

# Control de calidad de amortiguadores basado en inteligencia artificial y vision por computadora\*

Redolfi, Javier<sup>1,2</sup>, González Dondo, Diego<sup>2</sup>, and Araguás, R. Gastón<sup>2</sup>

<sup>1</sup> Facultad Regional San Francisco, UTN, San Francisco, Argentina

<sup>2</sup> Facultad Regional Córdoba, UTN, Córdoba, Argentina

**Resumen** El avance continuo y sostenido de los métodos basados en aprendizaje profundo están permitiendo realizar detecciones cada vez más complejas y específicas en el ámbito de la visión de máquina, o visión por computadoras aplicado a procesos industriales. Una situación muy común en industrias en las cuales se implementan sistemas de producción con intervención de operarios sometidos a tareas repetitivas durante largos períodos de tiempo es la aparición de fallas o faltantes en los productos, principalmente debido a distracciones o cansancio del personal. Muchas veces se intenta eliminar estas fallas agregando operarios que realicen las tareas otra vez tediosas y repetitivas de control de calidad pieza por pieza. En este trabajo se presenta el diseño, la implementación y la puesta a punto en planta de un sistema automático de detección de fallas en amortiguadores mediante el uso de cámaras convencionales y algoritmos de visión por computadora y aprendizaje profundo, con el que se logró detectar una falla de altísimo impacto en la producción de amortiguadores. El sistema permite relajar el control de calidad a cargo de un operario, inspeccionando en forma continua cada amortiguador producido y generando una alarma al detectar la falla. Actualmente el sistema se encuentra en funcionamiento en la línea de producción de una planta de la región con resultados altamente satisfactorios.

**Keywords:** Detección de Objetos · Control de Calidad · Visión por Computadora · Deep Learning.

## 1. Introducción

El estado actual de los algoritmos de aprendizaje profundo basados principalmente en redes neuronales está dando lugar a desarrollos de inspección de componentes industriales mediante la visión por computadoras de muy alta complejidad en comparación con los abordados por las técnicas clásicas de procesamiento de imágenes. El aumento de las prestaciones de los equipos informáticos necesarios para realizar los entrenamientos de las redes neuronales y el fácil acceso a cámaras de video de bajo costo y alta resolución permiten hoy dar soluciones a la industria de mucha confiabilidad y rápido recupero de la inversión.

---

\* Universidad Tecnológica Nacional, FRSFco y FRC.



Fig. 1: Detección automática de faltante de bujes en línea de producción de amortiguadores.

La incorporación de procesos innovativos al área productiva implica desarrollo e investigación, tiempos de puesta a punto y un arduo trabajo en planta para calibrar las distintas variables que distan su comportamiento real al de la simulación establecida en un software. Desde hace ya algunos años muchos autores [7, 11, 15, 22] hablan de la cuarta revolución industrial, también conocida como Industria 4.0 la cual consiste en una nueva manera de producir mediante la adopción de tecnologías 4.0. Dentro de estas tecnologías podemos nombrar a la simulación, la fabricación aditiva, los sistemas de integración horizontal y vertical, la ciberseguridad, la realidad aumentada, el cómputo en la nube, los robots autónomos, el internet industrial de las cosas, el big data, los dispositivos móviles, la inteligencia artificial y las tecnologías de la información, es decir, soluciones enfocadas en la interconectividad, la automatización y los datos en tiempo real [5].

Como uno de los habilitadores tecnológicos de esta nueva industria, nombramos a la inteligencia artificial (IA, por sus siglas) la cual intenta simular la inteligencia humana a través de algoritmos computacionales [22]. Una de las ramas de la IA es la visión por computadora (CV, por sus siglas en inglés), la cual se aplica a distintos dominios, con el propósito, entre otros, de identificar objetos en imágenes o videos para poder clasificarlos.

La CV tiene cada vez más aplicaciones en la industria, las cuales permiten automatizar tareas repetitivas, tediosas o peligrosas como por ejemplo control de calidad, manejo de inventario, ordenado de piezas, líneas de ensamble, seguridad de los operarios, etc. [1, 8, 14].

En la industria, el uso de cintas transportadoras, es un factor común utilizado en distintos procesos, organizando y agilizando los mismos; acoplar algoritmos basados en CV a estas cintas, permite identificar la posición de los productos que la misma transporta [3, 16, 21, 23–25, 27], su estructura [10], detectar fallas [19], analizar su distribución [25] e informar lo percibido a sistemas integrados que utilicen esta información para la toma de decisiones.

Las soluciones comerciales que ofrece el mercado (cámaras inteligentes) son capaces de procesar imágenes mediante métodos clásicos como detectores de borde, binarizados, mapas de color, etc, sin embargo la inclusión de algoritmos de IA en estos equipos pequeños es aún escasa y de muy alto costo. Esto se debe principalmente a la potencia de cálculo que los algoritmos requieren, pero

también a las técnicas de procesamiento que deben aplicarse para lograr resultados satisfactorios, las cuales exigen capturar, almacenar y etiquetar grandes cantidades de imágenes de muestras para realizar los entrenamientos.

En este trabajo se presenta el proceso de diseño, implementación y puesta a punto en planta de un sistema de detección de fallas en amortiguadores mediante el uso de cámaras convencionales. Dicho sistema fue solicitado por la empresa ZF Argentina S.A. y actualmente se encuentra en funcionamiento en la planta de producción de la Ciudad de San Francisco, provincia de Córdoba. ZF es una empresa multinacional con base en Alemania que suministra sistemas para vehículos de pasajeros, vehículos comerciales y tecnología industrial en cuatro dominios tecnológicos: control de movimiento de vehículos, seguridad integrada, conducción automatizada y movilidad eléctrica <sup>1</sup>. En la planta de Argentina, ZF se encarga de la producción de amortiguadores para terminales automotrices y para recambio en general.

El sistema implementado se encarga de detectar una falla muy común en la producción, la cual consiste en la falta de colocación de un buje en uno o ambos extremos del amortiguador. Esta falla fue la principal causa de reclamos de clientes que tuvo la empresa en el período de producción 2019-2020. En la Fig. 1 se muestran algunos ejemplos de la salida del sistema de detección implementado y funcionando en la línea de producción.

Como hipótesis se plantea que a través de técnicas de IA, CV y detección de objetos basadas en Aprendizaje Profundo (DL, por sus siglas en inglés) es posible detectar el faltante de buje en amortiguadores a través de imágenes para poder generar una alarma y alertar a los operarios para corregir la falla en la producción. En particular, se investiga la capacidad de un detector de objetos genérico para detectar faltante de partes en un objeto y su óptimo funcionamiento como eslabón en un sistema de producción en serie. Los detectores de objetos, normalmente pueden discriminar con mucha exactitud entre clases con mucha diferencia visual, pero está poco estudiada la capacidad de los mismos para detectar pequeños cambios o variaciones en los objetos a detectar. Este último problema normalmente se conoce como clasificación de grano fino [29], en donde la clasificación se realiza entre clases visualmente muy similares visualmente.

Además, si bien estos detectores de objetos son muy usados en ambientes urbanos para realizar tareas como detección de autos [20], patentes, peatones [12], señales de tránsito [4], evasión de obstáculos, uso de protectores-buconasales [6], detección de frutas [26] o conteo de objetos, entre otros; su uso en ambientes industriales está poco explorado y bastante menos como sistema para detección de fallas en la producción.

## 2. Descripción del Problema

En muchos procesos industriales en los cuales se producen piezas en serie, una etapa importante de dicha producción es la inspección visual de las piezas producidas. Esta inspección visual permite controlar varios aspectos de los productos

<sup>1</sup> <https://www.zf.com>

terminados, como por ejemplo correcto armado, estado de la pintura, defectos de ensamblaje, faltante de piezas, entre otros. Pero esta tarea normalmente es realizada por operarios que deben controlar cientos o miles de piezas por hora, convirtiendo dicha actividad en algo muy tedioso y propenso a errores causados por distracciones y por la propia fatiga que genera. Estos errores pueden significar que un lote producido sea devuelto o rechazado por los compradores para su correcto control debido a la detección de la falla en una pieza.

En la tabla 1 se muestra un reporte interno de fallas detectadas previas al control de calidad visual, en un período de menos de un mes, de las cuales alrededor de un 0,25 % no fueron detectadas por estos controles y resultaron en reclamos por parte de los clientes.

Tabla 1: Modos de fallas detectadas y reportadas por el área de calidad de la planta durante el proceso de verificación. Desde el 14/02/2020 hasta el 20/03/2020.

Defecto	Cantidad
Sin buje	406
Buje desplazado	267
Defecto de pintura	157
Guardapolvo golpeado	26
Sin buje metálico	23
Buje incorrecto	22
Guardapolvo mal soldado	20
Sticker despintado	15
Buje dañado	5
Sin sticker	4
Sticker mal pegado	2
Sticker dañado	2
Doble platillo	1
Sticker mal impreso	1

Para este caso, el armado del amortiguador consiste en una serie de procesos realizados manualmente por los operarios, como por ejemplo la colocación de un buje en la cadena de producción. Como se puede apreciar en la tabla 1, la producción de amortiguadores sin buje es la falla más frecuente de todas. La realización de este control involucra a un operario por cada línea de ensamble, que realiza una marca visible con una fibra en cada buje colocado con el fin de evitar pasar por alto la detección de la falla. A pesar de esto los faltantes de buje que se indican en la tabla 1 son aquellos que fueron detectados, sin embargo existe todavía un porcentaje de fallas que pasan inadvertidas por este minucioso y tedioso control humano y generan aún reclamos por parte de los clientes.

En este trabajo se presenta el diseño, implementación y puesta a punto en planta de un sistema de control de calidad el cual se encarga de observar en punta de línea los amortiguadores producidos y generar una alarma visual y sonora para

advertir a los operarios sobre el faltante de buje en alguno de los amortiguadores que están pasando sobre la línea transportadora. Este sistema está compuesto por una PC, un sistema de cámaras y un software que implementa un algoritmo de detección de fallas basado en IA y el control de los dispositivos de alarma.

### 3. Solución Propuesta

#### 3.1. Algoritmo de detección

Si bien existen muchos detectores de objetos en la literatura, los que mejores resultados dan necesitan de un gran poder de cómputo para lograr trabajar en tiempo real, lo cual es un problema si se pretende implementar sistemas con muchas cámaras distribuidas en diferentes ambientes. En base al análisis realizado en el trabajo [9] en donde se concluye que detectores de objetos genéricos como SSD [13] o Faster-RCNN [18] dan muy buenos resultados en la detección de caras, al análisis presentado en [17] donde se comparan diferentes redes neuronales convolucionales para la detección de elementos de protección personal (EPPs) en ambientes industriales y al trabajo [28] en donde se utiliza una red neuronal para detectar paquetes en movimiento sobre un cinta transportadora, es que para este desarrollo se decidió utilizar el detector YOLOv4 [2].

El algoritmo YOLO (You Only Look Once, en inglés) es un sistema de detección de objetos en imágenes que en una sola pasada predece las regiones en donde es más probable la presencia de objetos y las probabilidades para cada clase en esas regiones.

#### 3.2. Conjuntos de Datos

Una parte fundamental del algoritmo es el entrenamiento del modelo detector. Para dicho entrenamiento es necesario contar con un conjunto de imágenes de muestra convenientemente etiquetadas con las clases que se quieren detectar. Para el caso de detección de fallas en general se puede considerar la detección del producto con falla y del producto sin falla (2 clases de interés), o simplemente entrenar para detectar el producto con falla (1 sola clase de interés). Esto último exige una mayor precisión en el proceso de etiquetado, ya que el modelo no se ajusta para la detección del producto sin falla en forma explícita sino que asume que el resto de las muestras no etiquetadas son producto sin falla. Para este problema en un principio se trabajó con dos clases, las cuales representaban al amortiguador con buje y al amortiguador sin buje. Con esta configuración se realizó un análisis de factibilidad inicial de detección en condiciones de laboratorio. En la Fig. 2 se pueden apreciar algunos ejemplos de estas imágenes. Dichas imágenes no fueron capturadas sobre una cinta transportadora, sino que se tomaron sobre un fondo blanco de papel.

Después de realizado el análisis de factibilidad, el cual corroboró en principio la viabilidad de la hipótesis propuesta inicialmente se cambió el enfoque del problema y se decidió utilizar solo una clase. Dicha clase representa la falla a

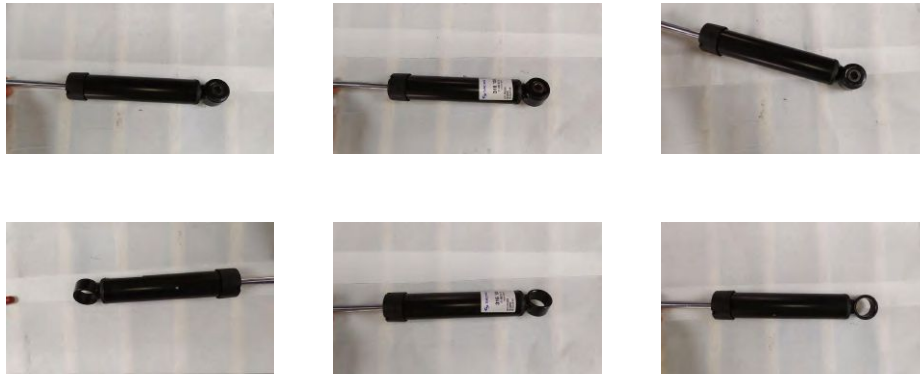


Fig. 2: Imágenes de amortiguadores en condiciones de laboratorio.



Fig. 3: Imágenes de amortiguadores pasando sobre la cinta transportadora.

detectar que consiste en el amortiguador sin buje. Este cambio trajo aparejado una ventaja muy importante debido a que la mayoría de los amortiguadores tienen el buje, entonces el etiquetado se volvía muy lento por la cantidad de amortiguadores con buje presentes en las imágenes; en cambio usando una sola clase, la cantidad de objetos fallados era mucho menor disminuyendo el tiempo de etiquetado. Antes de realizar este cambio se corroboró que la exactitud en la detección de amortiguadores fallados sea aproximadamente la misma. En la Fig. 3 se pueden ver algunos ejemplos del mismo amortiguador con y sin buje en condiciones reales sobre la cinta transportadora.

Uno de los problemas principales es que los amortiguadores pasan de manera desordenada, sin una posición fija y posiblemente muy cercano uno de otro causando oclusiones, cambios en la alineación, en las vistas, etc. En las Fig. 1 y Fig. 3 se puede apreciar la vista que tienen las cámaras de los amortiguadores pasando sobre la cinta. Por lo tanto, lo ideal es que la colección tenga imágenes donde se observen los amortiguadores desde diferentes puntos de vista, en la mayor cantidad de posiciones posibles y con amortiguadores sin falla o con falla.

Por lo tanto para poder generar los conjunto de datos, lo primero que se hizo fue buscar de manera empírica la cantidad de cámaras necesarias y las mejores ubicaciones para lograr minimizar los problemas de oclusiones del extremo de los amortiguadores en donde se encuentran los bujes a detectar. Basándonos en los primeros experimentos en planta se decidió utilizar 4 cámaras, 2 de ellas

ubicadas en la parte superior de la línea y 2 cámaras rasantes. La ubicación de las cámaras se puede apreciar en la Fig. 4. Las imágenes obtenidas por cada una de las cámaras se pueden apreciar en la Fig. 5.

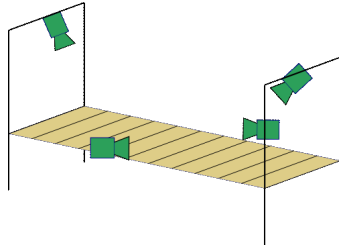


Fig. 4: Disposición de las cámaras del sistema.



Fig. 5: Vista de cada una de las cámaras.

Usando esta configuración de cámaras se capturaron 39 videos de los cuales se extrajeron 21322 imágenes. Estas imágenes se separaron en dos conjuntos, uno de entrenamiento compuesto por 17073 imágenes y otro de validación compuesto por 4249 imágenes (ver tabla 2). Para la creación de los videos, se implementó un script en python el cual se encarga de realizar en forma automática la captura en las 4 cámaras y la posterior conversión de los videos en una secuencia de imágenes. Para esta última conversión se generó una imagen por segundo de video.

Tabla 2: Conjuntos de datos utilizados. Número de clases y cantidad de muestras de entrenamiento y evaluación.

Clases	1
Cantidad de videos	39
Entrenamiento	17073
Evaluación	4249

### 3.3. Configuración Experimental

Para los experimentos se utilizó la librería llamada Darknet<sup>2</sup>, la cual permite configurar los modelos, entrenarlos y evaluarlos. El modelo YOLOv4 fue configurado con un tamaño de imagen de entrada de  $416 \times 416$ , y con un máximo número de iteraciones de  $50k$ . Se estableció un tamaño de lote de 32, con una subdivisión de 8, un momento de 0,949 y un decaimiento de 0,005. También se configuró la tasa de aprendizaje en 0,001 con una reducción de 0,1 veces del original a medida que se alcanzan los pasos de iteraciones  $40k$  y  $45k$  respectivamente. Para el entrenamiento de los distintos modelos se dividieron los diferentes conjuntos de datos en 2 partes en forma aleatoria: una parte formada por el 80 % de las imágenes para entrenamiento y otra compuesta por el 20 % restante para validación como se puede apreciar en la tabla 2. Para la evaluación y comparación de los experimentos se utilizó como medida la precisión promedio media (mAP, por sus siglas en inglés). La mAP se mide como la media de la precisión promedio obtenida para cada una de las clases:

$$\text{mAP} = \sum_{c=1}^C \text{AP}_c \quad (1)$$

donde  $C$  es el número de clases y  $\text{AP}_c$  es la precisión promedio para la clase  $c$ . Para este caso de una sola clase el cálculo es más simple. Esta precisión promedio ó AP (por sus siglas en inglés) se calcula como el área bajo la curva de precisión y exhaustividad<sup>3</sup> en donde los positivos verdaderos ó TP (por sus siglas en inglés) se consideran cuando la intersección sobre la unión entre la caja limitante de la detección y la ubicación verdadera del objeto es mayor a 0,5, es decir:

$$\text{IoU} = \frac{\text{Área de la Intersección}}{\text{Área de la Unión}} > 0,5 \quad (2)$$

Todos los entrenamientos fueron realizados en una PC de escritorio con un procesador AMD Ryzen 9 3900X 12-Core de 3,8GHz con 16GB de RAM y una GPU Nvidia Geforce RTX 2080Ti 11Gb.

### 3.4. Sistema de detección propuesto

Para la utilización del algoritmo en la aplicación real en la industria, se diseñó un sistema compuesto por una computadora de escritorio con un procesador Intel Core i9 10900F S1200, con 16GB de memoria RAM, una GPU Nvidia Geforce RTX 3080Ti 12Gb y 4 cámaras USB Webcam Logitech con resolución de 1080p. En esta PC se instaló un sistema operativo Debian<sup>4</sup> versión 10.5 y el entorno Darknet. El contiene también un monitor para visualizar las detecciones y una alarma sonora y visual para indicar a los operarios que hay un faltante de buje

<sup>2</sup> <https://github.com/AlexeyAB/darknet>

<sup>3</sup> En lenguaje inglés esta curva es conocida como curva de precision/recall.

<sup>4</sup> <https://www.debian.org/>





Fig. 6: Sistema de detección propuesto en condiciones normales de funcionamiento en la línea de montaje de la planta.

en algún amortiguador. El sistema de alarma se encuentra comandado por la PC a través de un relé con interfaz USB. También en el monitor se muestra cual es el amortiguador que presenta la falla para facilitar la tarea del operario. Además de esto, un requerimiento solicitado por la empresa era no utilizar teclado ni mouse, por lo tanto se utilizó un lector de código de barras LI2208 de la marca Zebra el cual se utiliza para prender y apagar el sistema, la PC y entrar en el modo grabación de videos para la generación de nuevos videos de entrenamiento. En la Fig. 6 se pueden observar dos imágenes del sistema en funcionamiento en planta, donde se pueden apreciar los amortiguadores pasando sobre la cinta en condiciones normales de funcionamiento, el monitor con las imágenes en tiempo real de las 4 cámaras (en celeste), a la derecha de este la alarma visual y sonora (en amarillo) y en la parte superior el gabinete de la PC en donde se realizan los cómputos (en rojo).

### 3.5. Software

Con respecto a la implementación del software el requerimiento fundamental fue que tenga una mínima interacción con los operarios para no generar distracciones adicionales. Para cumplir con esto se implementó una aplicación en lenguaje Python que se ejecuta de manera automática cuando la PC se enciende. La aplicación se encarga de cargar el modelo de detección entrenado, leer cuantas cámaras hay conectadas a la PC y aplicar el algoritmo de detección en cada una de las cámaras. Para la interacción con el usuario se utiliza una

lectora de código de barras mediante la cuál se puede indicar al sistema que dé inicio al proceso de detección utilizando un determinado modelo y que finalice con el proceso de detección. En caso de que se detecte una falla (amortiguador sin buje) se envía una señal a través del protocolo USB al sistema de alarma. En condiciones normales, el sistema de alarma mantiene una baliza lumínica en verde, pero en caso de que haya una falla comienza a sonar una sirena y la baliza se pone en color rojo. Esto último se mantiene hasta que los operarios quiten el producto defectuoso de la cinta. Además la aplicación muestra las capturas de las cámaras en el monitor conectado a la PC en tiempo real y en caso de que haya una falla, ésta se muestra a través de un recuadro sobre el amortiguador con buje faltante como se puede apreciar en la Fig. 1.

## 4. Resultados

### 4.1. Análisis de factibilidad en condiciones de laboratorio

Como se comentó en la sección anterior, el primer experimento realizado fue analizar la factibilidad de la detección en base a imágenes de laboratorio con un fondo blanco que facilite la detección (ver Fig. 2). Este conjunto de imágenes fue dividido en dos grupos y con el primer grupo compuesto por 50 imágenes se entrenó un modelo con el propósito de distinguir entre amortiguadores con buje y sin buje. El segundo grupo de 14 imágenes fue usado para evaluar el comportamiento de la red entrenada sobre imágenes nuevas que no fueron vistas nunca antes por esta red. Con esta configuración se obtuvo una precisión de  $AP = 75\%$  y si bien en los resultados se apreciaron algunas fallas en las detecciones realizadas por la red, es sabido que estos algoritmos basados en Redes Neuronales necesitan muchos datos de entrenamiento para funcionar en forma óptima. A pesar de utilizar tan pocos datos de entrenamiento los resultados fueron muy buenos por lo tanto se decidió avanzar con la empresa en la solución final. Pero lo más importante de este análisis inicial fueron las conclusiones que surgieron del mismo, las cuales forman parte del sistema construido. Para obtener una óptima detección se necesita cumplir con las siguientes condiciones de trabajo:

- Utilizar una cinta blanca o de un color que contraste mucho con el color del amortiguador.
- Utilizar más de una cámara para que los amortiguadores se vean desde diferentes ángulos y sea más robusta la detección.
- Capturar un número suficiente de imágenes en condiciones reales de trabajo para realizar el entrenamiento de la Red Neuronal.

### 4.2. Evaluación sobre el conjunto de validación

Después de realizar el ajuste del modelo usando la imágenes de entrenamiento capturadas en condiciones de trabajo reales (tabla 2), este fue evaluado en el conjunto de imágenes de validación. Los resultados obtenidos se observan en la

tabla 3. Sobre un total de 4249 imágenes se realizaron 4117 detecciones únicas con 472 falsas detecciones (Falsos positivos), obteniéndose un  $AP = 95,55\%$  y una  $IoU = 72,31\%$ . Si bien el  $AP$  obtenido con el modelo final es de  $95,55\%$ , se puede

Tabla 3: Métricas obtenidas sobre el conjunto de evaluación.

Modelo	Cantidad de detecciones	Veraderos Positivos	Falsos Positivos	Precision	Recall	F1-Score	IoU	AP
YOLOv4	6509	3913	492	0.89	0.95	0.92	72.31 %	<b>95.55 %</b>

notar en la tabla que las fallas se producen mayormente por falsos positivos (ver tabla 3) en condiciones de trabajo en planta un nivel de falsos positivos tan alto no es tolerable debido a que los operarios eran alertados en forma incorrecta. Por lo tanto, para disminuir las falsas alarmas se subió el umbral a un valor de 0,9 y se implementó un sistema de filtrado temporal mediante el cual la detección de la falla genera una alarma solamente si se mantiene durante una cantidad determinada de imágenes consecutivas. De esta manera se logró un sistema confiable que solo genera alarmas cuando detecta un amortiguador sin buje.

### 4.3. Evaluación operativa en la línea de producción

Después de la evaluación en laboratorio y de la realización de los ajustes nombrados en la sección anterior, el modelo se corrió sobre el sistema prototipo propuesto en 3.4 para la detección de amortiguadores con faltante de buje en condiciones normales de funcionamiento en planta. La velocidad de procesamiento obtenida con este equipo estándar fue de 15 FPS procesando 4 cámaras a la vez, con lo cual se logra cumplir en forma satisfactoria la respuesta esperada para un sistema de alarma de detección de fallas en tiempo real corriendo en una PC estándar.

Durante el año 2021 la verificación de la presencia del buje en cada amortiguador la realizaban los operarios a través de inspección visual, pieza por pieza y marcando el buje colocado en su posición con un marcador de color para asegurar su control. Con esto, el reporte de piezas defectuosas debido a falta de bujes durante ese año fue de 12 lotes reclamados. El sistema de detección de fallas se encuentra en funcionamiento en la línea de producción desde hace cinco meses (desde febrero de 2022), funcionando en modo automático y de forma no asistida. Durante este tiempo no se reportaron reclamos por esta falla.

## 5. Conclusiones

En este trabajo se presentó el diseño, implementación y puesta a punto en planta de un sistema de detección de fallas en la producción de amortiguadores.

En particular, para este caso la falla a detectar es el faltante de buje el cual según un análisis de la empresa es una de las fallas más comúnmente reportada. Si bien el tiempo de funcionamiento del sistema en forma no asistida es aún corto, los resultados obtenidos hasta el momento son muy alentadores.

La hipótesis planteada inicialmente fue que es posible detectar el faltante de buje en amortiguadores para poder generar una alarma y alertar a los operarios de la falla en la producción usando técnicas de CV y de detección de objetos basadas en DL. Dicha hipótesis fue validada en experimentos de laboratorio demostrando la factibilidad para la detección como se puede observar en la Tabla 3 logrando una mAP de aproximadamente 95 %. Además el sistema se encuentra en funcionamiento en planta desde febrero del 2022 con un comportamiento satisfactorio<sup>5</sup> alertando a los operarios sobre posibles fallas en la producción de amortiguadores. De esta manera se pudo corroborar el planteo inicial en condiciones de funcionamiento normal. Como trabajo a futuro la empresa nos solicitó dos nuevos sistemas de control de calidad en base a los modos de fallas de la tabla 1; el primero para determinar desperfectos en la etiqueta del amortiguador y el segundo para determinar el torque correcto<sup>6</sup> de algunos amortiguadores que en uno de sus extremos llevan un bástago roscado con una tuerca.

## Agradecimientos

Este trabajo fue desarrollado en el marco del proyecto homologado PID UTN - CCUTNCO 0007894 "Detección de objetos en ambientes industriales utilizando técnicas de aprendizaje profundo" de la Universidad Tecnológica Nacional, y el Convenio de Transferencia Tecnológica firmado con la empresa ZF. Los autores agradecen al Sr. Cristian Prunello, jefe del departamento de mejora continua de la empresa mencionada por su valiosa contribución al proyecto.

## Referencias

1. Ali, M.H., Aizat, K., Yerkhan, K., Zhandos, T., Anuar, O.: Vision-based robot manipulator for industrial applications. *Procedia computer science* **133**, 205–212 (2018)
2. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020)
3. Butters, L., Xu, Z., Klette, R.: Using machine vision to command a 6-axis robot arm to act on a randomly placed zinc die cast product. In: *Proceedings of the 2nd International Conference on Control and Computer Vision*. pp. 8–12 (2019)
4. Ćorović, A., Ilić, V., Đurić, S., Marijan, M., Pavković, B.: The real-time detection of traffic participants using yolo algorithm. In: *2018 26th Telecommunications Forum (TELFOR)*. pp. 1–4. IEEE (2018)

<sup>5</sup> Se aclara que este trabajo fue escrito a finales de junio de 2022, con lo cual estamos hablando de aproximadamente 150 días de funcionamiento satisfactorio de forma ininterrumpida.

<sup>6</sup> Este proceso consiste en pasar al amortiguador por una máquina que ajusta una tuerca de sujeción con un determinado torque.

5. Cortés, C.B.Y., Landeta, J.M.L., Chacón, J.G.B., Pereyra, F.A., Osorio, M.L.: El entorno de la industria 4.0: implicaciones y perspectivas futuras. *Conciencia tecnológica* (54) (2017)
6. Gonzalez Dondo, D., Redolfi, J.A., Araguás, R.G., Garcia, D.: Application of deep-learning methods to real time face mask detection. *IEEE Latin America Transactions* **19**(6), 994–1001 (2021). <https://doi.org/10.1109/TLA.2021.9451245>
7. Griffiths, F., Ooi, M.: The fourth industrial revolution-industry 4.0 and iot [trends in future i&m]. *IEEE instrumentation & measurement magazine* **21**(6), 29–43 (2018)
8. Hirano, Y., Garcia, C., Sukthankar, R., Hoogs, A.: Industry and object recognition: Applications, applied research and challenges. In: *Toward Category-Level Object Recognition*, pp. 49–64. Springer (2006)
9. Issenhuth, T., Srivastav, V., Gangi, A., Padoy, N.: Face detection in the operating room: Comparison of state-of-the-art methods and a self-supervised approach. *International Journal of Computer Assisted Radiology and Surgery* **14**(6), 1049–1058 (2019)
10. Jeong, S.H., Son, G.J., Son, J.H., Kim, Y.D.: A proposal of real-time foreign object detection for food through conveyor belt using hyperspectral camera. *Bulletin of Networking, Computing, Systems, and Software* **9**(1), 51–52 (2020)
11. Koh, L., Orzes, G., Jia, F.J.: The fourth industrial revolution (industry 4.0): technologies disruption on operations and supply chain management. *International Journal of Operations & Production Management* (2019)
12. Lan, W., Dang, J., Wang, Y., Wang, S.: Pedestrian detection based on yolo network model. In: *2018 IEEE international conference on mechatronics and automation (ICMA)*. pp. 1547–1551. IEEE (2018)
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: *European conference on computer vision*. pp. 21–37. Springer (2016)
14. Luan, S., Li, Y., Wang, X., Zhang, B.: Object detection and tracking benchmark in industry based on improved correlation filter. *Multimedia Tools and Applications* **77**(22), 29919–29932 (2018)
15. Martinelli, A., Mina, A., Moggi, M.: The enabling technologies of industry 4.0: examining the seeds of the fourth industrial revolution. *Industrial and Corporate Change* **30**(1), 161–188 (2021)
16. Montoya Holguin, C., Cortés Osorio, J.A., Chaves Osorio, J.A.: Sistema automático de reconocimiento de frutas basado en visión por computador. *Ingeniare. Revista chilena de ingeniería* **22**(4), 504–516 (2014)
17. Olmedo, M., Redolfi, J.A., González Dondo, D., Araguás, R.G.: EVALUACIÓN EMPÍRICA DE LA ROBUSTEZ DE DIFERENTES REDES NEURONALES USADAS PARA LA DETECCIÓN DE OBJETOS. XXIV Congreso sobre Métodos Numéricos y sus Aplicaciones-ENIEF 2019. Asociación Argentina. De Mecánica Computacional (AMCA). Santa Fe, Argentina, Noviembre 5-7. Mecánica computacional (2019)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*. pp. 91–99 (2015)
19. Sanver, U., Yavuz, E., Eyupoglu, C.: An image processing application to detect faulty bottle packaging. In: *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. pp. 986–989. IEEE (2017)

20. Sarıbaş, H., Çevikalp, H., Kahvecioğlu, S.: Car detection in images taken from unmanned aerial vehicles. In: 2018 26th Signal Processing and Communications Applications Conference (SIU). pp. 1–4. IEEE (2018)
21. Sarkar, A., Chakrabarty, S., Roy, B.: On online counting of cigarette in packets—an image processing approach. In: Proceedings of the International Conference on Nano-electronics, Circuits & Communication Systems. pp. 177–190. Springer (2017)
22. Schwab, K.: La cuarta revolución industrial. Debate (2016)
23. Shi, J., Tao, W.: Dynamic object counting application based on object detection and tracking. In: Tenth International Conference on Digital Image Processing (IC-DIP 2018). vol. 10806, p. 1080614. International Society for Optics and Photonics (2018)
24. Tripathy, D., Reddy, K.G.R.: Adaptive threshold background subtraction for detecting moving object on conveyor belt. Intl. Journal of Indestructible Mathematics and Computing **1**(1), 41–46 (2017)
25. Walam, S.S., Teli, S.P., Thakur, B.S., Nevarekar, R.R., Patil, S.M.: Object detection and separation using raspberry pi. In: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). pp. 214–217. IEEE (2018)
26. Wang, Y., Yan, G., Meng, Q., Yao, T., Han, J., Zhang, B.: Dse-yolo: Detail semantics enhancement yolo for multi-stage strawberry detection. Computers and Electronics in Agriculture **198**, 107057 (2022)
27. Yang, J., Li, S., Wang, Z., Yang, G.: Real-time tiny part defect detection system in manufacturing using deep learning. IEEE Access **7**, 89278–89291 (2019)
28. Yuan, R., Jaime, I., Chiabrando, B.J., Redolfi, J.A.: Detección de paquetes en movimiento sobre una cinta transportadora usando visión por computadora. In: 2020 IEEE Congreso Bienal de Argentina (ARGENCON). pp. 1–6. IEEE (2020)
29. Zhao, B., Feng, J., Wu, X., Yan, S.: A survey on deep learning-based fine-grained object classification and semantic segmentation. International Journal of Automation and Computing **14**(2), 119–135 (2017)