

Sistema inteligente para identificar defectos en tablillas electrónicas

Rubén Arturo Domínguez Altamirano ¹, M.C. Arturo Legarda Sáenz ¹, Dr. Hernán de la Garza Gutiérrez ¹, MISC Jesús Arturo Alvarado Granadino ¹

¹ División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México – Instituto Tecnológico de Chihuahua 2, Av. de las Industrias 11101, Complejo Industrial Chihuahua, C.P. 31130, Chihuahua, Chihuahua, México.

Correspondencia: ruben.dominguez.3792@gmail.com

Resumen

En la industria existen muchos equipos que realizan actividades repetitivas, las cuales resultan cansadas para una persona. Por ese motivo, suelen conseguirse equipos que realicen las tareas; sin embargo, las tareas siguen dependiendo de una persona para analizar o revisar lo que es detectado por el equipo. Estos equipos pueden ser complementados con las nuevas tecnologías como la Inteligencia Artificial (IA). En este trabajo de investigación se realiza el diseño y evaluación de un sistema inteligente (SI) para la identificación de posibles defectos en muestras físicas que son capturadas a imágenes por un equipo de inspección automática (AOI) modelo FX940 UV. Para lograrlo, se utilizan técnicas de inteligencia artificial que buscan aprovechar y ampliar el beneficio de un equipo AOI en una planta de manufactura, con el diseño y creación de su propio *dataset* ajustado al modelo que produce la planta de manufactura, en este caso el área de *conformal coating*. Además, se utiliza una técnica de transferencia de aprendizaje (*Transfer learning*) que aprovecha el entrenamiento previo de una red neuronal Mask R-CNN que tiene una velocidad más rápida de entrenamiento y pruebas. Esta investigación muestra que aún es posible obtener mayores beneficios de los equipos que se tienen actualmente en la industria y obtener mayor eficiencia en los procesos; así como que los actuales y futuros equipos que se encuentran disponibles en el mercado pueden mejorar y aprovechar las nuevas tecnologías.

Palabras clave

AOI, Mask R-CNN, dataset, sistema inteligente, inteligencia artificial, transferencia de aprendizaje, conformal coating.

Abstract

In industry, there are many pieces of equipment that perform repetitive activities, which are tiring for one person. For that reason, equipment is often procured to perform the tasks; however, the tasks still depend on a person to analyze or review what is detected by the equipment. This equipment can be complemented with new technologies such as Artificial Intelligence (AI). In this research work, the design and evaluation of an intelligent system (IS) for the identification of possible defects in physical samples that are captured in images by an automatic inspection equipment (AOI) model FX940 UV is carried out. To achieve this, artificial intelligence techniques are used to take advantage and extend the benefit of an AOI equipment in a manufacturing plant, with the design and creation of its own dataset adjusted to the model produced by the manufacturing plant, in this case the conformal coating area. In addition, a transfer learning technique is used that takes advantage of the previous training of a Mask R-CNN neural network that has a faster training and testing speed. This research shows that it is still possible to obtain greater benefits from the equipment currently available in the industry and obtain greater efficiency in the processes; as well as that the current and future equipment available in the market can improve and take advantage of new technologies.

Keywords

AOI, Mask R-CNN, dataset, intelligent system, artificial intelligence, transfer learning, conformal coating.

Artículo arbitrado

Recibido:	Aceptado:
15 de octubre de 2021	21 de octubre de 2021

Introducción

Los sistemas inteligentes suelen representarse como aquellos que tienen un comportamiento similar a la inteligencia humana o animal. Esto se debe a que se definen por su capacidad de representar, procesar y modificar su conocimiento en algún tema y a la capacidad de mejorar su desempeño con base en la retroalimentación, lo cual les permite resolver problemas concretos a través del establecimiento de acciones para lograr los objetivos propuestos, mediante interacciones y adaptaciones a las distintas situaciones del entorno (UNED, 2016).

Usualmente, los sistemas inteligentes que analizan imágenes o patrones suelen utilizar algunas técnicas de la IA como Visión Artificial (VA), Redes Neuronales (RN), entre otras. La visión artificial consiste en métodos para adquirir, procesar, analizar y comprender imágenes. Las redes neuronales son modelos computacionales que intentan imitar la forma como nuestro cerebro aprende a identificar patrones, se realiza a través parámetros que deben ajustarse de acuerdo con la actividad que se esté realizando. La unidad estructural de una red neuronal es la neurona, esta técnica emula la naturaleza biológica de nuestro cerebro, donde la neurona recibe señales de entrada y puede o no emitir una señal de salida, este comportamiento es emulado en las neuronas artificiales con la función de activación (Reyes, 2020).

En este trabajo de investigación se diseñó y evaluó un sistema inteligente que busca complementar a los equipos automáticos de AOI que están en las plantas de manufactura. Básicamente, los equipos de AOI capturan imágenes de las tablillas con *conformal coating* para determinar si se encuentra presente una *falla* que puede ser la ausencia de componentes, burbujas en *conformal coating*, desplazamiento de un componente, contaminación o de un *falso* que representa una alerta mas no un defecto real. Sin embargo, una persona debe estar revisando cada

llamada que hace el equipo para realizar un veredicto de acuerdo con sus estándares o capacitaciones adquiridas. Este sistema podría provocar alguna mala clasificación y el riesgo de un escape. Además, se consume el tiempo de una persona para la revisión. Por esa razón, la propuesta que realizamos, mediante el análisis de las imágenes obtenidas con el equipo AOI, realiza la identificación de los defectos en estas imágenes y toma decisiones más rápidas y con una mayor precisión para disminuir los escapes de fallas y reducción de falsos en las tablillas electrónicas de las producciones diarias.

Para que el sistema inteligente realice estas actividades debe ser entrenado. Con este objetivo, se creó un *dataset* exclusivo al tipo de tablilla electrónica y proceso de *conformal coating* en búsqueda de burbujas. Se seleccionaron las tarjetas disponibles con algún defecto, aunque se encontró una pequeña cantidad de tarjetas disponibles con defecto, ya que cuando se detecta algún defecto se detiene la producción y se soluciona el problema. Por esa razón, no es posible tener muchas tarjetas defectuosas. Sin embargo, existe el riesgo de que una tarjeta defectuosa no sea detectada y siga en el proceso hasta llegar al cliente, lo cual podría provocar un rechazo de calidad más adelante. utilizarse utiliza la transferencia de aprendizaje mediante el uso de redes neuronales como Mask R-CNN, las cuales están basadas en una red neuronal convolucional que se complementa con extracción de áreas delimitadas, las cuales ayudan a segregar la imagen obtenida e identificar mayores rasgos en una imagen, lo que beneficia la identificación de posibles defectos y que tiene una velocidad más rápida de entrenamiento y pruebas.

Como se describe en la investigación de esta red neuronal Mask R-CNN, la red, detecta eficientemente los objetos de una imagen y, al mismo tiempo, genera máscaras de segmentación de alta calidad para cada instancia, lo cual amplía una red Faster R-CNN al añadir una ramificación a fin de predecir una máscara de objeto en paralelo a la ramificación existente para el reconocimiento del

cuadro delimitador. Para este fin, mencionan que lograron los mejores resultados en los retos de COCO, como la segmentación de instancias, la detección de objetos en cajas delimitadas y la detección de puntos clave de personas. Mask R-CNN supera a todos los modelos existentes en cada tarea, incluidos los ganadores del desafío COCO 2016 (Kaiming, 2017).

COCO es un *dataset* de Microsoft que es ampliamente utilizado en proyectos de aprendizaje automático. COCO son las siglas de Common Objects in Context (Objetos Comunes en Contexto), fue creado con el objetivo de avanzar en el reconocimiento de imágenes, contiene conjuntos de datos visuales exigentes y de alta calidad para la visión artificial, en su mayoría redes neuronales de última generación (Vidushi, 2021).

Metodología

Se comenzó con la obtención de las muestras disponibles defectuosas en el proceso de *conformal coating* de la empresa, que fueron escaneadas en un equipo AOI modelo FX940 UV. El equipo guardó por carpeta imágenes de cada muestra procesada, las cuales fueron utilizadas para crear el *dataset* del modelo. Se procesaron las imágenes de las muestras en el software de delimitación, para identificar las zonas de defecto y así crear una base para el entrenamiento de la red neuronal convolucional y su posterior validación de cordura y pesos. De esa manera, posteriormente fue posible iniciar con el procesamiento de las nuevas muestras.

En este trabajo de investigación se utilizaron algunas herramientas existentes para la creación del *dataset*. Además, se transfirió el aprendizaje entre redes neuronales y se utilizaron muestras con defecto de burbuja en el proceso de *conformal coating*.

Identificación de los defectos en VGG Image Annotator.

El primer paso es la importación de las imágenes con la aplicación VGG Image Annotator (Dutta y Zisserman, 2019), donde se crea una clase llamada

“name” y se delimitan los defectos mediante puntos para que la identificación sea más restrictiva. Éstos se nombran como “*damage*” en la clasificación, como se puede apreciar en la Figura 1. Experimentalmente, se seleccionaron 22 muestras defectuosas que fueron pasadas por el equipo para obtener las imágenes. Más adelante, se identificó que era necesario incrementar la cantidad de estas imágenes, por lo que se realizó nuevamente el procedimiento ahora con 40 imágenes, las cuales tenían una variación en la cantidad de defectos presentes, desde uno hasta seis defectos.



Figura 1. Identificación de clases. Fuente: elaboración propia.

Posteriormente, se guardaron todas las anotaciones realizadas en un archivo de tipo JSON, junto a las imágenes que fueron clasificadas (ver Figura 2), las cuales sirvieron para iniciar con el entrenamiento de la red neuronal.

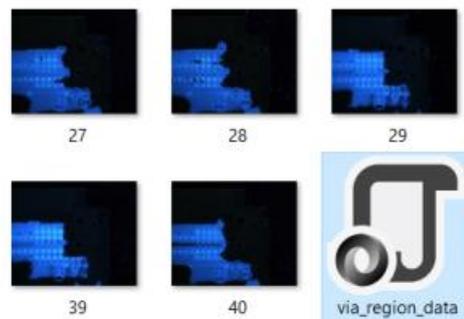


Figura 2. Imágenes con archivo de anotaciones (RPN). Fuente: elaboración propia.

Entrenamiento de la red Mask R-CNN

Primeramente, se utilizó una red ya existente de tipo Mask R-CNN que utiliza el lenguaje de programación Python 3.6 y la técnica de transferencia de aprendizaje. En el entrenamiento del modelo, se generaron cajas delimitadoras y máscaras de segmentación para cada instancia de un objeto en la imagen. Todo esto utilizó la Red de Pirámide de Características (FPN) y *backbone* de ResNet101. Para entrenar al modelo, se hizo uso de un *dataset* ya entrenado llamado COCO como punto de control para realizar la transferencia de aprendizaje. Aprovechando la investigación de Kaiming (2017) de la red Mask R-CNN, se eligió este modelo para realizar el presente proyecto.

Se configuraron las carpetas en sus rutas para cargar las imágenes de entrenamiento (ver Figura 3).

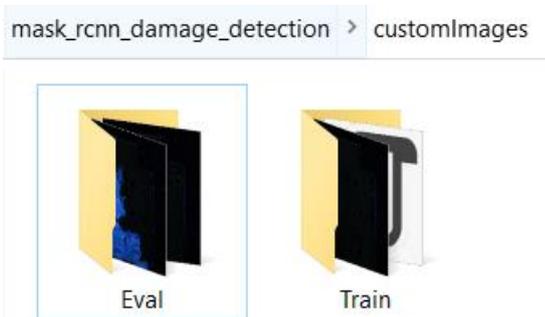


Figura 3. Carpetas de entrenamiento. Fuente: elaboración propia.

Posteriormente, se cargaron los datos de la carpeta de entrenamiento y se obtuvo la información presentada en la Figura 4. Tal como se puede observar, se trata de dos clases: una de BG o *background* y otra llamada *damage*. En esta información también se incluyeron las 40 imágenes obtenidas del equipo AOI para el paso de entrenamiento.

```
# Load dataset
# Get the dataset from the releases page
# https://github.com/matterport/Mask_RCNN/releases
dataset = custom.CustomDataset()
dataset.load_custom(CUSTOM_DIR, "train")

# Must call before using the dataset
dataset.prepare()

print("Image Count: {}".format(len(dataset.image_ids)))
print("Class Count: {}".format(dataset.num_classes))
for i, info in enumerate(dataset.class_info):
    print("{:3}. {:50}".format(i, info['name']))
```

Image Count: 40
Class Count: 2
0. BG
1. damage

Figura 4. Resultado de carpetas de entrenamiento. Fuente: elaboración propia.

Después, se cargó el *dataset* que había sido creado al inicio del trabajo, donde se encuentran las imágenes de validación, lo cual incluye tanto a las imágenes que se usaron para evaluación y el *dataset* COCO. Así, se inició el entrenamiento (ver Figura 5) y se creó el *dataset* “mask_rcnn_damage.h5”

```
#Train the head branches
#Passing layers="heads" freezes all layers
except the head layers.
#You can also pass a regular expression to
select which layers to train by name pattern.
model.train(dataset_train, dataset_val,
learning_rate=config.LEARNING_RATE,
epochs=1,
layers='heads')

model_path = os.path.join(ROOT_DIR, "mask_rcnn_damage.h5")
model.keras_model.save_weights(model_path)
```

Figura 5. Inicio del entrenamiento (Fuente: elaboración propia).

Después de terminar el entrenamiento de la época, obtuvimos los valores del resultado del entrenamiento, como la pérdida obtenida entre las distintas capas de la red (ver Figura 6).

```
Epoch 1/1
100/100 [=====] - 2651s 27s/step - loss: 1.3565 -
rpn_class_loss: 0.0300 - rpn_bbox_loss: 0.2892 - mrcnn_class_loss: 0.0732 -
mrcnn_bbox_loss: 0.5032 - mrcnn_mask_loss: 0.4610 - val_loss: 1.3652 -
val_rpn_class_loss: 0.0265 - val_rpn_bbox_loss: 0.2214 - val_mrcnn_class_loss: 0.0772 -
val_mrcnn_bbox_loss: 0.6442 - val_mrcnn_mask_loss: 0.3959
```

Figura 6. Resultados del entrenamiento. Fuente: elaboración propia.

Validación del *dataset* “mask_rcnn_damage.h5”

Después de obtener el modelo, se realizó una validación de los pesos entre las capas, para verificar si existía un control de cordura y pesos y determinar si los sesgos estaban bien distribuidos.

Una muestra de los resultados de salida se observa en la Figura 7 y Figura 8.

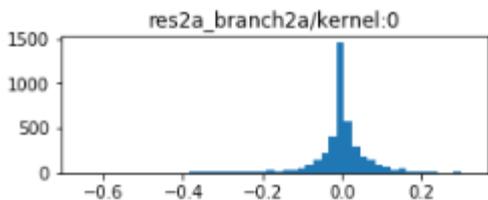
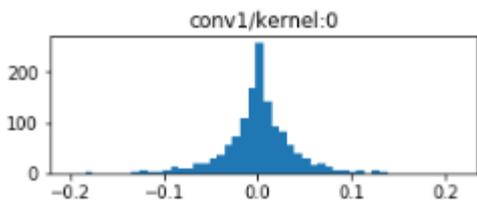


Figura 7. Resultados de validación del modelo. Fuente: elaboración propia.

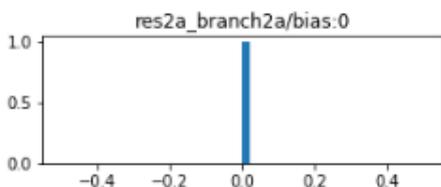
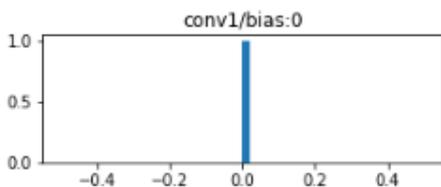


Figura 8. Resultados de validación del modelo. Fuente: elaboración propia.

Preprocesamiento de los datos

El sistema inteligente, a través del *dataset* entrenado previamente (*mask_rcnn_damage.h5*) calculó los cuadros delimitadores a partir de las máscaras (ver Figura 9), en cada imagen que fue proporcionada.

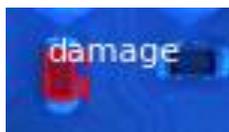


Figura 9. Imagen amplificada con máscara. Fuente: elaboración propia.

Esto permite manejar los cuadros delimitadores de forma consistente, independientemente del conjunto de datos de origen. Asimismo, facilita el cambio de tamaño, la rotación o el recorte de imágenes porque simplemente se generan los cuadros delimitadores a partir de las máscaras de actualización en lugar de computar la transformación de los cuadros delimitadores para cada tipo de transformación aplicado a la imagen (ver Figura 10)

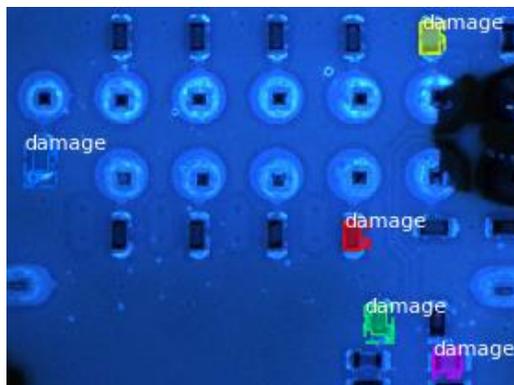


Figura 10. Cuadros delimitadores en defectos. Fuente: elaboración propia.

Para soportar múltiples imágenes por lote, las imágenes se redimensionaron a un tamaño (1024x1024), conservando la relación de aspecto. Si una imagen no es cuadrada, entonces se añade un relleno de cero en la parte superior/inferior o derecha/izquierda.

Se crean mini máscaras, las máscaras binarias de instancia pueden hacerse grandes al entrenar con imágenes de alta resolución. Para visualizar el efecto del redimensionamiento de la máscara y para verificar la corrección del código, se aumenta y redimensiona la máscara (ver Figura 11).



Figura 11. Imagen con máscara aumentada. Fuente: elaboración propia.

Se inició con el proceso de las anclas (*anchors*). El

orden de éstas es importante, pues deben usar el mismo en las fases de entrenamiento y de predicción y deben coincidir con el orden de la ejecución de la convolución, para una red FPN. Las anclas deben estar ordenadas de manera que sea fácil hacer coincidir las anclas con la salida de las capas de convolución que predicen las puntuaciones y los desplazamientos de las anclas, se debe organizar primero por nivel de pirámide. Todas las anclas del primer nivel, luego todas las del segundo y así sucesivamente. Esto hace más fácil separar las anclas por nivel, dentro de cada nivel, se acomodan las anclas por la secuencia de procesamiento del mapa de características.

Típicamente, una capa de convolución procesa un mapa de características comenzando desde la parte superior izquierda y moviéndose a la derecha fila por fila. Para cada celda del mapa de características, se elige cualquier orden de clasificación para las anclas de diferentes proporciones. Aquí se hace coincidir con el orden de las proporciones pasadas a la función. En la Figura 12, se visualizan las anclas de una celda en el centro del mapa de características de un nivel específico.

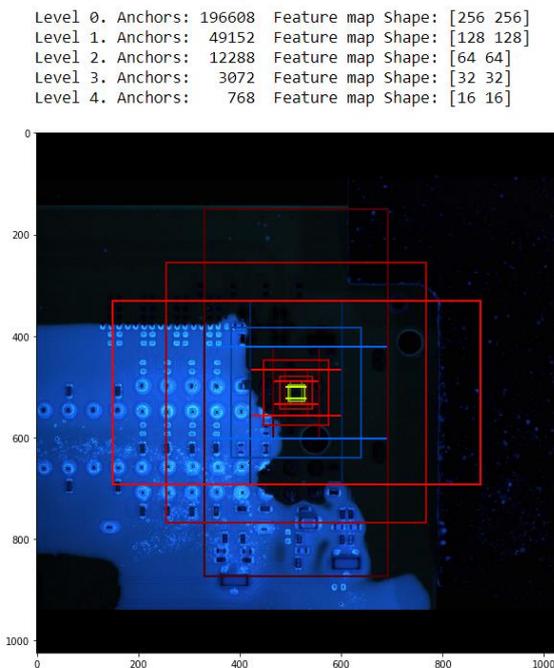


Figura 12. Anclas de una celda. Fuente: elaboración propia.

Se inició con el Generador de datos y se realizó la evaluación de todas las anclas, positivas (ver Figura 13), negativas (ver Figura 14) y neutrales (ver Figura 15).

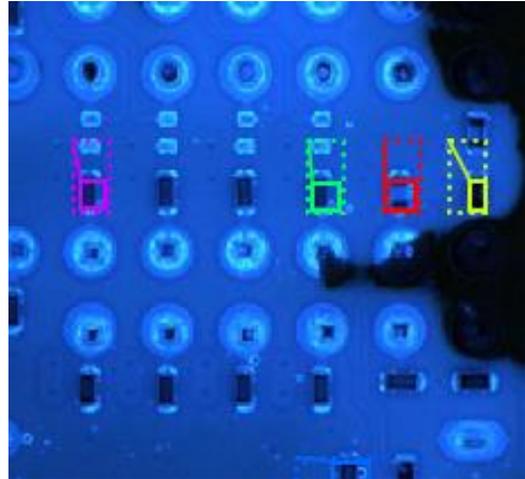


Figura 13. Anclas positivas. Fuente: elaboración propia.

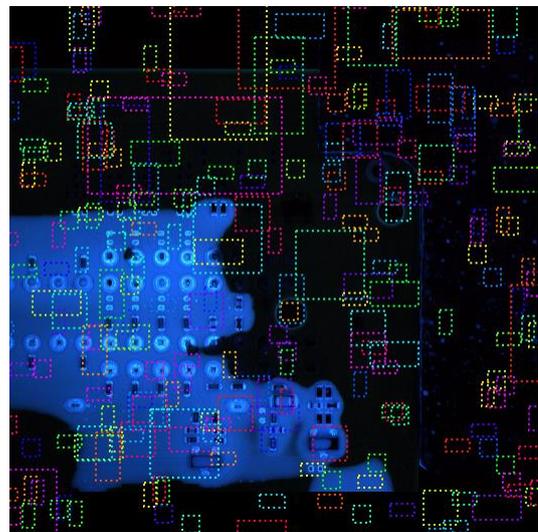


Figura 14. Anclas negativas. Fuente: elaboración propia.



Figura 15. Anclas neutras. Fuente: elaboración propia.

Posteriormente, se continuó con las Regiones de Interés (ROI), como se observa en la Figura 16. Aquí se muestran las diferentes ROIs, máscaras, cajas de detección y fondos de imagen, de imágenes muestra.

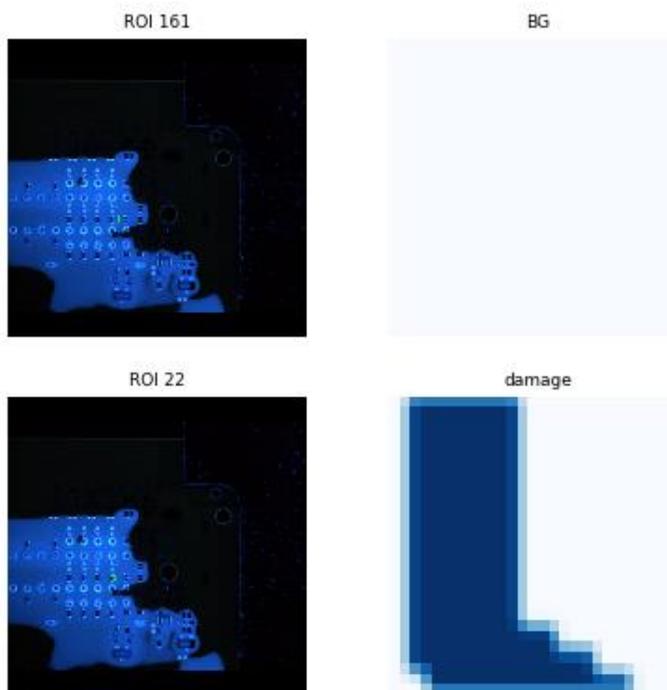


Figura 16. Detección de defectos y fondos en imágenes muestra. Fuente: elaboración propia.

Ejecución del modelo y predicciones en imágenes

Se realizó la ejecución del modelo en las imágenes del conjunto de pruebas/evaluación y se obtuvieron las predicciones entregadas por el modelo. Se muestra un ejemplo del resultado en la Figura 17, donde se despliegan valores que indican el porcentaje de similitud a defectos conocidos.

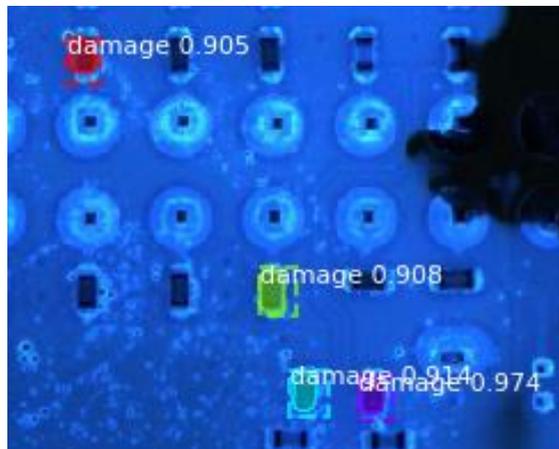


Figura 17. Resultado de la predicción. Fuente: elaboración propia.

Resultados

El análisis de todas las imágenes con el sistema inteligente en la primera fase de prueba, basado en pocas imágenes (cerca de 20), logró una detección superior al 50 % de defectos en las muestras de la predicción (ver Figura 18).

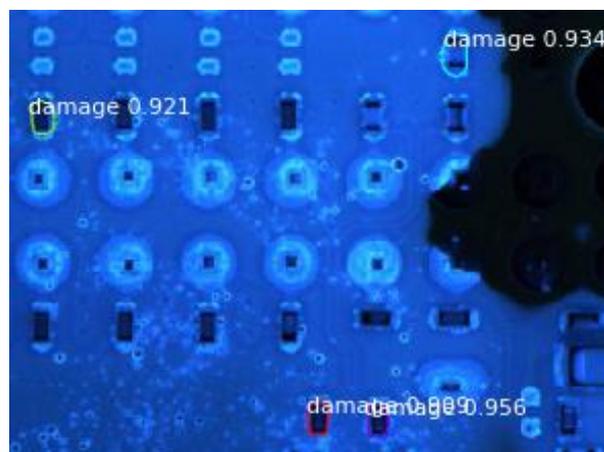


Figura 18. Predicción de muestras en primera fase. Fuente: elaboración propia.

A través del análisis del resultado anterior, se decidió realizar una segunda fase de pruebas, en la cual se incrementó la cantidad de muestras con defecto a 40. Esto permitió una mejor detección (cercana al 85 % como se muestra en la Figura 19). Si bien el resultado aumentó, se considera que existen todavía oportunidades de mejora por el gran universo de variaciones que se pueden tener en este proceso de *conformal coating*. Además, se observó que es fácil obtener tantas muestras con defecto y sus imágenes con defectos reales, ya que, en la industria, estas fallas tienden a ser eliminadas o reducidas.



Figura 19. Predicción de muestras en segunda fase. Fuente: elaboración propia.

Discusión y conclusiones

Durante el transcurso de este trabajo de investigación, se observaron muchas variables en los distintos tipos de defectos como: componentes ausentes, movidos. No obstante, únicamente se seleccionaron los defectos de *conformal coating* como burbujas. Las otras fallas fueron contenidas o descartadas, ya que era un universo muy grande de opciones disponibles que afectaban al desarrollo del trabajo de investigación; sin embargo, es fácilmente expandible a los demás tipos de fallas para su detección. Por el tipo de proceso que llevan las tarjetas electrónicas es muy difícil obtener muestras con defectos, ya que cuando se detecta alguna el proceso es detenido y se soluciona el problema. Por esa razón, se obtuvieron sólo unas 40 muestras disponibles del mismo modelo con defecto, para éstas se guardaron las imágenes que se obtuvieron

con el equipo AOI y se trató de obtener la mayor cantidad de variaciones en burbujas cerca de los componentes.

Como se observó durante las etapas de esta investigación, conforme se busque identificar algún determinado defecto, el *dataset* irá creciendo, para así obtener un mejor resultado en porcentaje de identificación de los defectos. Se pueden realizar mejoras en el proceso de creación de los *dataset* cuando se tienen más muestras, pues son muy benéficas para el futuro del trabajo de investigación, dado que permiten lograr una mejor detección de los defectos en la mayoría de las imágenes que se proporcionan para evaluación.

Además, se logró que el sistema inteligente identificara defectos de al menos un 85 % en las tablillas electrónicas con *conformal coating*. Este trabajo de investigación es muy extenso, ya que se requiere mayor tiempo y muestras para lograr un sistema casi perfecto y que pueda reemplazar la mayoría del tiempo a una persona en esta actividad. Se debe considerar que, en la industria, es muy difícil conseguir tantas tarjetas electrónicas con defectos para hacer un extenso *dataset*, puesto que se trataría de un gasto monetario extra y de material desperdiciado. Por ello, se trabajó con la mayor cantidad de tablillas electrónicas disponibles a fin de expandir el universo de detección y mejorar en las pruebas. Sin embargo, es importante considerar que, para obtener más muestras de tablillas electrónicas, se debe desechar el material defectuoso y eso es algo que se evita en cualquier industria.

Dentro de los métricos de las empresas se busca tener un número muy cercano a 0 en escapes de defectos a cliente final y una inspección sin defectos y falsos, ya que esto incrementa tiempo del proceso y reduce la cantidad de material producido. Las empresas actualmente destinan múltiples recursos para ejecutar sus funciones, pero muchas veces, no se aprovechan los recursos con los que ya cuentan. El propósito de este proyecto fue demostrar que se tienen los equipos, pero éstos podrían aprovecharse de una mejor manera o complementarse, a fin de

lograr un mejor desempeño de sus actividades, para crear una innovación en sus procesos y reducir una carga para el personal. Este desarrollo permitiría que solamente ocuparan su tiempo en las tarjetas que el equipo encuentra defectuosas casi con un 100 % de certeza y no en una variación del proceso.

Referencias

- Dutta A. y Zisserman A. (2019). The VIA Annotation Software for Images, Audio and Video. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, pp. 2276–2279. DOI: 3343031.3350535.
- Interactive Chaos (s.f.). Recuperado el 26 noviembre de 2020 de: <https://www.interactivechaos.com/manual/tutorial-de-machine-learning/arquitectura-de-redes-neuronales>
- Kaiming H., Georgia G., Piotr D. y Ross G. (2017). *Mask R-CNN*. Recuperado de: <https://arxiv.org/abs/1703.06870>
- Reyes Ochoa R. y López Aguayo S. (2020). *Representaciones del aprendizaje reutilizando los gradientes de la retropropagación*. Recuperado de: <https://arxiv.org/abs/2012.03188>
- UNED (2016). Recuperado el 18 noviembre de 2021: de http://portal.uned.es/EadmonGuiasWeb/htdocs/abrir_fichero/abrir_fichero.jsp?idGuia=86334
- Viso.ai (Vidushi Meel, 2021). Recuperado el 18 noviembre de 2021 de: <https://viso.ai/computer-vision/coco-dataset/>