



POSGRADOS

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-30-NO.507-2019

OPCIÓN DE TITULACIÓN:

ARTÍCULOS PROFESIONALES DE ALTO NIVEL

TEMA:

RECONOCIMIENTO DE COLOR EN PIEZAS
DE USO INDUSTRIAL POR MEDIO DE REDES
NEURONALES ARTIFICIALES USANDO UN
ROBOT SCARA

AUTORES:

FRANCISCO XAVIER PESÁNTEZ ZÚÑIGA
DARÍO BERNARDO PULLA IÑIGUEZ

DIRECTOR:

CHRISTIAN RAÚL SALAMEA PALACIOS

CUENCA – ECUADOR
2025

Autores:



Francisco Xavier Pesantez Zúñiga

Ingeniero en Mecatrónica.

Candidato a Magíster en Electrónica y Automatización,
Mención en Informática Industrial por la Universidad
Politécnica Salesiana - Sede Cuenca.

fpesantezz@est.ups.edu.ec



Darío Bernardo Pulla Iñiguez

Ingeniero en Mecatrónica.

Candidato a Magíster en Electrónica y Automatización,
Mención en Informática Industrial por la Universidad
Politécnica Salesiana - Sede Cuenca.

dpullai@est.ups.edu.ec

Dirigido por:



Christian Raúl Salmea Palacios

Ingeniero Electrónico.

Máster en Diseño, Gestión y Dirección de Proyectos.

Doctor en Ingeniería de Sistemas Electrónicos.

csalamea@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2025 Universidad Politécnica Salesiana.

CUENCA – ECUADOR – SUDAMÉRICA

FRANCISCO XAVIER PESANTEZ ZÚÑIGA

DARÍO BERNARDO PULLA IÑIGUEZ

Reconocimiento de color en piezas de uso industrial por medio de redes neuronales artificiales usando un robot SCARA

Reconocimiento de color en piezas de uso industrial por medio de redes neuronales artificiales usando un robot SCARA

F. Pesántez, D. Pulla, and C. Salamea

Universidad Politécnica Salesiana - Ecuador

Resumen

Dentro de la inteligencia artificial y la visión computacional, el reconocimiento de color constituye una técnica relevante que puede aplicarse en diversos procesos de automatización industrial. Por otra parte, los robots SCARA (Selective Compliance Assembly Robot Arm) se han posicionado como herramientas clave en entornos de manufactura por su precisión y velocidad en tareas de manipulación. En el presente trabajo, se explora la integración de estas dos áreas mediante el desarrollo de un sistema de clasificación de piezas industriales por color, utilizando redes neuronales convolucionales (CNN) y un robot SCARA modular. Esta integración permite automatizar tareas que tradicionalmente requieren inspección humana, mejorando la eficiencia y reduciendo errores. El sistema propuesto, evaluado en un conjunto de datos de piezas de colores amarillo, azul, verde y otros, alcanzó una precisión global del 91.3 %. El color que mejor se reconoció fue el azul con un 96 %, mientras que la clase "Otros" presentó la menor precisión con un 85 %. Estos resultados demuestran la viabilidad de la solución para optimizar procesos de selección y clasificación en la industria, particularmente en pequeñas y medianas empresas (Pymes).

Abstract

Within artificial intelligence and computer vision, color recognition constitutes a relevant technique that can be applied to various industrial automation processes. On the other hand, SCARA (Selective Compliance Assembly Robot Arm) robots have become key tools in manufacturing environments due to their precision and speed in handling tasks. This study explores the integration of these two areas through the development of a system for classifying industrial parts by color, using convolutional neural networks (CNNs) and a modular SCARA robot. This integration allows for the automation of tasks that traditionally require human inspection, improving efficiency and reducing errors. The proposed system, evaluated on a dataset of parts in yellow, blue, green, and other colors, achieved an overall accuracy of 91.3%. The most accurately recognized color was blue, with 96%, while the "Others" class showed the lowest accuracy at 85%. These results demonstrate the feasibility of the solution for optimizing selection and classification processes in industry, particularly in small and medium-sized enterprises (SMEs).

Palabras clave— Robótica Industrial, SCARA, Redes Neuronales Convolucionales, Clasificación de Color, Automatización Industrial.

1 Introducción

1 La transformación de los procesos industriales ha sido impulsada por la incorporación de tecnologías
2 como la visión por computadora, el aprendizaje profundo y la robótica colaborativa (Groover, 2008;
3 Kagermann et al., 2013; Monostori, 2014). Estos avances han transformado significativamente la
4 forma en que las empresas optimizan sus operaciones, permitiendo automatizar tareas repetitivas,
5 mejorar la eficiencia y reducir los errores humanos. En particular, la integración de sistemas
6 inteligentes ha facilitado la toma de decisiones en tiempo real mediante el análisis de datos visuales y
7 sensoriales, incrementando así la productividad y la trazabilidad en entornos industriales dinámicos
8 (Kapusi et al., 2022).

9 Una de las aplicaciones más críticas en las líneas de producción es la clasificación de objetos por
10 características visuales como el color. Esta tarea, antes dependiente de la percepción humana, puede
11 realizarse hoy con mayor precisión, velocidad y consistencia mediante técnicas de visión artificial
12 combinadas con inteligencia artificial (Richards, 2012; Song, 2025). La identificación cromática
13 resulta esencial en industrias como la alimentaria, farmacéutica, electrónica o de ensamblaje, donde
14 errores en la clasificación pueden comprometer la calidad del producto final o interrumpir el flujo
15 de trabajo.

16 El reconocimiento de color se ha beneficiado del desarrollo de arquitecturas avanzadas como
17 las redes neuronales convolucionales (CNN), que han demostrado un rendimiento sobresaliente
18 en tareas de clasificación visual, superando a los métodos tradicionales en precisión, robustez y
19 adaptabilidad (LeCun et al., 2015; Goodfellow et al., 2016). Estas redes han sido aplicadas con éxito
20 en entornos controlados para el reconocimiento de objetos, detección de defectos o clasificación de
21 productos. Sin embargo, su aplicación práctica en entornos industriales reales suele estar limitada
22 por el requerimiento de hardware especializado o de alto rendimiento computacional (Swaminathan
23 et al., 2024).

24 Adicionalmente, la eficacia de los modelos CNN puede verse afectada por variaciones en la
25 iluminación o el contexto visual, lo cual ha motivado investigaciones sobre espacios de color
26 alternativos, como HSV, que han demostrado mejorar la estabilidad del reconocimiento cromático
27 bajo condiciones no controladas (Xian et al., 2023). A esto se suma el uso de técnicas de aumento
28 de datos como rotaciones, desplazamientos y escalado, que permiten robustecer los modelos frente
29 a muestras no vistas durante el entrenamiento (Shorten and Khoshgoftaar, 2019).

30 Paralelamente, los robots tipo SCARA (Selective Compliance Assembly Robot Arm) se han
31 consolidado como una solución versátil y eficiente para tareas de ensamblaje y manipulación
32 horizontal debido a su velocidad, precisión y facilidad de integración en entornos industriales
33 (Koren, 1985). El desarrollo de diseños modulares de código abierto ha permitido la creación
34 de soluciones funcionales de bajo costo orientadas a entornos educativos y de prototipado
35 industrial (HowToMechatronics, 2023; Zhang et al., 2024). Estos avances abren la posibilidad de
36 integrar visión artificial y clasificación automática en brazos robóticos compactos, replicables y
37 económicamente accesibles.

38 En este contexto, el presente trabajo propone un sistema de clasificación automática de piezas
39 industriales basado en el reconocimiento de color mediante redes neuronales convolucionales y
40 un robot SCARA modular. La arquitectura del sistema fue diseñada para operar con recursos
41 computacionales accesibles, distribuyendo la carga entre un microcontrolador embebido y un
42 computador externo, y está orientada a su aplicación en entornos de producción de pequeña y
43 mediana escala. El objetivo principal es validar la viabilidad técnica y operativa de esta integración,
44 analizando su precisión, estabilidad y adaptabilidad en condiciones simuladas que representan
45 escenarios reales de manufactura.

46 El resto del artículo está organizado de la siguiente manera. En la Sección 2 se presenta el
47 estado del arte relacionado con el uso de visión artificial, redes neuronales convolucionales y robots
48 SCARA en tareas de clasificación industrial, lo que permite contextualizar y justificar la propuesta
49 desarrollada. En la Sección 3 se detallan los materiales y métodos utilizados en el desarrollo
50 del sistema propuesto, incluyendo la construcción del robot, el sistema de visión artificial y la
51 arquitectura de red neuronal. La Sección 4 presenta los resultados experimentales, acompañados
52 de un análisis comparativo del desempeño de los modelos evaluados. Finalmente, en la Sección 5
53 se discuten las principales conclusiones del estudio y se proponen líneas futuras de investigación
54 orientadas a mejorar la aplicabilidad del sistema en entornos industriales reales.

55 2 Estado del Arte

56 Diversas soluciones han sido desarrolladas para abordar la clasificación automática por color en
57 entornos industriales, utilizando sensores ópticos o cámaras para realizar tareas de identificación
58 cromática. En (Kesici et al., 2018) se analiza un sistema de pintura automatizado que emplea
59 un robot SCARA con visión artificial, demostrando la viabilidad de identificar múltiples colores
60 bajo condiciones de iluminación variables mediante procesamiento de imágenes. Por otro lado, en
61 (Nkomo and Collier, 2012) se describe un sistema de bajo costo basado en sensores TCS230 y
62 microcontroladores PSoC, enfocado en la clasificación cromática en tiempo real. Aunque ambos
63 trabajos evidencian soluciones funcionales y accesibles, presentan limitaciones importantes en
64 cuanto a capacidad de adaptación frente a variaciones cromáticas, condiciones de iluminación
65 o cambios geométricos, debido a la ausencia de técnicas de aprendizaje profundo.

66 En los últimos años, las redes neuronales convolucionales (CNN) han demostrado un desempeño
67 sobresaliente en tareas de clasificación visual, superando a los métodos tradicionales por su precisión
68 y capacidad de generalización (LeCun et al., 2015). Estas arquitecturas han sido aplicadas con éxito
69 al reconocimiento de objetos y defectos, incluso bajo condiciones de iluminación no controladas
70 o presencia de ruido. No obstante, muchas implementaciones requieren recursos computacionales
71 significativos o se desarrollan en entornos de laboratorio, lo que restringe su adopción en fábricas
72 de pequeña escala. Algunos enfoques recientes proponen estrategias más ligeras y adaptadas a
73 entornos reales, como el uso del espacio de color HSV para mejorar la robustez frente a variaciones
74 lumínicas en tareas de visión por computadora (Kang et al., 2021).

75 La integración de CNN con brazos robóticos ha sido explorada recientemente con aplicaciones
76 orientadas a la manipulación inteligente. Por ejemplo, (?) reporta una solución basada en redes
77 profundas sobre un robot SCARA industrial para clasificación en tiempo real, logrando mejoras
78 relevantes en precisión y velocidad. De manera complementaria, (?) plantea una arquitectura
79 modular con simulación digital para SCARA, diseñada para facilitar su implementación en
80 escenarios industriales. Además, (Tinoco et al., 2025) presenta una evaluación comparativa de
81 controladores aplicados a manipuladores SCARA de bajo costo, demostrando su viabilidad técnica
82 y operativa en entornos agrícolas, con potencial de extrapolación a procesos industriales ligeros.

83 Los robots del tipo SCARA (Selective Compliance Assembly Robot Arm) se utilizan
84 ampliamente en tareas de ensamblaje, manipulación y transporte horizontal de componentes debido
85 a su velocidad, precisión y facilidad de integración (Koren, 1985). Su arquitectura cinemática,
86 compuesta por dos articulaciones rotacionales y un eje lineal, ofrece un equilibrio adecuado entre
87 rigidez y flexibilidad. Gracias al auge de plataformas de código abierto, estos robots se han
88 convertido en una opción atractiva tanto en contextos académicos como industriales, especialmente
89 para desarrollos de bajo costo.

90 A diferencia de los enfoques descritos en (Kesici et al., 2018; Nkomo and Collier, 2012),
91 que emplean sensores discretos o sistemas ópticos convencionales, la solución propuesta en este
92 trabajo combina una red neuronal convolucional entrenada específicamente para clasificación
93 cromática con un robot SCARA modular, mejorando la robustez ante variaciones de iluminación y
94 forma. Asimismo, en contraste con desarrollos que requieren hardware especializado o condiciones
95 controladas para su implementación, el sistema planteado se basa en una arquitectura accesible
96 y replicable, orientada a pequeñas y medianas empresas con recursos limitados. Esta propuesta
97 responde directamente a las limitaciones observadas en la literatura y representa un aporte práctico
98 al avance de la automatización industrial inteligente.

99 3 Materiales y Métodos

100 3.1 Construcción del Robot SCARA

101 El robot SCARA mostrado en la Figura 1, utilizado en este proyecto, se basa en un diseño modular
102 de código abierto desarrollado por HowToMechatronics (HowToMechatronics, 2023), lo que permite
103 una construcción accesible y personalizable (Zhang et al., 2024). Cuenta con cuatro grados de
104 libertad: dos rotacionales en el plano XY, un movimiento lineal en el eje Z y un rotacional adicional
105 para el control del efector final. Esta configuración facilita tareas de clasificación y manipulación
106 de piezas en planos horizontales con precisión.

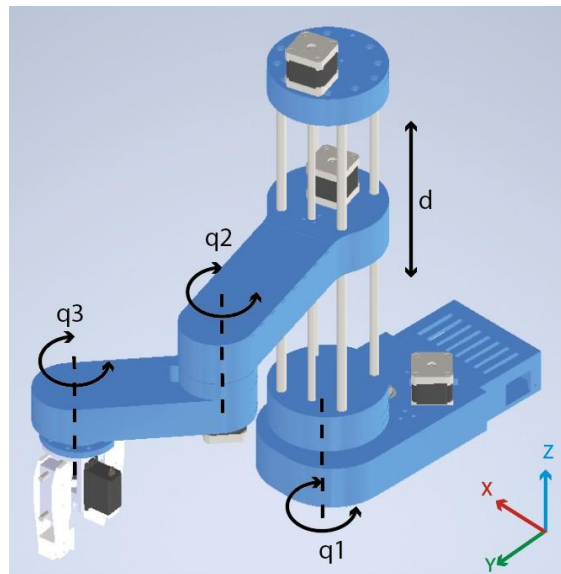


Figura 1: Robot SCARA

107 La estructura fue fabricada mediante impresión 3D utilizando filamento PLA, un material ligero
 108 y rígido, ideal para aplicaciones de prototipado funcional (Tymrak et al., 2014). Está compuesta
 109 por un total de 40 piezas impresas en 3D. La impresión se realizó con una impresora FLSUN V400,
 110 operando a una velocidad promedio de 200 mm/s, lo que permitió equilibrar tiempos de fabricación
 111 y calidad superficial. Para facilitar la manipulación y el acabado posterior, se utilizaron soportes
 112 de tipo árbol. Las articulaciones, tanto rotacionales como lineales, se implementaron mediante
 113 rodamientos lineales, rodamientos axiales y cojinetes de empuje, garantizando un movimiento suave
 114 y de baja fricción. El ensamblaje de las piezas se efectuó utilizando una variedad de tornillos y
 115 elementos de fijación estándar. Las dimensiones generales del robot son: una altura máxima de 508
 116 mm y una longitud máxima de alcance de 712 mm, medida en la máxima elongación del brazo.

117 Para permitir el movimiento de las articulaciones descritas, se seleccionaron motores paso a
 118 paso NEMA 17, controlados por drivers DQ420MA compatibles con micropasos, debido a su
 119 equilibrio entre costo, precisión y facilidad de uso en sistemas de automatización ligera. Estos
 120 motores híbridos, con un ángulo de paso de 1,8 grados (200 pasos/rev) y dimensiones compactas,
 121 son ideales para brazos robóticos tipo SCARA empleados en tareas de clasificación y manipulación
 122 de piezas pequeñas (Misra et al., 2020). El accionamiento mediante micropasos proporciona un
 123 control preciso de los movimientos, y cada eje cuenta con un interruptor de fin de carrera para su
 124 calibración automática. Además, su tamaño compacto y facilidad de control los hacen una opción
 125 práctica para aplicaciones que requieren precisión sin comprometer la simplicidad del sistema.

126 El sistema de agarre consiste en una pinza controlada por un servomotor, sincronizada con
 127 el sistema de visión para ejecutar acciones de clasificación. Este diseño modular permite futuras
 128 mejoras en el efector final, adaptándolo a distintas aplicaciones industriales. El diseño armado se
 129 ilustra en la Figura 2.

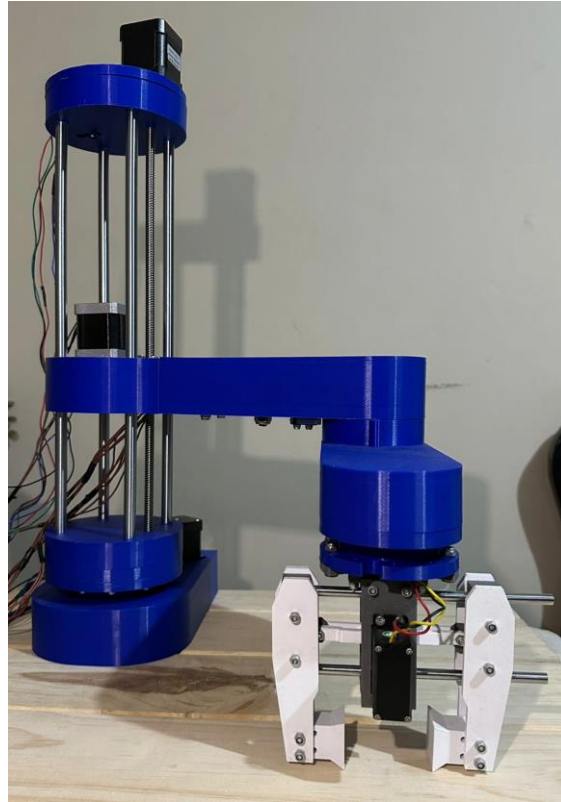


Figura 2: Vista General Robot SCARA

130 3.2 Cinemática y Control

131 Se implementó la cinemática directa e inversa del robot SCARA para convertir coordenadas
 132 cartesianas en ángulos articulares, lo que permite controlar con precisión la posición del efector
 133 final (Keras Team, 2024; Kingma and Ba, 2014). El sistema se ejecutó en un Arduino Mega 2560,
 134 utilizando la biblioteca AccelStepper (McCauley, 2020), la cual permite un movimiento fluido y
 135 preciso de los motores paso a paso mediante el control de la aceleración y velocidad.

136 La elección del Arduino Mega como plataforma de control se justifica debido a que las tareas
 137 de mayor demanda computacional, como el procesamiento de imágenes y la inferencia de modelos
 138 de red neuronal, se ejecutan en un computador externo. En consecuencia, el Arduino se limita a
 139 controlar los motores paso a paso y ejecutar movimientos en función de coordenadas recibidas, lo
 140 cual no requiere un sistema embebido de alto rendimiento. Además, el uso de esta plataforma ha
 141 sido validado en aplicaciones industriales de bajo costo, donde se demuestra su viabilidad como
 142 controlador en sistemas de manipulación automatizados (Sun and Bei, 2023).

143 La comunicación entre el sistema de visión y el robot se estableció mediante el protocolo UART;
 144 se envían al microcontrolador coordenadas cartesianas y ángulos articulares, que son interpretados
 145 y convertidos en pasos para el control de los motores. En la Tabla 1 se presenta la conversión entre
 146 el movimiento del robot y los pasos del motor, incluyendo las relaciones de reducción de las poleas
 147 en las articulaciones rotacionales.

Tabla 1: Relación entre Movimiento del Robot y Pasos del Motor

Modelo	Factor de Conversión	Reducción
Articulación 1 (q_1)	11.11 pasos/grado	20:1
Articulación 2 (q_2)	8.88 pasos/grado	16:1
Articulación 3 (q_3)	2.22 pasos/grado	4:1
Eje Z (d)	160 pasos/mm	-

148 La Figura 3 muestra el modelo geométrico planar de dos grados de libertad (2-DOF) utilizado
 149 para representar el brazo robótico SCARA. Este modelo está compuesto por dos eslabones rígidos
 150 de longitudes L_1 (228mm) y L_2 (136.5mm), unidos por articulaciones rotacionales. A partir de
 151 esta representación se obtienen las ecuaciones de cinemática directa e inversa, fundamentales para
 152 el control del movimiento del efector final en el plano xy .

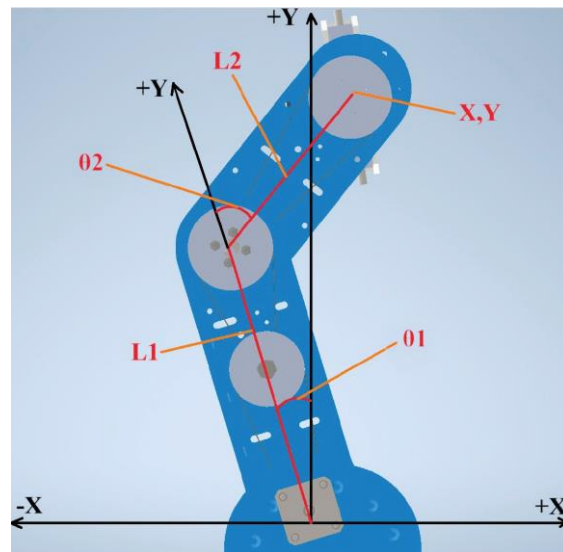


Figura 3: Modelo geométrico planar

153 La cinemática directa permite determinar la posición del efector final (x, y) a partir de los
 154 ángulos articulares θ_1 y θ_2 . Considerando la suma de los vectores de posición de cada eslabón
 155 proyectados en el plano cartesiano, se obtienen las Ecuaciones 1, 2.

$$x = L_1 \cdot \sin(\theta_1) + L_2 \cdot \sin(\theta_1 + \theta_2) \quad (1)$$

$$y = L_1 \cdot \cos(\theta_1) + L_2 \cdot \cos(\theta_1 + \theta_2) \quad (2)$$

156 Estas ecuaciones se derivan mediante el análisis trigonométrico del sistema, considerando que
 157 el segundo eslabón gira un ángulo relativo θ_2 respecto al primero.

158 La cinemática inversa permite calcular los ángulos articulares θ_1 y θ_2 requeridos para alcanzar
 159 una posición deseada (x, y) . Utilizando el teorema del coseno sobre el triángulo formado por los
 160 eslabones y la línea que une el origen con el efector final, se obtienen la Ecuación 3.

$$\theta_2 = \arccos \frac{\left(x^2 + y^2 - L_1^2 - L_2^2 \right)}{2L_1L_2} \quad (3)$$

161 A partir de θ_2 , el ángulo θ_1 puede determinarse mediante análisis geométrico y trigonométrico,
 162 obteniendo así la Ecuación 4.

$$\theta_1 = \arctan \frac{x}{y} - \arctan \frac{L_2 \cdot \sin(\theta_2)}{L_1 + L_2 \cdot \cos(\theta_2)} \quad (4)$$

163 Estas expresiones permiten implementar un control cartesiano del robot, que puede ser traducido
 164 a comandos articulares mediante la cinemática inversa.

165 3.3 Sistema de Visión

166 El sistema de visión artificial es el encargado de capturar y procesar imágenes de las piezas
 167 industriales para su posterior clasificación por color. Se utilizó una cámara HD conectada al
 168 computador mediante USB, y las imágenes fueron procesadas mediante la biblioteca OpenCV,

169 ampliamente utilizada en aplicaciones de visión por computadora (Song, 2025). El flujo de
170 procesamiento comprende las siguientes etapas:

- 171 • **Captura de imagen:** Se adquiere una imagen RGB en tiempo real desde la cámara.
- 172 • **Redimensionamiento:** La imagen se reduce a 100×100 píxeles para normalizar la entrada
173 del modelo.
- 174 • **Conversión de espacio de color:** Se transforma de RGB a HSV, lo que mejora la robustez
175 ante cambios de iluminación (Xian et al., 2023).
- 176 • **Normalización y expansión:** La imagen es normalizada al rango $[0,1]$ y expandida a una
177 dimensión adicional para adaptarse al formato requerido por la red neuronal convolucional
178 (CNN).
- 179 • **Clasificación con CNN:** El modelo clasifica la imagen en categorías predefinidas según el
180 color (amarillo, azul, verde, otros).
- 181 • **Postprocesamiento:** Se interpreta la salida de la red y se asigna la acción correspondiente
182 en el sistema físico (ej. ubicación de destino).

183 Este proceso se resume en la Figura 4, que ilustra cada una de las etapas desde la captura hasta
184 la clasificación final.

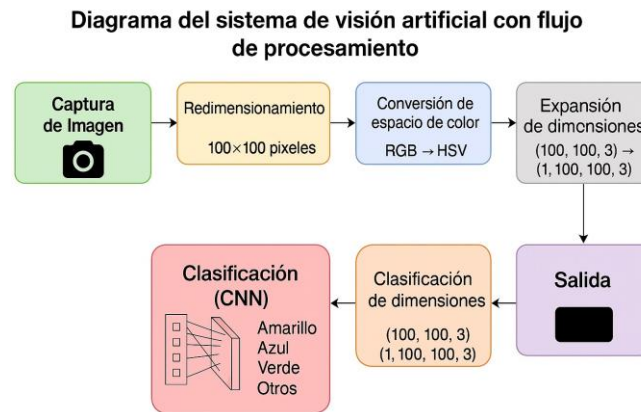


Figura 4: Diagrama del sistema de visión artificial con flujo de procesamiento

185 3.4 Dataset y Aumento de Datos

186 La base de datos utilizada en este proyecto está compuesta por un total de 620 imágenes de cubos
187 de colores, distribuidas equitativamente entre las cuatro clases objetivo: Amarillo, Azul, Verde y
188 Otros, con 155 imágenes por categoría. Las piezas utilizadas en la toma de imágenes son cubos con
189 dimensiones de $3.5 \text{ cm} \times 3.5 \text{ cm} \times 3.5 \text{ cm}$, colocados sobre un fondo neutro y registrados desde un
190 ángulo fijo, asegurando así la homogeneidad en el conjunto de datos. El color es el único patrón
191 empleado para la clasificación de las piezas. Algunas de las piezas empleadas pueden observarse en
192 la Figura 5.

193 Estos cubos pueden representar objetos típicos en industrias de manufactura, control de calidad
194 o robótica educativa, donde la identificación y manipulación precisa de piezas pequeñas es crucial.



Figura 5: Muestras de piezas utilizado en la base de datos

195 Con el fin de mejorar la capacidad de generalización del modelo y evitar el sobreajuste,
196 se aplicaron técnicas de aumento de datos (data augmentation) mediante la herramienta
197 ImageDataGenerator de la biblioteca Keras (Chollet, 2021). Es importante destacar que
198 ImageDataGenerator no genera un conjunto fijo de imágenes nuevas, sino que aplica
199 transformaciones aleatorias a las imágenes existentes durante el entrenamiento, aumentando así la
200 diversidad de las muestras que el modelo ve en cada época. Entre las transformaciones realizadas se
201 incluyen rotaciones aleatorias de hasta $\pm 30^\circ$, desplazamientos horizontales y verticales de hasta un
202 20 % del tamaño de la imagen, zoom de hasta un 15 %, así como volteos horizontales. Estas técnicas
203 permiten simular variaciones comunes en las condiciones de operación industrial, como pequeñas
204 rotaciones o cambios en la posición de las piezas, incrementando así la robustez del modelo ante
205 datos no vistos durante el entrenamiento (Shorten and Khoshgoftaar, 2019).

206 3.5 Arquitectura de Red Neuronal

207 Se diseñaron y entrenaron tres modelos de redes neuronales convolucionales (CNN), con el objetivo
208 de evaluar el impacto de distintas configuraciones arquitectónicas sobre el rendimiento del sistema
209 de clasificación por color. Cada modelo se diferencia por el tamaño del kernel convolucional, la
210 cantidad de capas convolucionales, el número de neuronas en la capa densa final y la proporción
211 de datos utilizados para entrenamiento y validación. Estos aspectos (véase la Tabla 2) fueron
212 seleccionados estratégicamente para analizar la influencia de la profundidad de la red, la capacidad
213 de extracción de características y la cantidad de datos de entrenamiento disponibles.

214 Dicha selección se fundamentó en criterios técnicos y teóricos orientados a representar
215 variaciones clave que afectan el desempeño de los modelos. En particular, la elección de tamaños
216 de kernel (3×3 , 4×4 y 5×5) se fundamentó en su efecto sobre la escala de las características
217 que la red puede captar: mientras los kernels pequeños permiten capturar detalles locales, los
218 más grandes pueden abarcar patrones espaciales más amplios, lo cual resulta crucial en tareas
219 de clasificación por color donde el contexto puede influir en la percepción del tono o la textura
220 (Szegedy et al., 2015). La variación en el número de capas y neuronas permitió evaluar la relación
221 entre la capacidad expresiva del modelo y su tendencia al sobreajuste, conforme a lo descrito en
222 estudios sobre regularización en redes profundas (Srivastava et al., 2014). Finalmente, las distintas
223 proporciones de datos de entrenamiento y validación (80/20, 70/30 y 90/10) se seleccionaron
224 para estudiar cómo la cantidad de datos disponibles para aprendizaje afecta la estabilidad de los
225 resultados y la capacidad de generalización del modelo, siguiendo enfoques clásicos de validación
226 cruzada y selección de modelos (Arlot and Celisse, 2009). Estas decisiones permitieron estructurar
227 un análisis comparativo riguroso, orientado a comprender el impacto de factores arquitectónicos y
228 de entrenamiento en el rendimiento del sistema.

Tabla 2: Arquitecturas comparadas de los modelos CNN

Modelo	Kernel	Capas Conv.	Neuronas Densas	Entrenamiento/Validación
1	3x3	3	512	80/20
2	5x5	2	256	70/30
3	4x4	2	512	90/10

Se empleó la función de pérdida sparse-categorical-crossentropy, adecuada para problemas de clasificación multiclase con etiquetas codificadas como enteros. Como optimizador se utilizó Adam (Adaptive Moment Estimation), conocido por su eficiencia computacional y rápida convergencia en tareas de aprendizaje profundo (Goodfellow et al., 2016). La métrica principal considerada para la evaluación del desempeño fue la precisión (accuracy), lo que permite valorar la proporción de clasificaciones correctas respecto al total de muestras procesadas. Además, se implementó la técnica de early stopping, que detiene el entrenamiento automáticamente cuando la métrica de validación deja de mejorar durante 3 épocas consecutivas. Esto contribuye a prevenir el sobreajuste (overfitting), mejorando la capacidad de generalización del modelo frente a datos nuevos y no vistos (Miseta et al., 2024).

3.6 Entrenamiento y Evaluación

El modelo fue configurado para entrenarse durante un máximo de 50 épocas; sin embargo, se incorporó una estrategia de early stopping con una tolerancia de 3 épocas sin mejoras en la pérdida de validación, lo que permitió detener el entrenamiento de forma anticipada y evitar el sobreajuste. Se utilizó un tamaño de lote (batch size) de 32 muestras por iteración, seleccionado como un equilibrio entre la estabilidad del gradiente y la eficiencia computacional, ya que tamaños mayores demandan más memoria, mientras que los menores pueden inducir oscilaciones durante el aprendizaje. A lo largo del proceso, se monitoreó el desempeño del modelo tanto en el conjunto de entrenamiento como en el de validación, lo que permitió analizar su evolución y capacidad de generalización en el tiempo.

Para evaluar el desempeño de los modelos, se recurrió a diversas métricas ampliamente utilizadas en tareas de clasificación multiclase. La matriz de confusión permitió visualizar la distribución de aciertos y errores por clase, revelando la capacidad del modelo para distinguir correctamente entre los diferentes colores. Asimismo, se calculó la precisión por clase, lo que resultó útil para identificar posibles desequilibrios, especialmente en categorías más ambiguas como la clase "Otros". Adicionalmente, se generaron curvas de aprendizaje que mostraron la evolución de la precisión y la pérdida a lo largo de las épocas de entrenamiento y validación, facilitando la detección de fenómenos como sobreajuste o subajuste (Gonzalez and Woods, 2008).

Los modelos de redes neuronales convolucionales fueron entrenados en un equipo portátil ASUS ROG Strix G15, con procesador Intel Core i7-10750H de seis núcleos y 16GB de memoria RAM. El proceso de entrenamiento se llevó a cabo localmente, utilizando Python y la biblioteca TensorFlow. En promedio, cada modelo fue entrenado durante aproximadamente 3 minutos por configuración, dependiendo de la complejidad arquitectónica. Para analizar el impacto de la disponibilidad de datos sobre el rendimiento del modelo, se aplicaron tres proporciones distintas de división entre los conjuntos de entrenamiento y validación: 80/20, 70/30 y 90/10. Esta estrategia permitió estudiar cómo la cantidad de datos destinada al aprendizaje afecta la estabilidad de los resultados y la capacidad de generalización del sistema.

Estas herramientas y procedimientos en conjunto permitieron una comparación objetiva y detallada del rendimiento entre los distintos modelos entrenados, brindando fundamentos sólidos para seleccionar la arquitectura más adecuada para la tarea de clasificación de color en entornos industriales.

3.7 Arquitectura de ejecución y comunicación del sistema

El sistema propuesto se basa en una arquitectura distribuida de control, en la que se asignan funciones específicas entre un microcontrolador de propósito general (ATmega2560, integrado en la plataforma Arduino Mega 2560) y una unidad de cómputo externa de mayor capacidad. Esta

274 organización permite desacoplar el control de bajo nivel, encargado de los actuadores, de las tareas
275 computacionalmente intensivas asociadas al procesamiento de imágenes y la clasificación mediante
276 redes neuronales.

277 El robot SCARA está equipado con cuatro motores paso a paso tipo NEMA 17: dos para las
278 articulaciones rotacionales del plano horizontal, uno para el desplazamiento vertical del eje Z, y
279 otro para la orientación del efector final. Adicionalmente, un servomotor gestiona la apertura y
280 cierre de la pinza de sujeción. Todos los actuadores se encuentran conectados a la placa Arduino
281 Mega, que actúa como interfaz de control del sistema embebido. Esta placa recibe coordenadas
282 cartesianas desde el computador anfitrión a través de una conexión serial mediante el protocolo
283 UART. Dichas coordenadas son transformadas internamente en ángulos articulares mediante un
284 algoritmo de cinemática inversa implementado con funciones trigonométricas. El control de los
285 motores paso a paso se realiza con la biblioteca AccelStepper, que permite generar perfiles de
286 aceleración y desaceleración, favoreciendo un desplazamiento suave y preciso.

287 Las tareas de percepción y clasificación se desarrollaron en Python y se ejecutan en el
288 computador externo. La adquisición de imágenes se realiza en tiempo real mediante una cámara
289 HD conectada vía USB. Las imágenes son procesadas mediante OpenCV, convertidas al espacio
290 de color HSV, redimensionadas y normalizadas como entrada a una red neuronal convolucional
291 (CNN) previamente entrenada. El modelo clasifica las piezas dentro de una de las cuatro clases
292 predefinidas: Azul, Verde, Amarillo u Otros.

293 Una vez que el sistema de visión, ejecutado en Python, identifica la clase de color
294 correspondiente, se envía una referencia codificada al microcontrolador ATmega2560 mediante
295 comunicación serial. En la plataforma Arduino, esta referencia es interpretada para determinar
296 la posición destino asociada a esa clase. A partir de este valor, se ejecuta la cinemática inversa
297 directamente en el microcontrolador, lo cual permite calcular los ángulos articulares necesarios para
298 posicionar el efector final sobre el área de clasificación correspondiente. Finalmente, el Arduino
299 genera las señales de control requeridas para accionar los motores paso a paso, y el servomotor
300 activa la pinza para liberar la pieza clasificada en el lugar adecuado.

301 La Figura 6 ilustra el esquema general del sistema, incluyendo el flujo de datos entre módulos,
302 la interacción hardware/software, y los procesos que conforman el ciclo percepción-decisión-acción.

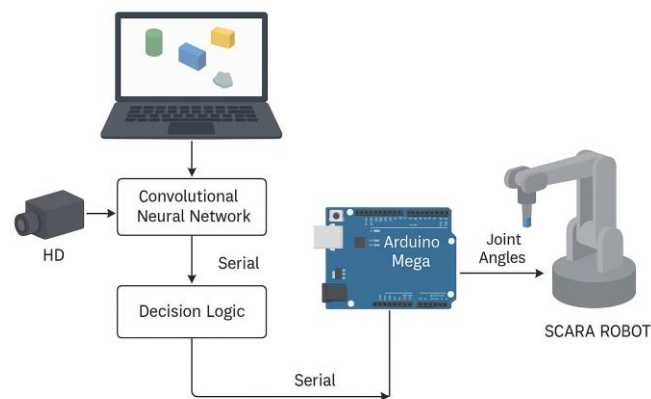


Figura 6: Esquema general de la arquitectura distribuida del sistema de visión y control del robot SCARA.

303 La Figura 7 muestra la interfaz desarrollada en Python, donde se visualiza en tiempo real la
304 imagen capturada por la cámara, la clase de color detectada y las coordenadas cartesianas actuales
305 del efector final.



Figura 7: Interfaz gráfica desarrollada en Python

306 4 Resultados y Discusión

307 4.1 Evaluación Cuantitativa y Discusión Técnica

308 En esta sección se presenta el análisis que evalúa el impacto de las variaciones en la arquitectura
 309 de las CNN sobre el desempeño del sistema de clasificación de color, con el fin de justificar las
 310 decisiones de diseño tomadas en la Sección 2.5. Específicamente, se investiga como la profundidad
 311 de la red, el tamaño del kernel de convolución y la cantidad de datos de entrenamiento influyen en
 312 la eficacia del sistema. Para cada uno de estos parámetros, se comparan los resultados obtenidos
 313 por los diferentes modelos, y se explica como las diferencias observadas en el desempeño (precisión,
 314 pérdida, etc.) respaldan la elección de una arquitectura sobre otra. Se entrenaron y evaluaron
 315 tres modelos de red neuronal convolucional (CNN), cada uno con diferente configuración de
 316 arquitectura y proporción de datos de entrenamiento/validación. Se analizaron métricas por clase,
 317 comportamiento de aprendizaje y errores mediante curvas y matrices de confusión.

318 La Tabla 3 muestra los porcentajes de precisión por clase obtenidos para cada uno de los tres
 319 modelos de red neuronal convolucional (CNN). Cada valor representa el porcentaje de muestras de
 320 una determinada clase que fueron correctamente clasificadas por el modelo correspondiente. Este
 321 análisis permite evaluar el desempeño del modelo de forma desagregada, identificando fortalezas
 322 y posibles limitaciones en la clasificación de cada categoría. Además, facilita la comparación entre
 323 modelos para determinar cuál presenta un rendimiento más equilibrado entre las distintas clases.

Tabla 3: Precisión por clase para cada modelo CNN

Clase	Modelo 1	Modelo 2	Modelo 3
Amarillo	94 %	92 %	93 %
Azul	96 %	95 %	91 %
Verde	91 %	90 %	90 %
Otros	85 %	82 %	86 %

324 Como puede observarse en las Figuras 8, 9, 10, las curvas de pérdida y precisión de cada
 325 modelo reflejan la evolución del entrenamiento, permitiendo identificar variaciones significativas
 326 en la dinámica entre los modelos evaluados. Estas curvas fueron generadas utilizando la librería
 327 Matplotlib en Python, a partir de los datos obtenidos durante el entrenamiento realizado con
 328 TensorFlow y Keras.

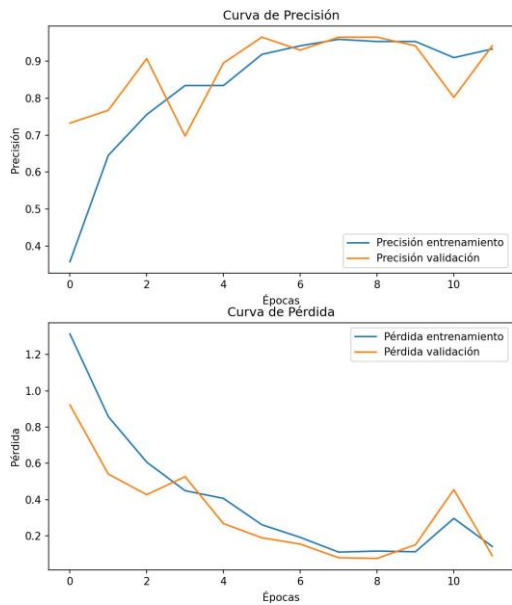


Figura 8: Curva de pérdida y precisión – Modelo 1

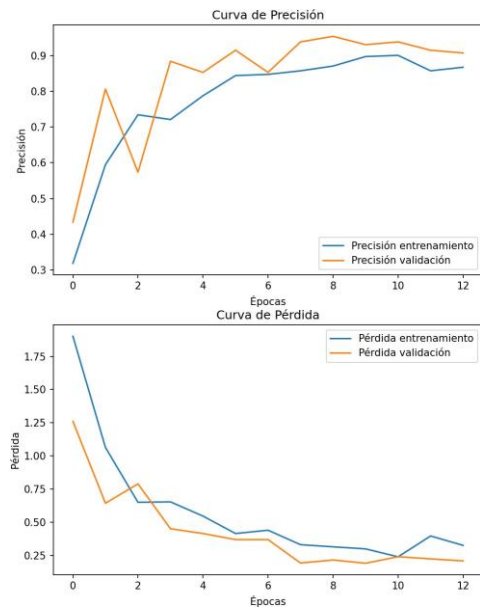


Figura 9: Curva de pérdida y precisión – Modelo 2

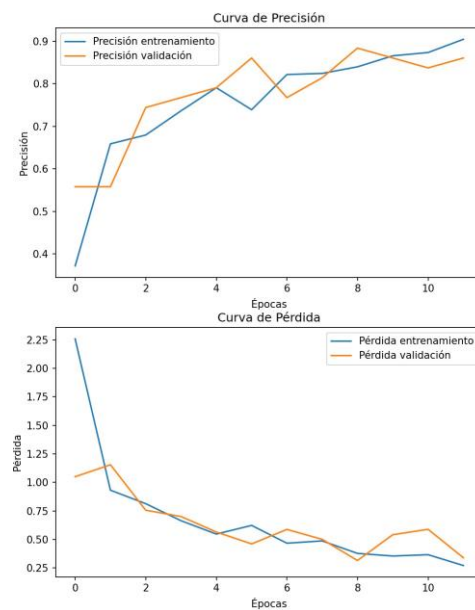


Figura 10: Curva de pérdida y precisión - Modelo 3

329 El Modelo 1 exhibe una convergencia robusta y consistente hacia una alta precisión, acompañada
 330 de una disminución sostenida de la función de pérdida, lo que denota un aprendizaje eficaz y una
 331 adecuada capacidad de generalización. En contraste, los Modelos 2 y 3 demuestran fluctuaciones
 332 perceptibles en las métricas de precisión y pérdida, particularmente en el conjunto de validación,
 333 lo cual sugiere una menor estabilidad en el proceso de optimización y un incremento en el riesgo
 334 de sobreajuste. Específicamente, el Modelo 3, caracterizado por una menor proporción de datos
 335 destinados a la validación, presenta la mayor variabilidad en su desempeño, lo que implica una
 336 generalización comprometida. En síntesis, el Modelo 1 se distingue por un perfil de entrenamiento
 337 más favorable en términos de estabilidad y capacidad de generalización, en comparación con los
 338 Modelos 2 y 3.

339 Además del análisis por clase, se calcularon métricas globales que permiten una evaluación
 340 agregada del desempeño de los modelos. En particular, se consideraron la precisión promedio, el
 341 *recall* y el *F1-score* en su forma macro, es decir, promediando uniformemente el rendimiento sobre
 342 todas las clases. Estas métricas ofrecen una visión complementaria al *accuracy*, ya que permiten
 343 evaluar la capacidad del modelo para identificar correctamente cada clase sin estar sesgado por la
 344 frecuencia de aparición de cada una (Sokolova and Lapalme, 2009). Como se resume en la Tabla 4,
 345 el Modelo 1 obtuvo los valores más altos y equilibrados en las tres métricas, lo que respalda su
 346 capacidad de generalización y robustez. Por el contrario, los Modelos 2 y 3, aunque alcanzan
 347 resultados aceptables, presentan un rendimiento inferior en términos de balance entre precisión y
 348 recuperación, evidenciando una mayor variabilidad en su capacidad predictiva.

Tabla 4: Métricas macro promedio para cada modelo CNN

Métrica	Modelo 1	Modelo 2	Modelo 3
Precisión	0.9478	0.8306	0.8875
Recall	0.9408	0.7865	0.8934
F1-score	0.9429	0.7646	0.8839

349 La Tabla 5 resume de forma cualitativa las observaciones clave en términos de estabilidad,
 350 pérdida, riesgo de sobreajuste y eficiencia arquitectónica de cada modelo.

Tabla 5: Comparación de modelos según métricas clave

Métrica	Modelo 1	Modelo 2	Modelo 3
Estabilidad y precisión	Alta, converge hacia época 6	Media, con fluctuaciones en validación	Media, varía más por set de prueba pequeño
Pérdida validación	Estable, baja progresiva	Oscilante hacia últimas épocas	Disminuye, pero con picos tardíos
Sobreajuste	Bajo	Moderado en clases minoritarias	Riesgo medio por menos validación
Generalización esperada	Alta	Media-Alta (Mejor en azul)	Media, especialmente en clase "Otros"
Eficiencia por arquitectura	Equilibrada entre profundidad y kernel	Kernels grandes mejoran captación global	Compensa con dropout, pero con ruido

351 Las matrices de confusión que se pueden observar en las Figuras 11, 12 y 13, muestran los
 352 resultados de clasificación, incluyendo tanto los aciertos como los errores entre las distintas clases.
 353 Se identifica que la clase Otros exhibe la precisión más baja (85%), lo que indica una mayor
 354 dificultad del modelo para clasificar correctamente las piezas que no correspondan a los colores
 355 primarios de amarillo, azul y verde. La complejidad inherente a la variada gama cromática presente
 356 en esta categoría "Otros" se refleja en la mayor tasa de errores observada. Un análisis más detallado
 357 de la matriz de confusión revela patrones específicos en estos errores. Por ejemplo, se encontraron
 358 casos en los que el modelo confundió tonos de gris y blanco, asignándolos incorrectamente a otras
 359 clases de color.

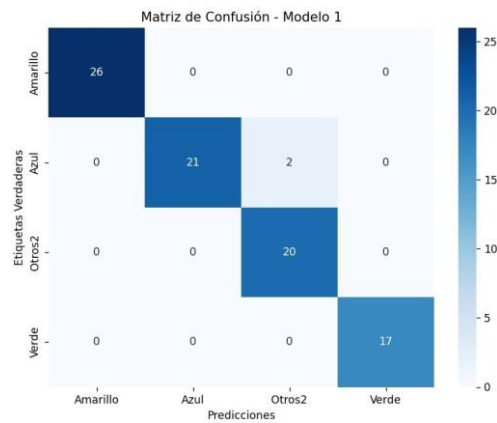


Figura 11: Matriz de Confusión - Modelo 1

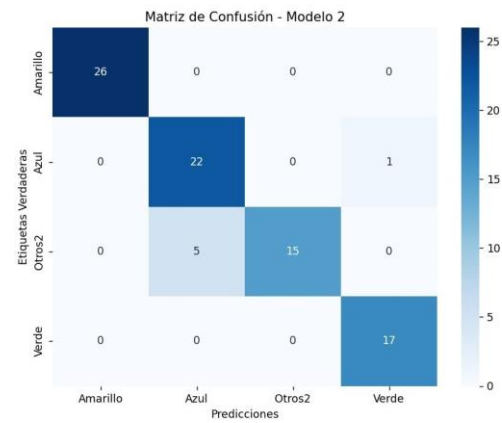


Figura 12: Matriz de Confusión - Modelo 2

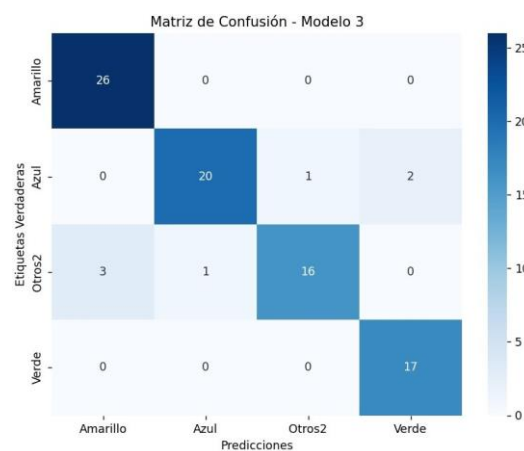


Figura 13: Matriz de Confusión - Modelo 3

360 El sistema demostró robustez ante condiciones de iluminación variables, atribuible a la
 361 conversión a espacio de color HSV y al aumento de datos realizado con técnicas como rotaciones,
 362 desplazamientos y zoom (Xian et al., 2023), lo cual contribuyó a mejorar la generalización de los
 363 modelos.

364 4.2 Implementación en el Brazo Robotico SCARA

365 Una vez validado el desempeño del Modelo 1 como el más balanceado, se integró al sistema de
 366 control del robot SCARA para realizar la clasificación de piezas por color. El sistema de visión
 367 artificial detecta el color de cada objeto sobre una plataforma de prueba, enviando la clase detectada
 368 al controlador basado en Arduino mediante comunicación UART. El robot SCARA, controlado
 369 mediante coordenadas calculadas con cinemática inversa, ejecuta una rutina de desplazamiento
 370 hacia el área de clasificación correspondiente. La pinza, accionada por un servomotor, realiza la
 371 captura y depósito de la pieza en el contenedor asignado. La Figura 14 muestra la implementación
 372 física del sistema en condiciones reales de operación. En condiciones simuladas, el sistema logró
 373 procesar una pieza cada 3.5 segundos, con una precisión global del 91.3 %, demostrando eficiencia
 374 y adaptabilidad para futuras aplicaciones industriales.

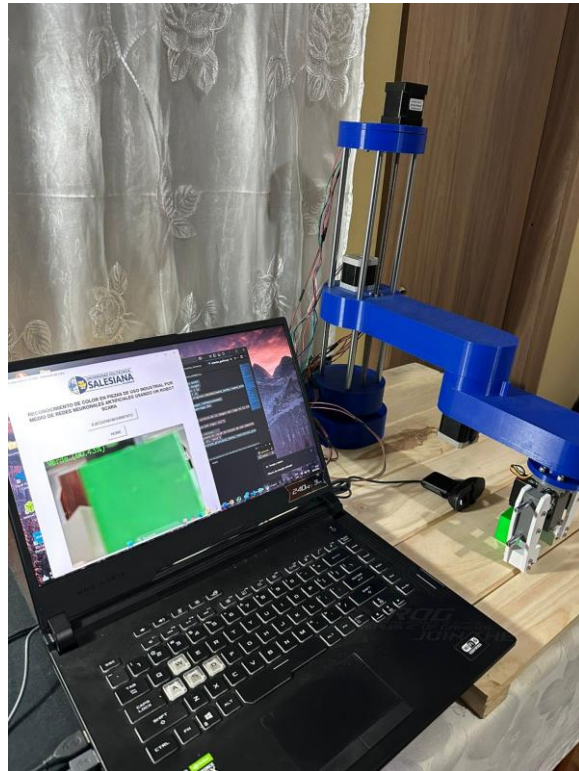


Figura 14: Implementación física del sistema de clasificación automática mediante robot SCARA y visión artificial

375 4.3 Limitaciones del Sistema

376 Aunque el sistema propuesto alcanzó una precisión global del 91.3 %, existen ciertos factores que
 377 podrí an limitar su aplicabilidad en entornos industriales más exigentes:

- 378 • **Sensibilidad a condiciones lumínicas extremas.** Si bien el uso del espacio de color HSV
 379 proporciona cierta tolerancia frente a variaciones moderadas de iluminación, el rendimiento
 380 disminuye significativamente en escenarios con luz natural cambiante, sombras proyectadas o
 381 reflejos especulares. Esto podrí a comprometer su estabilidad operativa en plantas sin control
 382 lumínico artificial.
- 383 • **Limitaciones del conjunto de entrenamiento.** El modelo fue entrenado con un conjunto
 384 de 620 imágenes de cubos sobre fondo neutro, lo que restringe su capacidad de generalización
 385 frente a objetos con textura, formas complejas o fondos visualmente contaminados. Este sesgo
 386 del *dataset* limita la escalabilidad del sistema a nuevas condiciones de trabajo.
- 387 • **Dependencia exclusiva de información cromática.** La red neuronal está diseñada para
 388 clasificar únicamente en función del color. Objetos de tonalidad similar pero diferentes en
 389 geometría o textura no pueden ser discriminados, lo que impide su aplicación en tareas que
 390 requieran identificación multimodal.
- 391 • **Requisitos de procesamiento externo.** La inferencia del modelo se realiza actualmente
 392 en un ordenador portátil conectado al sistema, lo que introduce dependencia de un equipo
 393 externo y podrí a dificultar su integración en líneas de producción autónomas o móviles.

394 Estas limitaciones no desvirtúan la viabilidad técnica del sistema, pero constituyen
 395 oportunidades claras de mejora que deben ser consideradas para futuras implementaciones
 396 industriales a mayor escala.

397 5 Conclusiones

398 El presente trabajo logró implementar un sistema funcional para el reconocimiento y clasificación de
399 piezas industriales por color, integrando visión artificial mediante redes neuronales convolucionales
400 (CNN) con un brazo robótico tipo SCARA de arquitectura modular y bajo costo. La clasificación
401 cromática se realizó a partir de imágenes transformadas al espacio de color HSV y enriquecidas
402 mediante técnicas de aumento de datos, lo que permitió alcanzar una alta precisión bajo condiciones
403 controladas que simulan un entorno industrial.

404 De los tres modelos evaluados, el Modelo 1 –compuesto por tres capas convolucionales con
405 kernel de 3×3 y una capa densa de 512 neuronas– demostró el mejor equilibrio entre precisión
406 y capacidad de generalización. Este modelo alcanzó una *accuracy* promedio de 94.78 %, un *recall*
407 de 94.08 % y un *F1-score* de 94.29 %, evidenciando un desempeño robusto y balanceado frente a
408 datos no vistos.

409 A nivel de sistema completo –desde la captura de imágenes hasta la ejecución robótica de la
410 clasificación– se logró una efectividad global del 91.3 % en condiciones simuladas de operación.
411 La categoría con mayor precisión fue “Azul” (96 %), mientras que la clase “Otros” presentó el
412 desempeño más bajo (85 %), debido a la mayor variabilidad cromática en los objetos que la
413 conforman.

414 La implementación práctica sobre un robot SCARA permitió validar de forma integral el proceso
415 de clasificación automática y actuación robótica, demostrando que el sistema es escalable, replicable
416 y adaptable a tareas de automatización en líneas de producción de pequeña y mediana escala.

417 Estos resultados permiten validar la hipótesis de partida, demostrando que es viable
418 implementar un sistema preciso, funcional y accesible para clasificación cromática automática
419 en entornos industriales simulados, utilizando herramientas tecnológicas de bajo costo y fácil
420 implementación.

421 Como líneas futuras de mejora se propone integrar un sistema de iluminación controlada que
422 garantice condiciones lumínicas constantes, lo que podría incrementar la robustez del modelo
423 frente a entornos cambiantes. Asimismo, la incorporación del reconocimiento de formas geométricas
424 permitiría ampliar la funcionalidad del sistema más allá del color, favoreciendo su aplicación en
425 tareas de clasificación multimodal. Finalmente, se plantea como evolución tecnológica la migración
426 del modelo a plataformas embebidas, como Jetson Nano, lo que permitiría realizar la inferencia
427 directamente desde el robot, eliminando la dependencia de equipos externos y mejorando la
428 eficiencia energética del sistema. Estas propuestas abrirían nuevas oportunidades de optimización
429 y adaptación del sistema en contextos industriales reales.

430 Referencias

- 431 S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics*
432 *Surveys*, 4:40–79, 2009. doi: 10.1214/09-SS054. URL <https://doi.org/10.1214/09-SS054>.
- 433 F. Chollet. *Deep Learning with Python*. Manning Publications, 2nd edition,
434 2021. ISBN 9781617296864. URL [https://www.manning.com/books/](https://www.manning.com/books/deep-learning-with-python-second-edition)
435 [deep-learning-with-python-second-edition](https://www.manning.com/books/deep-learning-with-python-second-edition).
- 436 R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson Education, 3rd edition, 2008.
437 ISBN 9780131687288.
- 438 I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. ISBN 9780262035613.
439 URL <https://www.deeplearningbook.org/>.
- 440 M. P. Groover. *Automation, Production Systems, and Computer-Integrated Manufacturing*.
441 Pearson Education, 3rd edition, 2008. ISBN 9780132393218.
- 442 HowToMechatronics. Scara robot - how to build your own arduino
443 based robot, 2023. URL [https://howtomechatronics.com/projects/](https://howtomechatronics.com/projects/scara-robot-how-to-build-your-own-arduino-based-robot/)
444 [scara-robot-how-to-build-your-own-arduino-based-robot/](https://howtomechatronics.com/projects/scara-robot-how-to-build-your-own-arduino-based-robot/). Accessed: 2025-04-02.

- 445 H. Kagermann, W. Wahlster, and J. Helbig. Recommendations for implementing the strategic
446 initiative industrie 4.0: Securing the future of german manufacturing industry. Technical
447 report, Industrie 4.0 Working Group, acatech – National Academy of Science and Engineering,
448 2013. URL [https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/
449 recommendations-for-implementing-industry-4-0-data.pdf](https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf). Accessed: 2025-04-02.
- 450 H.-C. Kang, H.-N. Han, H.-C. Bae, M.-G. Kim, J.-Y. Son, and Y.-K. Kim. Hsv color-space-based
451 automated object localization for robot grasping without prior knowledge. *Applied Sciences*, 11
452 (16):7593, 2021. doi: 10.3390/app11167593. URL <https://doi.org/10.3390/app11167593>.
- 453 T. P. Kapusi, T. I. Erdei, G. Husi, and A. Hajdu. Application of deep learning in the deployment
454 of an industrial scara machine for real-time object detection. *Robotics*, 11(4):69, 2022. doi:
455 10.3390/robotics11040069. URL <https://doi.org/10.3390/robotics11040069>.
- 456 Keras Team. Keras api reference, 2024. URL <https://keras.io/api/>. Accessed: 2025-03-15.
- 457 I. Kesici, S. Biskin, and A. Eleyan. Automatic object painting with scara robot using computer
458 vision. *Celal Bayar University Journal of Science*, 14(1):17–22, 2018. doi: 10.18466/cbayarfbe.
459 306950. URL <https://doi.org/10.18466/cbayarfbe.306950>.
- 460 D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint*
461 *arXiv:1412.6980*, 2014. doi: 10.48550/arXiv.1412.6980. URL [https://doi.org/10.48550/
462 arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980).
- 463 Y. Koren. *Robotics for Engineers*. McGraw-Hill, 1st edition, 1985. ISBN 9780070353999.
- 464 Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi:
465 10.1038/nature14539. URL <https://doi.org/10.1038/nature14539>.
- 466 M. McCauley. Accelstepper library for arduino, 2020. URL [http://www.airspayce.com/mikem/
467 arduino/AccelStepper/](http://www.airspayce.com/mikem/arduino/AccelStepper/). Accessed: 2025-03-01.
- 468 T. Miseta, A. Fodor, and A. Vathy-Fogarassy. Surpassing early stopping: A novel correlation-based
469 stopping criterion for neural networks. *Neurocomputing*, 567:127028, 2024. doi: 10.1016/j.
470 neucom.2023.127028. URL <https://doi.org/10.1016/j.neucom.2023.127028>.
- 471 A. Misra, A. Sharma, G. Singh, and A. Kumar. Design and development of a low-cost
472 cnc alternative scara robotic arm. *Procedia Computer Science*, 171:1146–1153, 2020. doi:
473 10.1016/j.procs.2020.04.266. URL <https://doi.org/10.1016/j.procs.2020.04.266>.
- 474 L. Monostori. Cyber-physical production systems: Roots, expectations and r&d challenges.
475 *Procedia CIRP*, 17:9–13, 2014. doi: 10.1016/j.procir.2014.03.115.
- 476 M. Nkomo and M. Collier. A color-sorting scara robotic arm. In *Proceedings of the 2nd International*
477 *Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 763–768.
478 IEEE, 2012. doi: 10.1109/CECNet.2012.6201967. URL [https://doi.org/10.1109/CECNet.
479 2012.6201967](https://doi.org/10.1109/CECNet.2012.6201967).
- 480 J. A. Richards. *Remote Sensing Digital Image Analysis: An Introduction*. Springer, 5th edition,
481 2012. ISBN 9783642300622.
- 482 C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning.
483 *Journal of Big Data*, 6(1):60, 2019. doi: 10.1186/s40537-019-0197-0. URL [https://doi.org/
484 10.1186/s40537-019-0197-0](https://doi.org/10.1186/s40537-019-0197-0).
- 485 M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification
486 tasks. *Information Processing & Management*, 45(4):427–437, 2009. doi: 10.1016/j.ipm.2009.03.
487 002. URL <https://doi.org/10.1016/j.ipm.2009.03.002>.
- 488 P. Song. Object detection based on hsv in opencv. *Applied and Computational Engineering*,
489 121:116–122, 2025. doi: 10.54254/2755-2721/2025.19740. URL [https://doi.org/10.54254/
490 2755-2721/2025.19740](https://doi.org/10.54254/2755-2721/2025.19740). Accessed: 2025-07-02.

- 491 N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A
492 simple way to prevent neural networks from overfitting. *Journal of Machine Learning*
493 *Research*, 15(1):1929–1958, 2014. URL [https://jmlr.org/papers/volume15/srivastava14a/
494 srivastava14a.pdf](https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf).
- 495 G. Sun and G. Bei. Arduino-based intelligent handling robot design. *Advances in Computer,*
496 *Signals and Systems*, 7:67–74, 2023. doi: 10.23977/acss.2023.070109. URL [https://doi.org/
497 10.23977/acss.2023.070109](https://doi.org/10.23977/acss.2023.070109).
- 498 T. P. Swaminathan, C. Silver, and T. Akilan. Benchmarking deep learning models on nvidia jetson
499 nano for real-time systems: An empirical investigation. *arXiv preprint arXiv:2406.17749*, 2024.
500 doi: 10.48550/arXiv.2406.17749. URL <https://doi.org/10.48550/arXiv.2406.17749>.
- 501 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and
502 A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on*
503 *Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.
504 7298594. URL <https://doi.org/10.1109/CVPR.2015.7298594>.
- 505 V. Tinoco, M. F. Silva, F. N. dos Santos, and R. Morais. Benchmarking controllers for low-cost
506 agricultural scara manipulators. *Sensors*, 25(9):2676, 2025. doi: 10.3390/s25092676. URL [https:
507 //doi.org/10.3390/s25092676](https://doi.org/10.3390/s25092676).
- 508 B. M. Tymrak, M. Kreiger, and J. M. Pearce. Mechanical properties of components fabricated
509 with open-source 3-d printers under realistic environmental conditions. *Materials & Design*, 58:
510 242–246, 2014. doi: 10.1016/j.matdes.2014.02.038. URL [https://doi.org/10.1016/j.matdes.
511 2014.02.038](https://doi.org/10.1016/j.matdes.2014.02.038).
- 512 Z. Xian, R. Huang, D. Towey, and C. Yue. The impact of color spaces on convolutional neural
513 network classification performance. *SSRN Electronic Journal*, 2023. doi: 10.2139/ssrn.4442933.
514 URL <https://doi.org/10.2139/ssrn.4442933>.
- 515 Z. Zhang, Q. Guo, M. A. Grigorev, and I. Kholodilin. Construction method of a digital-twin
516 simulation system for scara robots based on modular communication. *Sensors*, 24(22):7183,
517 2024. doi: 10.3390/s24227183. URL <https://doi.org/10.3390/s24227183>.

518 **6 Biografía**



Francisco Xavier Pesántez Zúñiga El autor es estudiante del programa de Maestría en Electrónica y Automatización. El obtuvo su título de Ingeniero en Mecatrónica en el 2022 por la Universidad Politécnica Salesiana. Actualmente se desempeña como Técnico de Mantenimiento y Soporte de ATMs en la Cooperativa Juventud Ecuatoriana Progresista.



Darío Bernardo Pulla Iñiguez El autor es estudiante del programa de Maestría en Electrónica y Automatización. El obtuvo su título de Ingeniero en Mecatrónica en el 2022 por la Universidad Politécnica Salesiana. Actualmente se desempeña como Docente en el área de mecatrónica de la Unidad Educativa Técnico Salesiano de la ciudad de Cuenca, Ecuador.