
Junio	157969	1346	128
Julio	233529	1526	101
Agosto	221101	1162	162
Septiembre	217886	1156	225
Octubre	229662	1003	106
Noviembre	214640	783	72
Diciembre	160208	607	83
	2168956	12578	1277
Total	**2182811**		

La tabla 4 y la figura 23, muestra el porcentaje de tableros en clase A, B y R, producidos durante el año 2021 en la línea 4 de laminado con un total de 2182811 tableros.

Figura 23

Porcentaje de tableros clase A,B y R 2021

Fuente: Elaboración propia



La tabla 5 y la figura 24, muestra el porcentaje de tableros en clase A, B y R, producidos durante el año 2022 en la línea 4 de laminado con un total de 2276250 tableros.

Tabla 5

Histórico De producción 2022 línea 4 de laminado

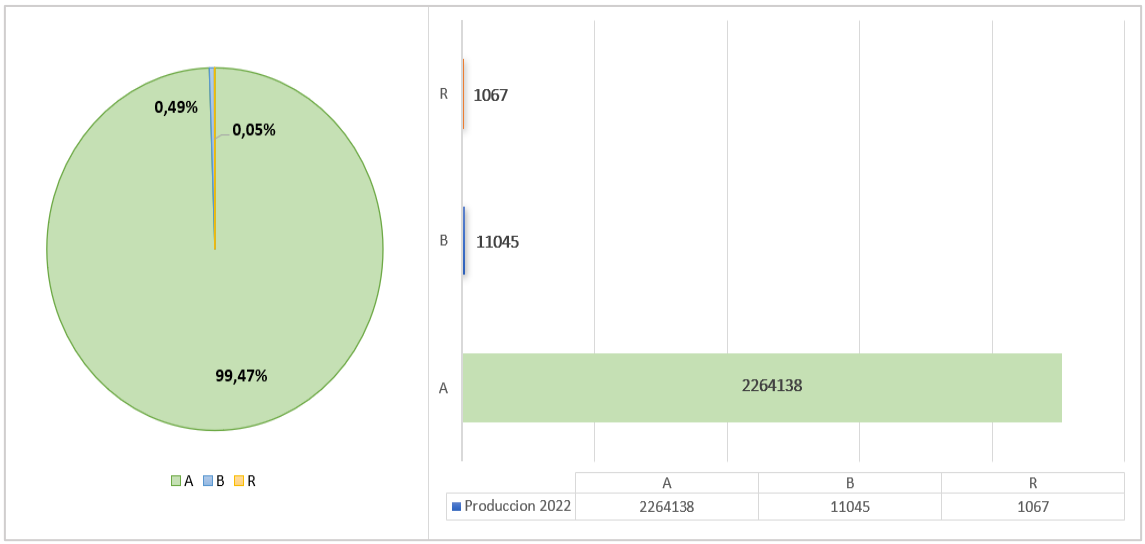
Fuente: Elaboración propia

	A	B	R
Enero	195994	681	79
Febrero	187744	779	54
Marzo	212958	947	98
Abril	206965	866	87
Mayo	161427	797	73
Junio	113051	535	65
Julio	196340	889	100
Agosto	215100	1071	99
Septiembre	217917	1405	126
Octubre	203958	873	66
Noviembre	188580	1093	95
Diciembre	164104	1109	125
	2264138	11045	1067
Total	**2276250**		

Figura 24

Porcentaje de tableros clase A,B y R 2022

Fuente: Elaboración propia



3.5. Análisis de datos

El análisis será realizado mediante software en este caso Google Colab donde se correrá el código de la red neuronal en lenguaje de Python, que se emplean para estimar parámetros diferentes de modelos metodológicos, con el fin de encontrar el algoritmo adecuado para la red neuronal y finalmente presentar los resultados.

Para la investigación es necesario conocer la cantidad y porcentaje de tableros con defectos de calidad, la tabla 6 y figura 25 indica los tableros con papel pegado y papel roto del acumulado de tableros producidos en la línea 4 de laminado, durante el año 2021.

Tabla 6

Histórico de tableros con papel pegado y roto (2021)

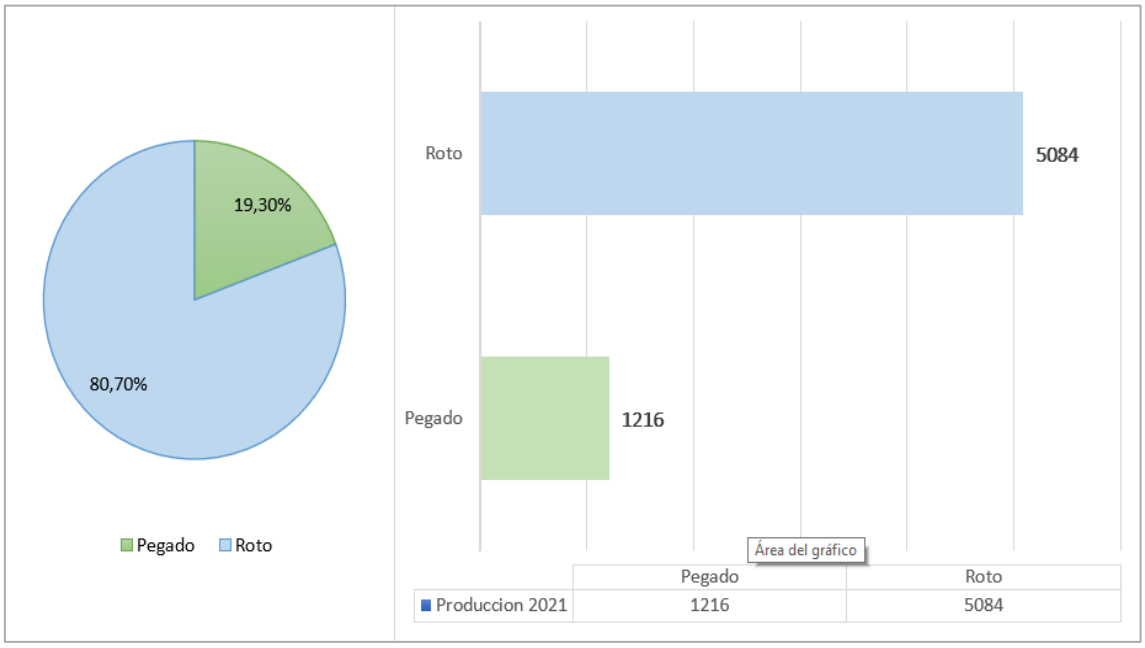
Fuente: Elaboración propia

	Pegado	Roto
Enero	1	
Febrero	99	553
Marzo	124	514
Abril	175	456
Mayo	132	530
Junio	123	587
Julio	101	434
Agosto	110	469
Septiembre	116	464
Octubre	110	396
Noviembre	86	384
Diciembre	39	297
	1216	5084
Total	**6300**	

Figura 25

Porcentaje de tableros con papel pegado y roto (2021)

Fuente: Elaboración propia



La tabla 7 y figura 26 indica los tableros con papel pegado y papel roto del acumulado de tableros producidos en la línea 4 de laminado, durante el año 2022.

Tabla 7

Histórico de tableros con papel pegado y roto (2022)

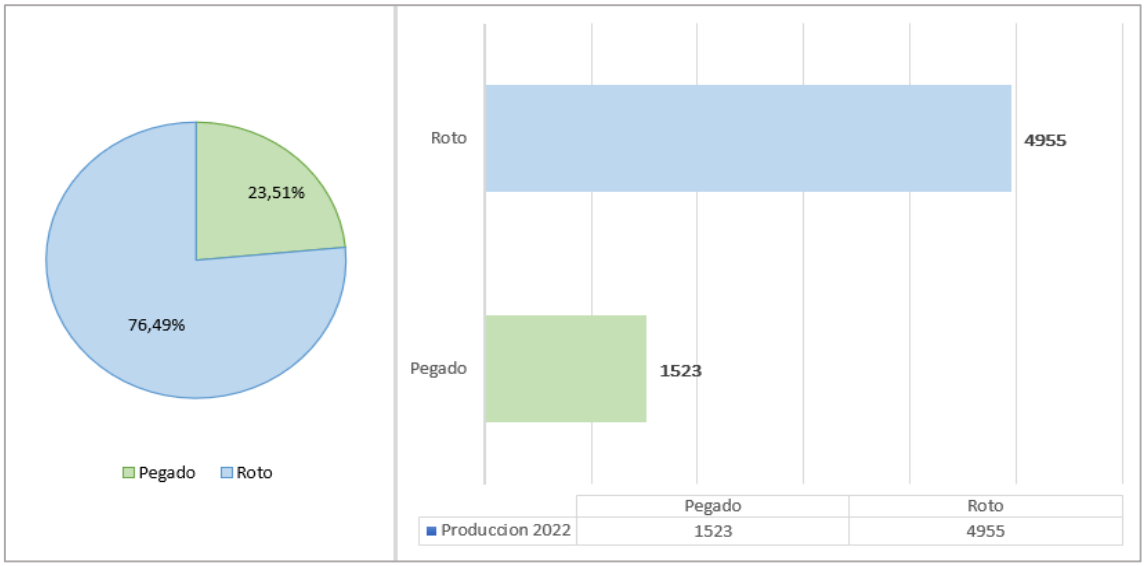
Fuente: Elaboración propia

	Pegado	Roto
Enero	89	357
Febrero	95	363
Marzo	114	580
Abril	118	524
Mayo	132	341
Junio	112	239
Julio	99	455
Agosto	139	418
Septiembre	154	503
Octubre	126	361
Noviembre	171	421
Diciembre	174	393
	1523	4955
Total	**6478**	

Figura 26

Porcentaje de tableros con papel pegado y roto (2022)

Fuente: Elaboración propia



La tabla 8 indica el acumulado de tableros clase B, de la línea 4 de laminado, con un total de 23623, de los cuales 2739 son por papel pegado y 10039 son por papel roto con un total de 12778 tableros en clase B, como muestra la tabla 9, lo que representa un 54.09% de fallas de calidad.

Tabla 8

Acumulado de tableros clase B (2021-2022)

Fuente: Elaboración propia

	Tableros clase B
2021	12578
2022	11045
TOTAL	**23623**

Tabla 9

Acumulado tableros papel pegado y roto (2021-2022)

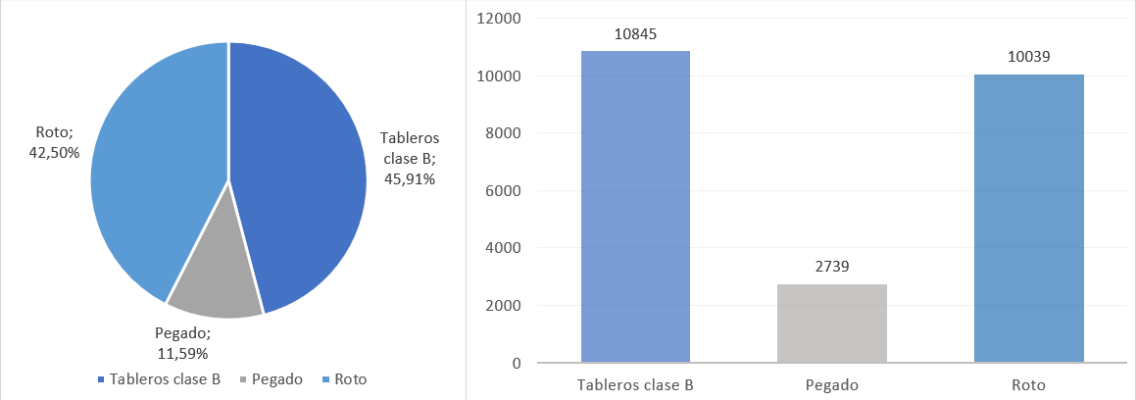
Fuente: Elaboración propia

	Tableros clase B	
	Pegado	**Roto**
2021	1216	5084
2022	1523	4955
	2739	**10039**
Total	**12778**	

Figura 27

Porcentaje de tableros papel pegado y roto (2021-2022)

Fuente: Elaboración propia



3.6. Fuentes de información

Las fuentes de información para la presente investigación son básicamente primarias, ya que se recogió de literatura de libros, artículos especializados, fuentes de internet en temas relacionados al ámbito de calidad y datos propios de una empresa para entrenar y probar una red neuronal para la clasificación de imágenes.

3.7. Desarrollo de red

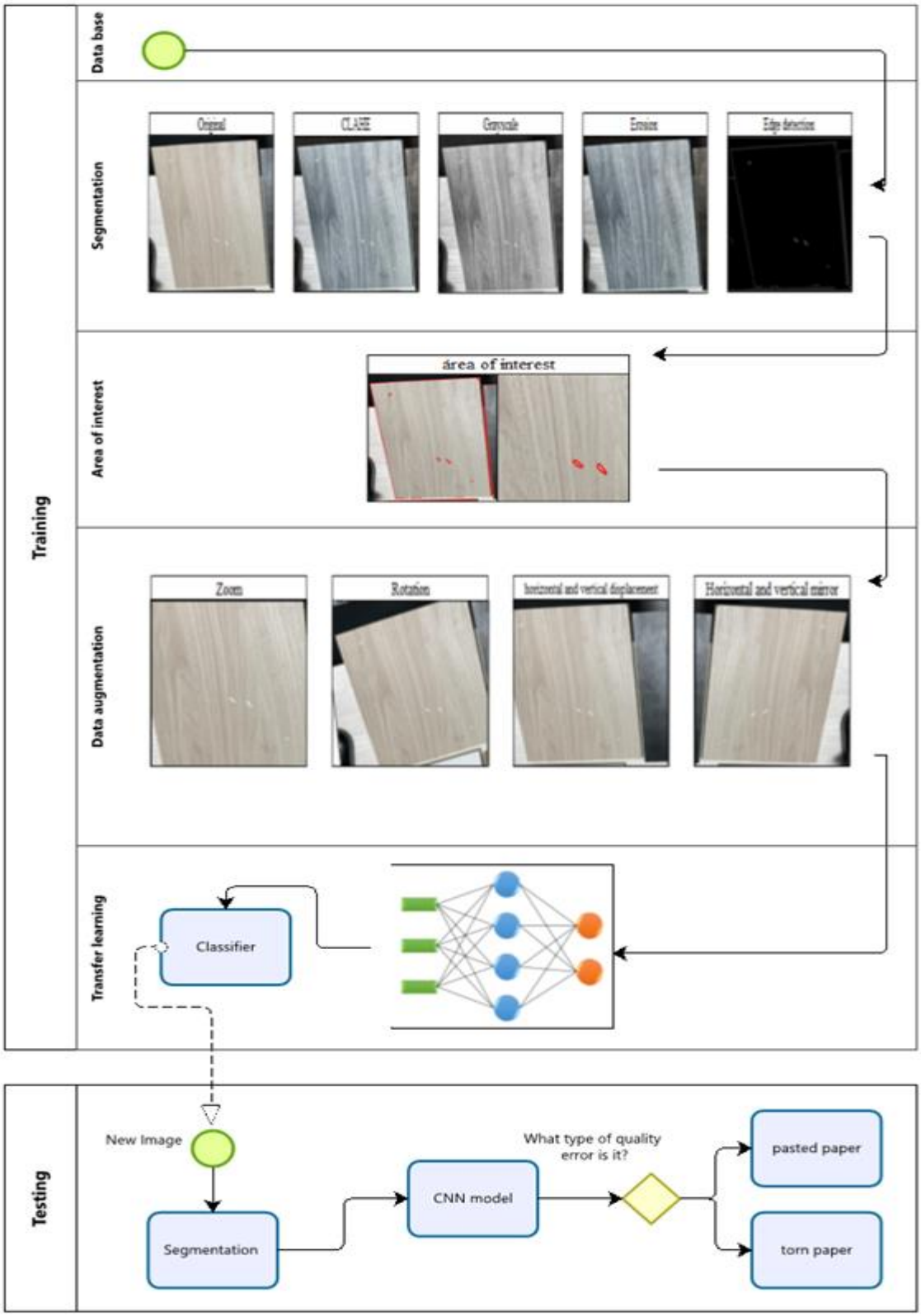
El desarrollo consiste en etapas, en la primera fase se recolecta imágenes con defectos de calidad, clasificados por personal de control de calidad de la organización donde se realiza el estudio. Estas imágenes se deben segmentar para así eliminar el ruido de fondo, de esta manera se consigue aislar los defectos de calidad, como siguiente paso se debe realizar un aumento de datos para que pueda discriminar el papel pegado del papel roto.

Como etapa final se tiene, el modelo ya entrenado donde es capaz de distinguir un tablero clase A, de un, clase B que tenga un defecto de calidad. Todo este proceso se puede visualizar en la figura 28.

Figura 28

Diagrama de proceso de método propuesto

Fuente: Elaboración propia



3.7.1. Segmentación

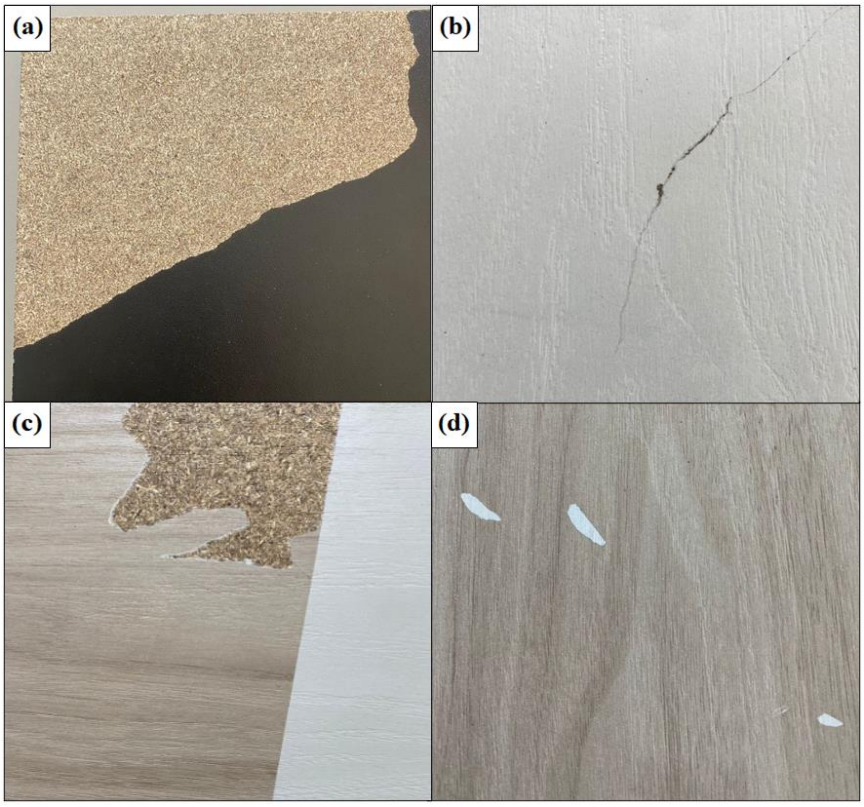
“En cualquier imagen se puede encontrar uno o varios objetos en un entorno, para lograr identificar la zona de interés se debe separar los objetos del entorno para que puedan ser distinguidos” (de la Escalera Hueso, 2001).

Para lograr esto se dispone de algunos métodos, los cuales son; segmentación por escala de grises, color, textura y bordes. Se cuenta con una base de datos para el entrenamiento de la red neuronal la cual se indicó en la tabla 1.

Figura 29

Ejemplos de fallas de calidad. (a) papel roto (b) papel trizado roto (c) papel roto y pegado (d) papel pegado

Fuente: Elaboración propia



En la figura 29 se puede observar como el entorno de la imagen se mezcla con la zona de interés, para este caso papel roto y papel pegado, para lograr

identificar la zona de interés de las distintas fallas de calidad se procesará dichas fallas como muestra la figura 30.

Figura 30

Errores de calidad para entrenamiento de la red

Fuente: Elaboración propia

Las imágenes deben ser procesadas con los pasos descritos a continuación:

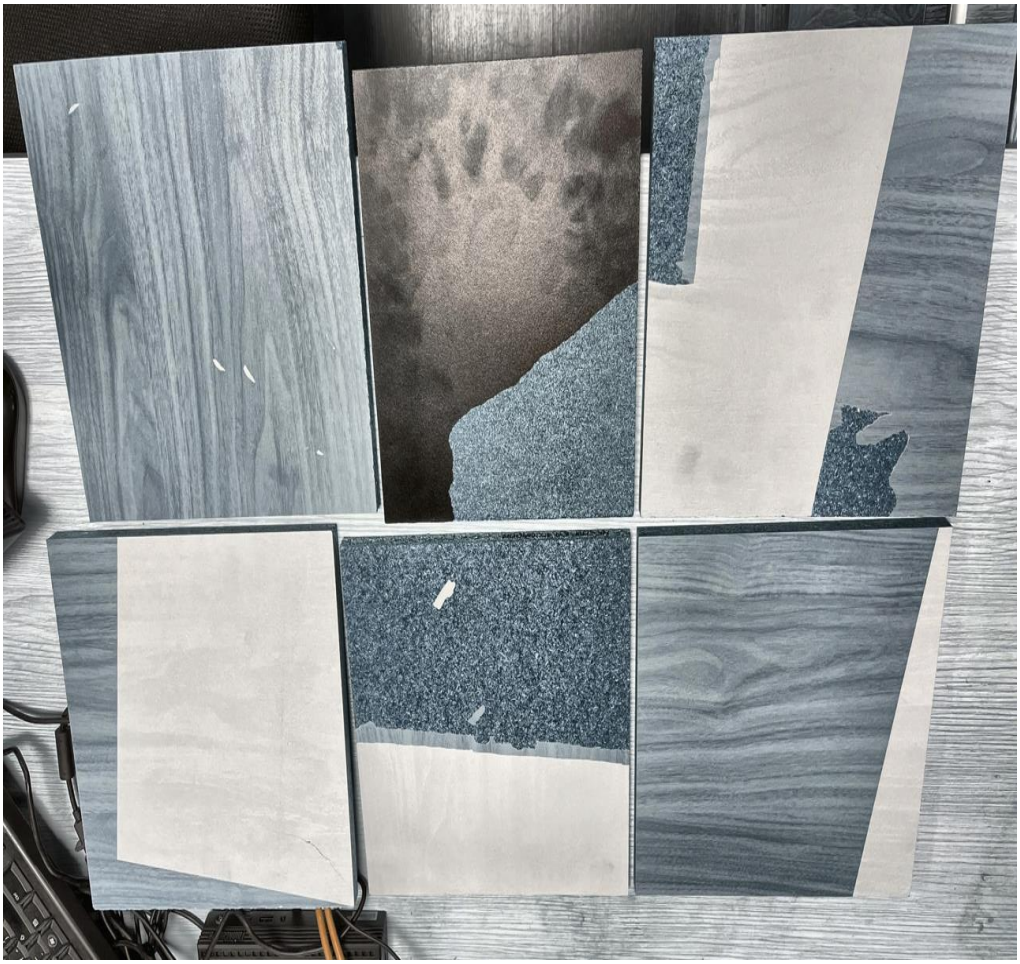
3.7.2. Método CLAHE

Como primera fase se aplica a la imagen el método CLAHE, (ecualización de histograma adaptable de contraste limitado), “es un método de procesamiento digital de imágenes que aumenta el contraste sin añadir ruido, se redistribuye los valores de brillo de pixeles de zonas seleccionadas de la imagen” (Pizer, y otros, 1987) mejorando así el contraste de la imagen como muestra la figura 31.

Figura 31

Aplicación de CLAHE

Fuente: Elaboración propia



3.7.3. Escala de grises

En la segunda fase se tiene que convertir la imagen de colores RGB “la cual se basa en la combinación de tres señales de luminancia cromática distinta” (de la Escalera Hueso, 2001) sus siglas representan los colores (red, green y blue), a escala de grises como muestra la figura 32.

Figura 32

Aplicación de escala de grises

Fuente: Elaboración propia

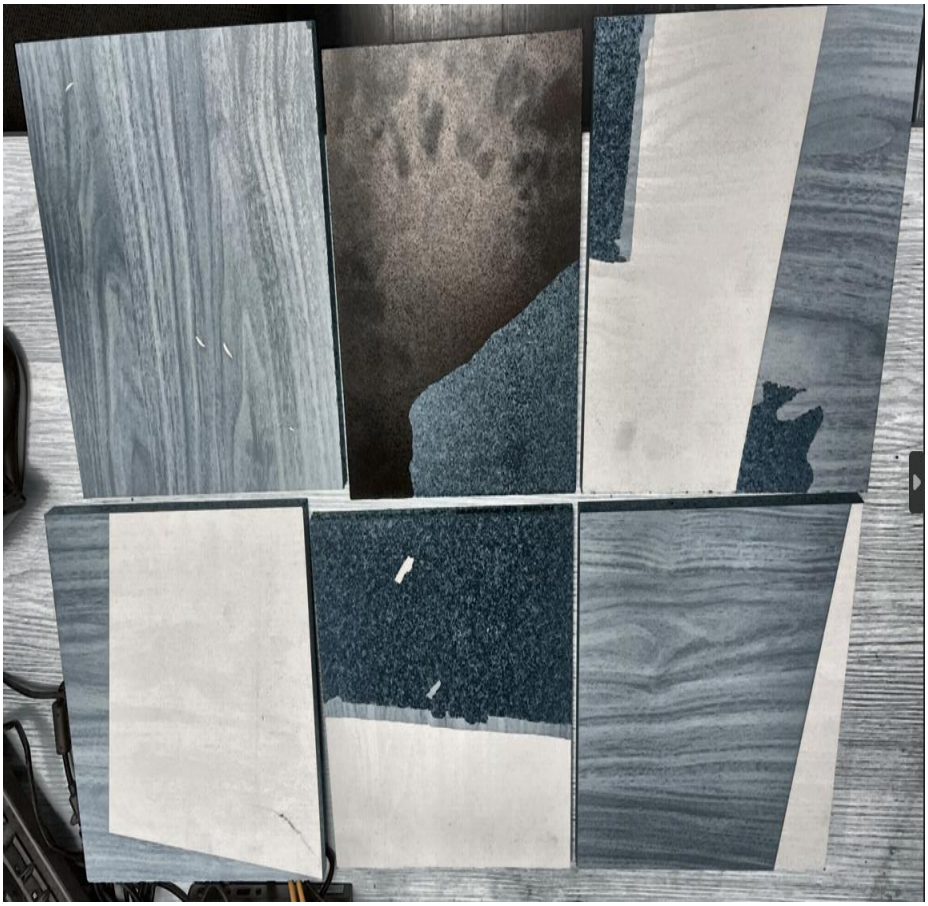
3.7.4. Erosión de imagen

Después de convertir la imagen RGB, a escala de grises se aplica la erosión “degradación progresiva de uno de los campos 0 o 1 siendo estos los niveles de gris en binario, un elemento a degradar seguirá perteneciendo al mismo si está rodeado de elementos iguales a el caso contrario pasaría al otro campo” (de la Escalera Hueso, 2001), esta mejora se puede apreciar en la imagen 33.

Figura 33

Aplicación de erosión

Fuente: Elaboración propia



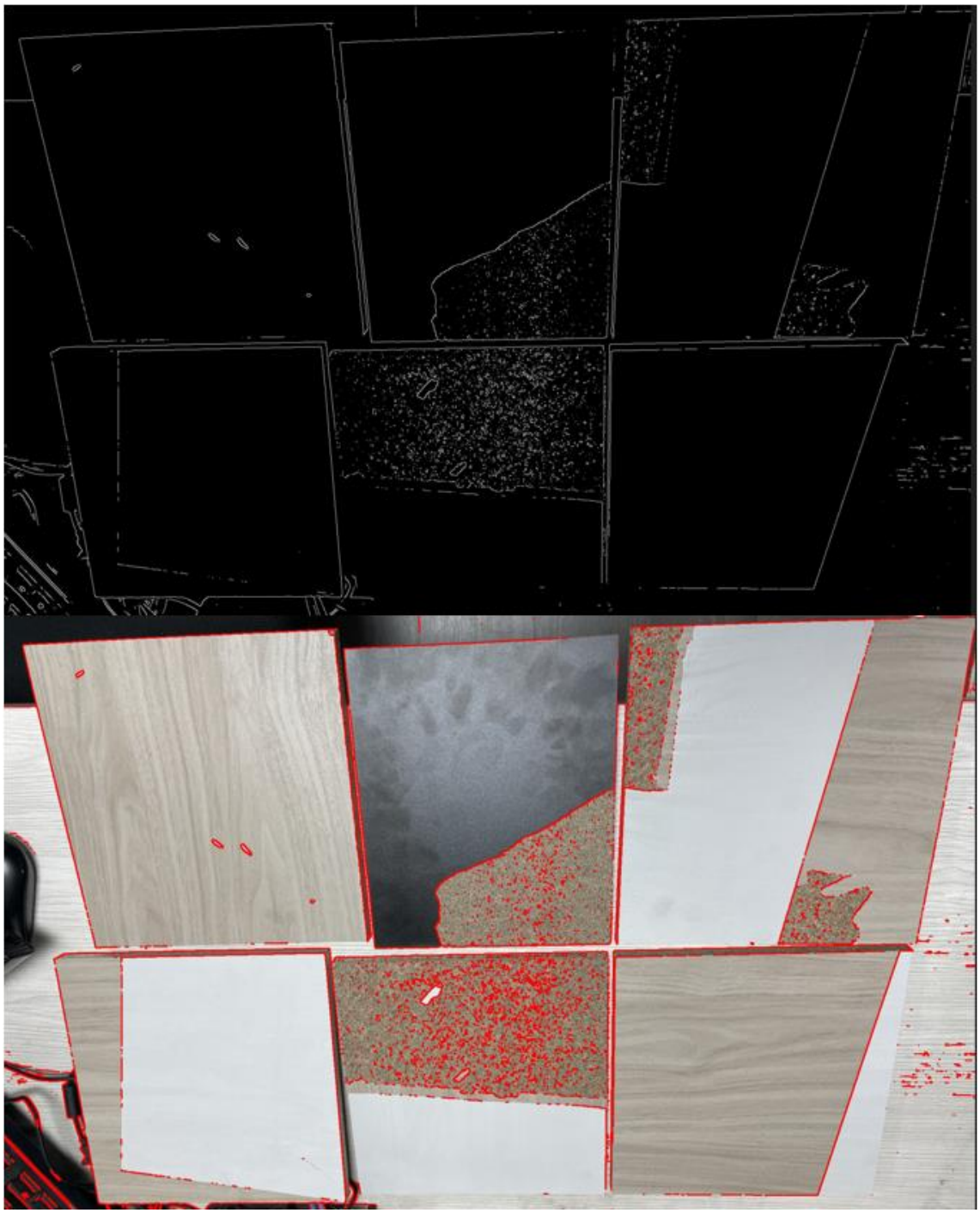
3.7.5. Detección de bordes

Finalmente, a la imagen erosionada se le aplica la detección de bordes, siendo estas de suma importancia “ya que al delimitar los objetos definen los límites entre ellos y el fondo y entre los objetos entre sí” (de la Escalera Hueso, 2001) la imagen 34 muestra la detección de bordes en la imagen erosionada.

Figura 34

Aplicación detección de bordes

Fuente: Elaboración propia



3.7.6. Aumento de datos

Consiste en “ampliar el conjunto de datos utilizado para el entrenamiento aplicando cambios a las imágenes existentes y, si es necesario, a los cuadros delimitadores que las acompañan. Estas imágenes modificadas se incluyen en la base de datos junto con las originales. De este modo se amplía la cantidad de datos para el entrenamiento de la red”. (Cuenca, Vázquez Martín, Anthony Mandow, Jesús Morales, & García Cerezo, 2021).

En la actualidad esta técnica es muy utilizada para el entrenamiento de una red neuronal, donde se necesita una base de datos de cantidades grandes de imágenes, para lo cual se plantea las siguientes estrategias:

- Zoom: “Consiste en el reescalado de una imagen, a un tamaño mayor o menor al original” (Gómez Sarasa & Ortega Pabón, J.D.) con un rango de variación de 0.2 es decir zoom al 80% y 120%.
- Rotación: es el cambio del rango de rotación aleatoria comprendido entre -180° y +180°.
- Desplazamiento vertical: Consiste en desplazar aleatoriamente una imagen en sentido de arriba y abajo dentro de un rango de 0.3.
- Desplazamiento Horizontal: Consiste en desplazar aleatoriamente una imagen en sentido de derecha o izquierda dentro de un rango de 0.3.
- Espejo vertical: Consiste en voltear la imagen verticalmente de modo aleatorio.
- Espejo horizontal: Consiste en voltear la imagen horizontalmente de modo aleatorio.

El aumento de datos se trabajó en la plataforma de Roboflow, donde permite, incrementar nuestra base de datos, incluyendo las imágenes originales, junto con las imágenes aumentadas logrando obtener el total de imágenes indicadas en la tabla 10.

Tabla 10

Total, de imágenes después del aumento de datos

Fuente: Elaboración propia

Tipo de falla	Acabado	Imágenes Originales	Imágenes aumentadas	Imágenes totales
Papel pegado	Fantasía	65	79	144
Papel roto	Fantasía	66	79	145
Total		**131**	**158**	**289**

3.8. Configuración experimental de la red neuronal

Las métricas de configuración para la red neuronal empleadas para la evaluación y comparación de las imágenes como la partición de datos, el aumento de imágenes, configuración de la red y métricas de evaluación se detalla a continuación.

3.9. Métrica de partición de datos

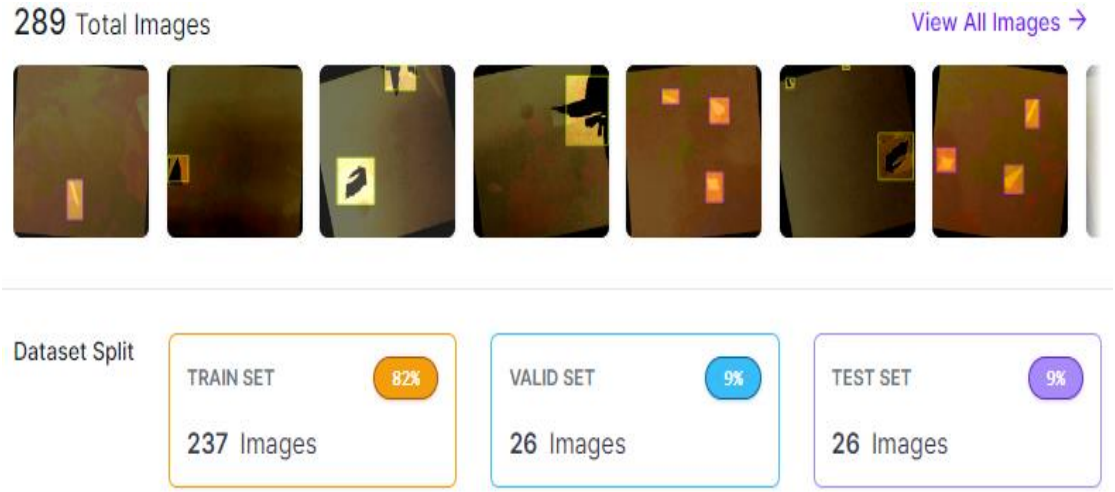
Para utilizar esta herramienta se utiliza la base de datos que está conformado por 289 imágenes las mismas incluyen, imágenes originales más imágenes aumentadas, como muestra la figura 35, donde se genera el código y segmentándolo en 3 clases los cuales son train set, valid set y test set, los cuales

se cargan y segementan para el entrenamiento de la red neuronal gracias a la plataforma de Roboflow.

Figura 35

Partición de datos para el entrenamiento.

Fuente: Elaboración propia



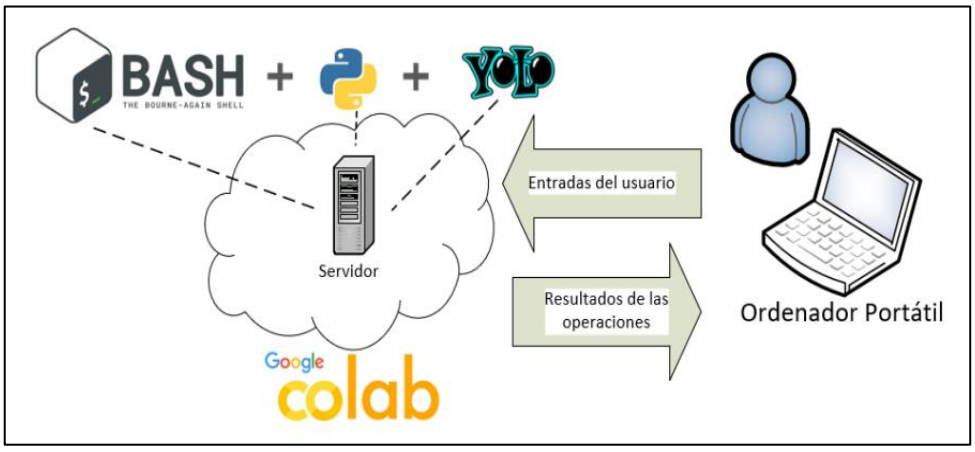
3.10. Entrenamiento en Google-colab

Esta herramienta permite entrenar a la red en la nube, la imagen 36 se aprecia el entorno de funcionamiento del servicio de entrenamiento en la nube.

Figura 36

Componentes de Google-colab

Fuente: (García, 2021)



De esta forma cualquier persona que cuente con un usuario de Google puede acceder con autentificación de cuenta a la plantilla del proyecto YOLOv5, con la posibilidad de cambiar los parámetros de entrada según la necesidad del sistema que se busca implantar, para finalmente entrenar el modelo.

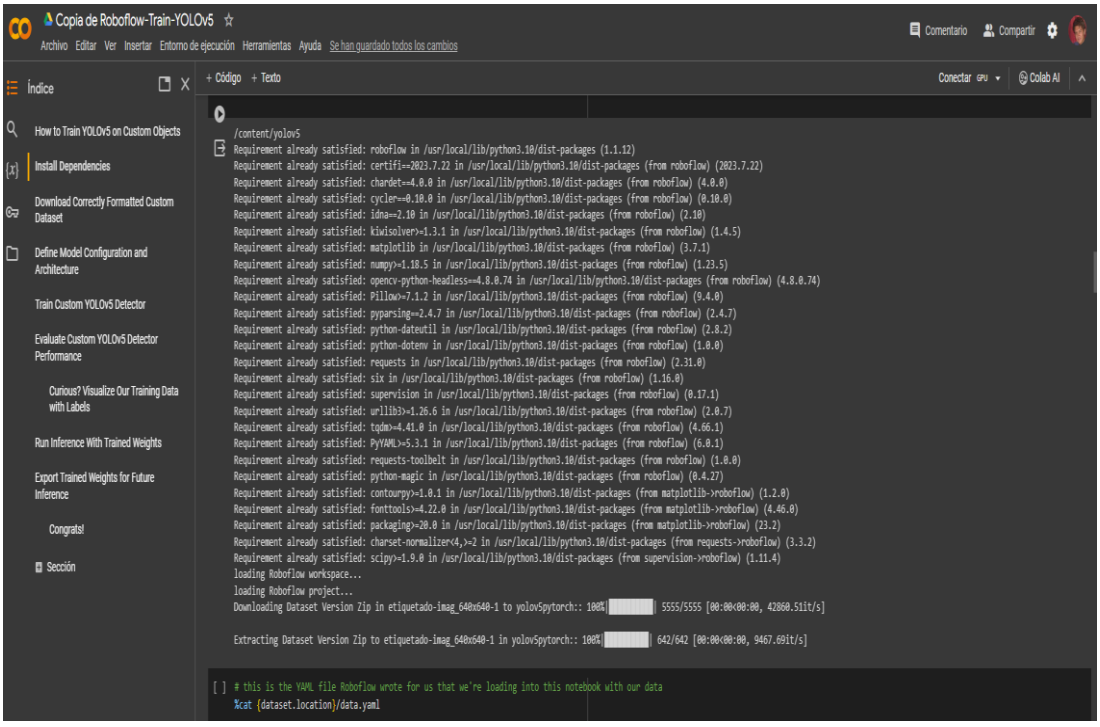
3.10.1. Interfaz Google-colab

Similar a Jupyter, Google Colaboratory es un servicio en la nube basado en notebook que permite emplear GPU y unidades de procesamiento tensorial (TPU) para la creación de servicios computacionales, estándares y software de código abierto. La imagen 37, muestra el interfaz de Google colab, con la plantilla de YOLOv5 y sus dependencias, así como los resultados tras la ejecución del código.

Figura 37

Interfaz de Google colab en la web

Fuente: Elaboración propia (Google colab)



Es indispensable emplear roboflow, que inyectará desde un repositorio previamente condicionado por el usuario, la base de datos de imágenes etiquetadas con la falla de calidad de papel roto y papel pegado, con dimensiones de 640x640 las cuales servirán para ejecutar el entrenamiento, para poder utilizar esta interfaz parametrizándola con nuestro conjunto de datos.

3.10.2. Roboflow

Roboflow es un entorno de desarrollo de visión artificial que proporciona métodos de preprocesamiento para el entrenamiento de modelos junto con servicios de recopilación y clasificación de datos, y el entrenamiento del modelo. Toda la información con respecto a esta plataforma se la puede encontrar en la siguiente referencia (Roboflow, s.f.)

Una vez subida el dataset, se clasifica las imágenes de forma individual, etiquetándolas con regiones de interés, donde se indica lo que se pretende clasificar, al igual es posible aumentar la base de datos, generando modificaciones en las imágenes tales como rotación, desplazamiento entre otros y explicados en el apartado 3.7.6 de la presente investigación.

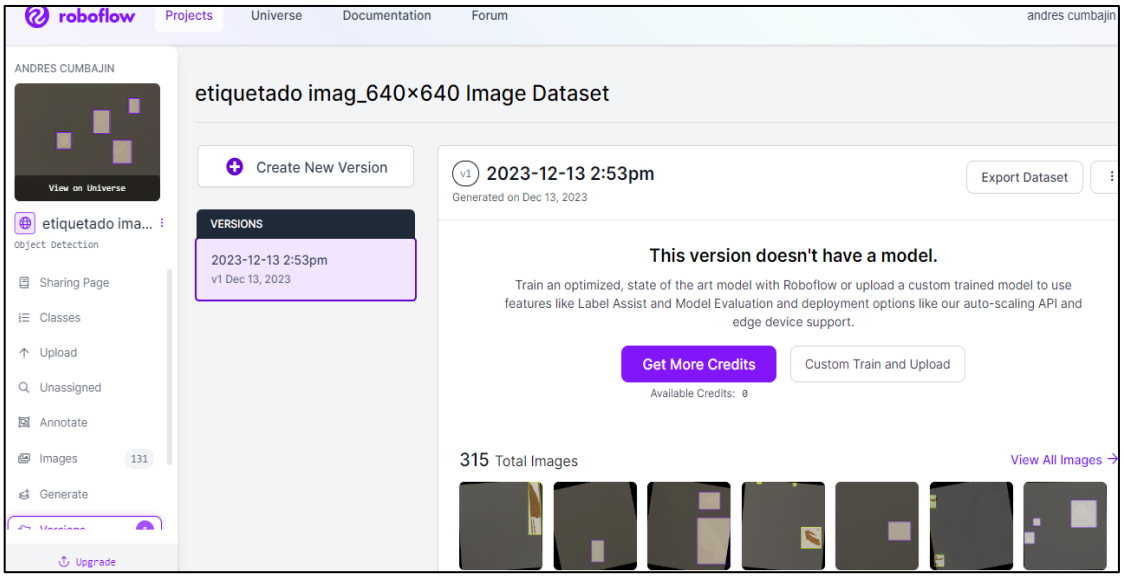
Terminado el etiquetado, es necesario llevarlo al entorno de entrenamiento de YOLOv5, el cual se segmenta en tres directorios de imágenes que roboflow provee y son los siguientes, training, valid y testing, el cual se indicó anteriormente en la figura 35.

En la figura 38 se puede apreciar como es el entorno de trabajo de roboflow, junto con el dataset ya etiquetado para generar el modelo para la detección de papel roto y papel pegado.

Figura 38

Interfaz de trabajo roboflow (web)

Fuente: Elaboración propia (roboflow)

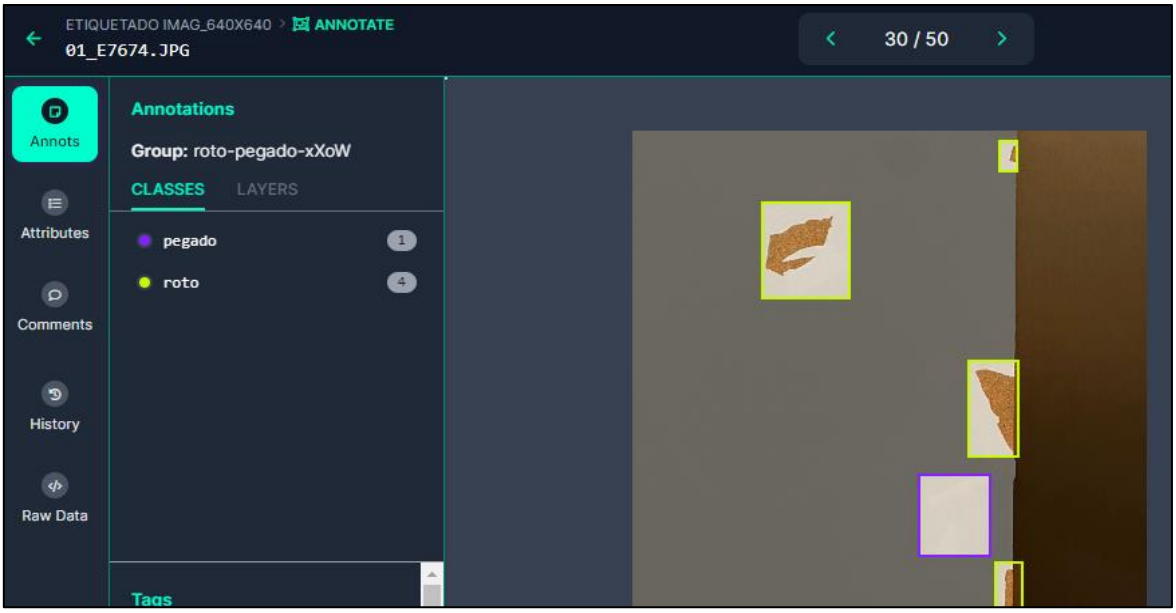


La figura 39 indica un ejemplo claro de el trabajo de etiquetado en cada imagen de la base de datos que se recopilo.

Figura 39

Etiquetado de imágenes en roboflow

Fuente: Elaboración propia (roboflow)



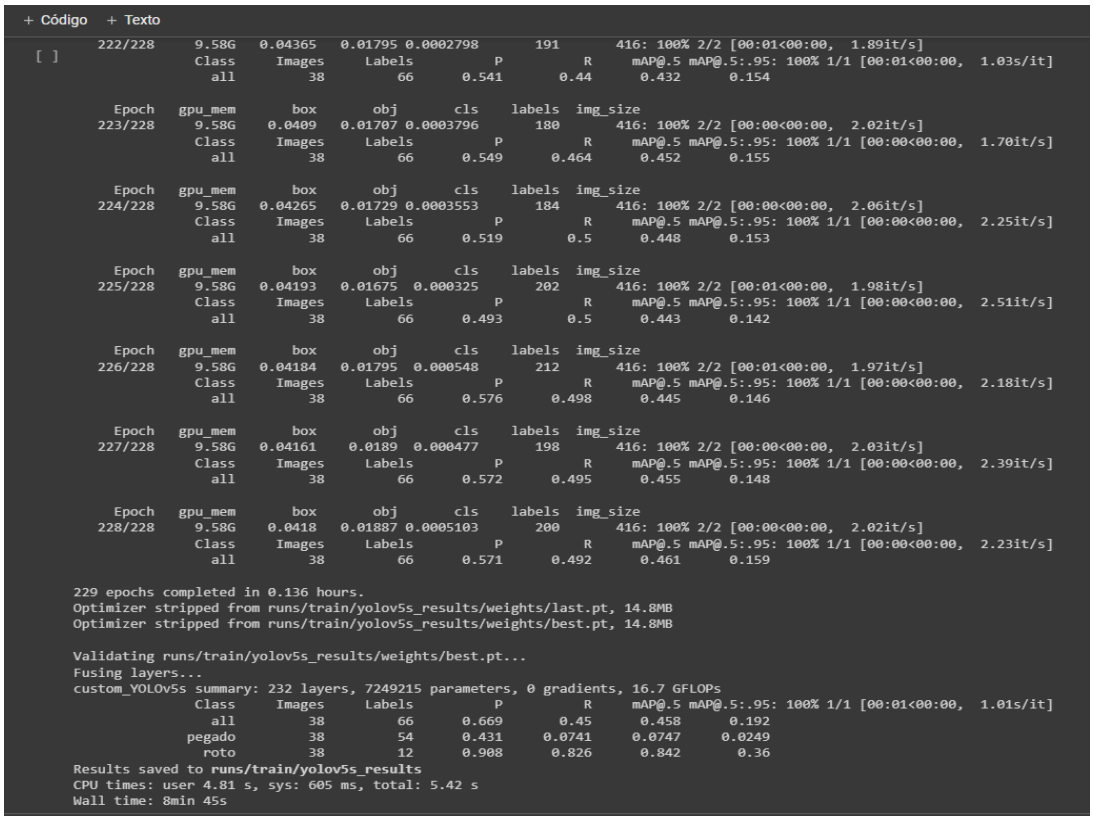
Finalmente se procede a exportar el dataset para empezar la etapa de entrenamiento en Google colab.

La figura 40 proporciona la información de la etapa y finalización de entrenamiento y su salida por pantalla en Google colab.

Figura 40

Resultado de entrenamiento y tiempo empleado

Fuente: Elaboración propia (Google colab)



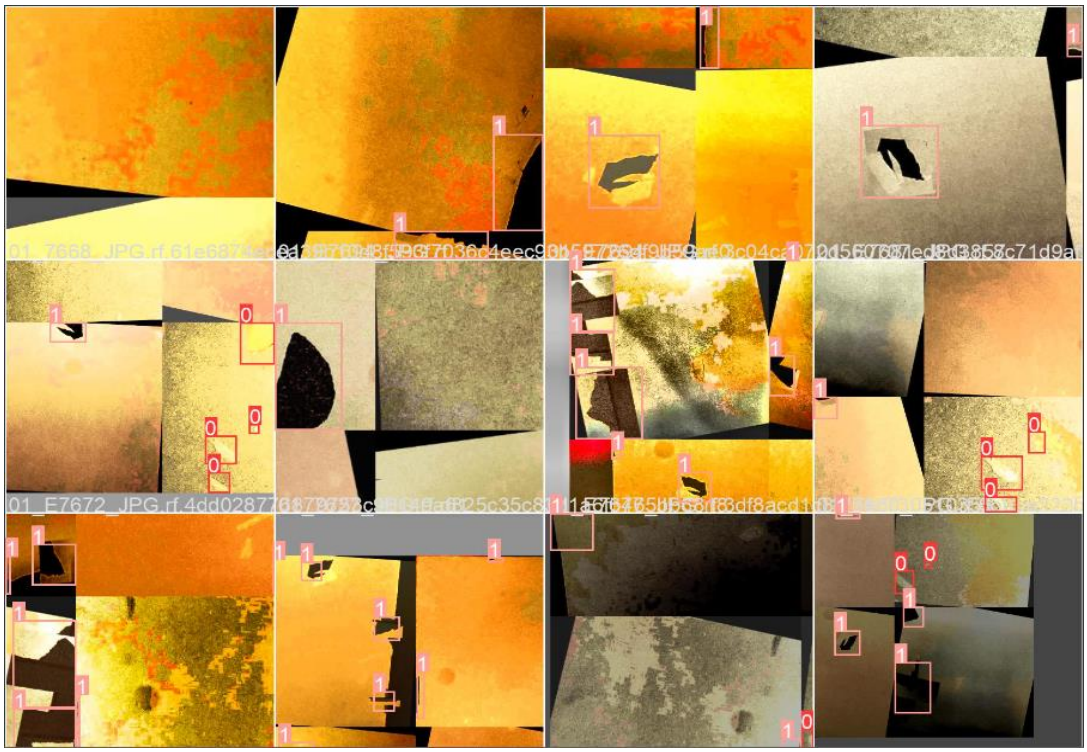
En resumen, la figura 40 indica que el servidor de Google le tomo 8 minutos y 45 segundos y 228 épocas para elaborar el modelo en conjunto con los pesos que YOLOv5 necesitara para la inferencia y se dividen en dos apartados, siendo el primero el (Best.pt) siendo los mejores pesos recolectados durante el entrenamiento y el segundo (Last.pt) siendo el peso de la última etapa del

entrenamiento. La figura 41 indica el entrenamiento realizado y por consecuencia el resultado es el proceso de inferencia.

Figura 41

Resultado del entrenamiento

Fuente: Elaboración propia (roboflow)



3.11. Red Neuronal (configuración)

Para la configuración se toma los siguientes aspectos:

- Algoritmo de optimización: es un algoritmo de optimización ampliamente utilizado en el entrenamiento de redes neuronales convolucionales según (Goodfellow , Bengio, & Courville, 2016) “ajusta la tasa de aprendizaje automáticamente en cada parámetro de la red. Esto significa que los parámetros que se actualizan con frecuencia tendrán una tasa de aprendizaje pequeña, ayudando a evitar oscilaciones y converger de manera rápida y estable”.

- Tasa de aprendizaje: La tasa de aprendizaje es un hiperparámetro critico el cual determina el ajuste de pesos de la red en cada paso de entrenamiento, se parte con un valor mínimo como sugiere (Wilson & Martinez, 2001) “un valor de 0.001sin embargo elegir el valor correcto requiere experimentación y ajuste”.
- Función de perdida: Esta función se utiliza para medir la discrepancia entre las predicciones del modelo y los valores reales o etiquetas en un conjunto de datos. Su objetivo principal es cuantificar cuán "equivocadas" son las predicciones del modelo en comparación con las respuestas correctas. Como se realizó en el estudio de (Nuñez Alverca, 2022) “el problema de clasificación es multiclase, se elige la entropía cruzada categórica como función de perdida, ya que permite un entrenamiento más rápido para las tareas de clasificación”.

3.12. Modelo de red (RNC)

Para la red se tiene como datos de entrada imágenes en un formato de jpg de formato 640x640 pixeles y dos clases de salida los que representa el papel pegado y roto, compuesta por los siguientes bloques y capas.

- Capas convolucionales de filtros de 3x3 son esenciales para el procesamiento de imágenes en redes neuronales convolucionales. “Permiten que la red aprenda y capture patrones de diferentes escalas en las imágenes, lo que es crítico para tareas como la clasificación de objetos en imágenes, se utilizan para extraer características de las imágenes, como bordes, texturas y patrones” (Goodfellow , Bengio, & Courville, 2016).

- Capa de normalización Batch Normalization, (BN), según (T. Tran, O. , -H. Kwon, K., -R. Kwon, S., & -H. Lee and , 2018)1 “mejora la estabilidad y el rendimiento del entrenamiento. Su objetivo principal es normalizar las activaciones intermedias de una capa en una red neuronal, lo que facilita el proceso de optimización y acelera la convergencia del entrenamiento”.
- Capa Leaky ReLU se utiliza como una función de activación en las capas convolucionales de una CNN indica que (Silva, 2020) “para introducir no linealidades en el modelo y permitir que el gradiente fluya incluso cuando las entradas son negativas. Esto ayuda a evitar el problema de unidades muertas (neuronas inactivas) y puede mejorar el rendimiento y la estabilidad del entrenamiento en redes neuronales convolucionales”.
- Capa Max Pooling ayuda a reducir dimensiones, mejorar la invariancia y simplificar la representación de características. “Estas propiedades son especialmente útiles en tareas de visión por computadora, como la clasificación de imágenes, la detección de objetos y la segmentación semántica”. (Goodfellow , Bengio, & Courville, 2016).
- Capa Dropout Layer se usa para prevenir el sobreajuste (overfitting) y mejorar la generalización del modelo. Según (Moreno, 2019) trabaja generalmente entre 0.2 y 0.5 “apagando aleatoriamente una fracción de neuronas en cada paso durante el entrenamiento. Esto significa que, durante el entrenamiento, algunas neuronas se "desconectan" temporalmente, evitando que dependan en exceso de ciertas características, permitiendo aprender representaciones más robustas y generalizables”

4. RESULTADOS

En base a los módulos de tensorBoard, generados en el entrenamiento de la red neuronal en Google Colab, donde se generó 8 test, en los cuales se varió parámetros para el entrenamiento indicados en la tabla 11.

Los resultados del **Test-7**, de entrenamiento de la red neuronal con arquitectura YOLOv5, donde se realizó la variación en el etiquetado simple de fallas de calidad en las imágenes de la base de datos para el entrenamiento, el aumento en las épocas de entrenamiento y agregar un filtro de imagen (histogram-equalization), ver figura 42, es la que presenta mejor balanceo en resultados de precisión y recall.

Tabla 11

Resultados de la variación de parámetros para entrenar a la red neuronal.

Fuente: Elaboración propia (TensorBoard).

Test - 1 Sin filtro 100 épocas			**Test - 2** Sin filtro 376 épocas			**Test - 3** Sin filtro 180 épocas			**Test - 4** Con filtro 180 épocas		
Class	P	R	Class	P	R	Class	P	R	Class	P	R
all	0,659	0,292	all	0,855	0,632	all	0,503	0,493	all	0,823	0,505
pegado	1,000	0,000	pegado	0,753	0,348	pegado	0,461	0,486	pegado	0,689	0,344
roto	0,317	0,583	roto	0,957	0,917	roto	0,545	0,500	roto	0,985	0,667
	66%	29%		86%	63%		50%	49%		82%	51%

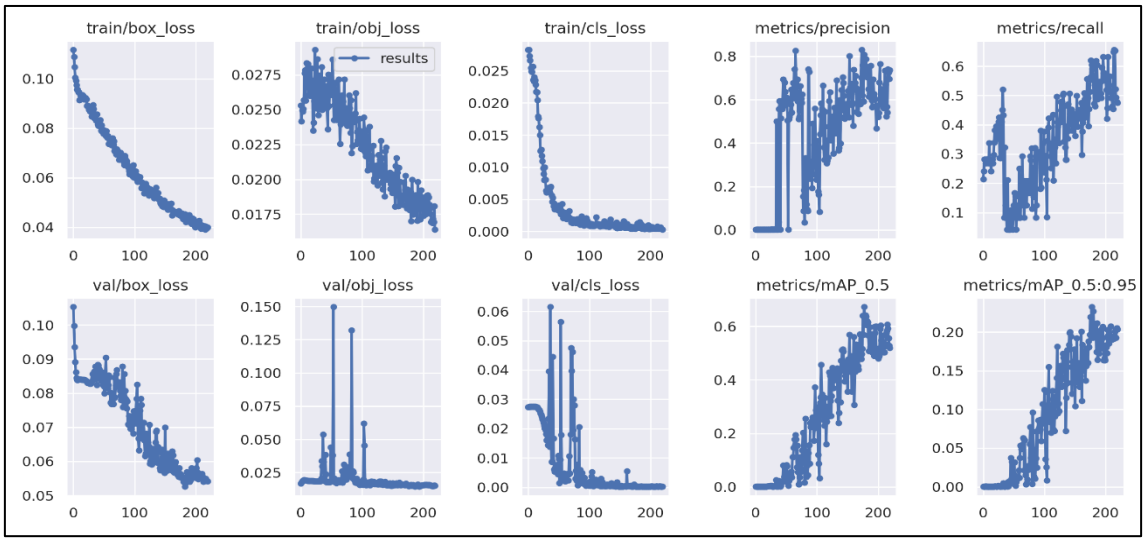
Test - 5 etiqueta simple + 220 épocas sin filtro			**Test - 6** etiqueta doble + 220 épocas sin filtro			**Test - 7** etiqueta simple + 376 épocas con filtro			**Test - 8** etiqueta doble + 220 épocas con filtro		
Class	P	R	Class	P	R	Class	P	R	Class	P	R
all	0,808	0,605	all	0,673	0,454	all	0,816	0,719	all	0,801	0,564
pegado	0,642	0,543	pegado	0,720	0,487	pegado	0,770	0,522	pegado	0,779	0,641
roto	0,975	0,667	roto	0,626	0,421	roto	0,862	0,917	roto	0,822	0,486
	81%	61%		67%	45%		**82%**	**72%**		80%	56%

En la figura 42 se puede observar las métricas de entrenamiento del Test-7, ya que fue la que mejor desempeño tuvo se procede analizar los resultados con este modelo y combinación de parámetros para el sistema de clasificación.

Figura 42

Métricas del resultado de entrenamiento del sistema de clasificación.

Fuente: Elaboración propia (TensorBoard).



La figura 43 indica el ejemplo de como al aplicar un filtro o preprocesamiento de una imagen, se puede resaltar los contornos y en este caso las fallas de calidad del papel pegado ya que, al ser del mismo color del papel en este caso blanco, tiende a confundirse con el fondo del tablero.

Figura 43

Filtro (Histogram-Equalization)

Fuente: Elaboración Propia (Roboflow)

Los resultados de la simulación del clasificador al realizar los ocho test, muestran nuevamente como el Test-7, muestra mejor desempeño frente a los demás test, ya que tiene un mejor balanceo entre precisión y recall.

El sistema se considera bueno, si es directamente proporcional, en otras palabras, si se aumenta el recall la precisión se va manteniendo alta, estos valores se pueden observar en la tabla 12.

Tabla 12

Resultados de la variación de parámetros en la simulación del clasificador

Fuente: Elaboración propia (TensorBoard).

Test - 1 Sin filtro 100 épocas			**Test - 2** Sin filtro 376 épocas			**Test - 3** Sin filtro 180 épocas			**Test - 4** Con filtro 180 épocas		
precisión	0,596	59,60%	precisión	0,6942	69,42%	precisión	0,6257	62,57%	precisión	0,7064	70,64%
recall	0,25	25,00%	recall	0,5563	55,63%	recall	0,45	45,00%	recall	0,5035	50,35%
Test - 5 etiqueta simple + 220 épocas sin filtro			**Test - 6** etiqueta doble + 220 épocas sin filtro			**Test - 7** etiqueta simple + 376 épocas con filtro			**Test - 8** etiqueta doble + 220 épocas con filtro		
precisión	0,736	73,60%	precisión	0,5643	56,43%	precisión	0,7282	**72,82%**	precisión	0,7578	75,78%
recall	0,4749	47,49%	recall	0,4541	45,41%	recall	0,7761	**77,61%**	recall	0,5084	50,84%

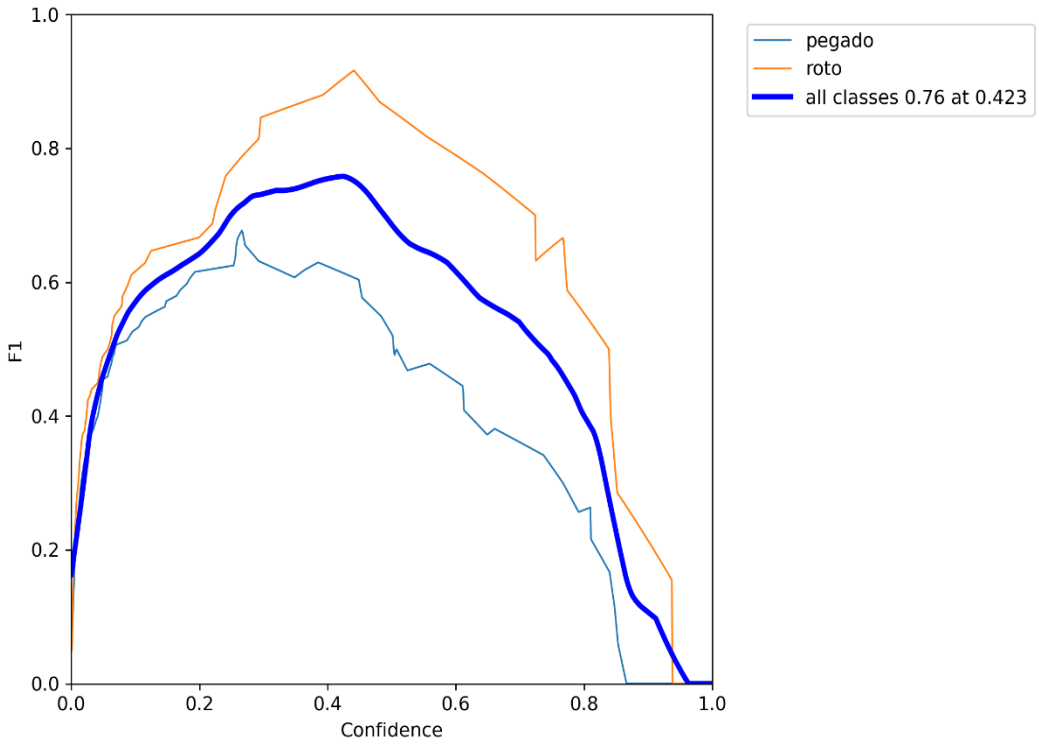
De la curva F1, figura 44, es de gran interés observar el comportamiento ya que muestra la media armónica, en otras palabras, el equilibrio entre la precisión y el recall para determinar un punto de diseño de confianza,

A partir de la curva F1, el valor de confianza (confidence) que optimiza la precisión y el recall es de 0.423 en el modelo entrenado, en varios casos es deseable que el valor de confianza sea más alto y aproximado a 1.

Figura 44

F1_curve

Fuente: Elaboración propia (TensorBoard)



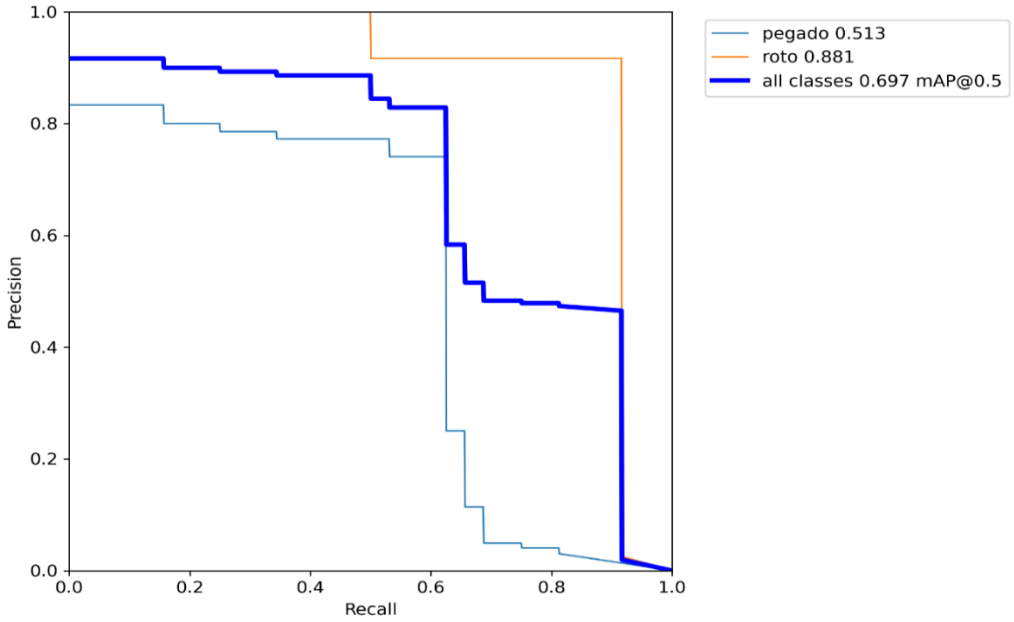
A continuación, se muestra la curva PR_curve (precisión–recall), ver figura 45, esta curva permite ver que a partir del recall, la precisión puede disminuir o aumentar.

El resultado óptimo para esta curva es que se acerque lo máximo posible a la esquina superior derecha, donde se obtiene una alta precisión y alto recall. La figura 45 indica como se traza la curva por cada etiqueta y la curva general del modelo.

Figura 45

PR_Curve

Fuente: Elaboración propia (TensorBoard)



Las figuras 46 y 47 indica como el modelo ha ido cobrando precisión (0.7282) durante el transcurso de las 376 épocas, disminuyendo los errores en la predicción y al mismo tiempo aumentando la sensibilidad (0.7761), a la vez se observa la existencia de oscilaciones en las etapas intermedias de las épocas, debido a que el muestreo inicial fue de 289 imágenes.

La precisión se calculó a partir de la división del número de verdaderos positivos entre la suma de verdaderos positivos y falsos positivos mientras que la sensibilidad (recall), se calculó mediante la división de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos.

Figura 46

Precisión del modelo

Fuente: Elaboración propia (TensorBoard)

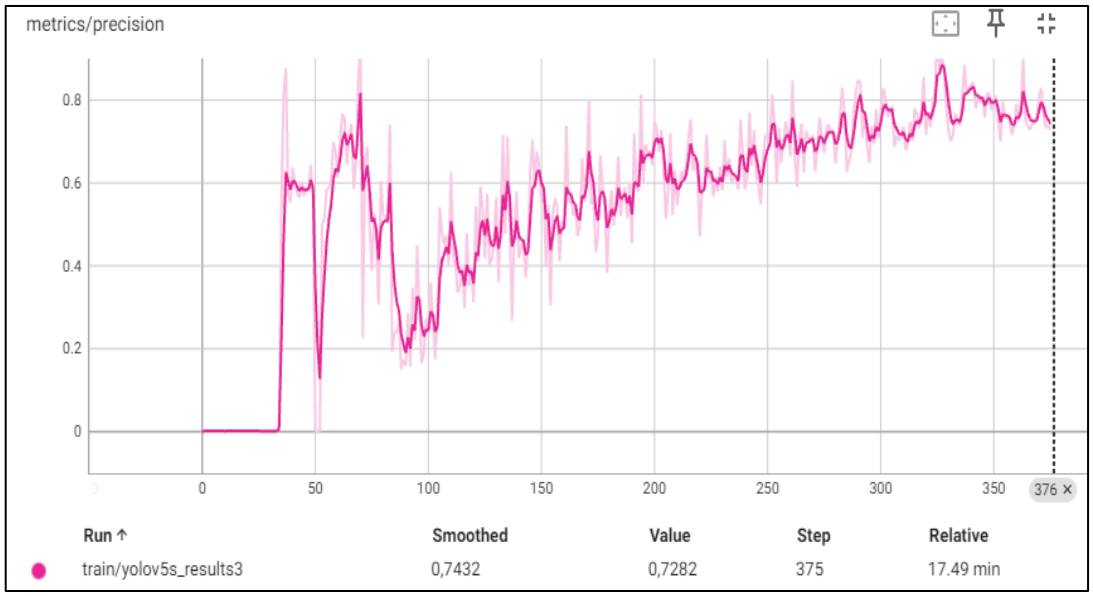
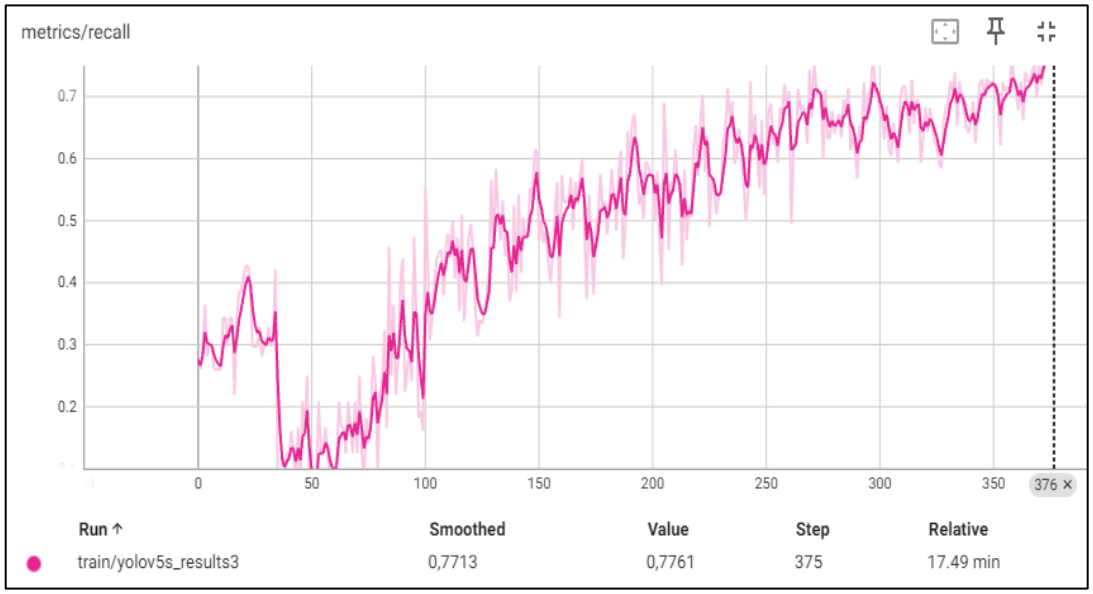


Figura 47

Recall del modelo

Fuente: Elaboración propia (TensorBoard)



En resumen, el sistema de clasificación cuenta con un 72% de precisión al detectar si es papel pegado o papel roto, mientras cuenta con un 77% de sensibilidad es decir la cantidad de errores de calidad que detecta en una misma imagen.

Estos porcentajes podrían mejorarse aumentando el dataset, de esta manera la precisión y sensibilidad del sistema aumentará ya que como se mencionó con anterioridad se usó un número limitado de imágenes para el entrenamiento, debido a las limitaciones que proveen los software libres que son gratuitos, (García, 2021) nos indica que “un dataset que sea considerablemente extenso para proporcionar una buena inferencia es cuando su tamaño se encuentra entre las 1000 imágenes, lo que genera un mayor retardo en la convergencia”

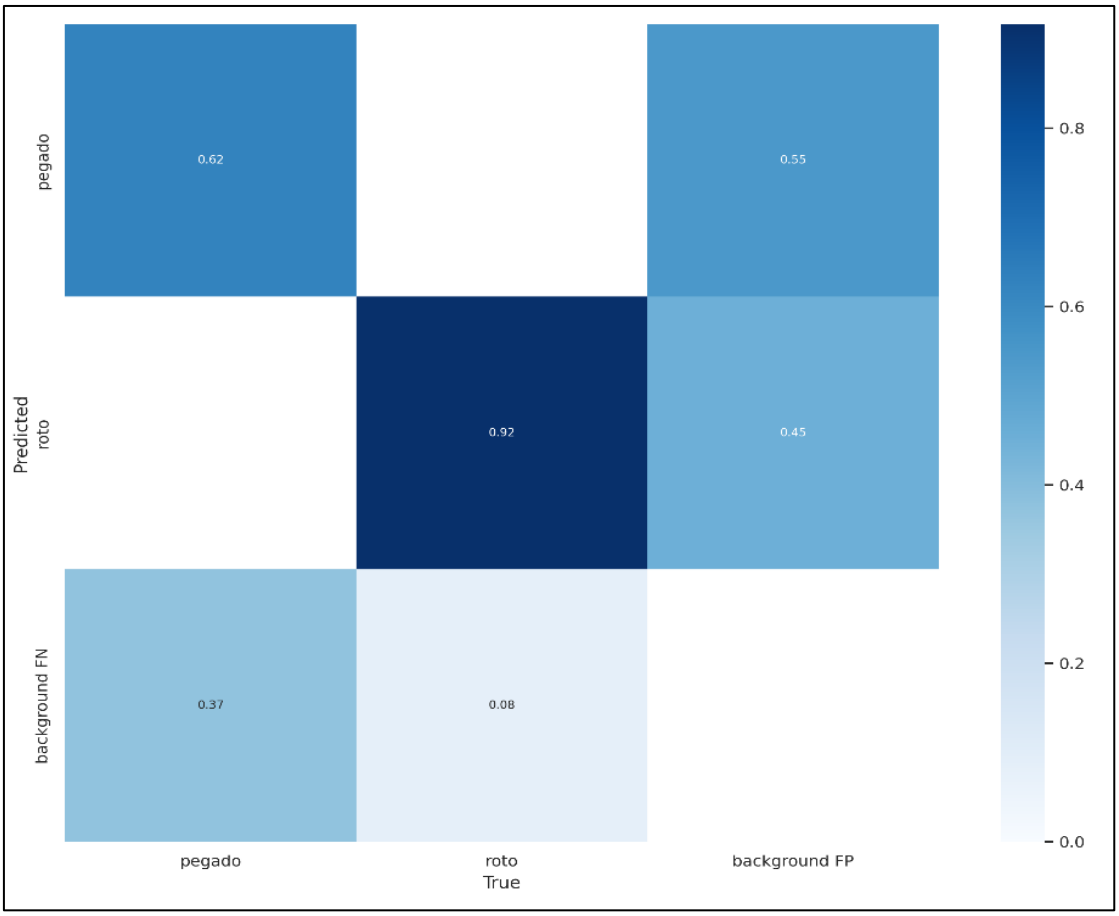
El uso de cámaras de alta resolución o cámaras de escáner lineales con filtros e iluminación apropiado contribuirá a mejorar e intensificar las características que presenta el papel pegado, de esta manera se espera que los porcentajes de detección y aciertos en la clasificación durante el entrenamiento se incrementen ya que el sistema actual durante el entrenamiento no tuvo un resultado muy satisfactorio con una precisión del 77% y sensibilidad del 52% de papel pegado versus una precisión del 86% y sensibilidad del 91% de papel roto muy aceptable.

Finalmente se realiza la matriz de confusión del modelo, la figura 48 proporciona un aspecto visual de como el modelo es capaz de cometer errores sistemáticos en la predicción de papel roto y papel pegado.

Figura 48

Matriz de confusión del modelo entrenado

Fuente: Elaboración propia (TensorBoard)



La imagen 48 muestra que el modelo de clasificación tuvo un 92% de aciertos identificando la falla de calidad de papel roto fue efectivamente papel roto, y un 8% de papel roto que no fue detectado y lo interpreto como tablero normal sin falla.

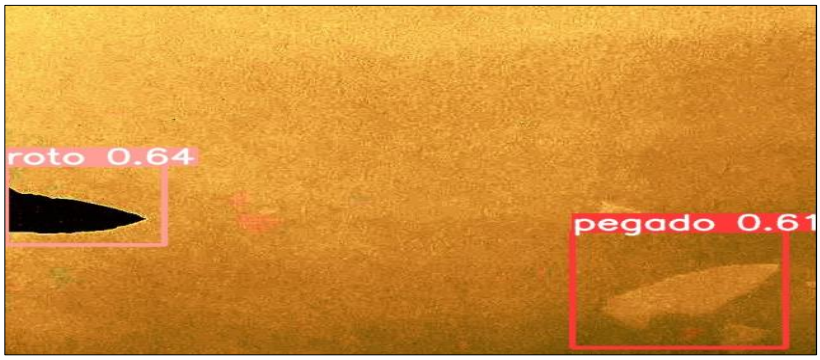
Mientras que tuvo un 63% de aciertos identificando la falla de calidad de papel pegado que efectivamente es papel pegado y un 37% que no fue identificado como papel pegado y lo interpreto como tablero normal sin falla.

En ninguno de los casos el papel roto fue identificado como papel pegado y viceversa. Los ejemplos de la clasificación se pueden apreciar en la tabla 13.

Tabla 13

Resultados de la clasificación de fallas de calidad en la simulación

Fuente: Elaboración propia (TensorBoard)

Fallas	Resultado de la simulación
Papel roto	
Papel pegado	
Papel pegado y roto	

Es interesante observar como el sistema de clasificación basado en una red neuronal convolucional con arquitectura YOLOv5, es capaz de detectar este tipo de fallas, que la limitante para este estudio aparte de no contar con un dataset amplio, fue la de no poder aislar el área de interés del papel pegado para que el sistema

pueda familiarizarse y tener una precisión y sensibilidad alta durante el entrenamiento.

Para el estudio es necesario determinar la factibilidad del sistema de clasificación para que la organización pueda tomar la decisión de invertir en cámaras, filtros e iluminación con un alto costo y si bien es cierto que no se tienen resultados de una implementación debido a las complicaciones de adquirir imágenes en tiempo real durante el proceso de producción debido a que no deben perder continuidad las líneas de laminado y el tiempo de respuesta por parte del clasificador oscila entre 2 y 3 segundos, se presenta los resultados de la simulación del sistema de clasificación y observando cómo se comportó durante el test de simulación con las imágenes de prueba de la tabla 13, podemos asumir que con un 70% de precisión siendo este valor el peor de los casos mas no es el real, ya que se demostró en la simulación del sistema que la precisión fue superior a 70% pero en términos prácticos y determinar la factibilidad se va a realizar los cálculos con este valor.

De esta manera podemos decir que los tableros que no fueron identificados por parte de los operadores de forma manual, indicados anteriormente en la tabla 9 el sistema los hubiera detectado en un 70% y deteniendo la producción para corregir la causa y en el peor de los casos un 30% no serían detectados, obteniendo los resultados de la tabla 14.

Tabla 14

Resultados clasificación manual vs sistema de clasificación automático

Fuente: Elaboración propia.

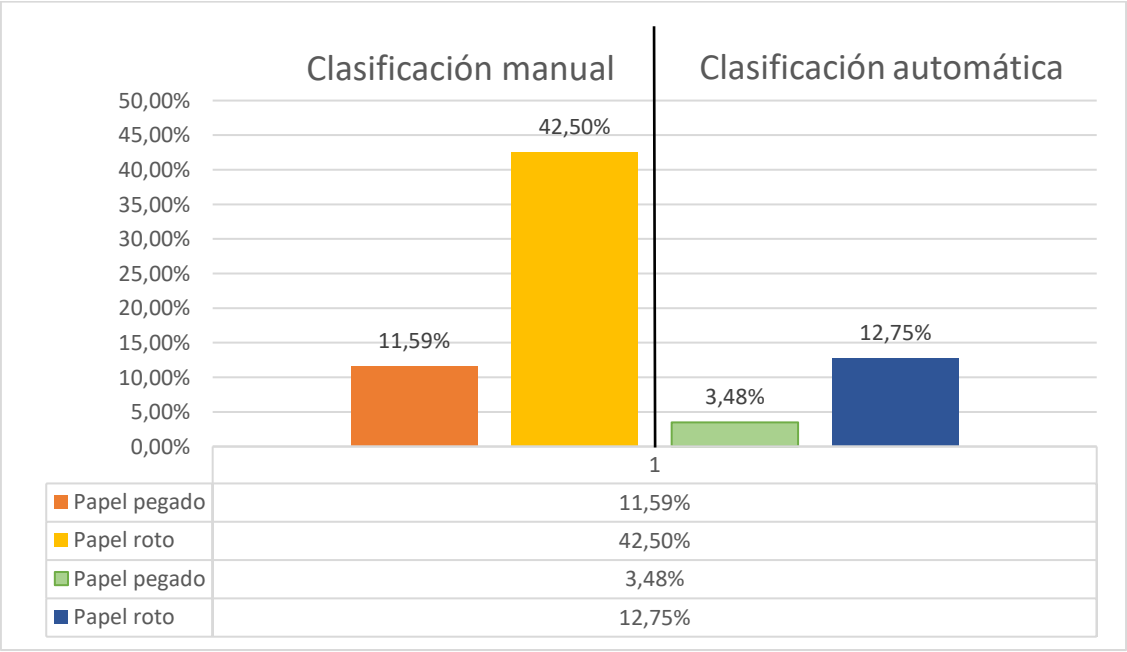
	Resultados de la clasificación manual en dos periodos anuales			Simulación con 70% de precisión	
	Total, de tableros clase B	% de tableros con papel pegado	% de tableros con papel roto	% de tableros con papel pegado (simulación)	% de tableros con papel roto (simulación)
Periodo	**Clase B**	**Papel pegado**	**Papel roto**	**Papel pegado**	**Papel roto**
2021	12578	1216	5084	365	1525
2022	11045	1523	4955	457	1487
Total	23623	2739	10039	822	3012
%	100%	11,59%	42,50%	3,48%	12,75%

En el capítulo 3 en la tabla 8 se indicó que el total de tableros clase B es de 23623, para este estudio solo se toma en cuenta al papel pegado y roto que representan el 11.59% y 42.50% respectivamente del global de tableros clase B.

Figura 49

Disminución del porcentaje de tableros rotos y pegados con el sistema de clasificación automática

Fuente: Elaboración propia.



De la tabla 14 y figura 49, podemos concluir que en el papel pegado se obtiene una disminución del 11.59% al 3.48% y en el papel roto que básicamente es el mayor defecto de calidad que presenta la línea de laminado disminuye del 42.50% al 12.75% incluso este porcentaje podría aún disminuir ya que en el papel roto el sistema de clasificación de comporta casi como un sistema ideal de clasificación con porcentajes de precisión de 86% y sensibilidad de 92%. Demostrando la factibilidad del proyecto en caso de que la organización decida implementar el sistema.

La carta de control tipo U, es una herramienta estadística utilizada en el control de calidad, con el objetivo de monitorear y gestionar el proceso de producción, para este caso la producción de tableros laminados y el control de calidad en el proceso de clasificación.

La carta de control tipo U permite visualizar patrones y tendencias en los datos del proceso de clasificación de tableros laminados. Esto facilita la detección temprana de posibles desviaciones o problemas en la calidad del producto antes de que se vuelvan más significativos. Al mostrar claramente la variabilidad del proceso, la carta de control tipo U ayuda a identificar las causas de las variaciones y permite tomar medidas correctivas de manera más eficiente. Esto es esencial para mantener un proceso productivo consistente.

La aplicación de la carta de control tipo U, se realiza con los históricos de producción indicados en las tablas 4, 5, 6 y 7 indicado en el capítulo 3. Las cantidades presentadas en los defectos son calculadas con el mismo criterio con el que se mide la factibilidad en la tabla 14, de que el sistema de control de calidad

hubiera detectado el 70% y el 30% en el peor de los casos no. Los datos para los cálculos se indican en la tabla 15 a continuación.

Tabla 15

Históricos de producción para la construcción de la carta tipo U

Fuente: Elaboración Propia.

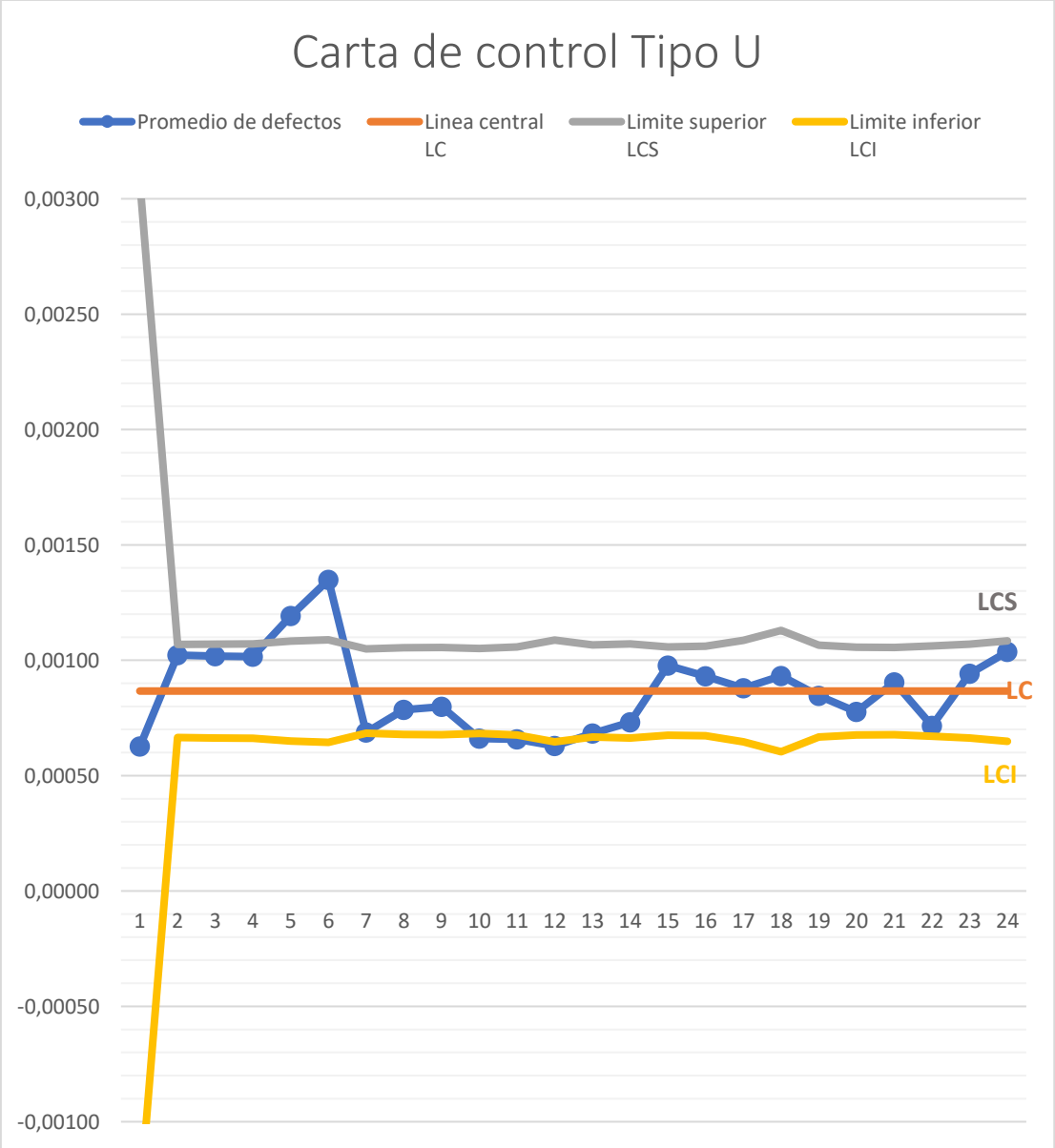
Subgrupo (mes)	Tamaño (Unidades Tableros)	# de defectos (roto y pegado)	Promedio de defectos	Línea central LC	Límite superior LCS	Límite inferior LCI
1	1597	1	0,00063	0,00087	0,0030764	-0,0013433
2	191426	196	0,00102	0,00087	0,0010684	0,0006647
3	188027	191	0,00102	0,00087	0,0010702	0,0006629
4	186320	189	0,00102	0,00087	0,0010711	0,0006619
5	166591	199	0,00119	0,00087	0,0010829	0,0006502
6	157969	213	0,00135	0,00087	0,0010887	0,0006443
7	233529	161	0,00069	0,00087	0,0010493	0,0006838
8	221101	174	0,00079	0,00087	0,0010543	0,0006787
9	217886	174	0,00080	0,00087	0,0010557	0,0006773
10	229662	152	0,00066	0,00087	0,0010508	0,0006823
11	214640	141	0,00066	0,00087	0,0010571	0,0006759
12	160208	101	0,00063	0,00087	0,0010872	0,0006459
13	195994	134	0,00068	0,00087	0,0010660	0,0006671
14	187744	137	0,00073	0,00087	0,0010703	0,0006627
15	212958	208	0,00098	0,00087	0,0010579	0,0006752
16	206965	193	0,00093	0,00087	0,0010606	0,0006724
17	161427	142	0,00088	0,00087	0,0010863	0,0006467
18	113051	105	0,00093	0,00087	0,0011292	0,0006039
19	196340	166	0,00085	0,00087	0,0010658	0,0006672
20	215100	167	0,00078	0,00087	0,0010569	0,0006761
21	217917	197	0,00090	0,00087	0,0010557	0,0006774
22	203958	146	0,00072	0,00087	0,0010621	0,0006710
23	188580	178	0,00094	0,00087	0,0010699	0,0006632
24	164104	170	0,00104	0,00087	0,0010845	0,0006485

La interpretación de los resultados calculados en la tabla 15, son presentados gráficamente en la figura 50, la cual es la carta de control tipo U para el sistema de control de calidad de tableros laminados.

Figura 50

Carta de control tipo U

Fuente: Elaboración propia.



De la carta de control se interpreta de la siguiente manera; el mes 1 se entiende a enero del 2021 y el mes 24 a diciembre del 2022, vemos que el mes 1 los límites de control son amplios debido a que solo presenta una falla de calidad y la producción fue baja en comparación de los siguientes meses.

Los meses 2, 3 y 4 se encuentran dentro de los límites, sin embargo, es muy cercano al límite superior lo que indica que estadísticamente no se encuentra bajo control. Los meses 5 y 6 se encuentran fuera de los límites de control, que indica que debió existir un suceso que origino demasiados tableros en B, y una producción baja, estadísticamente no se encontró bajo control.

Del mes 7 al 14 presenta una tendencia cercana al límite de control inferior, si bien es cierto no se acerca a la línea central, se mantuvo lineal. Del mes 15 al 21 y el mes 23 el proceso se encuentra bajo control, debido a la cercanía a la línea central con leves picos. Los meses 22 y 24 no se comportan con normalidad ya que presentan demasiada cercanía a los límites, indicando nuevamente que el proceso fue irregular.

Al identificar y abordar problemas de calidad de manera oportuna, se evitan desperdicios y retrabajos. Esto contribuye a la eficiencia del proceso y ahorra recursos que de otro modo se podrían haber utilizado para corregir defectos más adelante en la cadena de producción. La carta de control tipo U es una herramienta clave en el enfoque de mejora continua. Facilita la retroalimentación constante sobre el rendimiento del proceso, lo que permite implementar cambios graduales para optimizar la calidad y eficiencia a lo largo del tiempo. Facilita la toma de decisiones ya que proporciona información visual clara y objetiva sobre el comportamiento del proceso.

En resumen, la carta de control tipo U es una herramienta valiosa para el monitoreo y control de la calidad en procesos productivos, contribuyendo a la eficiencia operativa y a la mejora continua.

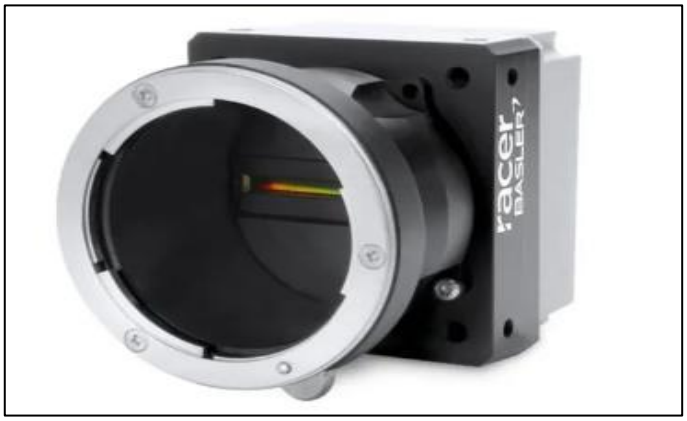
Para finalizar el estudio se presenta un presupuesto en caso de que la organización decida realizar la implementación, que consta de dos etapas y se detalla a continuación:

Las cámaras ideales para la línea de laminado y según el tipo de producción son las (Basler racer - escaneo por área) ver figura 51, marca con la que trabajan empresas como National Instruments para aplicaciones de visión artificial.

Figura 51

Cámaras Basler racer – escaneo por área

Fuente: NI-National Instruments (web)



En la primera etapa tabla 16, se necesitará de una sola cámara para poder tomar registros y crear un dataset de imágenes para poder entrenar al sistema de clasificación automático.

Tabla 16

Presupuesto de implementación etapa 1.

Fuente: Baslerweb.com

Item	Descripción	Cantidad	Valor	Total
1	Cámaras de video Basler racer	1	$1.639,00	$ 1.639,00
2	Tarjeta de video de 4 puertos Gigabit	1	$ 344,50	$ 344,50
3	Pc +Tarjeta gráfica + monitor	1	$2.500,00	$ 2.500,00
4	Materiales varios, cables conectores, encoder	1	$ 500,00	$ 500,00
				$ 4.983,50

En la segunda etapa con la red ya entrenada es necesario colocar un total de 4 cámaras debido a que la línea de laminado produce dos tableros a la vez y es necesario colocar 2 cámaras para un solo tablero tanto para la cara superior e inferior del tablero dando así un total de 4 cámaras, la tabla 17 indica lo explicado.

Tabla 17

Presupuesto de implementación etapa 2.

Fuente: Baslerweb.com

Item	Descripción	Cantidad	Valor	Total
1	Cámaras de viseo Basler racer	3	$1.639,00	$ 4.917,00
2	Lampara de iluminación	4	$1.200,00	$ 4.800,00
4	Materiales varios, cables conectores, encoder	3	$ 200,00	$ 600,00
				$10.317,00

El costo total de la implementación es de $15.300,50 un valor considerable pero teniendo en cuenta que cada tablero genera un reproceso y tiempos de entrega prolongados por la logística de transporte desde la planta hacia los centros de distribución, cuando un cliente decide devolver los tableros y aplicar una reposición los valores de la propuesta de implementación son factibles ya que se espera que todos los valores generados por reclamos disminuyan y a largo plazo se recupere la inversión, eliminando la causa raíz del problema de reclamos por parte de los clientes, la cual es enviar tableros con fallas de calidad no detectados en planta.

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Una vez realizado el análisis del proceso de clasificación de tableros, se detecta que la causa principal de la mala clasificación es la rapidez de la línea y el poco tiempo de respuesta que tiene el operador para determinar si un tablero es de calidad optima, debido a que en muchas ocasiones las imperfecciones pueden ser diminutas y difíciles de observar en un tiempo limitado, en conjunto con el exceso de confianza del operador.
- El diseño y desarrollo de un algoritmo basado en el modelo de YOLOv5 en redes neuronales para la clasificación de imágenes con el objetivo de identificar tableros en tiempo real con fallas de calidad representan un avance significativo en la automatización y mejora de procesos de evaluación de calidad. La implementación de capas convolucionales, capas de pooling y capas completamente conectadas, junto con funciones de activación adecuadas, contribuirá a la extracción eficiente de características y patrones distintivos. Este enfoque ofrece la capacidad de optimizar la eficiencia y precisión en la identificación de fallas, contribuyendo a un control de calidad más robusto y eficaz.
- La segmentación de imágenes para aislar las áreas de papel pegado y roto es un paso crítico en la evaluación de la calidad de los tableros. La implementación de métodos apropiados, la optimización continua de parámetros y la validación constante contribuyen a la identificación y manejo de imperfecciones en la producción. La aplicación de técnicas de preprocesamiento, como normalización, ajuste de tamaño y aumento de datos, contribuirá a la mejora de la robustez del modelo frente a variaciones

en las condiciones de las imágenes, ya que el papel pegado a pesar de que es detectable a la vista humana, en el modelo de la red, no fue eficaz.

- Terminado la simulación, se concluye que la mejor manera de entrenar a la red neuronal es alimentándola de imágenes mejoradas mediante filtros, debido a que el dataset, fue con imágenes tomadas con celular, y posterior se mejoró aplicando métodos de procesamiento donde se mejoró el contraste, escala de grises, erosión y detección de bordes, sin embargo, el modelo no fue capaz de tener una buena sensibilidad en las imágenes donde la falla de calidad era la de papel pegado.
- Finalmente, la mejor arquitectura de red neuronal convolucional que se tuvo acceso por su código abierto y plataformas de entrenamiento gratuito fue la de YOLOv5, la cual se caracteriza por identificar objetos de imágenes en tiempo real, lo cual es un requisito indispensable si se decide implementar en la línea de laminado, debido a la rapidez y el tiempo de decisión que se necesita para no bajar el rendimiento por hora en la producción de tableros. Su precisión y sensibilidad son equilibrados ya que se encuentran sobre el 73% un valor nada despreciable.

5.2. Recomendaciones

- En trabajos futuros, seria de interés para la organización que se pueda detectar más fallas de calidad, como las estrías o huellas de impregnación que no son fáciles de detectar a simple vista, ya que es necesario tener un ángulo de visión y luz específico para que estas fallas aparezcan, como se mencionó al inicio del presente trabajo se tomó en cuenta solo al papel roto y papel pegado.

- Dado que la obtención de imágenes para el dataset, es una tarea manual, y para este estudio fue limitado, debido a que solo se puede evidenciar cuando las fallas de calidad aparecen cuando hay un defecto en la línea de laminado, es de gran interés explorar nuevas técnicas de aumento de datos más avanzados que permita generar un dataset significativo a partir de uno pequeño.
- Es recomendable mejorar las imágenes de entrada para el entrenamiento de la red neuronal, ya que como se evidenció en los resultados, la identificación de papel pegado aumento significativamente después de tratar la imagen de entrada con un filtro, sin embargo, existen cámaras con filtros y espectros de luz que proporcionarían imágenes con mayor detalle de fallas, permitiendo tener una mayor precisión y sensibilidad.
- Sería interesante poder obtener imágenes en tiempo real con fallas de calidad durante la producción mediante una cámara de escaneado lineal, ideales para realizar inspecciones de objetos en movimiento, de esta manera no se afectaría en el tiempo de inspección del tablero ya que se ubicaría antes de la zona de clasificación, así cuando llegue mediante las bandas transportadoras el algoritmo ya tendría el tiempo suficiente de haber procesado la imagen y haber identificado si tiene o no fallas.

BIBLIOGRAFÍA

Arce, J. I. (26 de Julio de 2019). *Health BIG DATA La matriz de confusión y sus métricas*. Obtenido de https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

Badaro, S., Ibañez, L. J., & Aguero, M. J. (2013). Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones. *Universidsad de Palermo, Facultad de Ingeniería*.

Benavides, A. R. (2017). *Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones.* Sevilla: Universidad de Sevilla. Departamento de Estadística e Investigación Operativa. Obtenido de https://idus.us.es/handle/11441/63201

Bernal Torres, C. A. (2010). *Metodología de la investigacion, administracion, economía, humanidades y ciencias sociales.* Colombia: PEARSON EDUCACIÓN.

Betancourt, D. (2018). *Cómo hacer una matriz de priorización.*

Carrión, C. B. (2020). *REDES CONVOLUCIONALES .* Sevilla: Universidad de Sevilla.

Chávez, O. C. (2018). *PRODUCTIVIDAD, GESTIÓN DE LA CALIDAD Y PRODUCTIVIDAD.* QUITO: Universidad de las Fuerzas Armadas Espe.

Costa Albuquerque, V. H., Auzuir, R., & Cortez, P. C. (2009). *Evaluation of Multilayer Perceptron and Self-Organizing Map Neural Network Topologies*

applied on Microstructure Segmentation from Metallographic Images. Portugal: NDT & E International.

Cuenca, Á., Vázquez Martín, R., Anthony Mandow, Jesús Morales, & García Cerezo, A. (2021). *Análisis de técnicas de aumento de datos y entrenamiento en YOLOv3 para detección de objetos en imágenes RGB y TIR del UMA-SAR Dataset.* Universidad de Malaga.

de la Escalera Hueso, A. (2001). *VISIÓN POR COMPUTADOR. Fundamentos y métodos.* Madrid: Pearson Education.

Ferrari, A., Lombardi, S., & Signoroni, A. (2017). *Bacterial colony counting with Convolutional Neural Networks in Digital Microbiology Imaging.* Sciencedirect. Obtenido de https://www.sciencedirect.com/science/article/abs/pii/S0031320316301650

Garcia , S., Fernandez, A., Luengo, J., & Herrera, F. (2009). *A study of statistical techniques and performance measures for genetics based machine learning: accuaracy and interpretability* (3 ed.).

García Villanueva, M., & Romero Muñoz , L. (2020). *Diseño de una arquitectura de red neuronal convolucional para la clasificacion de objetos.* Ciencia Nicolaita 81.

García, L. M. (2021). *Viabilidad y rendimiento de YOLOv5 en Raspberry Pi 4 modelo B.* Sevilla: Dpto. Ingeniería Telemática Escuela Técnica Superior de Ingeniería Universidad de Sevilla. Obtenido de https://biblus.us.es/bibing/proyectos/abreproy/93731/fichero/TFG-3731%20MU%C3%91IZ%20GARC%C3%8DA,%20LUIS.pdf

Gómez Sarasa, C., & Ortega Pabón, J.D. . (s.f.). *Técnicas de aumento de datos para imágenes aéreas y evaluación de rendimiento en modelos de deep learning.* Revista Universidad Católica de Oriente. Obtenido de https://revistas.uco.edu.co/index.php/uco/article/view/285/371

Gonzales, R., & Richard E, W. (2018). *Digital Image Processing* (4 ed.). England: Pearson Education. Obtenido de https://dl.icdst.org/pdfs/files4/01c56e081202b62bd7d3b4f8545775fb.pdf

Goodfellow , I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press. Obtenido de https://www.deeplearningbook.org/

Gutiérrez Pulido, H. (2014). *Calidad Total y productividad.* Mexico: McGRAW-HILL.

Gutierrez Pulido, H., & De la Vara Salazar, R. (2013). *Control Estadistico de la Calidad y Seis Sigma.* México: Mc Graw Hill.

Helena, G., Cezar, J., & Miranda, B. (2015). *Computer-aided diagnosis system based on fuzzy logic for breast cancer categorization.* Science Direct.

Hernández Sampieri, R., Fernández Collado, C., & Pilar Baptista, L. (2014). *Metodología de la Investigación.* México: The McGraw-Hill Companies.

Hui, J. (13 de marzo de 2018). *SSD object detection: Single Shot Multibox Detector for real-time processing*. Obtenido de https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

Ian , G., Yoshua , B., & Aaron , C. (2016). *Deep learning book*. (M. Press, Editor) Obtenido de www.deeplearningbook.org

Islam , M., Hannan , M., Basri , H., & Hussain , A. (2014). *Solid waste bin detection and classification using Dynamic Time Warping and MLP classifier.* Waste Manag.

Juran, J., Dr. Frank M, & Bingham, R. (2021). *Manual de Control de la Calidad* (Segunda ed.). México: REVERTE.

Li, Y. a.-O. (2015). Joint embeddings of shapes and images via CNN image purification. *ACM Trans Journal*. Obtenido de https://dl.acm.org/doi/10.1145/2816795.2818071

LIDERLOGO (SEO y Machine Learning: Cómo la Inteligencia Artificial Está Cambiando la Optimización de Búsqueda). (2023).

López, J. F. (2018). *Control estricto de matrices de confusion por medio de distribuciones multinomiales.* España: Revista Internacional de Ciencia y Tecnología de la Información Geográfica.

Maldonado, J. A. (2015). *Fudamentos de calidad total .*

Management, F. S. (31 de Agosto de 2022). *FORE School of Management*. Obtenido de https://www.fsm.ac.in/blog/an-introduction-to-machine-learning-its-importance-types-and-applications/

Maya Carillo, M., Vallejo Villarreal, A., Ramos, V., & Borsic Laborde, Z. (2018). CULTURA ORGANIZACIONAL E INNOVACIÓN EN LAS EMPRESAS. *CinciAmerica*. Obtenido de https://cienciamerica.edu.ec/index.php/uti/article/view/215/331

Mejía, M., & Alzate, M. (2015). *Automatic classification of pathological shapes.* Colombia: Revista de Ingenieria Universidad Distrital FJC.

Moreno, Á. A. (2019). *Clasificación de imágenes usando redes neuronales.* Sevilla: Escuela Técnica Superior de Ingeniería.

Nuñez Alverca, B. (2022). *Clasificación automática de parásitos de reptiles mediante redes neuronales convoluvionales.* Quito: EPN.

Ocaña Raza, E., Lara Calle , A., Mayorga Paredes , R., & Saá Tapia , F. (2017). Rediseño de procesos utilizando herramientas técnicas alineadas al enfoque Harrington y ciclo PHVA. *CienciAmérica, 6*. Obtenido de https://cienciamerica.edu.ec/index.php/uti/article/view/126/108

Olabe, X. B. (2008). *Redes neuronales artificiales y sus aplicaciones.* Escuela Superior de Ingenieria de Bilbao, EHU. Obtenido de https://ocw.ehu.eus/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf

Olabe, X. B. (s.f.). *Redes Neuronales Artificiales y sus Aplicaciones.* Bilbao: Escuela Superior de Ingeniería de Bilbao, EHU .

Pizer, S., E, P., John D, A., Robert Cromartie, Ari Geselowitz, Trey Greer, . . . Karel Zuiderveld. (1987). Adaptive histogram equalization and its variations. *Science Direct, 39*(3), 355-368. Obtenido de www.sciencedirect.com/science/article/pii/S0734189X8780186X

Ramírez, J. (19 de Julio de 2018). *Medium* . Obtenido de "Curvas PR y ROC": https://medium.com/bluekiri/curvas-pr-y-roc-1489fbd9a527

Raneros, S. R. (2021). *Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning.* Valladolid,: UNIVERSIDAD DE VALLADOLID .

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. *Cornell University*, 10. Obtenido de https://arxiv.org/abs/1506.02640

Robalino Lopez, A., Ramos, V., Franco , A., & Undoa , X. (2017). Diseño de un modelo-herramienta para la medicion de la innovacion en la industria ecuatoriana. *CienciAmerica, 6*, http://cienciamerica.uti.edu.ec/openjournal/index.php/uti/article/view/97/8. Obtenido de https://cienciamerica.edu.ec/index.php/uti/article/view/97/83

Roboflow. (s.f.). *Use Roboflow to build powerful computer vision models.* Obtenido de https://docs.roboflow.com/

Silva, A. H. (2020). *Diseño de Redes Neuronales Antagónicas para Estegoanálisis.* Madrid: Universidad Politecnica de Madrid .

Solutions, M. (2018). *Machine learning, una pieza clave en la transformacion de los modelos de negocio.* España.

Steve, H., Douglas G, A., & Susan, M. (2015). *Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: A discussion and proposal for an alternative approach.* Euro Radiol. Obtenido de https://doi.org/10.1007/s00330-014-3487-0

T. Tran, O. , -H. Kwon, K., -R. Kwon, S., & -H. Lee and . (2018). *Blood Cell Images Segmentation using Deep Learning Semantic Segmentation.* China: IEEE Xplore. Obtenido de IEEE Xplore: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8644754&isnumber=8644726

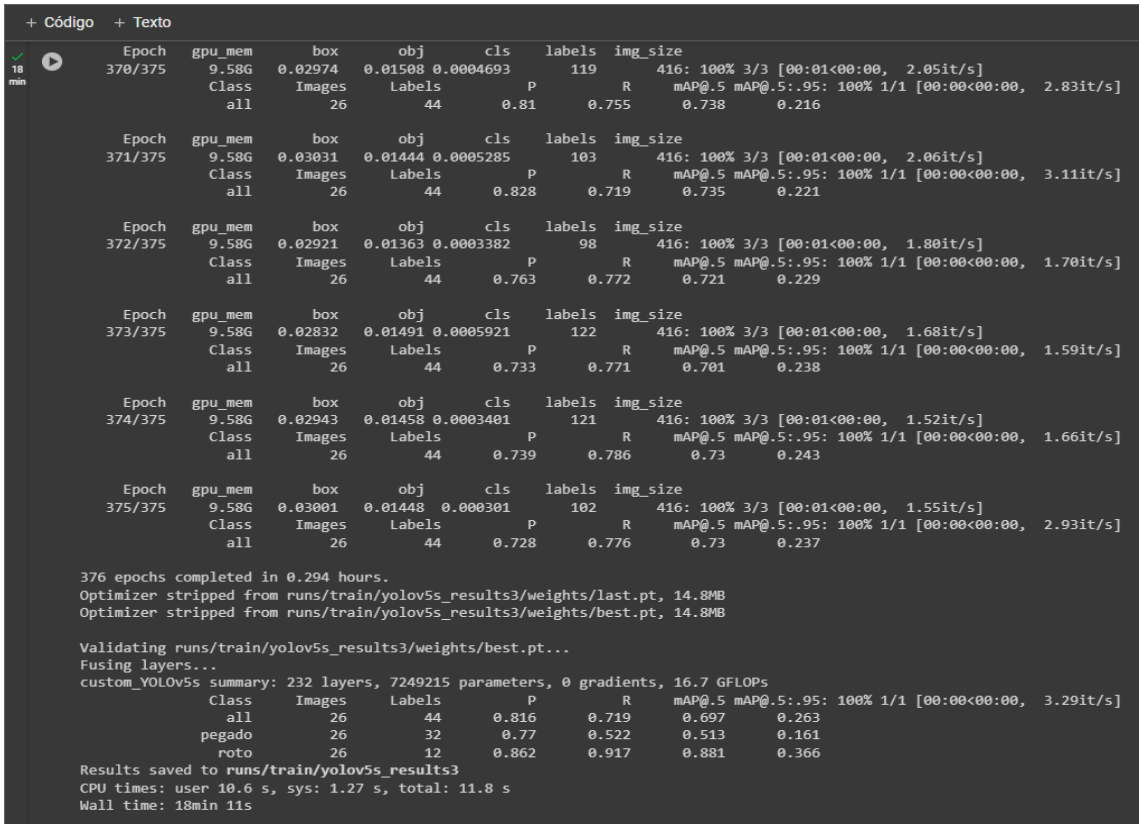
Unipython, Ecualización de histogramas. (2018). Obtenido de https://unipython.com/ecualizacion-de-histogramas/

Wilson, D., & Martinez, T. (2001). *The need for small learning rates on large problems " IJCNN'01. International Joint Conference on Neural Networks.* Washington, DC, USA,: IEEE. Obtenido de https://ieeexplore.ieee.org/document/939002

Zhang, G., Zeng, Z., Zhang, S., Zhang, Y., & Wu, W. (2017). *SIFT Matching with CNN Evidences for Particular Object Retrieval.* Sciencedirect. Obtenido de https://www.sciencedirect.com/science/article/abs/pii/S0925231217302680

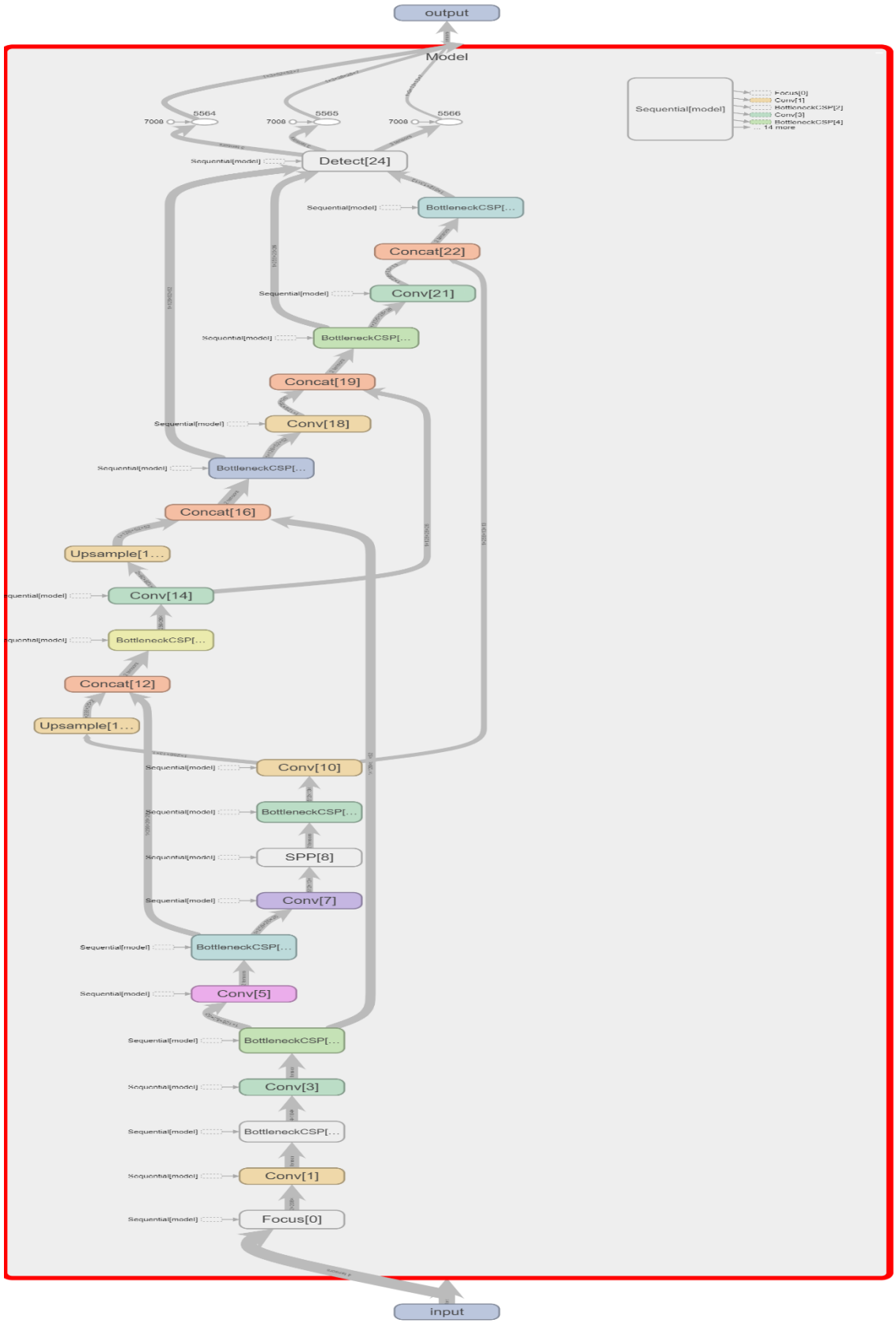
ANEXOS

Anexo 1.- Resultado del entrenamiento de la red con las 375 épocas.



```
+ Código  + Texto

18 min
      Epoch    gpu_mem       box       obj       cls    labels  img_size
    370/375     9.58G   0.02974   0.01508 0.0004693       119       416: 100% 3/3 [00:01<00:00,  2.05it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.83it/s]
                  all         26         44       0.81      0.755      0.738      0.216

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    371/375     9.58G   0.03031   0.01444 0.0005285       103       416: 100% 3/3 [00:01<00:00,  2.06it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.11it/s]
                  all         26         44      0.828      0.719      0.735      0.221

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    372/375     9.58G   0.02921   0.01363 0.0003382        98       416: 100% 3/3 [00:01<00:00,  1.80it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.70it/s]
                  all         26         44      0.763      0.772      0.721      0.229

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    373/375     9.58G   0.02832   0.01491 0.0005921       122       416: 100% 3/3 [00:01<00:00,  1.68it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.59it/s]
                  all         26         44      0.733      0.771      0.701      0.238

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    374/375     9.58G   0.02943   0.01458 0.0003401       121       416: 100% 3/3 [00:01<00:00,  1.52it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.66it/s]
                  all         26         44      0.739      0.786       0.73      0.243

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    375/375     9.58G   0.03001   0.01448  0.000301       102       416: 100% 3/3 [00:01<00:00,  1.55it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.93it/s]
                  all         26         44      0.728      0.776       0.73      0.237

376 epochs completed in 0.294 hours.
Optimizer stripped from runs/train/yolov5s_results3/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results3/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results3/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.29it/s]
                  all         26         44      0.816      0.719      0.697      0.263
              pegado          26         32       0.77      0.522      0.513      0.161
                roto          26         12      0.862      0.917      0.881      0.366
Results saved to runs/train/yolov5s_results3
CPU times: user 10.6 s, sys: 1.27 s, total: 11.8 s
Wall time: 18min 11s
```

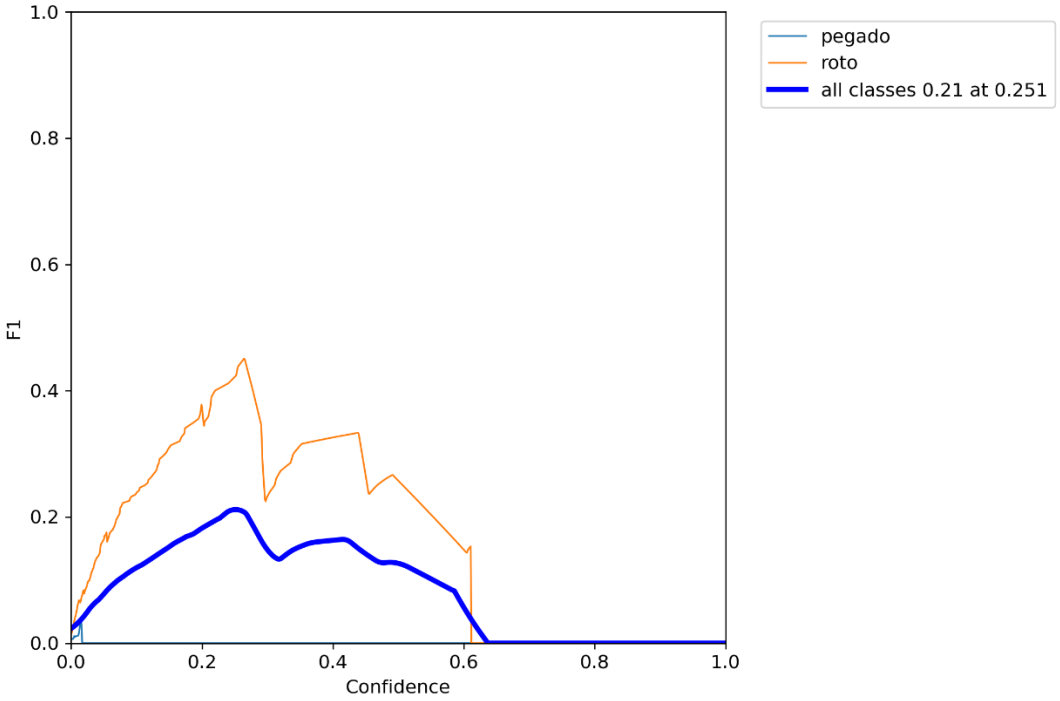
Anexo 2.- Estructura de red neuronal YOLOv5 del modelo propuesto



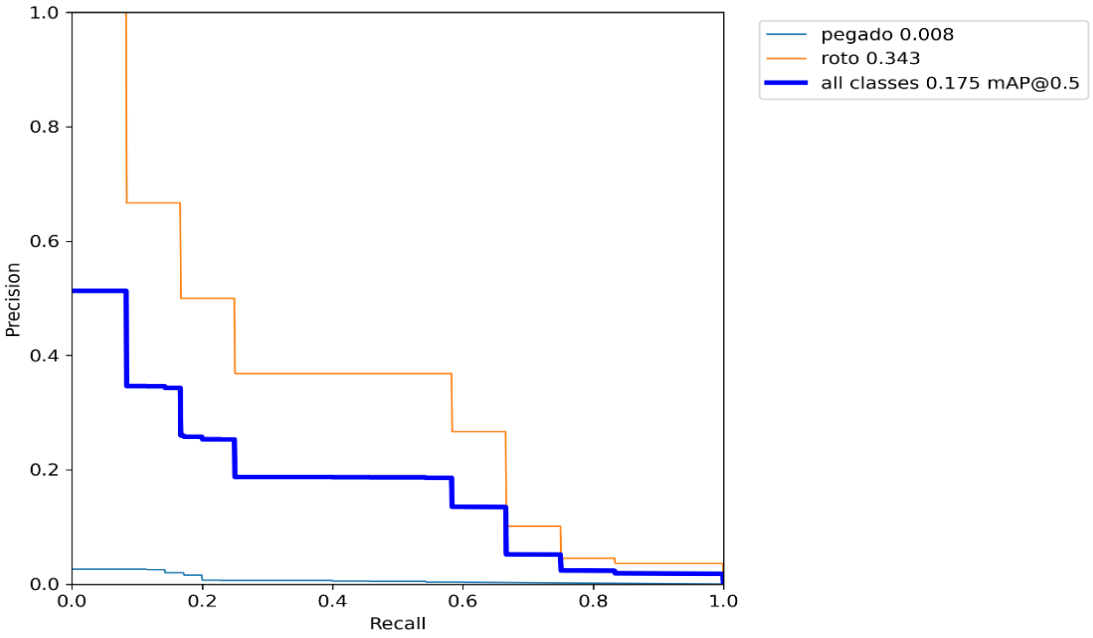
Anexo 3.- Resultado del entrenamiento de la red Test 1

```
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.52it/s]
                 all         27         47      0.182      0.306       0.15     0.0523

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      96/99     9.58G   0.06568   0.02565  0.001533       108       416: 100% 3/3 [00:01<00:00,  2.18it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.59it/s]
                 all         27         47      0.187      0.314      0.162     0.0483

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      97/99     9.58G   0.06432   0.02439  0.001483        86       416: 100% 3/3 [00:01<00:00,  2.17it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.18it/s]
                 all         27         47      0.095      0.292      0.138     0.0399

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      98/99     9.58G   0.06524   0.02321  0.001102        91       416: 100% 3/3 [00:01<00:00,  1.97it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.61it/s]
                 all         27         47      0.612      0.292      0.174     0.0497

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      99/99     9.58G   0.06445   0.02437  0.001389        82       416: 100% 3/3 [00:01<00:00,  2.03it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.95it/s]
                 all         27         47      0.596       0.25      0.163      0.049

100 epochs completed in 0.073 hours.
Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.27it/s]
                 all         27         47      0.659      0.292      0.175     0.0595
              pegado         27         35          1          0    0.00751    0.00183
                roto         27         12      0.317      0.583      0.343      0.117
Results saved to runs/train/yolov5s_results
CPU times: user 2.55 s, sys: 337 ms, total: 2.88 s
Wall time: 4min 57s
```

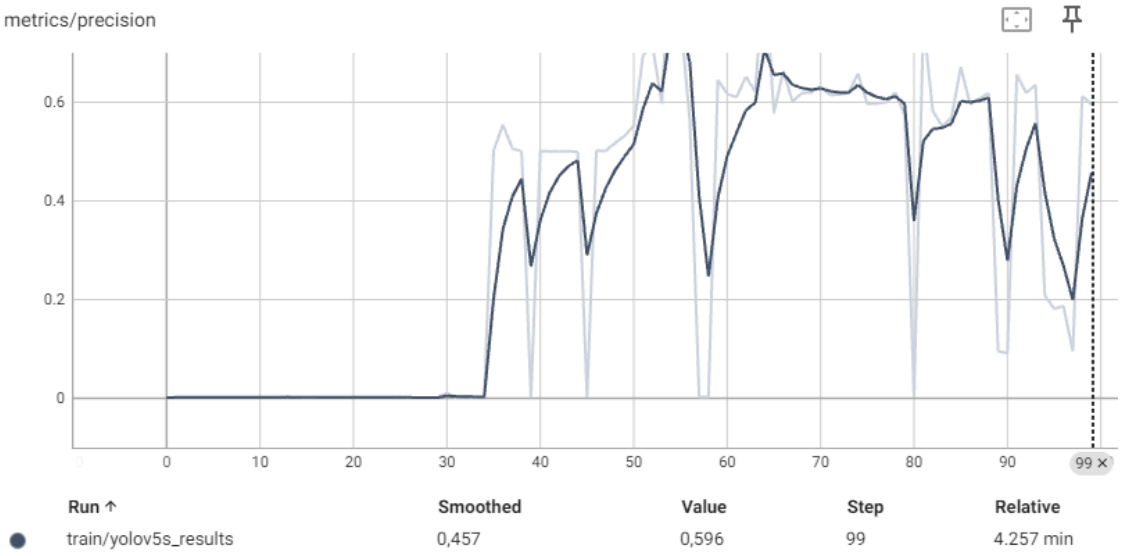
Anexo 4.- F1_curve Test 1



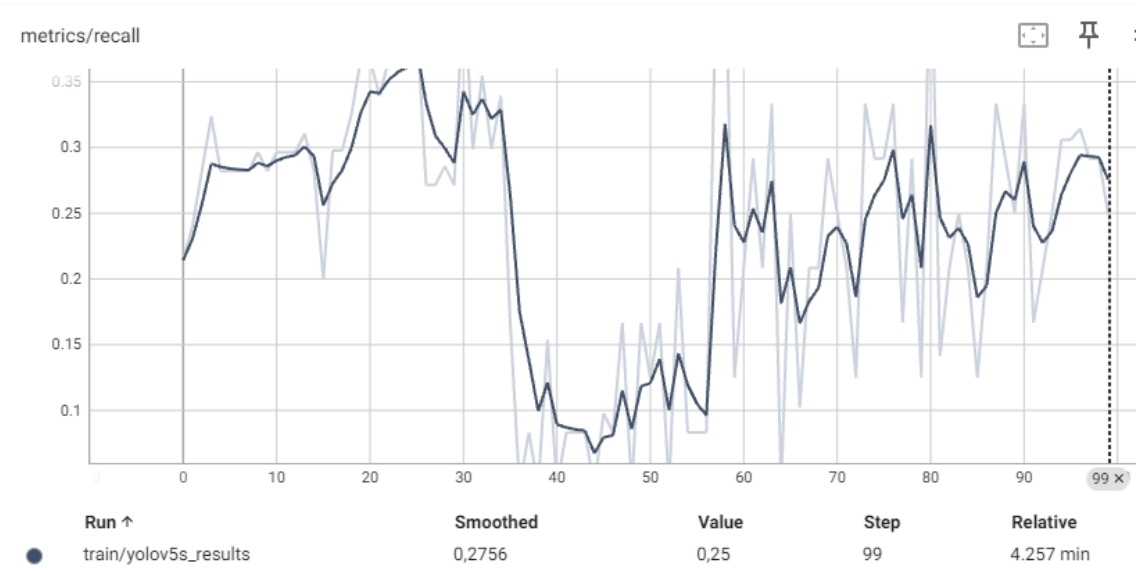
Anexo 5.- PR_ curve Test 1



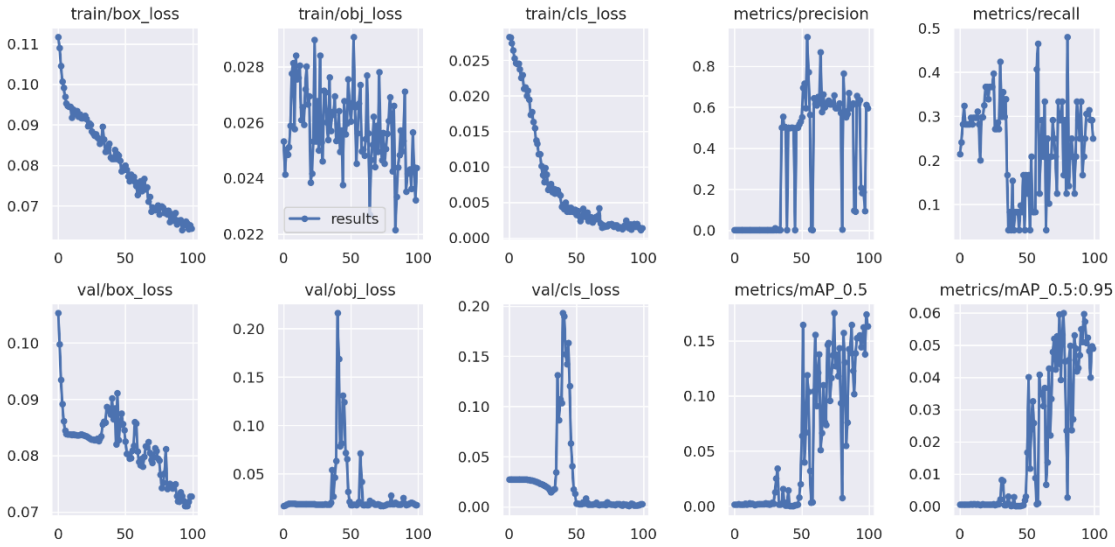
Anexo 6.- Precisión Test 1.



Anexo 7.- Recall Test 1.



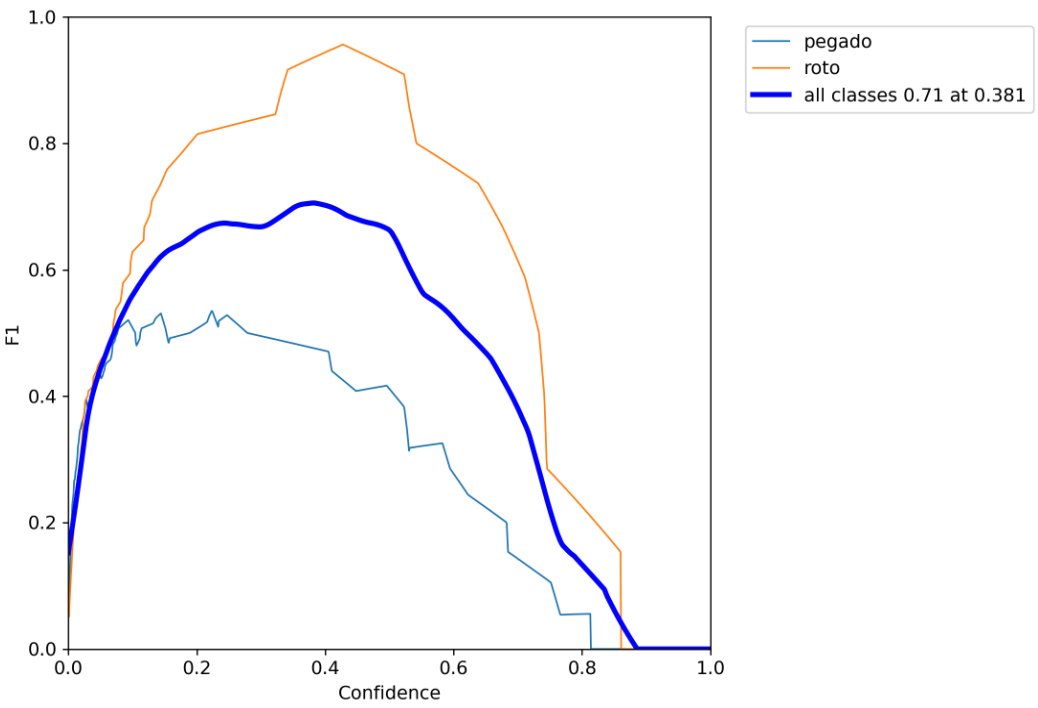
Anexo 8.- Métricas de entrenamiento del Test 1.



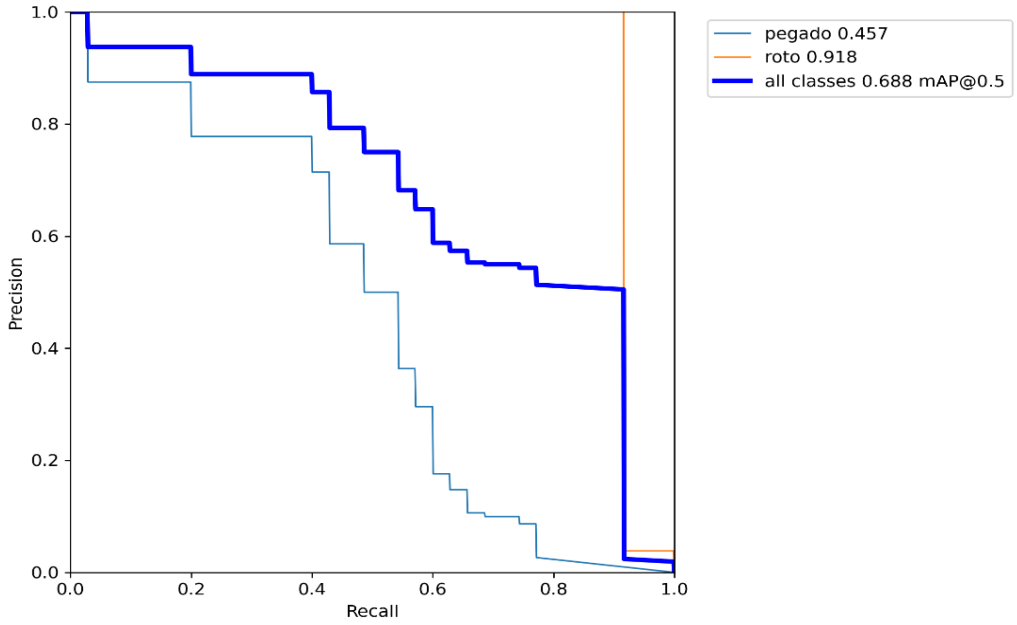
Anexo 9.- Resultado del entrenamiento de la red del Test 2

```
+ Código  + Texto

                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.63it/s]
                   all         27         47      0.765      0.561      0.574      0.194

      Epoch   gpu_mem       box       obj       cls    labels  img_size
    372/499     9.58G   0.03086   0.01313 0.0005731        74       416: 100% 3/3 [00:01<00:00,  2.05it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.92it/s]
                   all         27         47      0.752      0.674      0.663      0.221

      Epoch   gpu_mem       box       obj       cls    labels  img_size
    373/499     9.58G   0.03238   0.01414 0.0002966        82       416: 100% 3/3 [00:01<00:00,  2.10it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.92it/s]
                   all         27         47      0.683       0.66      0.639      0.234

      Epoch   gpu_mem       box       obj       cls    labels  img_size
    374/499     9.58G   0.03104   0.01445 0.0006199        90       416: 100% 3/3 [00:01<00:00,  1.76it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.56it/s]
                   all         27         47        0.8      0.715      0.728      0.225

      Epoch   gpu_mem       box       obj       cls    labels  img_size
    375/499     9.58G   0.02926    0.0148 0.0004872       107       416: 100% 3/3 [00:01<00:00,  2.03it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.72it/s]
                   all         27         47      0.795       0.65       0.68      0.211

      Epoch   gpu_mem       box       obj       cls    labels  img_size
    376/499     9.58G   0.03044   0.01325 0.0006157        79       416: 100% 3/3 [00:01<00:00,  2.10it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.79it/s]
                   all         27         47      0.694      0.556      0.573      0.168
Stopping training early as no improvement observed in last 100 epochs. Best results observed at epoch 276, best model saved as best.pt.
To update EarlyStopping(patience=100) pass a new patience value, i.e. `python train.py --patience 300` or use `--patience 0` to disable EarlyStopping.

377 epochs completed in 0.290 hours.
Optimizer stripped from runs/train/yolov5s_results2/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results2/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results2/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.22it/s]
                   all         27         47      0.855      0.632      0.688      0.293
                pegado         27         35      0.753      0.348      0.457      0.147
                  roto         27         12      0.957      0.917      0.918       0.44
Results saved to runs/train/yolov5s_results2
CPU times: user 10.1 s, sys: 1.24 s, total: 11.3 s
Wall time: 17min 59s
```

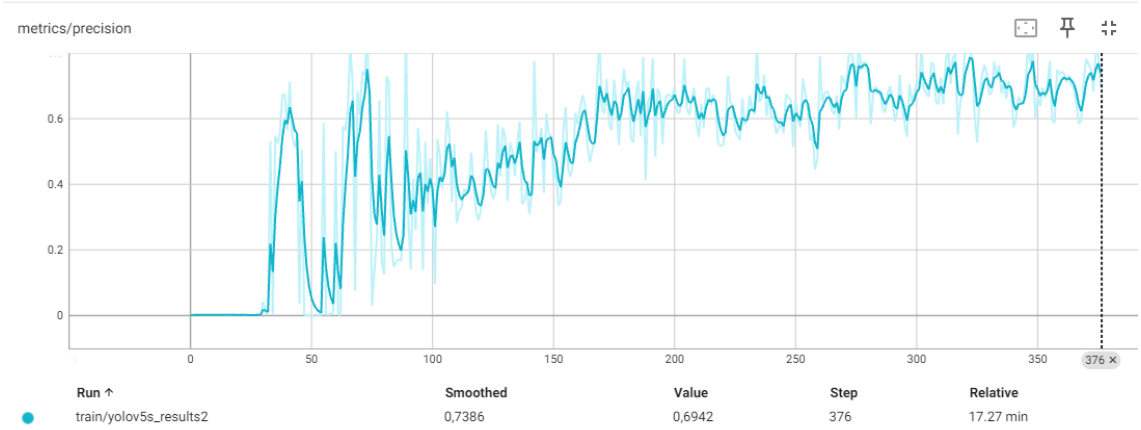
Anexo 10.- F1_curve Test 2.



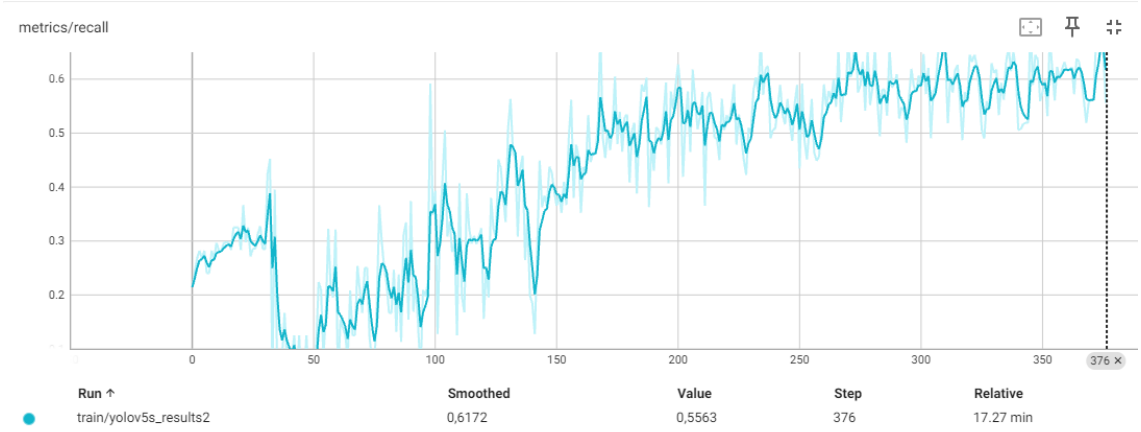
Anexo 11.- PR_curve Test 2.



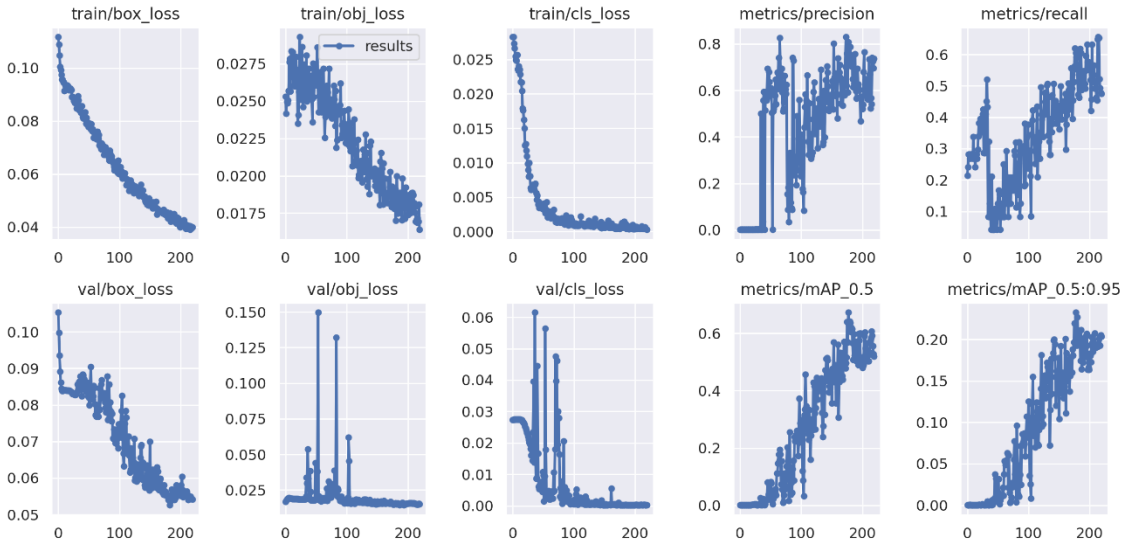
Anexo 12.- Precisión Test 2.



Anexo 13.- Recall Test 2.



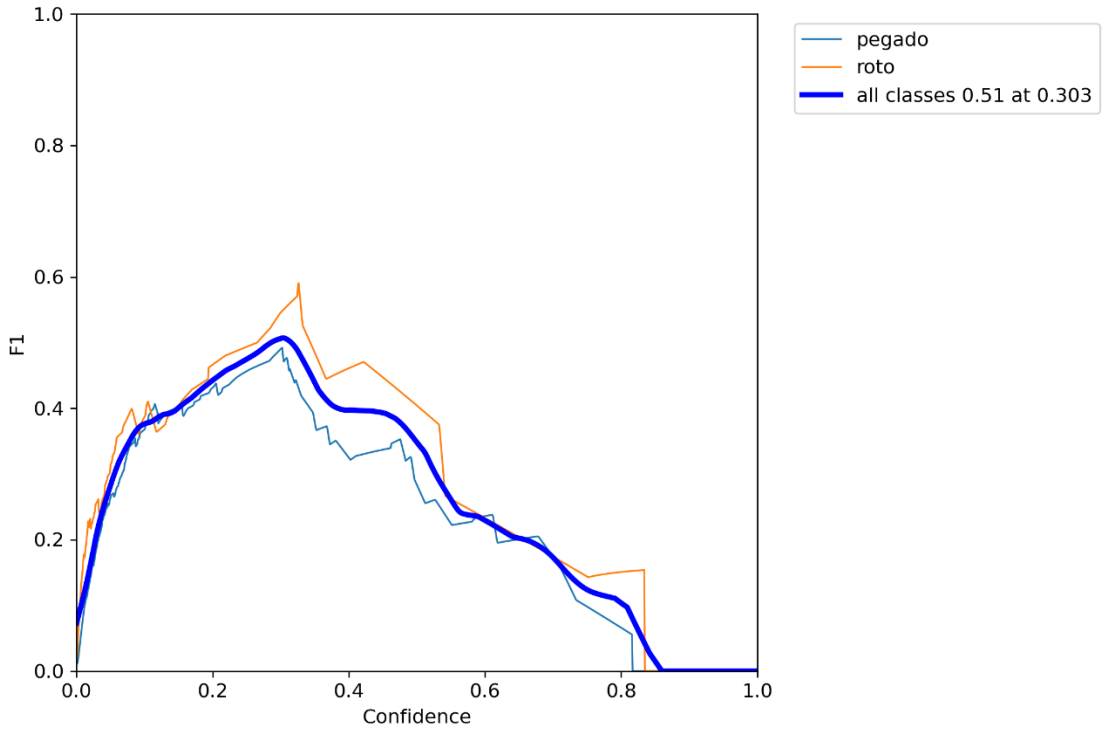
Anexo 14.- Métricas de entrenamiento del Test 2.



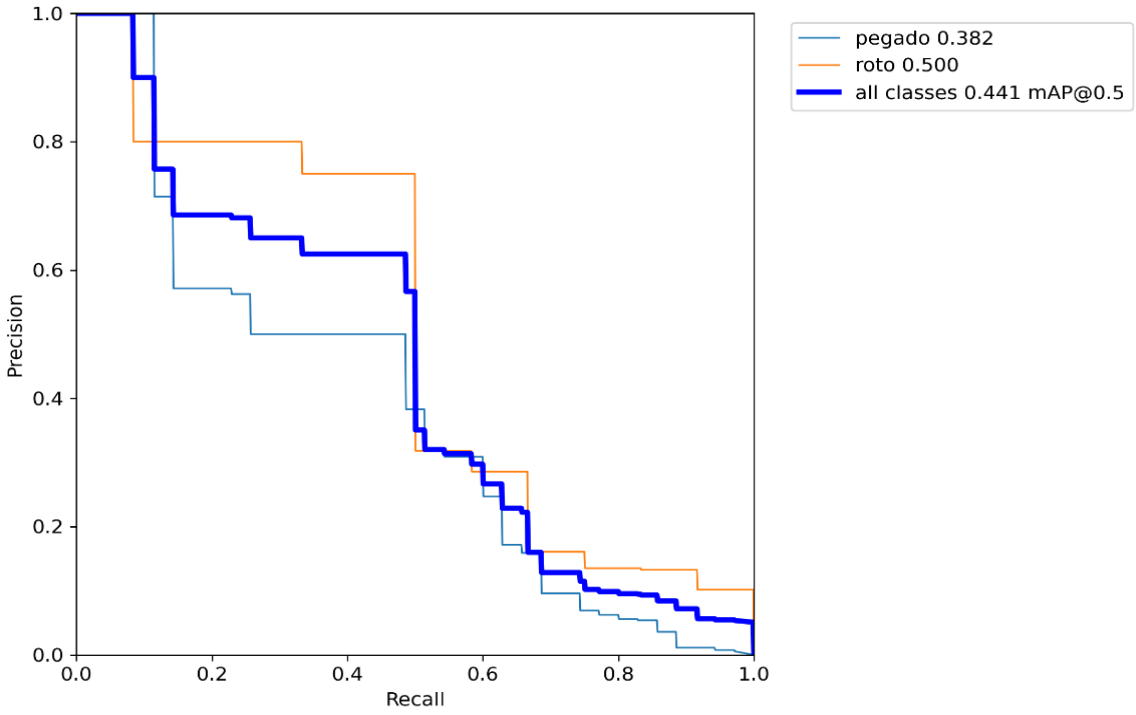
Anexo 15.- Resultado del entrenamiento de la red test 3

```
+ Código  + Texto

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    174/179      9.58G   0.04781   0.02039  0.000426        70       416: 100% 3/3 [00:01<00:00,  1.90it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.81it/s]
                   all         27         47      0.581      0.488      0.457      0.154

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    175/179      9.58G   0.04681    0.0203  0.000437        95       416: 100% 3/3 [00:01<00:00,  1.81it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.56it/s]
                   all         27         47      0.545       0.52      0.468      0.163

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    176/179      9.58G    0.0468   0.01953 0.0005677       103       416: 100% 3/3 [00:01<00:00,  2.17it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.92it/s]
                   all         27         47      0.526       0.52      0.467      0.168

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    177/179      9.58G   0.04817   0.01906  0.000373        81       416: 100% 3/3 [00:01<00:00,  2.20it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.02it/s]
                   all         27         47      0.531      0.509      0.453      0.162

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    178/179      9.58G   0.04637   0.01906 0.0008233        65       416: 100% 3/3 [00:01<00:00,  2.13it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.00it/s]
                   all         27         47      0.525       0.52      0.457      0.156

      Epoch    gpu_mem       box       obj       cls    labels  img_size
    179/179      9.58G   0.04565   0.02024 0.0004778        96       416: 100% 3/3 [00:01<00:00,  2.19it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.05it/s]
                   all         27         47      0.626       0.45      0.479      0.171

180 epochs completed in 0.130 hours.
Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.26it/s]
                   all         27         47      0.503      0.493      0.441      0.177
               pegado         27         35      0.461      0.486      0.382      0.143
                 roto         27         12      0.545        0.5        0.5      0.211
Results saved to runs/train/yolov5s_results
CPU times: user 4.39 s, sys: 552 ms, total: 4.94 s
Wall time: 8min 23s
```

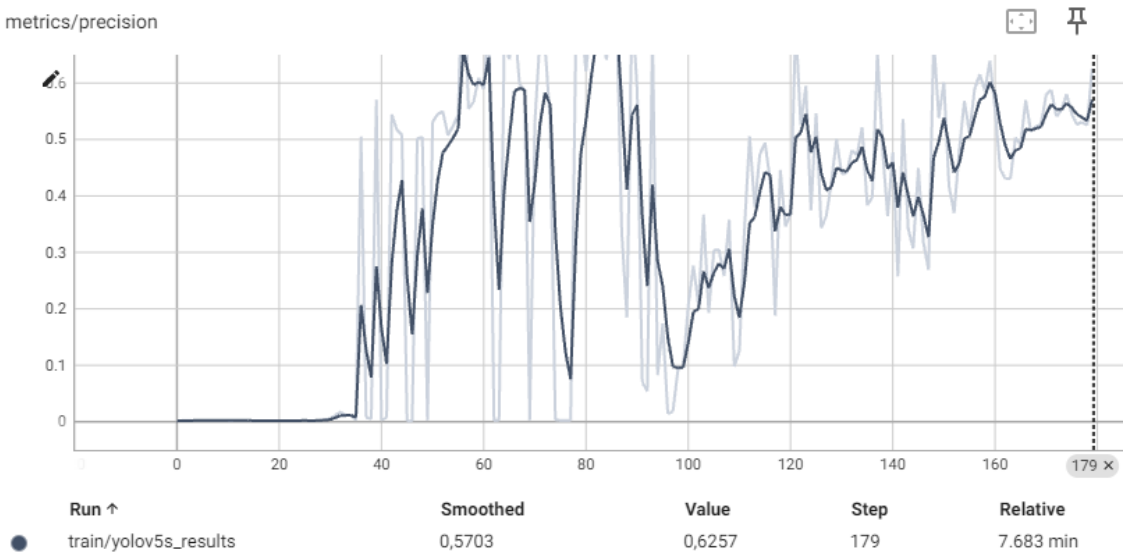
Anexo 16.- F1_Curve Test 3



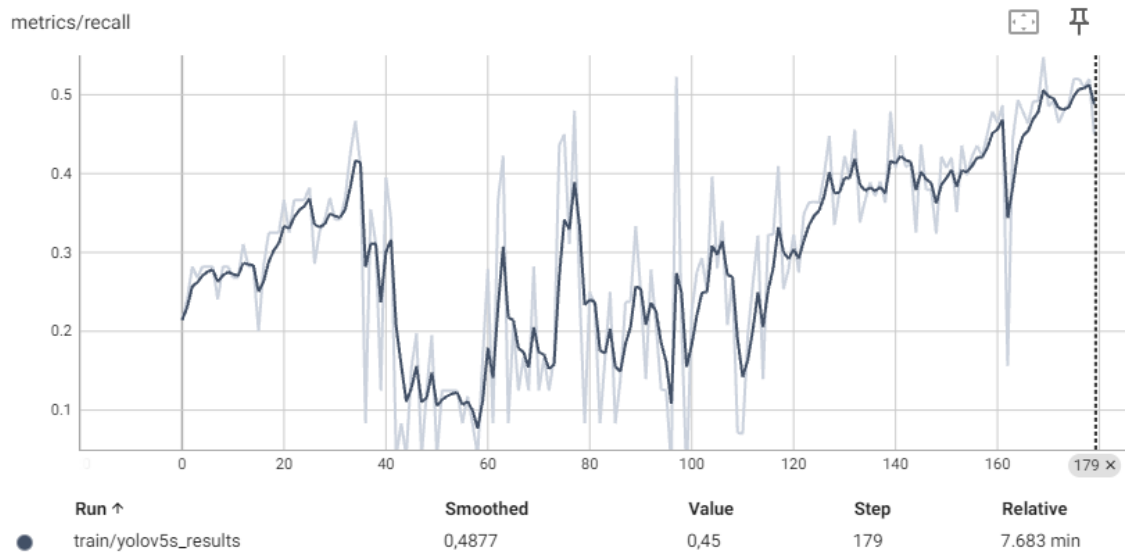
Anexo 17.- PR_Curve Test 3.



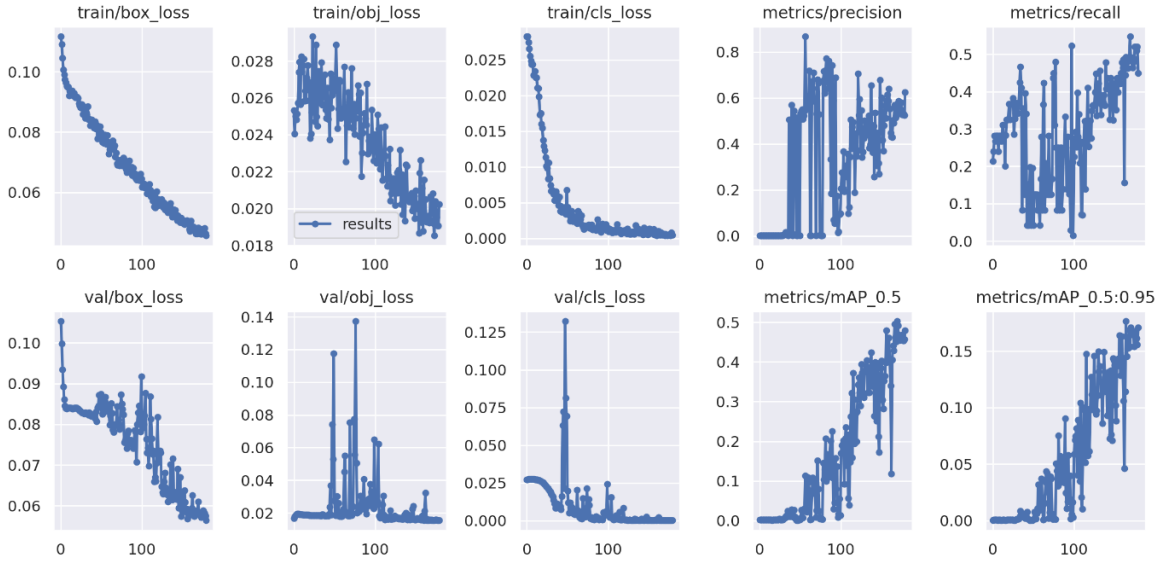
Anexo 18.- Precisión Test 3.



Anexo 19.- Recall test 3.



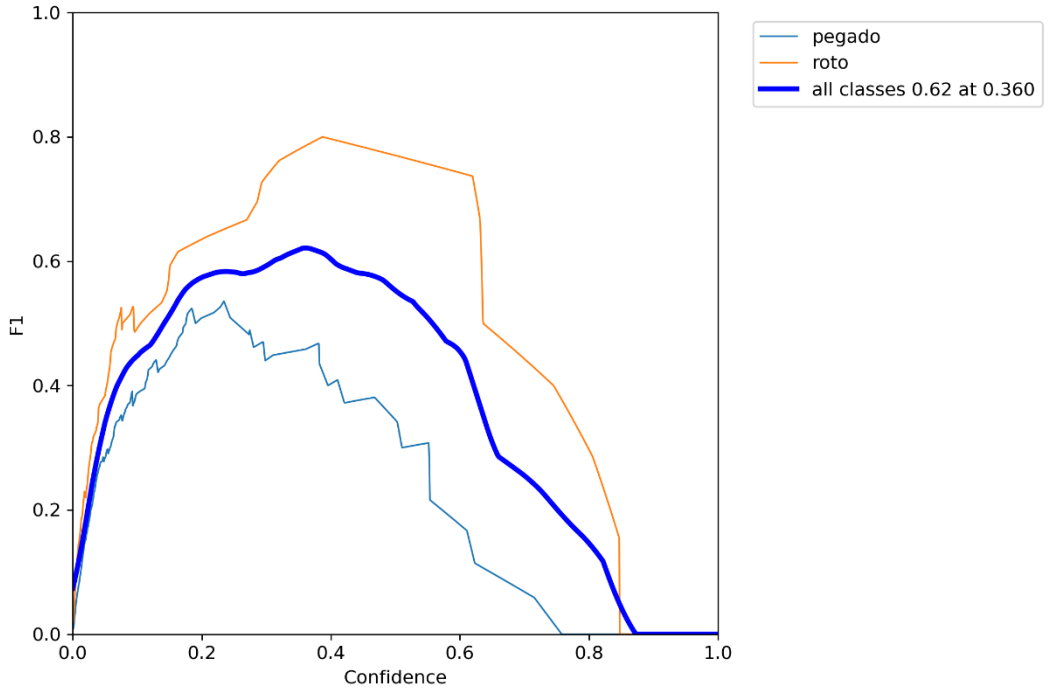
Anexo 20.- Métricas de entrenamiento Test 3.



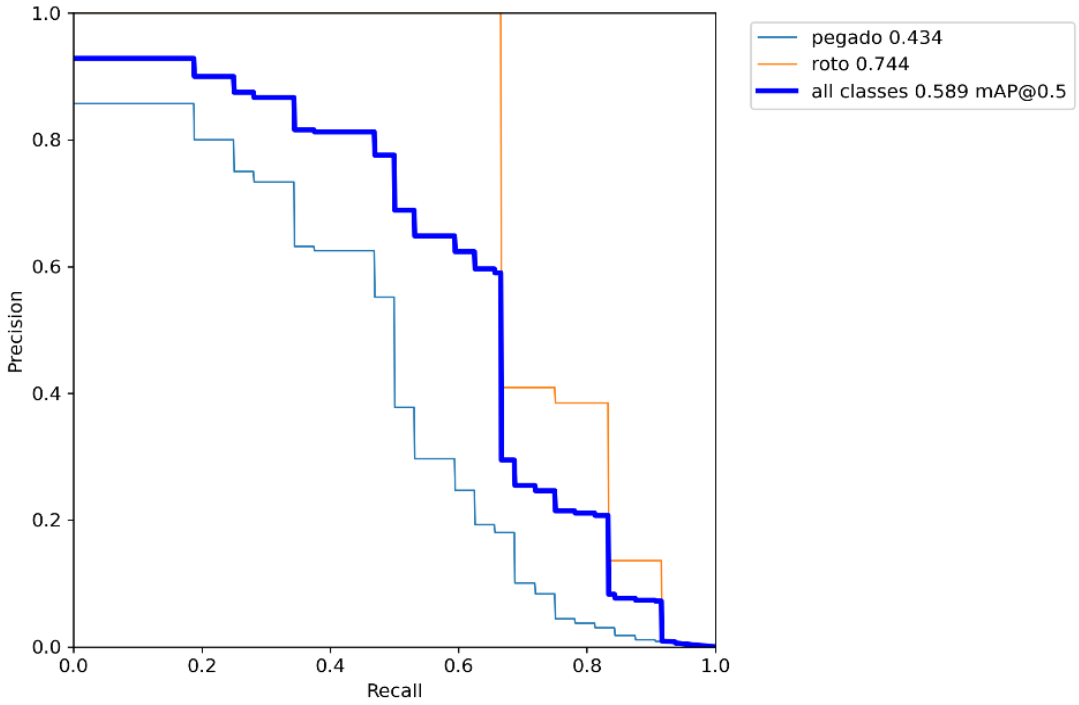
Anexo 21.- Resultado del entrenamiento de la red test 4.

```
+ Código  + Texto

[11]
6 min
      173/179     9.58G   0.04706   0.02121 0.0008904       107       416: 100% 3/3 [00:01<00:00,  2.15it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.41it/s]
                   all         26         44      0.842      0.495      0.578      0.215

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      174/179     9.58G   0.04741   0.02149   0.00135       112       416: 100% 3/3 [00:01<00:00,  1.93it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.06it/s]
                   all         26         44      0.823      0.505      0.589       0.23

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      175/179     9.58G   0.04641   0.01917 0.0008612       100       416: 100% 3/3 [00:01<00:00,  2.08it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.23it/s]
                   all         26         44      0.804      0.552      0.606      0.191

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      176/179     9.58G   0.04586   0.01888 0.0006789        81       416: 100% 3/3 [00:01<00:00,  1.85it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.24it/s]
                   all         26         44      0.822       0.53      0.579      0.226

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      177/179     9.58G   0.04564   0.02062 0.0007269       121       416: 100% 3/3 [00:01<00:00,  2.14it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.26it/s]
                   all         26         44      0.696      0.484      0.556      0.228

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      178/179     9.58G   0.04758   0.02153 0.0008776       119       416: 100% 3/3 [00:01<00:00,  2.10it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.17it/s]
                   all         26         44      0.713      0.479      0.552      0.225

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      179/179     9.58G   0.04773   0.02051  0.001071       100       416: 100% 3/3 [00:01<00:00,  2.16it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.34it/s]
                   all         26         44      0.706      0.503      0.555      0.215

180 epochs completed in 0.127 hours.
Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.33it/s]
                   all         26         44      0.823      0.505      0.589       0.23
                pegado         26         32      0.689      0.344      0.434      0.135
                  roto         26         12      0.958      0.667      0.744      0.325
Results saved to runs/train/yolov5s_results
CPU times: user 3.54 s, sys: 417 ms, total: 3.96 s
Wall time: 8min 10s
```

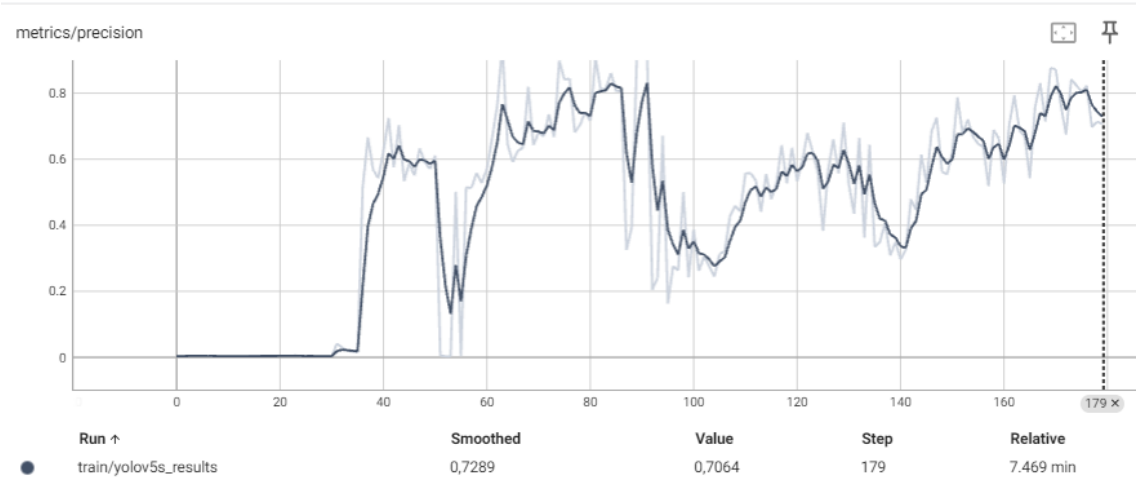
Anexo 22.- F1_Curve Test 4.



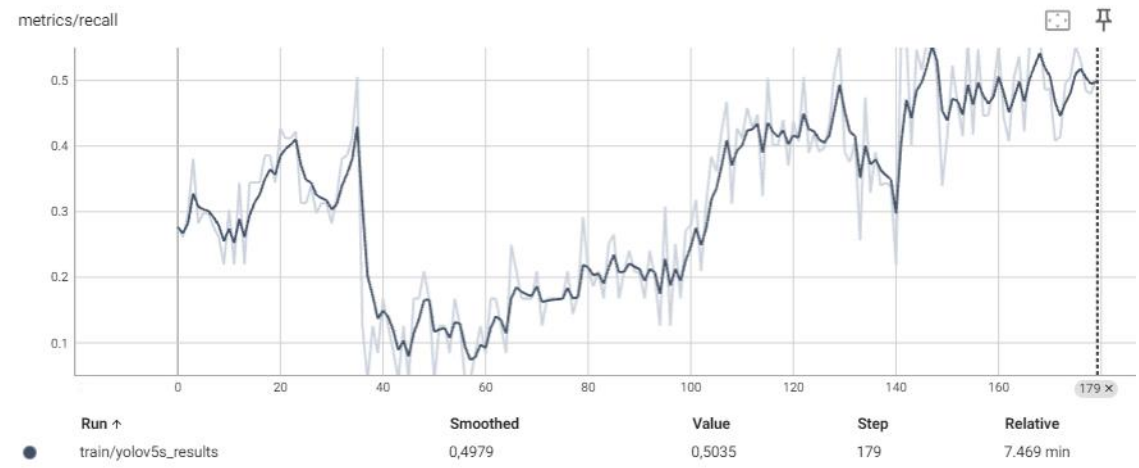
Anexo 23.- PR_Curve Test 4.



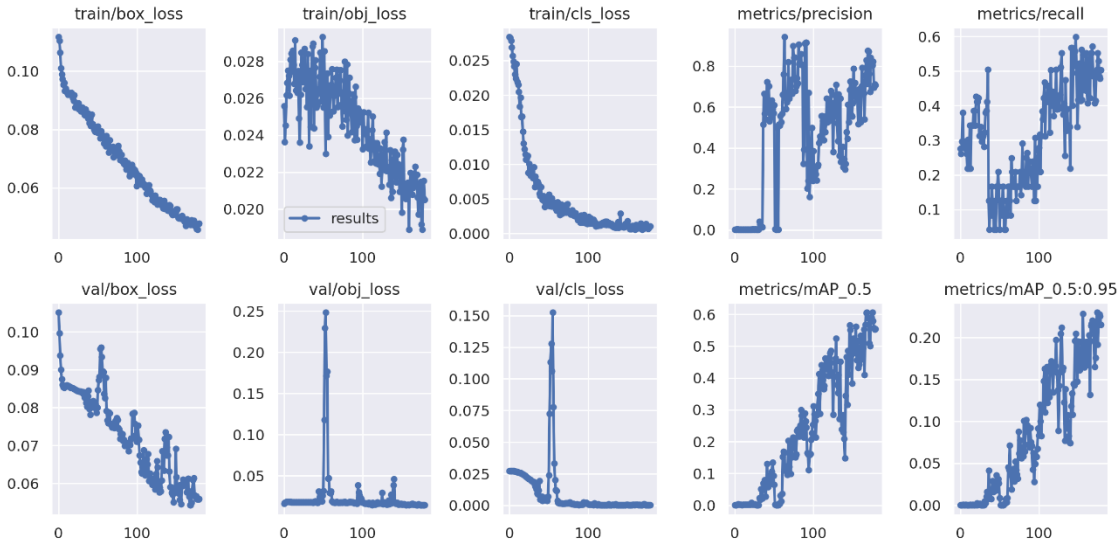
Anexo 24.- Precisión Test 4.



Anexo 25.- Recall test 4.



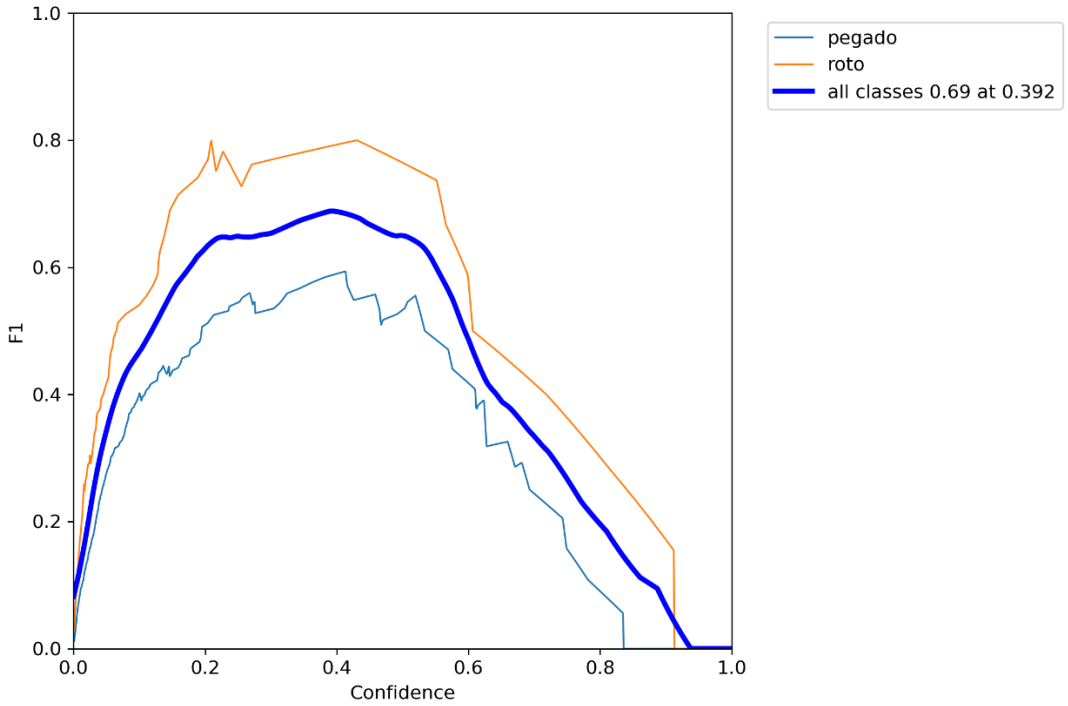
Anexo 26.- Métricas de entrenamiento Test 4.



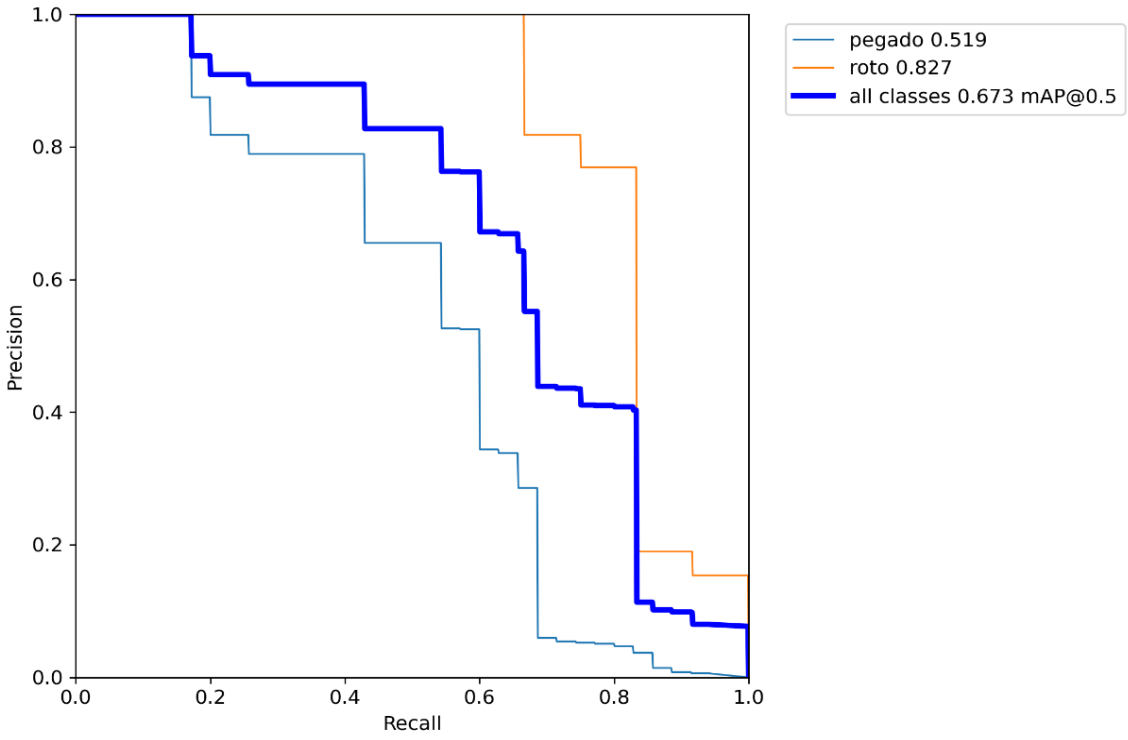
Anexo 27.- Resultado del entrenamiento de la red test 5.

```
+ Código  + Texto

     Epoch   gpu_mem       box       obj       cls    labels  img_size
 214/219     9.58G    0.0406   0.01706  0.000427        80       416: 100% 3/3 [00:01<00:00,  2.20it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.05it/s]
                  all         27         47      0.567      0.656      0.584      0.191

     Epoch   gpu_mem       box       obj       cls    labels  img_size
 215/219     9.58G   0.03905   0.01687 0.0004704        65       416: 100% 3/3 [00:01<00:00,  2.16it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.85it/s]
                  all         27         47      0.542      0.653      0.607      0.201

     Epoch   gpu_mem       box       obj       cls    labels  img_size
 216/219     9.58G   0.04026   0.01716 0.0006605        87       416: 100% 3/3 [00:01<00:00,  2.00it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.85it/s]
                  all         27         47      0.741      0.521      0.593      0.203

     Epoch   gpu_mem       box       obj       cls    labels  img_size
 217/219     9.58G   0.03947   0.01695 0.0003973        78       416: 100% 3/3 [00:01<00:00,  2.20it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.14it/s]
                  all         27         47      0.727      0.493      0.556      0.193

     Epoch   gpu_mem       box       obj       cls    labels  img_size
 218/219     9.58G   0.03968   0.01807 0.0003827        77       416: 100% 3/3 [00:01<00:00,  2.15it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.14it/s]
                  all         27         47      0.696      0.479      0.528      0.205

     Epoch   gpu_mem       box       obj       cls    labels  img_size
 219/219     9.58G      0.04    0.0164 0.0002733        68       416: 100% 3/3 [00:01<00:00,  2.19it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.07it/s]
                  all         27         47      0.736      0.475       0.52      0.204

220 epochs completed in 0.154 hours.
Optimizer stripped from runs/train/yolov5s_results2/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results2/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results2/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.34it/s]
                  all         27         47      0.808      0.605      0.673      0.233
               pegado         27         35      0.642      0.543      0.519      0.161
                 roto         27         12      0.975      0.667      0.827      0.304
Results saved to runs/train/yolov5s_results2
CPU times: user 4.6 s, sys: 486 ms, total: 5.09 s
Wall time: 9min 42s
```

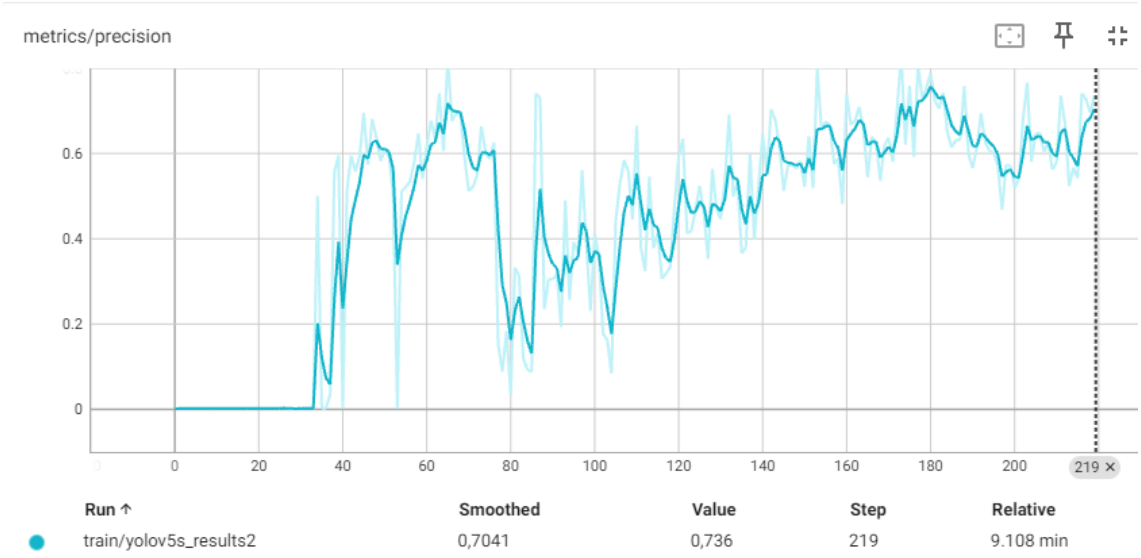
Anexo 28.- F1_Curve Test 5.



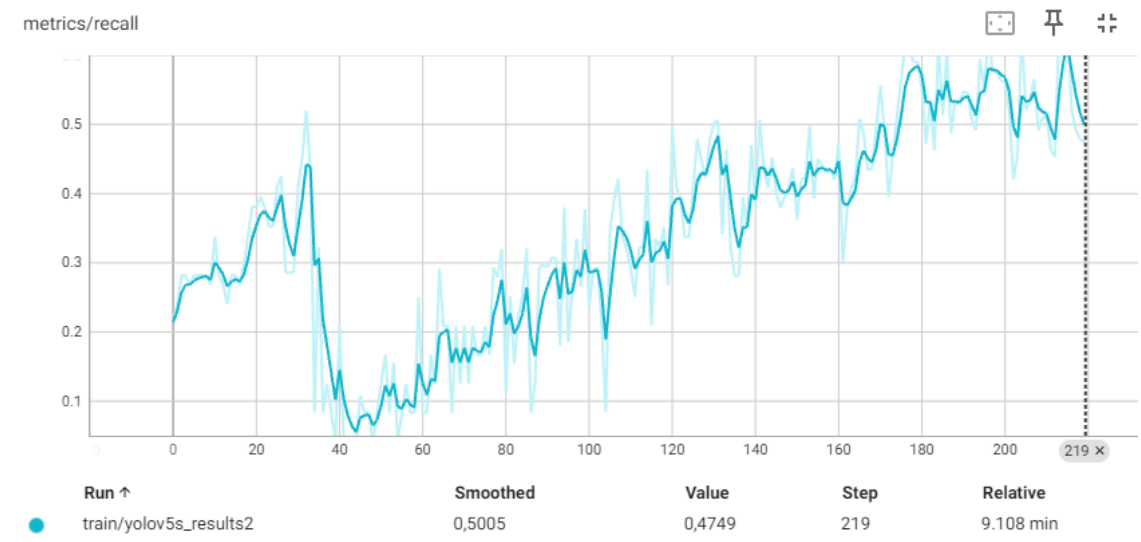
Anexo 29.- PR_Curve Test 5.



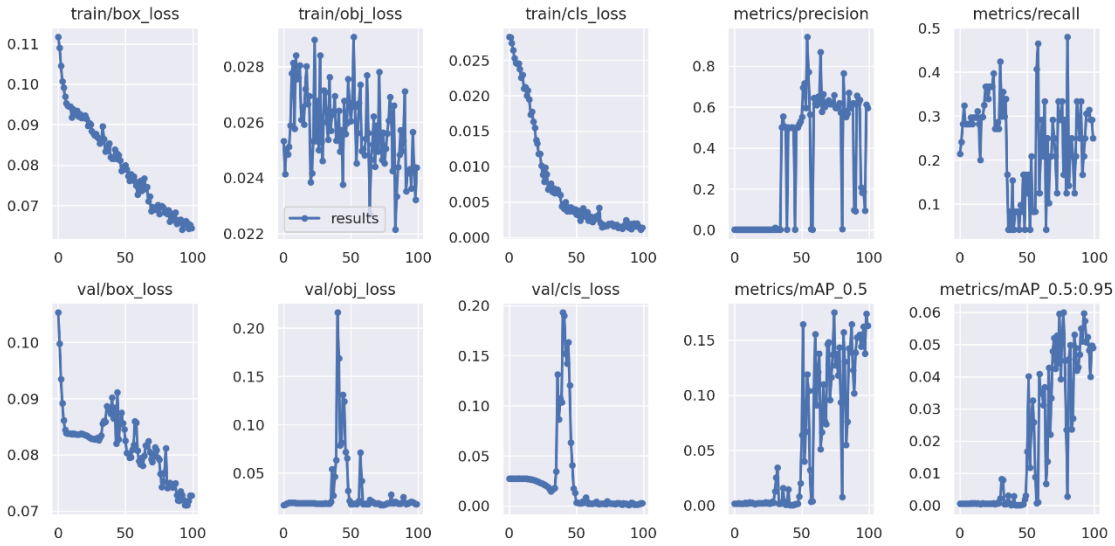
Anexo 30.- Precisión Test 5.



Anexo 31.- Recall test 5.



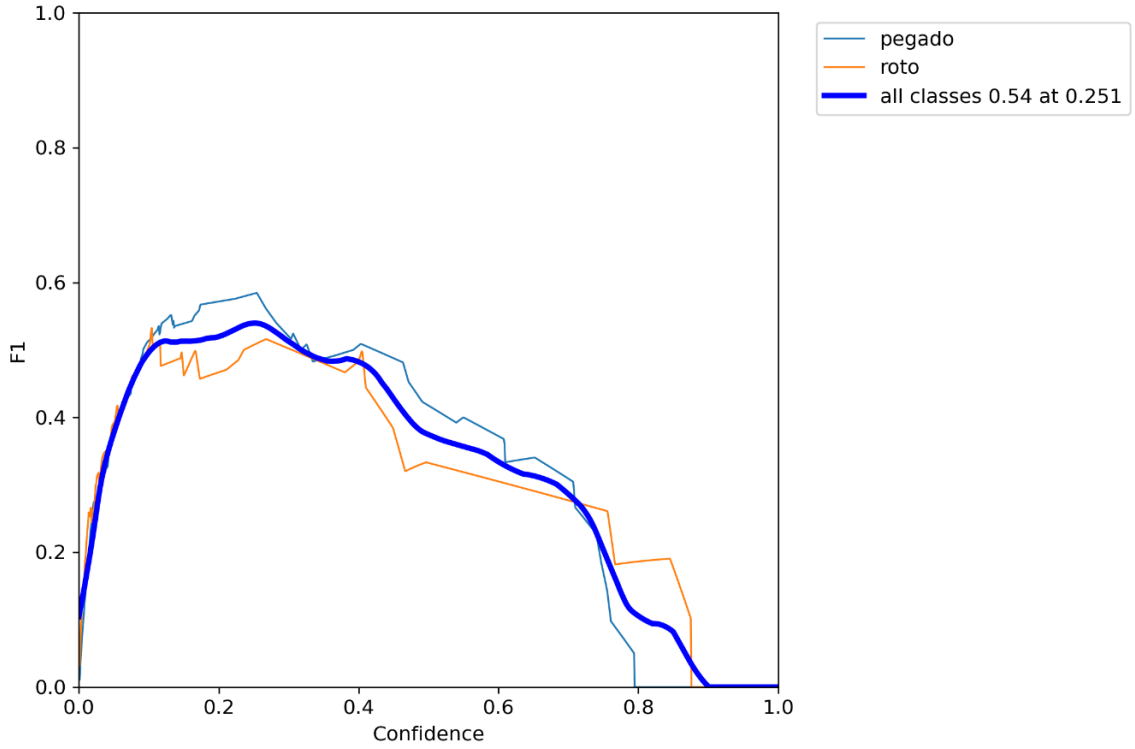
Anexo 32.- Métricas de entrenamiento Test 5.



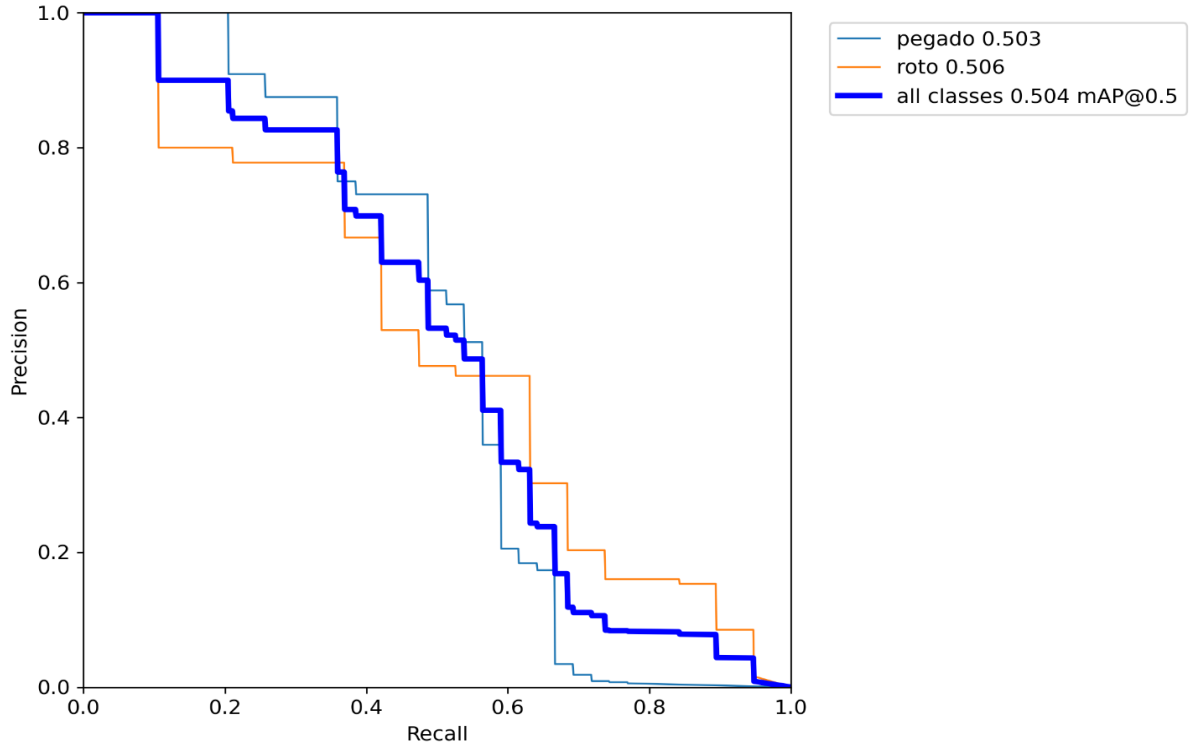
Anexo 33.- Resultado del entrenamiento de la red test 6.

```
+ Código  + Texto

✓ [44]  1 min
      215/219     9.58G   0.04397   0.02256 0.0006511        82       416: 100% 3/3 [00:01<00:00,  1.90it/s]
                  Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.70it/s]
                    all         27         58      0.623      0.454      0.484       0.19

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      216/219     9.58G   0.04375   0.02154  0.001032        78       416: 100% 3/3 [00:01<00:00,  2.10it/s]
                  Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.50it/s]
                    all         27         58      0.618      0.462      0.477      0.187

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      217/219     9.58G   0.04528   0.02163 0.0007603        82       416: 100% 3/3 [00:01<00:00,  2.06it/s]
                  Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.58it/s]
                    all         27         58      0.635      0.454      0.474      0.187

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      218/219     9.58G   0.04437   0.02262 0.0006704       102       416: 100% 3/3 [00:01<00:00,  2.04it/s]
                  Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.70it/s]
                    all         27         58      0.591      0.437      0.478      0.182

      Epoch   gpu_mem       box       obj       cls    labels  img_size
      219/219     9.58G   0.04553   0.02356  0.001783        96       416: 100% 3/3 [00:01<00:00,  1.83it/s]
                  Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.58it/s]
                    all         27         58      0.564      0.454      0.482      0.179

220 epochs completed in 0.173 hours.
Optimizer stripped from runs/train/yolov5s_results4/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results4/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results4/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
                  Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.12it/s]
                    all         27         58      0.673      0.454      0.504      0.204
                pegado          27         39       0.72      0.487      0.503      0.188
                  roto          27         19      0.626      0.421      0.506       0.22
Results saved to runs/train/yolov5s_results4
CPU times: user 5.33 s, sys: 619 ms, total: 5.95 s
Wall time: 10min 53s
```

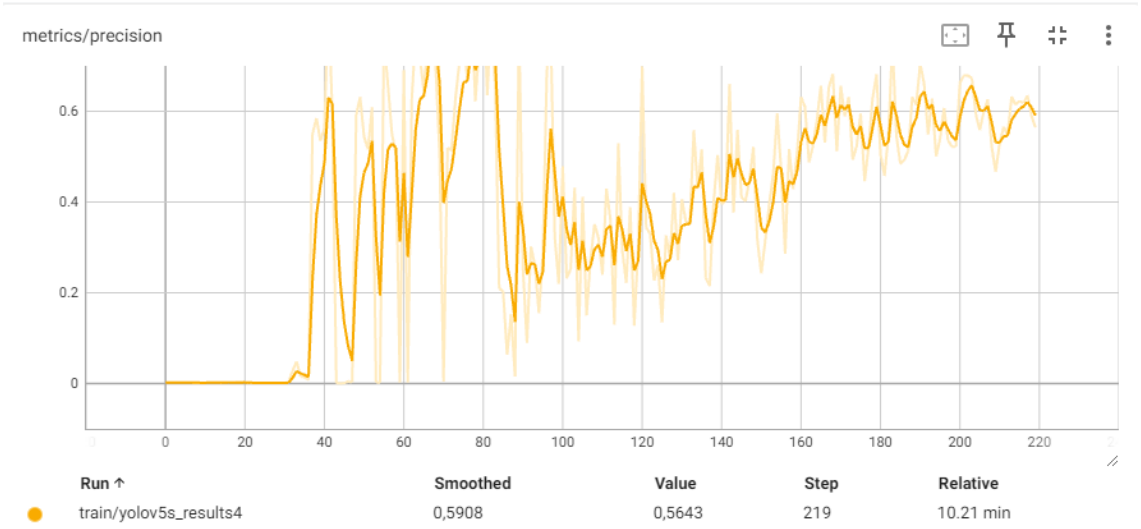
Anexo 34.- F1_Curve Test 6.



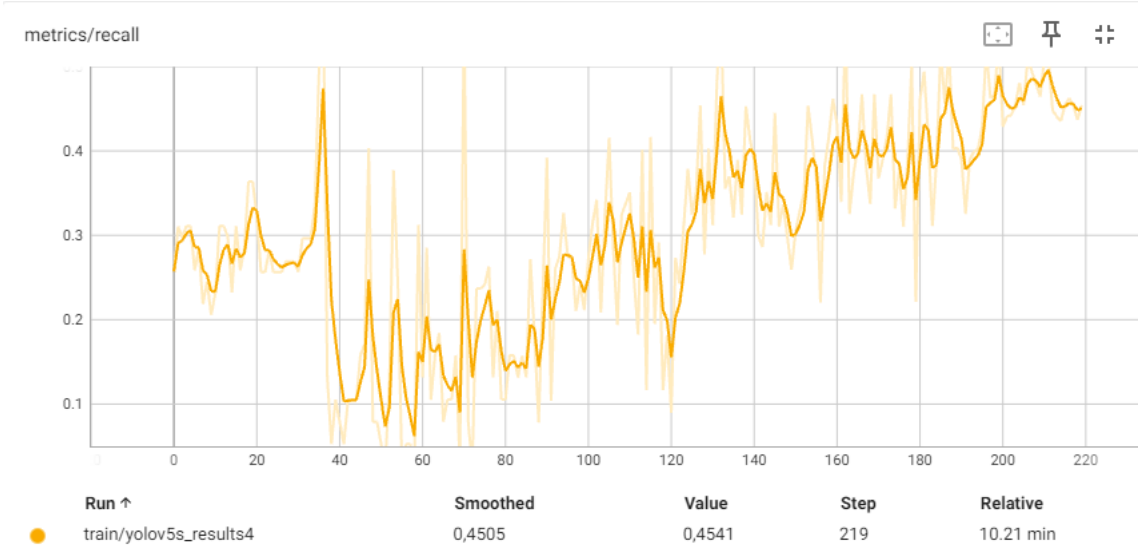
Anexo 35.- PR_Curve Test 6.



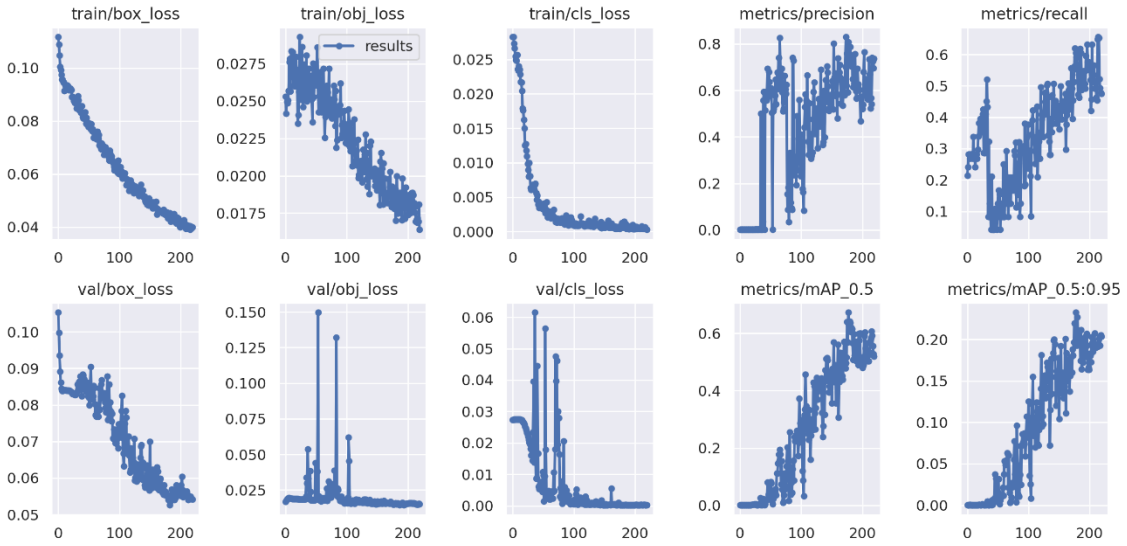
Anexo 36.- Precisión Test 6.



Anexo 37.- Recall test 6.



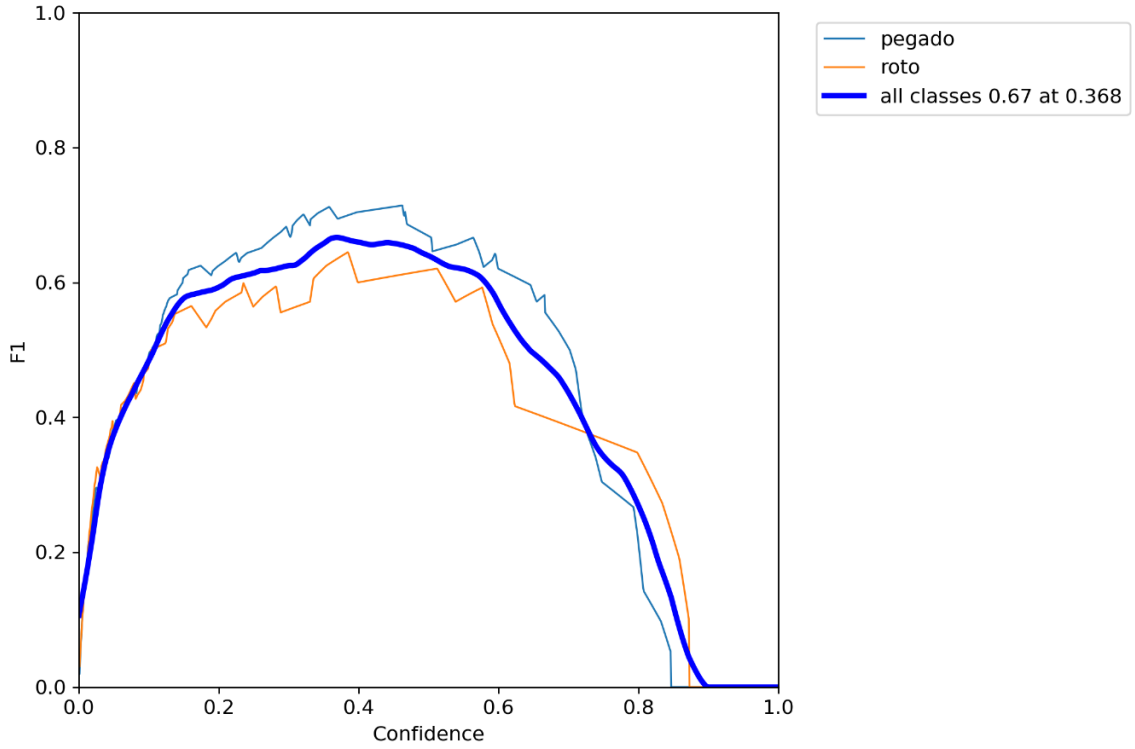
Anexo 38.- Métricas de entrenamiento Test 6.



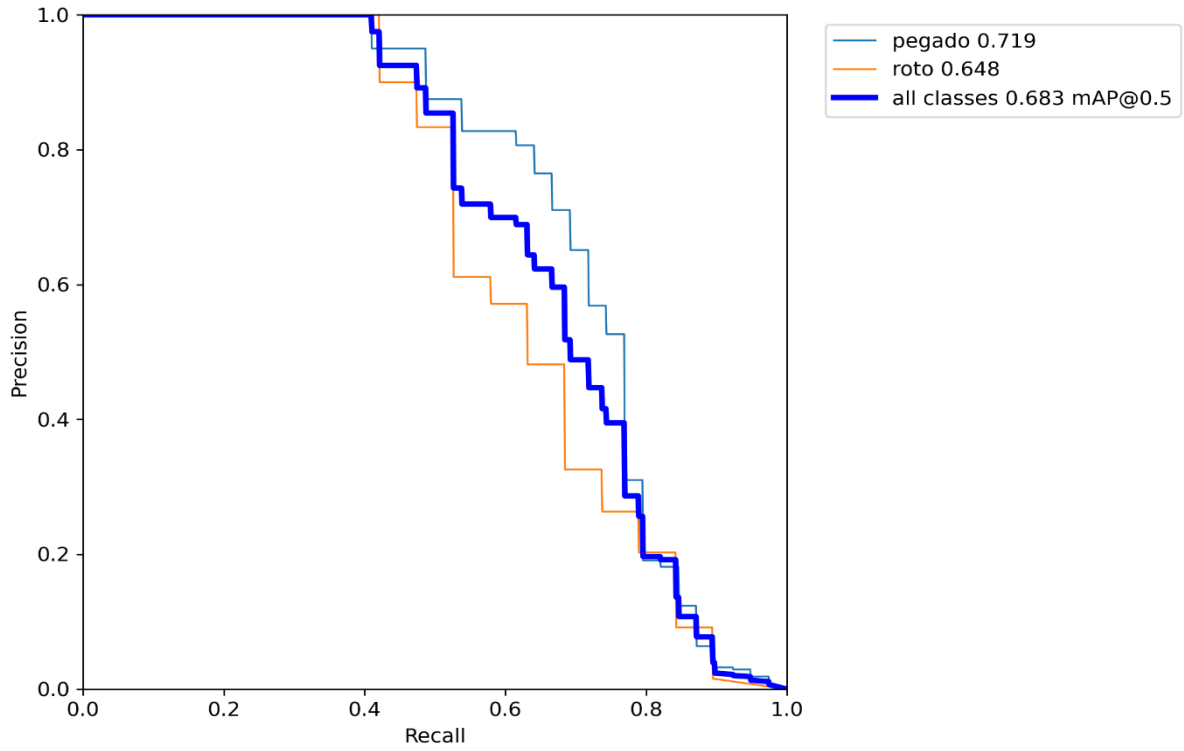
Anexo 39.- Resultado del entrenamiento de la red test 8.

```
                 all         27         58      0.877      0.545       0.67      0.254

      Epoch   gpu_mem       box       obj       cls    labels  img_size
  214/219     9.58G    0.0426   0.02173  0.001143        97       416: 100% 3/3 [00:01<00:00,  2.06it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.35it/s]
                   all         27         58      0.801      0.564      0.683      0.255

      Epoch   gpu_mem       box       obj       cls    labels  img_size
  215/219     9.58G   0.04081   0.02078 0.0005817        86       416: 100% 3/3 [00:01<00:00,  2.10it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.56it/s]
                   all         27         58      0.723      0.566      0.652      0.242

      Epoch   gpu_mem       box       obj       cls    labels  img_size
  216/219     9.58G   0.04246   0.01888 0.0005655        81       416: 100% 3/3 [00:01<00:00,  2.07it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.85it/s]
                   all         27         58      0.623      0.532      0.573      0.224

      Epoch   gpu_mem       box       obj       cls    labels  img_size
  217/219     9.58G   0.04204   0.02033 0.0005823        79       416: 100% 3/3 [00:01<00:00,  2.01it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.36it/s]
                   all         27         58      0.714      0.526       0.62      0.236

      Epoch   gpu_mem       box       obj       cls    labels  img_size
  218/219     9.58G   0.04125   0.02125 0.0004703       102       416: 100% 3/3 [00:01<00:00,  1.81it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.85it/s]
                   all         27         58      0.766      0.516      0.651      0.234

      Epoch   gpu_mem       box       obj       cls    labels  img_size
  219/219     9.58G   0.04006   0.02102 0.0008446        91       416: 100% 3/3 [00:01<00:00,  1.83it/s]
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.33it/s]
                   all         27         58      0.758      0.508      0.645      0.221

220 epochs completed in 0.178 hours.
Optimizer stripped from runs/train/yolov5s_results5/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results5/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results5/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7249215 parameters, 0 gradients, 16.7 GFLOPs
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.99it/s]
                   all         27         58      0.801      0.564      0.683      0.254
                pegado         27         39      0.779      0.641      0.719      0.256
                  roto         27         19      0.822      0.486      0.648      0.253
Results saved to runs/train/yolov5s_results5
CPU times: user 6.47 s, sys: 842 ms, total: 7.31 s
Wall time: 11min 11s
```

Anexo 40.- F1_Curve Test 8.

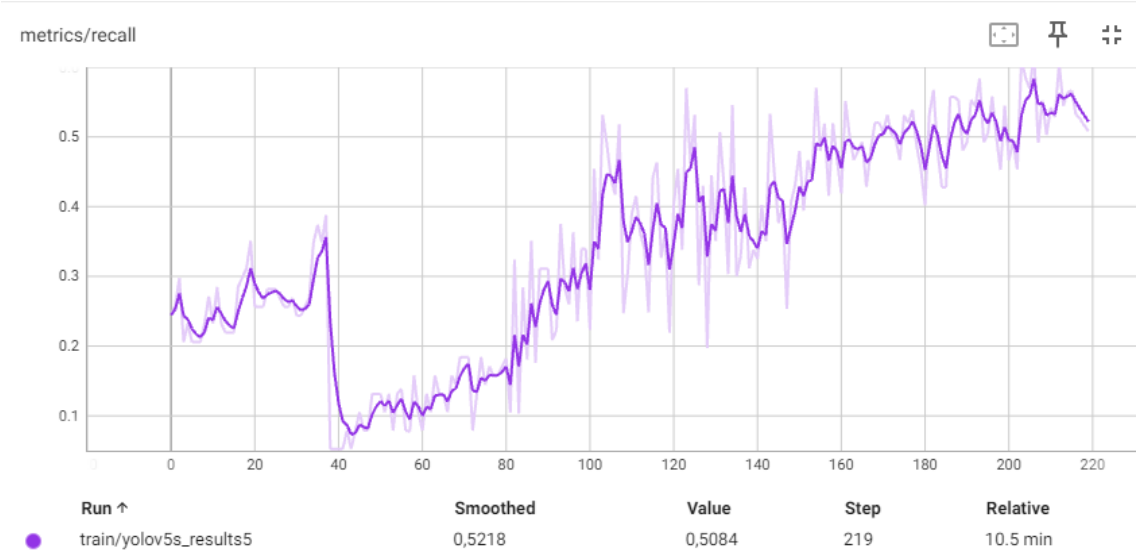


Anexo 41 .- PR_Curve Test 8.



Anexo 42.- Precisión Test 8.

Anexo 43.- Recall test 8.



Anexo 44.- Métricas de entrenamiento Test 8.