

Modelo de aprendizaje profundo para la inspección de tarjetas de circuitos impresos

José Antonio Lara Chávez, Carlos Avilés Cruz, Miguel Magos Rivera

Universidad Autónoma Metropolitana, Unidad Azcapotzalco,
México

{jalch, mrm, caviles}@azc.uam.mx

Resumen. En este artículo, se presenta el desarrollo de un modelo específico de aprendizaje profundo para la detección de errores en el montaje de componentes electrónicos sobre una tarjeta de circuito impreso. El diseño de este modelo se compone por dos etapas: extracción de atributos y clasificación. La primera de estas se basa en redes neuronales convolucionales, utilizadas bajo un enfoque inspirado en la visión humana: primero se perciben estructuras gruesas, luego medianas y finalmente detalles finos. Por su parte, la clasificación se lleva a cabo mediante una red neuronal totalmente conectada. Para el entrenamiento y evaluación correspondiente, fue construida una base de datos propia, compuesta por 2376 imágenes que corresponden a seis tipos de tablillas. Finalmente, el modelo fue implementado en un equipo de cómputo, el cual recibe una imagen digital de la placa de circuito impreso que se requiere inspeccionar para, en tiempo real, indicar si presenta algún tipo de defecto.

Palabras Clave: Aprendizaje profundo, inspección, tarjeta de circuito impreso.

Deep Learning Model for Printed Circuit Board Inspection

Abstract. In this article, the development of a specific deep learning model for detecting errors in the assembly of electronic components on a printed circuit board is presented. The design of this model consists of two stages: feature extraction and classification. The first of these is based on convolutional neural networks, used under an approach inspired by human vision: first, coarse structures are perceived, then medium ones, and finally fine details. The classification is carried out using a fully connected neural network. For the corresponding training and evaluation, a proprietary database was constructed, consisting of 2376 images corresponding to six types of PCBs. Finally, the model was implemented on a computing device, which receives a digital image of the PCB to be inspected and, in real time, indicates if it presents any type of defect.

Keywords: Deep learning, inspection, printed circuit boards.

1. Introducción

La detección de defectos en productos ensamblados y/o empacados, es una tarea importante en muchos procesos industriales. Su realización en forma manual la convierte en una actividad con un alto grado de incertidumbre, además de complicada y tediosa para el personal encargado [1]. Lo anterior puede ocasionar un impacto negativo, tanto en la calidad de los productos, como en los costos de producción, lo que podría traducirse en pérdidas significativas para las empresas.

Por tal motivo, desde hace algunos años se han reportado diversos trabajos en los cuales se realizan inspecciones automáticas, utilizando generalmente, visión por computadora [2]. Las primeras metodologías empleadas en este tipo de tareas se basan principalmente en el procesamiento digital de imágenes, complementadas en ocasiones con técnicas de optimización, algoritmos genéticos, etc. [3].

La principal desventaja de este tipo de procedimientos, radica en la dependencia frente a condiciones externas, tales como: iluminación, posición de la cámara, enfoque, color de fondo, etc. [4].

Por su parte, estas deficiencias suelen no presentarse cuando la inspección visual automática se realiza mediante modelos computacionales creados a partir de aprendizaje automático y aprendizaje profundo [5].

Por tal motivo, en el último quinquenio se han estudiado diferentes métodos basados en este enfoque para realizar inspecciones autónomas [6]. Las máquinas de soporte vectorial, redes neuronales, redes neuronales convolucionales y redes neuronales recurrentes, han sido aplicadas en control de calidad a nivel industrial. En este artículo, se presenta el desarrollo de un modelo específico de aprendizaje profundo para la detección y clasificación de errores en el montaje de componentes electrónicos sobre una tarjeta de circuito impreso (PCB, por sus siglas en inglés).

Este modelo se divide en dos principales etapas: extracción de atributos y clasificación. La primera de estas, se basa en redes neuronales convolucionales (CNN, por sus siglas en inglés), utilizadas bajo el esquema grueso-mediano-fino [7]. Este enfoque está inspirado en la visión humana: primero se perciben estructuras gruesas, luego medianas y finalmente detalles finos. La etapa de clasificación, se llevó a cabo mediante una red neuronal totalmente conectada.

Posterior al diseño de dicho modelo, éste se entrenó y evaluó a partir de una base de datos propia, compuesta por 2376 imágenes. Una de estas placas no tiene ningún defecto, mientras que el resto presenta uno de cinco distintos errores de montaje, ver Tabla 1.

El modelo desarrollado, se implementó en un equipo de cómputo, al cual se le envía una imagen digital de la tablilla de circuito impreso para detectar si presenta defectos. Un diagrama de flujo de la metodología seguida se muestra en la Figura 1.

En la sección dos del artículo, se describen los trabajos relacionados con el mismo. Posteriormente se abordan la construcción de la base de datos utilizada y el diseño del modelo. Enseguida se detalla el entrenamiento y evaluación, para después describir su implementación. El artículo finaliza con las conclusiones y la bibliografía del mismo.

Tabla 1. Tipos de tarjetas y sus errores correspondientes.

Tipo de tarjeta	Defecto
Tarjeta correctamente ensamblada	No presenta defecto
Tarjeta con error A	Circuito integrado ausente
Tarjeta con error B	Circuito integrado montado en sentido opuesto
Tarjeta con error C	Diodo con la polaridad invertida
Tarjeta con error D	Circuito integrado en lugar erróneo
Tarjeta con error E	Resistor con valor erróneo

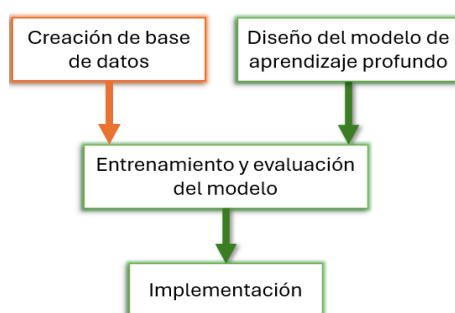


Fig. 1. Metodología utilizada.

2. Estado del arte

En la mayoría de los casos, los modelos de aprendizaje profundo empleados para inspecciones automáticas se componen por redes neuronales convolucionales [8]. Estos pueden generarse, modificando alguno ya existente, o bien, mediante la creación de uno específico para un trabajo en particular [9]. En Kulkarni et al. [10], utilizaron y evaluaron por separado, algoritmos de seguimiento de línea y agrupamiento, algoritmos de detección de objetos y redes neuronales convolucionales, para encontrar errores de fabricación en tapas de botellas de plástico. Un trabajo similar es presentado por Rajan et al. [11], quienes desarrollaron un sistema de identificación y clasificación de pernos dañados, utilizando redes neuronales convolucionales y una Raspberry Pi4. En Tran et al. [12], emplearon YOLO v4, como parte fundamental de un sistema de inspección física de redes eléctricas aéreas de alto voltaje. En la industria farmacéutica, Ficzero et al. [13], mediante ajustes a la versión 5 de YOLO, en combinación con técnicas de procesamiento de imágenes, desarrollaron un sistema identificador y clasificador de imperfectos en la superficie de píldoras médicas.

Las empresas de fabricación y ensamble de tarjetas de circuito impreso, son un ejemplo de área industrial en la cual la inspección manual resulta ser bastante complicada [14]. Lo anterior se debe a la gran cantidad, y tamaño reducido de los elementos que éstas contienen. Los tipos de errores que una PCB presentan, se pueden clasificar como: defectos superficiales, defectos en soldaduras y defectos de montaje.

Se consideran defectos superficiales a aquellos errores en las pistas de los circuitos impresos, tales como: cortocircuitos y circuitos abiertos, por mencionar solo algunos.

Un método de inspección automática para localizar este tipo de imperfecciones, es presentado en Kim et al. [15], en donde se desarrolló un modelo de aprendizaje profundo al que los autores llamaron: “Skip-Connected Autoencoder”. Este se basa en una CNN que codifica y decodifica una imagen de la placa a inspeccionar, posteriormente, el resultado se resta a la de una imagen sin defecto, para así dar con el error.

De forma similar, Park et al. [16] proponen un método para detectar desperfectos en la superficie de una PCB a través de un Auto-Encoder convolucional variable, en combinación con un modelo de aprendizaje profundo. Para este último, se implementaron, evaluaron y compararon resultados, utilizando por separado los modelos: ResNet 101, EfficientNet_b3 y DenseNet 121.

Así mismo, Austria et al [17], lograron encontrar y clasificar 10 tipos distintos de defectos superficiales, utilizando el modelo ResNet-101, como extractor de características, y Mask R-CNN como delimitador semántico.

Los defectos en soldaduras, están asociados, entre otros, a escasa cantidad de estaño o exceso de la misma en las terminales de los componentes, o a cortocircuitos generados por una mala técnica de soldadura. Kim et al. [18], desarrollaron un modelo de aprendizaje profundo al que denominaron “Dual-Stream Convolutional Networks”, el cual se caracteriza por tener dos CNN que trabajan en paralelo y que después se fusionan. Ambas redes reciben imágenes seccionadas de la tablilla a inspeccionar, detectando en cuál de éstas se presenta un error.

De forma semejante, Sezer et al. [19] presentan un método, en el cual, mediante una red convolucional 2D diseñada para distintas clases, se ubican elementos mal soldados. Esta red fue entrenada en tres ocasiones, utilizando respectivamente, las funciones de optimización: Adam, AdaMax y RMSprop, posteriormente, los resultados obtenidos en cada caso, fueron analizados para determinar la mejor función.

Por último, los defectos de montaje corresponden a un incorrecto ensamblaje de algún componente electrónico en una PCB, por ejemplo: elementos faltantes, mal ubicados o con polaridad invertida, entre otros. Debido a la dificultad que lo anterior representa, y con el objetivo de dar solución a este tipo de problemática, diversos autores han trabajado en metodologías enfocadas en reconocer, únicamente, los dispositivos presentes en estas placas.

En Lian et al. [20], mediante una modificación al modelo Mask R-CNN, se desarrolló un método para identificar, por su forma geométrica, piezas de montaje superficial. A su vez, con el objetivo de localizar la totalidad de circuitos integrados montados sobre una PCB, Gang et al. [21], implementaron y compararon los modelos: ResNet 56 y EfficientNet, para detectar y clasificar caracteres en microchips.

Por su parte, Zhang [22] mediante procesamiento de imágenes digitales, algoritmos de corrección de posicionamiento y YOLO, desarrolló un sistema capaz de distinguir y ubicar componentes presentes, ausentes o mal colocados en una placa de circuito impreso.

En Jia et al. [23], se presenta el desarrollo de un modelo de aprendizaje profundo al que llamaron “ICChaNet”, el cual es capaz de detectar los caracteres presentes en los circuitos integrados, y así determinar si éstos han sido colocados correctamente. Este modelo se basa en una red neuronal convolucional profunda, con una estructura LeNet-5 mejorada.

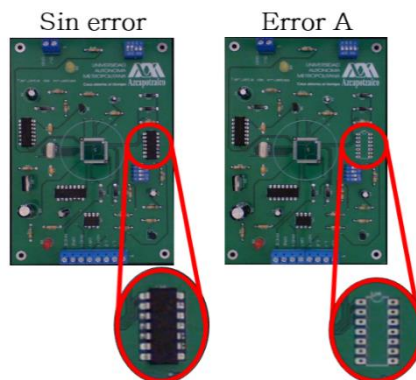


Fig. 2. Ejemplo de error en tarjeta de circuito impreso.

3. Base de datos

Para el entrenamiento y evaluación del modelo de aprendizaje profundo que se describe en este artículo, se creó una base de datos propia; se trata de un conjunto de imágenes digitales que corresponden a seis tipos de tarjetas de circuito impreso, ver Tabla 1. Estas placas miden 15 cm de largo por 10 de ancho, fueron fabricadas con fibra de vidrio, tienen una sola capa de cobre y no contienen componentes electrónicos de montaje superficial. La Figura 2, muestra una comparación entre la tablilla sin defecto y otra en la cual se tiene la ausencia de un circuito integrado. Cabe mencionar que, algunos errores presentes en estas tarjetas son relativamente fáciles de detectar, mientras que para otros la dificultad aumenta considerablemente. Por lo que, para aumentar la precisión en la clasificación de defectos para todas las clases, se decidió emplear una red neuronal profunda.

Inicialmente, se capturaron 36 fotografías con una dimensión de 1024 x 1280 píxeles, en formato de color RGB y una rotación de 10° entre ellas, lo anterior, para cada uno de los seis tipos de tarjetas fabricadas. Posteriormente, se generaron 10 nuevas muestras para cada una de las 36 imágenes de una clase de tablilla. Lo anterior, mediante la técnica de aumento de datos por variación de brillo, en un rango de 20% a 99%. De esta forma, se obtienen 396 imágenes por tipo de tarjeta, dando un total de 2376, las cuales, para facilitar el entrenamiento del modelo, fueron redimensionadas a 682 x 853 píxeles. Finalmente, la base de datos fue dividida aleatoriamente en: 80% para entrenar y 20% para evaluar.

4. Diseño del modelo de aprendizaje profundo

El diseño del modelo de aprendizaje profundo reportado en este artículo, se compone de dos partes: extracción de atributos y clasificación. La primera, consta de tres redes neuronales convolucionales, que trabajan bajo el esquema Grueso-Mediano-Fino [7]. Por otro lado, en la clasificación, se analizan los atributos extraídos, mediante una red neuronal totalmente conectada, para de esta forma obtener un resultado, ver Figura 3. Las características específicas del diseño de las dos partes que componen a la red

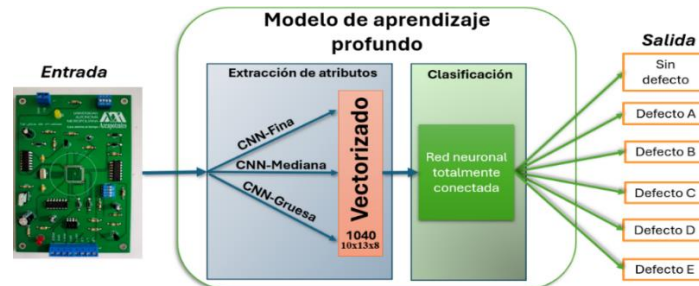


Fig. 3. Modelo de aprendizaje profundo.

neuronal profunda presentada en este artículo, son descritas en las siguientes subsecciones.

4.1. Extracción de atributos

La etapa de extracción de atributos se compone, por tres redes neuronales convolucionales, bajo el esquema Grueso-Mediano-Fino [7]. Este enfoque está inspirado en la visión humana: primero se perciben estructuras gruesas, luego medianas y finalmente detalles finos. Estas tres redes se encuentran conectadas en paralelo y se les llamó: CNN-Fina, CNN-Mediana y CNN-Gruesa.

CNN-Fina. Esta red cuenta con seis capas convolucionales 2D para la extracción de atributos. La primera de estas, también llamada de entrada, recibe una imagen de 682 x 853 píxeles en formato RGB y está compuesta por 32 filtros. La segunda y tercera, contienen 64, mientras que para la cuarta, quinta y sexta se emplearon respectivamente 32, 16 y 8 filtros. Cabe mencionar que, para la primera capa, la dimensión de los filtros es de 7x7, mientras que para el resto es de 5x5. Adicionalmente, al final de cada bloque de extracción de atributos, se utilizó una función de activación Luky Relu, con alfa de 0.2 y un Max Pool de 2x2, ver Figura 4.

CNN-Mediana. Esta red cuenta con cuatro capas convolucionales 2D para la extracción de atributos. La primera de estas, recibe una imagen de 682 x 853 píxeles en formato RGB y está compuesta por 64 filtros. La segunda contiene 32, mientras que para la tercera y cuarta, se emplearon respectivamente 16 y 8 filtros. Cabe mencionar que, para la primera capa, la dimensión de los filtros es de 7x7, mientras que para el resto es de 5x5. Adicionalmente, al final de cada bloque de extracción de atributos, se utilizó una función de activación Luky Relu, con alfa de 0.2 y un Max Pool de 2x2, ver Figura 5. Por último, para que la dimensionalidad a la salida coincida con la red fina, para las capas convolucionales tres y cuatro se utilizó un paso o Stride de 2x2, es decir, los filtros utilizados para la convolución se desplazaron dos píxeles a la vez.

CNN-Gruesa. Esta red cuenta con tres capas convolucionales 2D para la extracción de atributos. La primera de estas, recibe una imagen de 682 x 853 píxeles en formato RGB y está compuesta por 64 filtros de 7x7. Por su parte, la segunda y tercera contienen 8 filtros de 5x5, ver Figura 6.

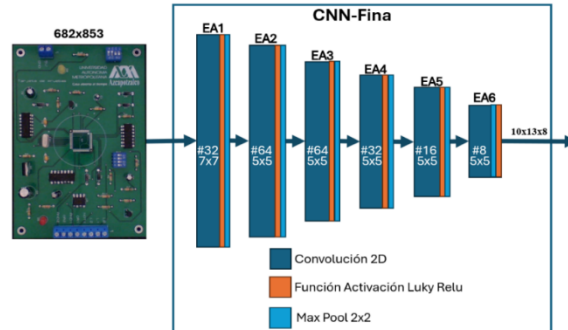


Fig. 4. Extracción de atributos: CNN-Fina.

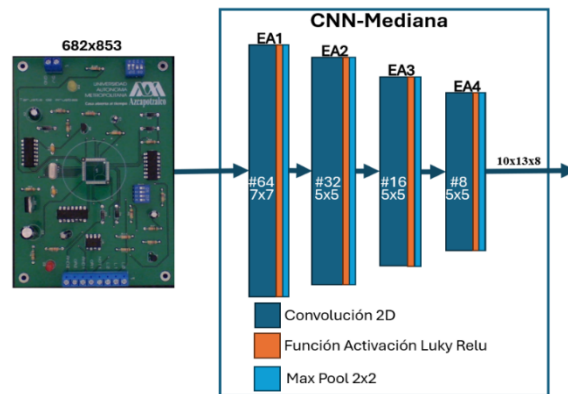


Fig. 5. Extracción de atributos: CNN-Mediana.

Adicionalmente, al final de cada bloque de extracción de atributos, se utilizó una función de activación LUKY ReLU, con alfa de 0.2 y un Max Pool de 2x2, Por último, para que la dimensionalidad a la salida coincida con la red fina y la mediana, en las tres capas convolucionales se empleó un Stride de 2x2.

Total de atributos extraídos. Las características extraídas por cada red, fueron utilizadas de acuerdo a las siguientes proporciones: 50% CNN-Fina, 30% CNN-Mediana y 20% CNN-Gruesa, ver Figura 7. De esta manera se genera un vector de 1040 elementos, que corresponde a la totalidad de los atributos extraídos.

4.2. Clasificación

La etapa de clasificación, mediante una red neuronal, tal y como se muestra en la Figura 8, analiza los atributos extraídos. Esta red está compuesta por tres capas totalmente conectadas con 520, 130 y 65 neuronas, respectivamente, además de una función de activación Selu. Adicionalmente, se tiene una capa de salida con seis neuronas y una función de activación SoftMax. Lo anterior, se debe a que se requiere, como resultado, una clasificación para este número de clases.

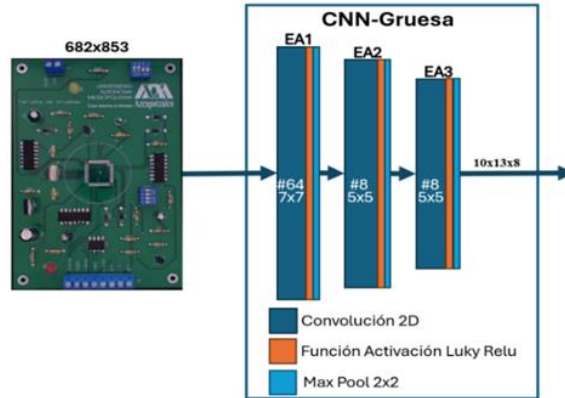


Fig. 1. Extracción de atributos: CNN-Gruesa.

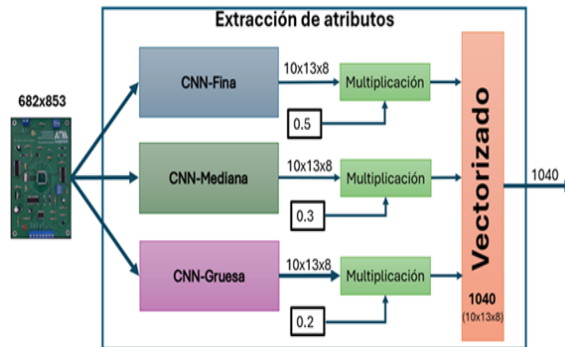


Fig. 7. Total de atributos extraídos.

5. Entrenamiento y evaluación

En esta sección se describe el equipo, los programas de cómputo y las métricas utilizadas para entrenar y evaluar el modelo de aprendizaje profundo, cuyo diseño ha sido descrito. Así mismo, se detalla el procedimiento y resultados obtenidos al realizar estas dos tareas.

5.1. Equipo y programas de cómputo utilizados

El entrenamiento y evaluación del modelo de aprendizaje profundo, fue realizado en una computadora marca Dell Alienware Aurora R13 Gaming Desktop, equipada con un procesador Intel Core i9 de doceava generación, 128 GB de memoria RAM, 2 TB SSD y 2 TB HDD. Adicionalmente, se empleó una GPU: Nvidia Geforce Rtx 3090 con 16 núcleos, a 5.2 GHz y memoria RAM de 24 GB GDDR6X. El sistema operativo utilizado fue Ubuntu 22.04 LTS de 64-bits. Como lenguaje de programación, se utilizó Python 3.9, complementado con las bibliotecas: Tensorflow y Keras versión 2.13, scikit-learn, Matplotlib y NumPy.

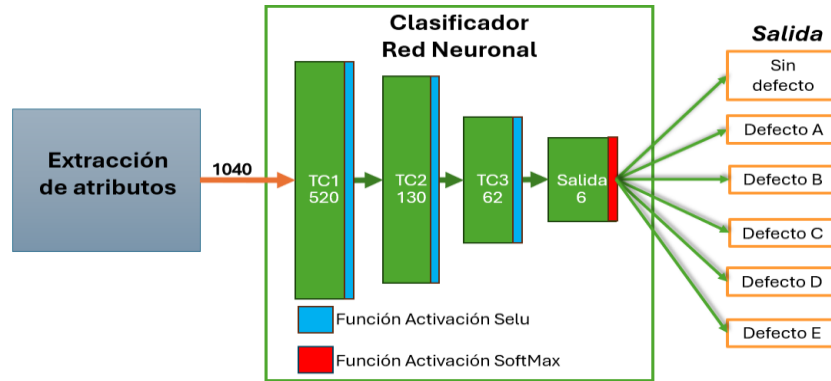


Fig. 8. Clasificación.

5.2. Métricas de entrenamiento y evaluación

La función de pérdida de tipo entropía cruzada categórica (categorical cross-entropy), fue utilizada durante el entrenamiento y la evaluación, para determinar el error del modelo, ecuación 1:

$$f_{p_{\text{categorical cross-entropy}}} = -\frac{1}{N} \sum_{i=0}^N \sum_{c=0}^N y * \log(\hat{y}) , \quad (1)$$

donde:

N = Número de ejemplos en el conjunto de datos.

c = Número de clases.

y = Valor binario cuyo valor es 1 si i pertenece a la clase c , y 0 si no.

\hat{y} = Probabilidad de que i pertenezca a la clase c .

Por su parte, para la evaluación fueron utilizadas las métricas: exactitud, precisión, sensibilidad y F1 score [24]. Estas, básicamente dependen de la cantidad de veces que se acertó o no en la predicción de la clasificación, de tal forma que dan información sobre el rendimiento del modelo y determinan si este es o no funcional.

Exactitud. Permite evaluar el rendimiento general de un modelo de clasificación. Se define como el número de predicciones correctas dividido entre el número total de intentos, ecuación 2:

$$\text{Exactitud} = \frac{\text{Verdaderos positivos} + \text{Verdaderos negativos}}{\text{Total de predicciones}}. \quad (2)$$

Precisión. Representa la cantidad de predicciones correctas del total de intentos, para una clase en particular, ecuación 3:

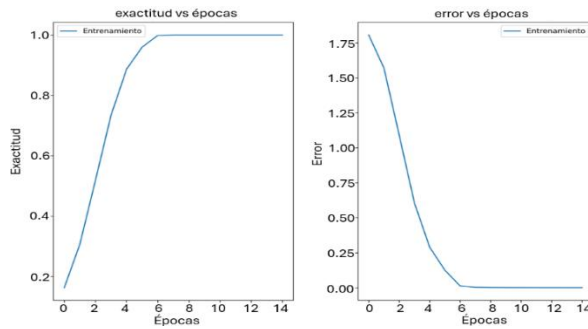


Fig. 9. Gráficas obtenidas durante el entrenamiento del modelo de aprendizaje profundo.

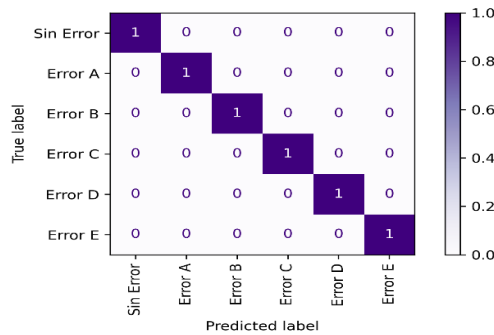


Fig. 10. Matriz de confusión correspondiente a la evaluación del modelo de aprendizaje profundo.

$$Precision = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ positivos}. \quad (3)$$

Sensibilidad. Se refiere a la capacidad del modelo para identificar la cantidad de casos positivos correctamente clasificados, para una clase en particular, ecuación 4:

$$Sensibilidad = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos}. \quad (4)$$

F1 score. Se utiliza comúnmente para conocer el equilibrio entre la precisión y la sensibilidad, para una clase en particular, en un modelo computacional de clasificación. Un valor bajo de F1 score puede indicar falta de calidad en las predicciones, este se calcula mediante la ecuación 5:

$$F1_{score} = \frac{2(Precision * Recall)}{Precision + Recall}. \quad (5)$$

Adicionalmente, para completar la evaluación del modelo, fue generada la matriz de confusión correspondiente. Esta herramienta, es una representación de los aciertos obtenidos en la clasificación de un conjunto de datos, para los cuales se conoce a que

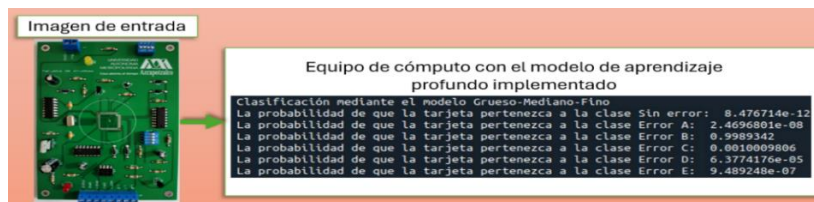


Fig. 11. Implementación del modelo de aprendizaje profundo.

Tabla 2. Resultados de las pruebas de implementación del modelo de aprendizaje profundo.

Tipo de Tarjeta	Sin Error	Error A	Error B	Error C	Error D	Error E
Sin Error	10	0	0	0	0	0
Error A	0	10	0	0	0	0
Error B	0	0	10	0	0	0
Error C	0	0	0	10	1	0
Error D	0	0	0	0	9	0
Error E	0	0	0	0	0	10

clase pertenecen. Las filas de esta matriz representan las clases reales, mientras que las columnas corresponden a las clases predichas, cada celda contiene el conteo correspondiente a la clasificación realizada por el modelo.

5.3. Entrenamiento

Para su entrenamiento, el modelo de aprendizaje profundo fue compilado utilizando una función de pérdida de tipo entropía cruzada categórica, un optimizador Adam, con una tasa de aprendizaje de 0.0005.

Posteriormente, el modelo se entrenó durante 15 épocas con un tamaño de lote (Batch size) de 10 muestras, utilizando el 80% de la base de datos descrita anteriormente.

De manera que, de las 2376 imágenes, 1900 fueron utilizadas para el entrenamiento. Al término de esta tarea, el error fue de cero, con una exactitud del 100%. Las gráficas de la Figura 9 muestran el comportamiento de las métricas por época.

5.4. Evaluación

La evaluación del modelo de aprendizaje profundo fue realizada utilizando el 20% de la base de datos descrita anteriormente. De manera que, de las 2376 imágenes, 476 fueron utilizadas para esta tarea. Al término de este proceso de valoración, los resultados corroboraron lo conseguido en el entrenamiento, ya que se obtuvo un error de 0.00034, con una exactitud del 100%. Los valores de precisión, sensibilidad y F1 score obtenidos para cada clase fueron del 100%. Por último, en la Figura 10 se muestra la matriz de confusión normalizada, obtenida a partir de la evaluación de dicho modelo.

6. Implementación

Posterior al entrenamiento y evaluación del modelo de aprendizaje profundo, descrito en las secciones anteriores, se realizó su implementación en un equipo de cómputo. El modelo recibe una imagen de la tarjeta de circuito impreso, la cual previamente se reduce de 1024x1280 a 682x853 píxeles. Posteriormente, se realiza la predicción correspondiente y, en tiempo real, se indica la probabilidad de que la tablilla presenté, o no, algún tipo de defecto. La Figura 11 muestra la información que el equipo de cómputo despliega en pantalla al realizar la predicción.

En pruebas realizadas con 10 imágenes por cada tipo de tarjeta de circuito impreso, se encontró una precisión en la clasificación del 90% para la clase: Error D, mientras que para todas las demás fue del 100%. La Tabla 2 muestra los resultados obtenidos en este procedimiento, las filas representan las clases reales, mientras que las columnas corresponden a las clases predichas y cada celda contiene el conteo correspondiente a la clasificación realizada por el modelo.

7. Conclusiones

El diseño, entrenamiento, evaluación e implementación de un nuevo modelo de aprendizaje profundo, basado en redes neuronales convolucionales, aplicado en la inspección de tarjetas de circuitos impresos, fue descrito en este artículo. Este modelo, el cual es exclusivo a un único diseño de PCB, está compuesto por dos partes: extracción de atributos y clasificación. El modelo es capaz de determinar si una placa presenta, o no, alguno de los 5 defectos propuestos.

Los resultados obtenidos en el entrenamiento, muestran un error de cero y una exactitud del 100%, en la clasificación de las tarjetas. Por su parte, los valores que estas métricas alcanzaron en la evaluación realizada, fueron prácticamente iguales, corroborando el buen rendimiento del modelo. Por lo que, en comparación con otros métodos, se tiene una eficacia similar y en ocasiones mayor. Por su parte, el esquema Grueso-Mediano-Fino, empleado en la extracción de atributos, funcionó correctamente en el problema planteado en este artículo.

7.1. Trabajo a futuro

Como trabajo a futuro, se plantea la modificación del modelo de aprendizaje profundo presentado en esta publicación académica, para añadir una etapa de segmentación, la cual sea capaz de identificar la totalidad de componentes presentes en la tarjeta de circuito impreso y con esto mejorar la inspección. Adicionalmente, se buscará aplicar el modelo en la identificación de componentes electrónicos de tamaños y naturaleza distintos a los presentados en este artículo.

Referencias

1. Pastor-López, I., Santos, I., Santamaría-Ibirika, A., Salazar, M., De-la-Peña-Sordo, J., Bringas, P.G.: Machine-learning-based surface defect detection and categorisation in high-

- precision foundry. In: 7th IEEE Conference on Industrial Electronics and Applications pp. 1359–1364 (2012). DOI: 10.1109/ICIEA.2012.6360934.
2. Neethu, N., Anoop, B.: Role of computer vision in automatic inspection systems. *International Journal of Computer Applications*, vol. 123, no. 13, pp. 28–31 (2015)
 3. Wu, C., Wang, D., Ip, A., Wang, D.W., Chan, C., Wang, H.: A particle swarm optimization approach for components placement inspection on printed circuit boards. *Journal of Intelligent Manufacturing*, vol. 20, no. 5, pp. 535–549 (2009). DOI: 10.1007/s10845-008-0140-2.
 4. Tornero-Montserrat, J., Armesto, L., Mora-Aguilar, M.C., Montés-Sánchez, N., Herraiz-Martínez, A., Asensio, J.M.: Detección de defectos en carrocerías de vehículos basado en visión artificial: Diseño e implantación. *Revista Iberoamericana de Automática e Informática Industrial*, vol. 9, no. 1, pp. 93–104 (2012). DOI: 10.1016/j.riai.2011.11.010.
 5. Kumar, R., Rani, P.: Machine Learning strategies in real-world engineering applications: A comprehensive survey. In: *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 16, pp. 131–140 (2024)
 6. Aparicio-Pico, L.E., Amaya-Marroquín, O.J., Devia-Lozano, P.A: Aplicación de Deep Learning para la identificación de defectos superficiales utilizados en control de calidad de manufactura y producción industrial: Una revisión de la literatura. *Ingeniería*, vol. 28, no. 1 (2023). DOI: 10.14483/23448393.18934.
 7. Avilés-Cruz, C., Ferreyra-Ramírez, A., Zúñiga-López, A., Cortéz, J.: Coarse-Fine convolutional Deep-Learning strategy for human activity recognition. *Sensors*, vol. 19, no. 7, pp. 1556 (2019). DOI: 10.3390/s19071556.
 8. Yang, J., Li, S., Wang, Z., Dong, H., Wang, J., Tang, S.: Using Deep Learning to detect defects in manufacturing: A comprehensive survey and current challenges. *Materials*, vol. 13, no. 24, pp. 5755 (2020). DOI: 10.3390/ma13245755.
 9. Janiesch, C., Zschech, P., Heinrich, K.: Machine Learning and Deep Learning. *Electronic Markets*, vol. 31, no. 3, pp. 685–695 (2021). DOI: 10.1007/s12525-021-00475.
 10. Kulkarni, R., Kulkarni, S., Dabhane, S., Lele, N., Paswan, R.S.: An automated computer vision based system for bottle cap fitting inspection. In: 2019 Twelfth International Conference on Contemporary Computing, pp. 1–5 (2019). DOI: 10.1109/IC3.2019.8844942.
 11. Rajan, A.J., Jayakrishna, K., Vignesh, T., Chandradass, J., Kannan, T.T.M.: Development of computer vision for inspection of bolt using convolutional neural network. *Materials Today: Proceedings*, vol. 45, no. 7, pp. 6931–6935 (2021). DOI: 10.1016/j.matpr.2021.01.372.
 12. Tran, H.T., Tran, M.Q., Tran, Q.H., Pham, V.C.: A computer vision system for power transmission line inspection robot. In: 2021 International Conference on System Science and Engineering, pp. 289–294 (2021). DOI: 10.1109/ICSSE52999.2021.9538440.
 13. Ficzer, M., Mészáros, L.A., Kállai-Szabó, N., Kovács, A., Antal, I., Nagy, Z.K., and Galata, D.L.: Real-time coating thickness measurement and defect recognition of film coated tablets with machine vision and Deep Learning. In: *International Journal of Pharmaceutics*, vol. 623, no. 1, pp. 121957 (2022). DOI: 10.1016/j.ijpharm.2022.121957.
 14. Aggarwal, N., Deshwal, M., Samant, P.: A survey on automatic printed circuit board defect detection techniques. In: 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 853–856, India (2022). DOI: 10.1109/ICACITE53722.2022.9823872.
 15. Kim, J., Ko, J., Choi, H., Kim, H.: Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors*, vol. 21, no. 15, pp. 4968 (2021). DOI: 10.3390/s21154968.

16. Park, J., Shin, S.Y.: Printed circuit board defect detection using generative Deep Learning model. In: 13th International Conference on Information and Communication Technology Convergence, pp. 463–466 (2022). DOI: 10.1109/ICTC55196.2022.9952939.
17. Austria, Y.D., Fajardo, A.C.: Defect detection and classification in printed circuit boards using convolutional neural networks. In: 2nd International Conference on Edge Computing and Applications, pp. 1498–1504 (2023). DOI: 10.1109/ICECAA58104.2023.10212195.
18. Kim, Y.G., Park, T.H.: SMT Assembly inspection using dual-stream convolutional networks and two solder regions. *Applied Sciences*, vol. 10, no. 13, pp. 4598 (2020). DOI: 10.3390/app10134598.
19. Sezer, A., Altan, A.: Detection of solder paste defects with an optimization-based Deep Learning model using image processing techniques. *Soldering and Surface Mount Technology*, vol. 33, no. 5, pp. 291–298 (2021). DOI: 10.1108/SSMT-04-2021-0013.
20. Lian, J., Wang, L., Liu, T., Ding, X., Yu, Z.: Automatic visual inspection for printed circuit board via novel Mask R-CNN in smart city applications. *Sustainable Energy Technologies and Assessments*, vol. 44, no. 1, pp. 101032 (2021). DOI: 10.1016/j.seta. 2021.101032.
21. Gang, S., Fabrice, N., Chung, D., Lee, J.: Character recognition of components mounted on printed circuit board using Deep Learning. *Sensors*, vol. 21, no. 9, pp. 2921 (2021). DOI: 10.3390/s21092921.
22. Zhang, K.: Using deep learning to automatic inspection system of printed circuit board in manufacturing industry under the internet of things. *Computer Science and Information Systems*, vol. 20, no. 2, pp. 723–741 (2023). DOI: 10.2298/CSIS220718020.
23. Jia, X., Liu, Z.: Character identification for integrated circuit components on printed circuit boards using Deep Learning. *Journal of Electrical Engineering and Technology*, vol. 17, no. 1, pp. 601–616 (2022). DOI: /10.1007/s42835-021-00885-4.
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Duchesnay, É.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, no. 1, pp. 2825–2830 (2011)