

## Implementación de una red neuronal para la detección de anomalías en bandejas

Sánchez Santalices, J.<sup>a</sup>, Moya de la Torre, Eduardo J<sup>b</sup>, Poncela Méndez Alfonso V.<sup>b\*</sup>

<sup>a</sup> Departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingenierías Industriales de la Universidad de Valladolid

<sup>b</sup> Instituto de las Tecnologías Avanzadas de la Producción, ITAP, de la Universidad de Valladolid

**To cite this article:** Sánchez Santalices, J., Moya de la Torres, E.J., Poncela Méndez, A.V. 2023.  
Implementation of a neural network for the detection of anomalies in trays.  
XLIV Jornadas de Automática, 873-878 <https://doi.org/10.17979/spudc.9788497498609.873>

### Resumen

El objetivo de este trabajo es la detección de diversos tipos de defectos que aparecen en las bandejas biodegradables durante su fabricación, más concretamente a la entrada y salida del proceso de laminado. Para ello, se emplea un sistema de visión artificial, que se va a encargar de realizar un análisis de las bandejas que vayan pasando por la línea de producción y expulsando las que considere defectuosas.

Dicho sistema de visión está compuesto por tres cámaras, dos situadas a la entrada del proceso y otra a la salida, un PLC que comanda las cámaras, un sistema neumático que expulsa las bandejas defectuosas, y un ordenador exclusivamente destinado al análisis de cada imagen tomada mediante el lanzamiento de una red neuronal.

El modelo desarrollado permite detectar anomalías de las bandejas durante la producción con una precisión, velocidad y efectividad muy elevada, produciendo un aumento muy significativo de la calidad de la producción.

*Palabras clave:* Visión artificial, red neuronal, Darknet, YOLO, Python.

### Implementation of a neural network for the detection of anomalies in trays

#### Abstract

The aim of this paper is the detection of several types of defects which appear at the biodegradable trays during its fabrication, specifically at the input and output of the laminate process. For that, it is used an artificial vision system that: analyzes the trays through the production line, selects the defectives and ejects them if necessary.

This artificial vision system is has: three cameras, two of them located at the input of the process and the other one at the output; a PLC that controls the cameras; a pneumatic system that ejects the defective trays; and a computer encharged only of analyzing every single image taken using a neuronal net.

The developed model allows to detect the anomalies of the trays during the production with a very high accuracy, speed and effectiveness, leading to a very significant increase of the quality production.

*Keywords:* Artificial vision, neuronal net, Darknet, YOLO, Python.

## 1. Introducción

Este trabajo es resultado de la participación del autor en un proyecto de la empresa PackBenefit S.L. [1], donde se realizaron las prácticas curriculares y extracurriculares, y que posteriormente dio lugar a su TFG [2] (protegido por acuerdo de confidencialidad según deseo expreso de la empresa). Está basado en un proyecto desarrollado en la empresa, la cual ha facilitado todos los medios necesarios para que pueda llevarse a cabo, otorgándole una gran importancia para seguir allanando el camino hacia convertirse en un ejemplo de las Industrias 4.0.

PackBenefit S.L es una empresa con sede en Aldeamayor de San Martín, Valladolid, perteneciente al sector de la Industria Alimentaria, más concretamente en el sector del "packaging compostable" para alimentos y tiene como objetivo principal la fabricación de unos envases sostenibles basados en una disolución de celulosa para reducir el uso de los plásticos, que suponen una gran amenaza al medio ambiente.

Estos envases sostenibles no sólo son biodegradables y reciclables, sino que también son 100% compostables, siendo PackBenefit S.L., la única empresa de este sector que cumple con estos tres pilares medioambientales de forma simultánea, lo que la convierte en una empresa vital en la lucha por conseguir cada día un mejor planeta y mejorar la vida de las personas.

El trabajo ha sido organizado como sigue. En la introducción se habla de la temática y justificación del proyecto implicado, junto con los elementos del sistema de empleados. Posteriormente, se incluye la fase de detección de anomalías con inteligencia artificial, donde se comentan las fases de etiquetado y clasificación de imágenes, diseño y entrenamiento del modelo, su evaluación y las diferentes problemáticas que aparecen. A continuación, aparece la fase de implementación del modelo en la línea de producción, los resultados prácticos de dicho sistema en funcionamiento y las conclusiones.

### 1.1. Tema y Justificación del proyecto

PackBenefit S.L. fabrica bandejas compostables que pasan por varios procesos antes de alcanzar su estado final. Es crucial verificar el estado de estas bandejas en diferentes etapas del proceso, ya que son susceptibles de tener defectos. La detección de anomalías es fundamental en el sector alimentario, evitando que las bandejas lleguen a los clientes con suciedad, objetos extraños o puntos negros, lo cual perjudicaría la imagen de la empresa.

La detección temprana de defectos también ayuda a identificar parámetros mal ajustados, permitiendo corregirlos a tiempo y evitar la producción de bandejas defectuosas. Actualmente, esta tarea es realizada por operarios cualificados, pero resulta repetitiva, lo que puede afectar la objetividad, calidad y velocidad de detección, disminuyendo la capacidad de producción y aumentando el riesgo de pasar bandejas defectuosas.

Gracias al desarrollo de la tecnología de visión artificial y los avances en inteligencia artificial, estas tareas repetitivas pueden ser realizadas por sistemas de visión. Por ello, la

empresa decidió implementar un sistema de detección basado en visión artificial, liberando a los operarios de esta tarea y permitiéndoles centrarse en otras áreas de producción.

El sistema de detección se encargará de identificar los defectos en las bandejas antes y después del proceso de laminado, que consiste en recubrirlas con un "film" compostable para aumentar su impermeabilidad y permitir el almacenamiento de líquidos. La ubicación del sistema de detección antes y después de este proceso se debe a que los defectos previos están relacionados con la disolución de celulosa de las bandejas, mientras que los defectos posteriores se deben al "film". De esta manera, el sistema puede detectar todos los posibles errores en la fabricación de las bandejas.

La implementación de este sistema de detección requiere el uso de varios componentes que formarán el sistema de visión artificial. Todos los elementos son fundamentales y críticos para el correcto funcionamiento del sistema, por lo que es importante asegurar una buena coordinación entre ellos para obtener resultados eficientes en la detección

### 1.2. Elementos del Sistema de Visión Artificial

El sistema de visión está compuesto por los siguientes elementos:

- **Tres cámaras:** Se utilizan cámaras monocromáticas para capturar imágenes de las bandejas desde diferentes ángulos. Dos cámaras se colocan en la entrada de la laminadora para examinar las dos caras de las bandejas, mientras que la tercera cámara se sitúa en la salida para capturar la parte superior de las bandejas (Figura 1).
- **Sistema neumático expulsor:** Este sistema se encarga de expulsar las bandejas defectuosas de la línea de producción. Se activa mediante una válvula que permite el paso de aire comprimido a través de tubos de plástico situados en diferentes puntos de la cinta transportadora.
- **PLC S7-1500:** Este autómata programable de Siemens controla el sistema neumático expulsor y la captura de imágenes de las cámaras. Se programa utilizando el TIA Portal [3] y se interactúa con él a través de una interfaz hombre-máquina (HMI).
- **PC para visión:** Un ordenador dedicado exclusivamente a esta aplicación se encarga de la comunicación entre las cámaras y el PLC, la ejecución de la red neuronal y el almacenamiento de los resultados. Cuenta con una potente GPU NVIDIA para acelerar la detección y está conectado a un sistema de alimentación ininterrumpida (SAI) para garantizar su funcionamiento continuo.
- **Red neuronal:** Se utiliza el framework Darknet con el modelo YOLOv4 [4] para la detección de anomalías. Esta red convolucional basada en aprendizaje profundo se ejecuta en la GPU del PC y utiliza una única CNN para localizar y etiquetar las detecciones mediante "bounding boxes" [5].



Figura 1: Cinta de salida, cámara de salida y sistema expulsor

Estos elementos se combinan para crear un sistema de visión que permite la detección automática de anomalías en las bandejas compostables durante los procesos de fabricación. La integración y coordinación de estos componentes mejoran la objetividad, calidad y velocidad en la identificación de defectos, optimizando así la capacidad de producción y la calidad del producto final.

## 2. Detección de anomalías con inteligencia artificial

En esta sección se indican y explican las fases más importantes que han ido marcando el devenir de la exactitud y precisión del modelo final de detección de anomalías.

### 2.1. Clasificación y etiquetado de imágenes

El sistema de detección tiene como objetivo identificar y desechar de forma autónoma las bandejas que presenten diferentes tipos de anomalías. Además, se pretende almacenar la información sobre los defectos detectados para realizar ajustes en los parámetros de producción y corregir los defectos en las bandejas futuras.

El departamento de calidad de la empresa distingue entre defectos primarios y secundarios. Los defectos primarios implican desechar la bandeja, mientras que los secundarios permiten aceptarla, pero se deben corregir los parámetros de producción para eliminar el defecto en futuras bandejas.

Los defectos se dividen en dos grupos: defectos de formación y defectos de laminado.

Los defectos de formación hacen referencia a las anomalías que han sido generadas durante el proceso de formación de las bandejas. Los principales defectos que aparecen en este grupo son: manchas de humedad; calvas; superficies irregulares; grietas y puntos negros.

Los defectos de laminado incluyen las anomalías generadas durante el proceso de laminado de la bandeja. Los principales defectos que se incluyen en este grupo son: arrugas;

deslaminado; ampollas; troquel descentrado y cuerpos extraños.

Una vez que se han explicado los defectos a identificar por parte del sistema de detección, es necesario realizar un etiquetado de imágenes de una forma rigurosa y precisa, para posteriormente introducir este volumen de imágenes etiquetadas (dataset) en el entrenamiento de Yolo [4].

### 2.2. Diseño y entrenamiento del modelo

Una vez que se ha generado el dataset inicial hay que indicar el porcentaje de imágenes que van a ser destinadas al entrenamiento y al test del modelo. Para el entrenamiento se ha destinado un 85%, por lo que el 15% restante serán empleadas para la validación del modelo. Además, se ha fijado el *Learning rate* a 0.005, el *momentum* a 0.949 y un número máximo de iteraciones igual a 24000.

En total fueron realizados tres entrenamientos, con el fin de obtener el mejor modelo posible. La causa principal por la que los dos primeros entrenamientos no estaban ofreciendo los resultados esperados era porque el dataset empleado contaba con cierto desequilibrio en cuanto al número de instancias por cada clase.

Esto es algo difícil de resolver, ya que el dataset cuenta con 12 clases (las anomalías que el sistema ha de detectar), lo que dificulta obtener un número considerablemente grande de instancias para cada clase, ya que hay anomalías que son mucho menos frecuentes y por lo tanto no se pueden conseguir tantas imágenes.

Por lo tanto, para conseguir un dataset equilibrado, se tuvo que reducir considerablemente el número de imágenes respecto al primer dataset, empleando finalmente aproximadamente 12000 imágenes, con un número de instancias por clase del orden 5000 con una variación de  $\pm 200$ .

En la Figura 2, se puede observar la curva de aprendizaje del modelo con un coeficiente de pérdida (*average loss*) de 0.7817, el cual se considera un buen resultado, dado que la documentación de Yolo [4] marca como frontera entre un buen y un mal entrenamiento que dicho valor se encuentre por debajo de 1.

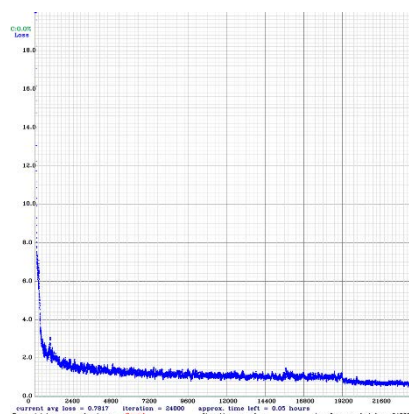


Figura 2: Curva de aprendizaje del último entrenamiento

### 2.3. Evaluación del modelo

Para determinar la calidad del modelo, este tiene que ser sometido a imágenes que no haya visto y valorar sus

detecciones. Para ello se ha desarrollado un programa basado en Python [6] con la implementación de OpenCV [7] que carga los pesos del modelo entrenado e infiere las detecciones que realice sobre las imágenes que procesa. Este programa se ejecuta sobre la GPU del computador del sistema de visión, permitiendo así optimizar el rendimiento del programa y obtener los resultados más rápido.

En la Figura 3 se muestra una imagen procesada por el modelo en la que se reflejan las detecciones con su porcentaje de confianza. Cabe destacar que sólo se tienen en cuenta las detecciones del modelo que superan un determinado umbral de confianza. En la Figura 3, se fijó el valor de este umbral a 0 para observar todo el número potenciales anomalías que identifica el modelo. Las anomalías que tengan un tamaño muy pequeño por lo general serán identificadas con un nivel de confianza bajo (inferior al 50%).

Esto no supone un problema grave, ya que el objetivo principal de este sistema es que identifique y clasifique los defectos primarios, que son los que provocan la expulsión de la bandeja, y por lo general, estos defectos son de mayor tamaño.

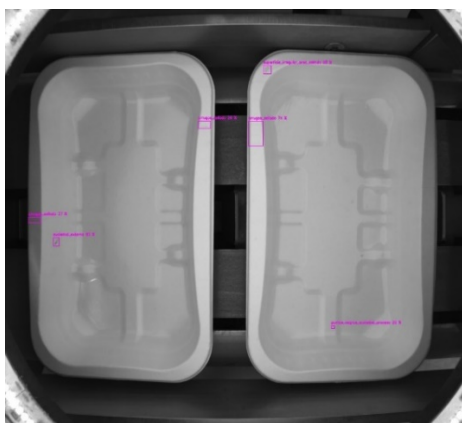


Figura 3: Imagen procesada por el modelo entrenado

Durante la prueba de evaluación, el sistema procesó 504 imágenes con una media de tiempo de procesamiento por imagen de 9,72 ms. El conseguir un tiempo de detección tan bajo permite que el sistema de detección pueda ser implementado en la línea de producción, ya que va a ser capaz de trabajar en tiempo real sin retrasar el avance de las bandejas.

#### 2.4. Problemáticas

Antes de realizar la implementación del sistema de detección sobre la línea de producción hay que tener en cuenta algunos de los problemas que se han detectado durante el proceso de desarrollo de dicho sistema y cómo se han solucionado.

##### 2.4.1. Lentitud del proceso previo de etiquetado

Esta parte requiere mucho tiempo si se quiere introducir un volumen grande de imágenes al entrenamiento, lo cual es aconsejable si se quieren obtener mejores resultados. Por lo que, para realizar próximos entrenamientos y seguir mejorando los resultados del modelo, sería necesario que una persona

estuviese etiquetando de forma manual durante un largo período de tiempo, lo cual resulta improductivo.

La solución adoptada fue desarrollar un programa de “auto etiquetado” que se encargara de realizar esta parte de forma automática y que el usuario tan sólo debiera supervisar las etiquetas y corregirlas si es necesario.

##### 2.4.2. Análisis de bandejas en diferentes formatos

En la línea de producción se tiene previsto que el sistema de detección sea capaz de identificar defectos en dos formatos diferentes de los que se fabrican. Estos formatos son los denominados 2315, con el cual aparecen dos bandejas por cada imagen tomada, y 2518, con el que aparece una bandeja en cada imagen.

Estos formatos de bandeja son bastante similares, aunque tienen diferencias especialmente en sus dimensiones, lo cual puede provocar confusiones en el sistema a la hora de la detección. Pero el principal inconveniente es el hecho de que en un caso haya dos bandejas por cada imagen y en otra una.

Esto se tuvo en cuenta para el entrenamiento, y las imágenes de entreno empleadas contenían ambos formatos, ya que si solo se alimenta el entrenamiento con un formato cuando se introduzca en el sistema una imagen con el otro formato de bandeja las detecciones que realice serán muy poco precisas.

### 3. Integración del modelo en la línea de producción

Una vez que el sistema de detección diseñado ha sido probado de forma aislada y se han dado como válidos los resultados de sus detecciones, puede comenzarse la integración del modelo en la línea de producción.

Cabe destacar que el uso o no del sistema de detección podrá ser activado o desactivado cuando se desee por medio del HMI (*Human Machine Interface*) de la laminadora de la fábrica (máquina en la que se implementa el sistema), ya que en un principio se utilizará para llevar a cabo diferentes pruebas de su funcionamiento, pero no estará constantemente activado hasta que se contraste su efectividad.

Para esta sección del trabajo fue necesario desarrollar programa muy extenso en el que se gestionan todos los elementos que componen el sistema de visión artificial. En este apartado se van a mencionar los principales módulos de este programa y una breve descripción.

En la Figura 4 se puede observar un layout de la conexión entre los elementos del sistema de visión.

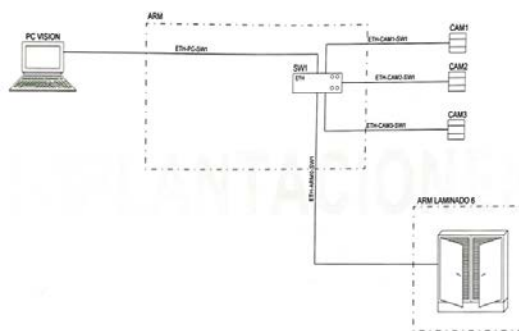


Figura 4: Layout conexión elementos sistema de visión

### 3.1. Conexión al PLC y cámaras de la máquina

El sistema de visión artificial utiliza el PLC de la laminadora para comandar el funcionamiento de la máquina y gestionar las órdenes de captura de imágenes. La conexión entre el PLC y el programa de Python se realiza utilizando la librería Snap7 [8].

Para establecer la conexión con el PLC desde Python, se crea una clase que actúa como cliente de comunicación con el PLC, que funciona como servidor. Se proporciona la dirección IP del PLC para establecer la conexión.

El ordenador que ejecuta el programa está conectado al PLC de la laminadora y al controlador de las cámaras a través de un switch industrial de Siemens, utilizando direcciones IP correspondientes para cada dispositivo. Esto permite la comunicación entre todos los elementos.

Las cámaras son controladas mediante la API Spinnaker SDK FLIR [9], y se utiliza la librería PySpin [10] para interactuar con ella desde Python. Se inicializa el sistema de las cámaras y se obtiene el listado de cámaras detectadas. Si no se detecta ninguna cámara, el sistema se cierra y se interrumpe el programa. En caso de que se detecten cámaras, se generan las variables necesarias para cada una de ellas, permitiendo su posterior uso.

### 3.2. Comunicación con el PLC

Se establece una conexión utilizando la librería Snap7 [8] y se proporciona la dirección IP del PLC. Se crea una clase cliente en Python para interactuar con el PLC como servidor. Se utiliza una función para leer información específica de un bloque de datos (DB) en el PLC. Se lee el DB de comunicación (DB 124) para obtener las órdenes del PLC, como capturar imágenes o actualizar parámetros.

Si se recibe la orden de actualizar parámetros, se leen los índices de confianza y áreas para calcular los umbrales. Se inicia un bucle de comunicación con el PLC para verificar constantemente las órdenes. Cuando se recibe una orden de captura de imágenes, se toman las imágenes y se procesan. Los resultados de la detección se escriben en el DB de comunicación, indicando si la bandeja es buena o mala y el tipo de defecto.

El PLC utiliza estos resultados para tomar decisiones, como la expulsión de la bandeja. El bucle de comunicación continúa hasta que se detenga el programa o se produzca un error. En resumen, la comunicación con el PLC implica establecer la conexión, leer y escribir datos en el DB de comunicación y realizar acciones en función de las órdenes recibidas del PLC.

### 3.3. Lanzamiento de imágenes y envío a Yolo

Después de leer las órdenes y parámetros del PLC, se verifica si alguna cámara solicita la captura de imágenes. Si es así, se crea un hilo para cada cámara y se lanza una función que se encarga de conectarse a las cámaras implementadas y capturar las imágenes, para posteriormente pasarlas al modelo Yolo para realizar detecciones.

En esta función se establece el modo de adquisición de las cámaras como continuo. Luego, se captura un número definido

de imágenes (por ejemplo, 3 por bandeja). Cada imagen se transforma al formato compatible con Yolo y se envía a la red para obtener las detecciones correspondientes.

Una vez que las imágenes son procesadas por Yolo, se reciben los parámetros de las *bounding boxes*, las confianzas de los defectos y los defectos detectados. Estos resultados se acumulan ya que se envían 3 imágenes por bandeja. Luego, se pasa al siguiente punto.

Si el formato de las bandejas es 2315 (2 bandejas por foto), se extraen las coordenadas "x" que separan el borde de cada bandeja. Se utiliza la función de OpenCV basada en la técnica de "Non Max Supression" con las detecciones de las 3 imágenes para eliminar las detecciones cercanas y quedarse con la de mayor confianza.

A continuación, se realiza un bucle para guardar los datos relevantes de las detecciones, como las áreas de los defectos, la clase de los defectos, la coordenada "x" y el tono del defecto. Para extraer el tono del defecto, se emplea una función que calcula el valor medio de los píxeles del defecto. Estos datos se guardan en un vector global del programa.

En resumen, se lanzan hilos para capturar imágenes de las cámaras, se procesan las imágenes utilizando Yolo para obtener las detecciones de defectos, se acumulan los resultados de las detecciones de las imágenes y se extraen los datos relevantes para su posterior análisis.

### 3.4. Calificación de las detecciones

Al igual que para la toma de imágenes, la calificación de los resultados recibidos del modelo y ya filtrados como se acaba de explicar, se ejecuta a través del lanzamiento de hilos, uno para los resultados de las imágenes tomadas en la salida y otros 2 para las imágenes tomadas desde la entrada.

En la función creada para la calificación de imágenes se analiza cada una de las detecciones finales obtenidas y en función del defecto que sea se valoran unas características u otras.

Esta calificación varía también dependiendo del formato, ya que si es 2315 (2 bandejas por imagen) hay que ver en que bandeja se encuentra el defecto, para ello se compara con la coordenada "x" intermedia que ya se explicó cómo se obtenía anteriormente.

### 3.5. Registro de los resultados

Para que sea posible analizar cada una de las detecciones que se van realizando a medida que se van tomando imágenes durante el paso de bandejas por la línea de producción y comprobar si la bandeja se ha expulsado o no, se van generando ficheros de texto (para el caso de la salida, para la entrada no se ha llegado a este punto).

## 4. Resultados prácticos del sistema en funcionamiento

El estudio comparativo entre la producción con y sin la activación del sistema de detección diseñado arroja los siguientes resultados:

**Primera jornada de test de control (sistema desactivado):**

- Producción OK: 8140 bandejas en buen estado.
- Scrap Previo: 1385 bandejas en mal estado que llegaron a la entrada de la laminadora.
- **Scrap Posterior:** 860 bandejas en mal estado que salieron de la laminadora.
- Scrap Total: 2245 bandejas en mal estado durante la producción.
- % Scrap: 18.96% del total de producción.
- Producción Total: 10385 bandejas.

#### Segunda jornada de test de control (sistema activado):

- Producción OK: 8360 bandejas en buen estado.
- Scrap Previo: 1218 bandejas en mal estado que llegaron a la entrada de la laminadora.
- **Scrap Posterior:** 1350 bandejas en mal estado que salieron de la laminadora.
- Scrap Total: 2568 bandejas en mal estado durante la producción.
- % Scrap: 23.50% del total de producción.
- Producción Total: 10928 bandejas.

El parámetro más relevante es el **% Scrap posterior**, que muestra el porcentaje de bandejas defectuosas que salen de la laminadora. El día en que se activó el sistema de detección, este valor aumentó en un 4.07% en comparación con el día sin el sistema activado. Sin embargo, es importante tener en cuenta que el operario todavía realizaba la extracción manual de bandejas defectuosas en la entrada de la laminadora, lo cual no fue detectado por el sistema de detección diseñado.

Es fundamental destacar que todas las bandejas, tanto las consideradas buenas como las defectuosas, deben pasar por un exhaustivo control de calidad antes de llegar al cliente. Por lo tanto, el aumento en el % Scrap posterior no se considera una pérdida de producción, ya que las bandejas detectadas como defectuosas por el sistema de detección realmente requerían ser descartadas.

El departamento de calidad analizó las **1350 bandejas consideradas defectuosas por el sistema de detección** y determinó que **1286** de ellas eran **realmente defectuosas**, lo cual indica una tasa de **acierto del 95.25%** en la expulsión de bandejas defectuosas. Las 64 bandejas restantes fueron consideradas erróneamente como defectuosas por el sistema (falsos positivos).

**La pérdida de producción debido al uso del sistema de detección se estima en un 0.58%**, calculada como el porcentaje de las 64 bandejas erróneamente expulsadas respecto al total de producción. Esta pérdida es insignificante en comparación con las ventajas que aporta la implementación del sistema de detección, como permitir que el operario se enfoque en otras tareas, aumentar la precisión en la detección de defectos y agilizar los controles de calidad posteriores.

En resumen, la implementación del sistema de detección de defectos en la línea de producción ha mostrado beneficios significativos. Aunque se observó un aumento en el % Scrap posterior, este incremento se debe a la detección más precisa de bandejas defectuosas que antes pasaban desapercibidas. La pérdida de producción atribuible al sistema de detección fue mínima (0.58%), lo que demuestra que los beneficios en términos de eficiencia y calidad superan ampliamente esta

pérdida. Además, el sistema permite al operario centrarse en otras tareas y reduce la carga de trabajo en los controles de calidad posteriores.

## 5. Conclusiones

El objetivo principal del trabajo aquí mostrado era desarrollar un sistema de *detección de anomalías en las bandejas*, para mejorar la eficiencia y calidad de la producción en la máquina.

Se llevaron a cabo las siguientes tareas:

- Etiquetado y clasificación de imágenes para obtener un conjunto de datos de entrenamiento adecuado.
- Diseño y entramiento un modelo *Yolov4* para la detección de defectos en las bandejas, logrando un entrenamiento válido con buenos resultados.
- Interacción con el automatista de la fábrica para la integración del sistema de detección desarrollado en la línea de producción de la laminadora.
- Desarrolló un programa en Python que comunica con el PLC de la máquina y las cámaras para analizar las imágenes de las bandejas.

*Se logró desarrollar e implementar con éxito un sistema de detección de anomalías en las bandejas.* El sistema permitió identificar y rechazar bandejas defectuosas, sin afectar significativamente la producción, lo que permitió al operario centrarse en otros aspectos para obtener bandejas de mayor calidad.

## Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo y facilidades mostradas por parte de la empresa PackBenefit, así como el tutor de la empresa asignado al autor. El autor agradece así mismo al ITAP UVa y a los tutores asignados el apoyo dado para la culminación del trabajo.

## Referencias

- [1] Pack Benefit, «Pack Benefit Web site,» 2018. [En línea]. Available: <https://packbenefit.eco/>. [Último acceso: 15 mayo 2023].
- [2] Escuela de Ingenierías Industriales, UVa, «EII UVa Titulaciones», Available: Escuela de Ingenierías Industriales - UVa [Último acceso: 28 mayo 2023].
- [3] Siemens AG, «Soporte Industrial Siemens,» 2009. [En línea]. Available: <https://support.industry.siemens.com/cs/document/109752566/simatic-step-7-and-wince-v15-trial-download-?dti=0&dl=en&lc=es-CL>. [Último acceso: 15 mayo 2023].
- [4] J. C. Redmon, «YOLO: Real-Time Object Detection,» 2018. [En línea]. Available: <https://pjreddie.com/darknet/yolo/>. [Último acceso: 7 mayo 2023].
- [5] S. D. R. G. A. F. Joseph Redmon, «You Only Look Once: Unified, Real-Time Object Detection,» de CVPR, 2016.
- [6] Python Software Foundation, «Python,» 2001. [En línea]. Available: <https://www.python.org/downloads/>. [Último acceso: 28 mayo 2023].
- [7] OpenCV team, «Open CV,» 2019. [En línea]. Available: <https://opencv.org/>. [Último acceso: 28 mayo 2023].
- [8] S. P. Gijss Molenaar, «python-snap7 Documentation,» 2017.
- [9] Teledyne FLIR LLC, «Spinnaker SDK | Teledyne FLIR,» Teledyne FLIR LLC, 2021. [En línea]. Available: <https://www.flir.es/products/spinnaker-sdk/?vertical=machine+vision&segment=iis>. [Último acceso: 7 mayo 2023].
- [10] Python Software Foundation, «PySpin,» 2017. [En línea]. Available: <https://pypi.org/project/pyspin/> [Último acceso: 28 mayo 2023]