

Automated Classification Platform with SCARA Robot and Artificial Vision

Adán Farfán Manrique¹, , Gabriel Ojeda Villalta² , Gian Rojas Zevallos³ , Enrique Tejada Villanueva⁴ 
Marcelo Quispe Ccachuco, Dr.⁴  y Sergio Mestas Ramos, Dr.⁶ ,

^{1,2,3,4,5,6} Universidad Católica de Santa María, Perú, adan.farfan@ucsm.edu.pe, gabriel.ojeda@ucsm.edu.pe, gian.rojas@ucsm.edu.pe, enrique.tejada@ucsm.edu.pe, mquispec@ucsm.edu.pe, smestast@ucsm.edu.pe

Abstract- The automation of industrial processes has given rise to innovative solutions that combine robotics and artificial vision to optimize the sorting of parts. In this context, the development of a SCARA robot capable of selecting and sorting parts using an artificial vision system based on an ESP32 as the main camera is presented. After analyzing the images, the SCARA robot controller executes precise trajectories that allow parts to be sorted according to predefined criteria. This type of approach ensures efficient and repetitive handling, optimizing operation times in production lines. The mechanical structure of the SCARA has been designed to offer high speed and accuracy, essential characteristics in industrial environments where reliability is a priority. The ESP32 as a camera offers good resolution, a powerful processor, and high connectivity, making it a compact and cost-effective solution that facilitates its implementation in applications where cost reduction is essential. Additionally, an intuitive graphical interface was designed, allowing the user to configure classification parameters and monitor system performance in real time, enhancing its flexibility and adaptability. This proposal integrates areas such as computer vision, robotic control, and mechatronic design, standing out for its ability to efficiently solve complex tasks, as well as for its functionality and accessibility. In this way, the prototype is consolidated as a practical and versatile tool for automated piece classification. This work demonstrates the potential of accessible technologies in developing advanced solutions for the industry, positioning the system as a promising alternative in the field of industrial automation.

Keywords- SCARA robot, Artificial vision, Part classification, Mechatronic design.

Plataforma de Clasificación Automatizada con Robot SCARA y Visión Artificial

Adán Farfán Manrique¹, , Gabriel Ojeda Villalta², , Gian Rojas Zevallos³, , Enrique Tejada Villanueva⁴, 
Marcelo Quispe Ccachuco, Dr.⁴,  y Sergio Mestas Ramos, Dr.⁶, 

^{1,2,3,4,5,6} Universidad Católica de Santa María, Perú, adan.farfan@ucsm.edu.pe, gabriel.ojeda@ucsm.edu.pe,
gian.rojas@ucsm.edu.pe, enrique.tejada@ucsm.edu.pe, mquispec@ucsm.edu.pe, smestasr@ucsm.edu.pe

Resumen- *La automatización de procesos industriales ha dado lugar a soluciones innovadoras que combinan robótica y visión artificial para optimizar la clasificación de piezas. En este contexto, se presenta el desarrollo de un robot SCARA capaz de seleccionar y clasificar piezas mediante un sistema de visión artificial basado en un ESP32 como cámara principal. Tras el análisis de las imágenes, el controlador del robot SCARA ejecuta trayectorias precisas que permiten clasificar las piezas según criterios predefinidos, este tipo de enfoque asegura una manipulación eficiente y repetitiva, optimizando los tiempos de operación en líneas de producción, la estructura mecánica del SCARA ha sido diseñada para ofrecer alta velocidad y exactitud, características esenciales en entornos industriales donde la confiabilidad es prioritaria. El ESP32 como cámara aporta una buena resolución, un buen procesador y una alta conectividad, lo que la convierte en una solución compacta y económica, lo que facilita su implementación en aplicaciones donde la reducción de costos es fundamental. Adicionalmente, se diseñó una interfaz gráfica intuitiva que permite al usuario configurar parámetros de clasificación y monitorear el rendimiento del sistema en tiempo real, mejorando su flexibilidad y adaptabilidad. Esta propuesta integra áreas como visión artificial, control robótico y diseño mecatrónico, destacando por su capacidad de resolver tareas complejas de manera eficiente, además de destacar por su funcionalidad y accesibilidad, consolidando de esta manera al prototipo como una herramienta práctica y versátil para la clasificación automatizada de piezas. Este trabajo muestra el potencial de las tecnologías accesibles en el desarrollo de soluciones avanzadas para la industria, posicionando al sistema como una alternativa prometedora en el ámbito de la automatización industrial.*

Palabras Clave- *Robot SCARA, Visión artificial, Clasificación de Piezas, Diseño mecatrónico.*

I. INTRODUCCIÓN

La integración de robots SCARA (Selective Compliance Assembly Robot Arm) con sistemas de visión artificial está transformando los procesos de clasificación en la industria, permitiendo operaciones más precisas, rápidas y eficientes. Este proyecto se centra en el desarrollo de una plataforma automatizada de clasificación de figuras geométricas utilizando un robot SCARA y visión artificial, combinando tecnologías de procesamiento de imágenes, cálculos cinemáticos y control robótico para crear un sistema completamente funcional y adaptable a un entorno automatizado [1], [2].

El sistema completo que se planificó tiene las siguientes partes:

Hardware

1. ESP32-CAM: Este módulo fue seleccionado debido a su capacidad de capturar imágenes en tiempo real con resolución adecuada y bajo consumo energético. La integración de su cámara OV2640 con un microcontrolador ESP32 permitió combinar procesamiento local con transmisión inalámbrica de datos, una característica clave para sistemas de visión artificial compactos y de bajo costo.
2. Robot SCARA: El diseño del robot SCARA fue elegido por su capacidad para realizar movimientos rápidos y precisos en aplicaciones de "pick and place". Sus características de flexibilidad y repetibilidad lo hacen ideal para tareas de ensamblaje y manipulación en planos horizontales.
3. Electroimán: Para la manipulación de objetos, se implementó un electroimán que proporciona un

agarre estable y seguro, eliminando la necesidad de sistemas complejos de grippers mecánicos o neumáticos.

4. Controladores y Sensores: Los servomotores, sensores de distancia, y un controlador L298N aseguraron movimientos controlados y retroalimentación adecuada durante el proceso de clasificación.
5. Faja Transportadora: Esta componente facilitó el transporte de los objetos desde el área de entrada hacia la zona de clasificación, proporcionando un flujo continuo y eficiente.

Software

1. Python con OpenCV: Este lenguaje y biblioteca se utilizaron para realizar el procesamiento de imágenes, detectando contornos, formas y colores de los objetos con precisión.
2. Arduino IDE: Proporcionó una plataforma para programar el SOC (System On Chip) ESP32, encargándose del control del robot SCARA y la gestión de los datos recibidos desde el sistema de visión artificial.
3. MATLAB: Se empleó para simular trayectorias y verificar modelos cinemáticos antes de implementarlos en el sistema físico.

A. El Robot SCARA

La elección del robot SCARA se basa en sus características de precisión y velocidad en la ejecución de tareas. Gracias a su estructura, que permite movimientos rápidos en el plano horizontal y cierta flexibilidad en el eje vertical, estos robots son ideales para aplicaciones que requieren alta repetibilidad y eficiencia además de tener cálculos de posición más simples respecto a su modelo cinemático directo [3]. Según Adept Technology, los SCARA destacan por su capacidad para manejar tareas de ensamblaje y manipulación con alta repetibilidad, lo que los convierte en una opción ideal para operaciones industriales automatizadas [4]. Además, investigaciones recientes resaltan su integración con sistemas de planificación de trayectorias para garantizar movimientos optimizados y suaves en aplicaciones industriales [4], [5].

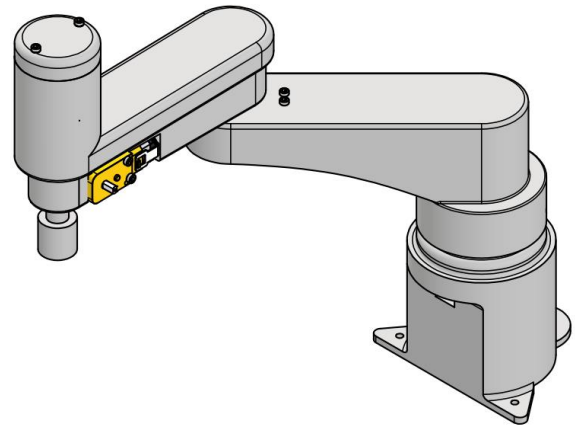


Fig. 1 Ensamble Final del Robot SCARA.

B. Sistema de Visión Artificial

En este proyecto, se utilizó una ESP32-CAM para capturar imágenes en tiempo real, que fueron enviadas a un programa en Python utilizando la librería OpenCV para su procesamiento. Las operaciones de preprocesamiento incluyeron la conversión a escala de grises, la eliminación de ruido y la segmentación de bordes, permitiendo identificar figuras geométricas como triángulos, cuadrados y círculos [8], [9]. Este procesamiento es fundamental para garantizar una identificación precisa de los objetos, como lo demuestra Celis en el uso de técnicas avanzadas de umbralización y filtrado para mejorar la calidad de las imágenes capturadas [2].



Fig. 2 Reconocimiento de imágenes por visión artificial.

C. Sistema de Comunicación

La posición de las figuras fue calculada y enviada a una ESP32 en formato JSON a través de comunicación WiFi, permitiendo un control dinámico y preciso [11], [12]. Para la manipulación física de las figuras, el robot SCARA utiliza un electroimán, controlado por un transistor BC457 y protegido mediante un diodo 1N4007 para evitar picos de voltaje. El sistema realiza cálculos cinemáticos (directo e inverso) para calcular los ángulos de sus articulaciones, garantizando movimientos precisos y trayectorias optimizadas [1], [2]. Según González, la implementación de un modelo cinemático inverso mejora significativamente la precisión en la planificación y ejecución de trayectorias, haciendo que los sistemas SCARA sean más efectivos en aplicaciones industriales [4].

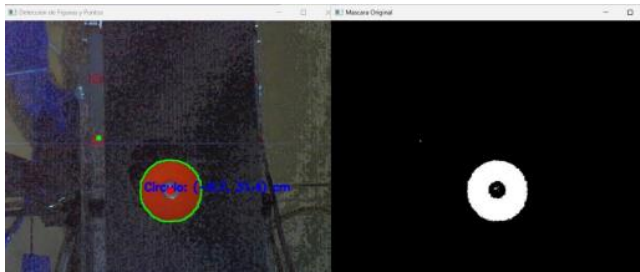


Fig. 3 Detección de la posición y la forma.

El diseño del sistema incluye modelos matemáticos del robot SCARA, planificación de trayectorias, procesamiento de imágenes y su integración en un único circuito de control [13], [14], [15]. Este enfoque garantiza que el robot pueda identificar, recoger y depositar figuras geométricas en compartimientos predefinidos de manera precisa y autónoma [6]. Además, el uso de software como MATLAB y Arduino IDE facilita tanto la simulación como el control práctico del sistema, destacando su relevancia en entornos industriales y educativos [12], [13].

La ESP32-CAM detecta y envía la posición y el tipo de figura que se detecta, a la ESP32 que gobierna el funcionamiento del robot SCARA y el funcionamiento de la faja de transmisión, sincronizando así los dos procesos y haciéndolo totalmente automatizado y sin la intervención externa de sistemas computacionales [3].

II. MATERIALES Y MÉTODOS

A. Diseño del Robot SCARA

Para el diseño del robot SCARA se consideró las velocidades de trabajo del robot, las cuales estarán limitadas por la velocidad de los actuadores, en este caso los servomotores y el sistema piñón-cremallera del actuador final, además del peso de cada uno de los eslabones y el material del cual estarán hechos, en este caso se optó por el PLA+ y proporcionar un porcentaje de relleno alto en los primeros eslabones y menor en los siguientes para que los primeros eslabones sean más resistentes y los últimos tengan menor peso. El ensamblaje de los eslabones se realizaría por medio de pernos, para tener una unión rígida y sin que estas lleguen a ceder por el peso.



Fig. 4 Robot SCARA ensamblado.

Los actuadores seleccionados para el proceso son dos servomotores MG995 ya que nos proporcionan un alto torque y el rango de movimiento es suficiente para el proceso que se va a realizar, otorgándonos resultados excelentes en su aplicación, para el sistema piñón cremallera se diseñó ambos elementos para adaptarlo con un motor amarillo de 9 voltios.

Para el control de la posición del efector final se recurrió a un sensor de posición infrarrojo VL43LOX el cual con la aplicación de filtro se pudo diseñar un controlador digital PID para este sistema piñón-cremallera, así dando un punto fijo para el recojo de los objetos y su posterior subida para el posicionamiento de los objetos.

El modelado cinemático directo e inverso se realizaron a partir de la tabla DH que se encuentra en la tabla N°1, en el cual mediante el procedimiento matemático de las matrices de transformación homogénea:

Tabla I Diagrama D-H Robot Scara

Articulación	$\theta(\theta)$	d	a	$\alpha(\alpha)$
1	q_1	l_1	l_2	0
2	q_2	0	l_3	0
3	0	$-q_3$	0	0

Con ello hallamos las ecuaciones que rigen el mundo del robot, en el cual se relaciona las posiciones angulares y las posiciones en el plano del robot, las cuales son las siguientes:

$$x = L3 * \cos(q1 + q2) + L2 * \cos(q1) \dots \dots (1)$$

$$y = L3 * \sin(q1 + q2) + L2 * \sin(q1) \dots \dots (2)$$

$$z = L1 - q3 \dots \dots (3)$$

$$q1 = \tan^2\left(\frac{y}{x}\right) - \tan^2\left(\frac{L3 * \sin(q2)}{L2 + L3 * \cos(q2)}\right) \dots \dots (4)$$

$$q2 = \tan^2\left(\frac{\sqrt{1 - \left(\frac{x^2 + y^2 - L2^2 - L3^2}{2 * L2 * L3}\right)^2}}{\frac{x^2 + y^2 - L2^2 - L3^2}{2 * L2 * L3}}\right) \dots \dots (5)$$

$$q3 = L1 - z \dots \dots (6)$$

Para el resto de las partes de la plataforma de clasificación se realizaron la fijación por medio de pernos y tuercas asegurándolos así que no se muevan y que estén fijados a la plancha de madera principal.

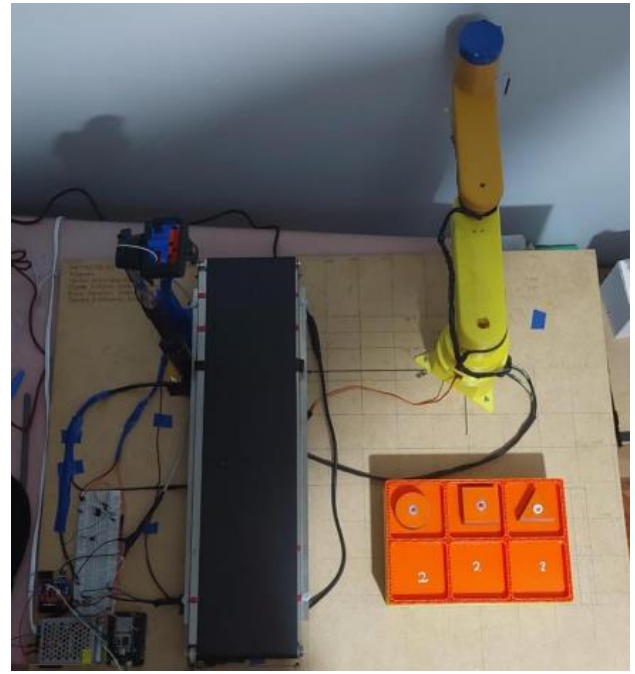


Fig. 5 Plataforma completamente ensamblada.

B. Visión Artificial

Con el fin de identificar tanto la posición real y el tipo de forma de la pieza, se vio viable utilizar la visión artificial. De este modo se buscó adquirir imágenes en tiempo real, procesarlas, evaluarlas y finalmente brindar información de la misma.

El desafío más importante fue la obtención de imágenes claras y fáciles de procesar, no obstante, por la simplicidad de la cámara ESP32CAM, una iluminación variable, ya que no se contó con un sistema de iluminación propio del sistema, es que se tuvo que tener en consideración ciertas técnicas de preprocesamiento como la segmentación por color, transformaciones geométricas y trabajar con el sistema HSV ya que resultó más viable trabajar la segmentación por color que con el método RGB.

Para la captura de imágenes, se utilizó una ESP32-CAM, que cuenta con una cámara integrada y conectividad WiFi, permitiendo transmitir un flujo de video continuo al programa en Python. Este dispositivo fue configurado con una resolución de 640x480 píxeles, lo que proporcionó un balance adecuado entre calidad de imagen y velocidad de procesamiento. Las figuras utilizadas en las pruebas eran geométricas (triángulos, cuadrados y círculos), fabricadas en

plástico y de colores naranja. Estas figuras se colocaron sobre un plano de trabajo gris mate, propia de la faja, lo cual minimizó los reflejos y facilitó la segmentación por color.

El software principal fue desarrollado en Python para el procesamiento de imágenes. Además, se empleó la librería requests para gestionar la comunicación entre Python y el SOC ESP32, encargado de controlar el robot SCARA. Para transformar las coordenadas de píxeles obtenidas en coordenadas reales dentro del plano de trabajo, se calculó previamente una matriz de homografía mediante la captura de puntos de referencia en el espacio físico.

La visión artificial se implementó en una serie de etapas diseñadas para garantizar una identificación precisa y eficiente de las figuras.

1. Captura de Imágenes

El flujo de video fue transmitido al programa en Python mediante el protocolo HTTP, donde cada cuadro fue capturado individualmente mediante la función `cv2.VideoCapture()` de OpenCV, lo que permitió realizar un procesamiento independiente en tiempo real para cada imagen.

2. Preprocesamiento de Imágenes

El preprocesamiento de las imágenes capturadas fue esencial para mejorar la detección de las figuras. En primer lugar, las imágenes en formato RGB se convirtieron al modelo de color HSV (Hue, Saturation, Value) utilizando la función `cv2.cvtColor()`. Este modelo fue elegido por su capacidad para separar el tono de color de las variaciones de iluminación, lo que facilita la segmentación precisa del color naranja.

Una vez convertido al modelo HSV, se definieron rangos de color para crear una máscara binaria que resaltara únicamente las figuras de color naranja mediante la función `cv2.inRange()`. Para eliminar ruido y mejorar la calidad de las máscaras, se aplicaron operaciones morfológicas como el cierre (`cv2.morphologyEx()` con `MORPH_CLOSE`), que rellena huecos dentro de las figuras, y la apertura (`cv2.morphologyEx()` con `MORPH_OPEN`), que elimina elementos pequeños no deseados. Finalmente, se utilizó un filtro gaussiano (`cv2.GaussianBlur()`) para suavizar los bordes de las figuras y mejorar la detección de contornos.

3. Detección de Figuras

La detección de figuras se realizó mediante la identificación de contornos en la máscara preprocesada utilizando la función `cv2.findContours()`. Cada contorno detectado fue aproximado a un polígono mediante `cv2.approxPolyDP()`, lo que permitió clasificar las figuras según el número de vértices: triángulos (3 vértices), cuadrados (4 vértices) y círculos (más de 10 vértices aproximadamente). Solo se consideraron contornos con un área mayor a un umbral predefinido, lo que ayudó a ignorar elementos irrelevantes o ruido residual.

4. Cálculo de Coordenadas Reales

Para determinar la posición de las figuras en el plano de trabajo, se calcularon sus centroides utilizando momentos geométricos mediante la función `cv2.moments()`. Las coordenadas del centroide obtenidas en píxeles fueron transformadas a coordenadas reales en centímetros utilizando una matriz de homografía previamente calculada. Este proceso permitió mapear la posición detectada en la imagen al sistema de referencia físico del robot SCARA, asegurando una localización precisa, con la ayuda de puntos ya conocidos procedemos a trazar una distancia en X y Y del punto central de cada figura.

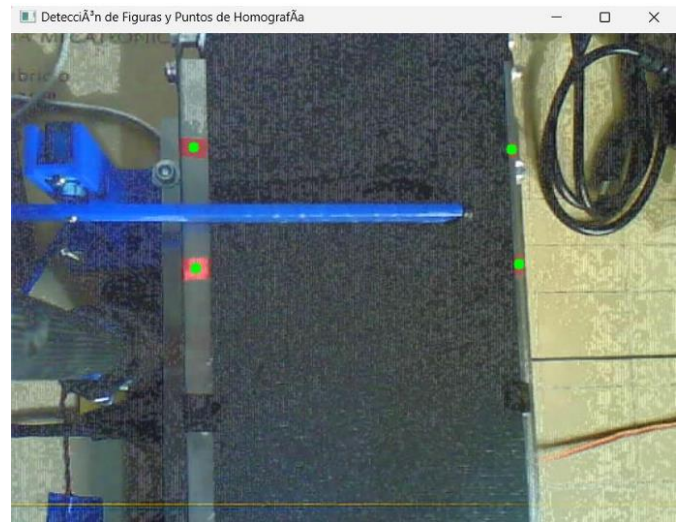


Fig. 6 Puntos ya conocidos para la detección de distancia.

5. Validación de Estabilidad

Para evitar errores debidos a fluctuaciones en las detecciones, se implementó un sistema de validación de estabilidad. Este sistema comparaba las posiciones detectadas en cuadros consecutivos y solo consideraba como estables aquellas que permanecían dentro de un rango definido durante

un número específico de cuadros. Esto redujo la probabilidad de enviar datos incorrectos al robot SCARA.

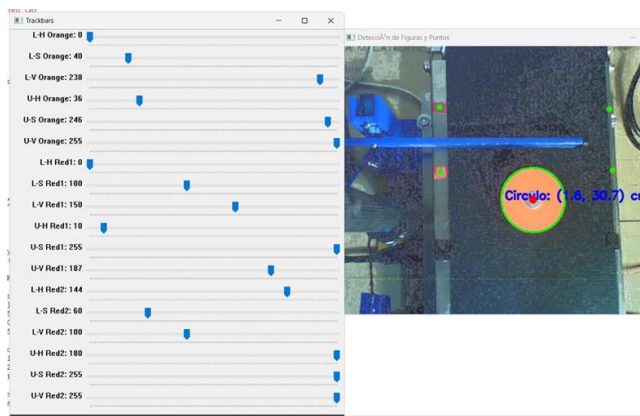


Fig. 7 Resultados obtenidos de la visión artificial

C. Sistema de Comunicación

El proceso de comunicación se hace mediante la librería Json de Arduino y la comunicación de estos fue mediante WiFi utilizando el Punto de Acceso de un teléfono celular, de esta manera la ESP32-CAM pasa dos números enteros (X y Y) y una letra (C, S, T), la C significa que la pieza es un círculo, la S que es un cuadrado y la T que es un triángulo.

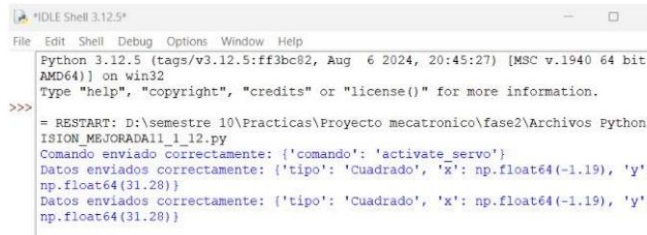


Fig. 8 Ejemplo de transmisión de datos.

Con ello y la integración de los modelados cinemáticos del SCARA se realizó el movimiento de los actuadores con la librería ESPServo, así moviendo las articulaciones 1 y 2. La articulación 3 tiene 2 medidas específicas, arriba y abajo, las cuales se colocarán mediante una función para hacer más simple el funcionamiento del robot.



Fig. 9 Transmisión de datos en tiempo real.

Con las coordenadas son almacenadas en el ESP32 del robot SCARA, así teniendo un ciclo de funcionamiento que se puede asemejar a la programación de un robot industrial, en la cual se le designa los puntos y el tipo de figura, con ello se obtiene el punto inicial y el punto final viene dado por la localización de los compartimentos de colocación que se muestra en la figura 6.

```
void asignarPosicion(char letra, float &xFinal, float &yFinal) {
    if (letra == 'T') {
        // Ciclo entre posT1, posT2
        if (contadorT == 0) {
            xFinal = posT1x;
            yFinal = posT1y;
            contadorT++; // Aumentar el contador de T
            Serial.println("Posición T1 asignada.");
        }
        else if (contadorT == 1) {
            xFinal = posT2x;
            yFinal = posT2y;
            contadorT = 0; // Reiniciar el contador de T para que vuelva a posT1
            Serial.println("Posición T2 asignada.");
        }
    }
    else if (letra == 'S') {
        // Ciclo entre posS1, posS2
        if (contadorS == 0) {
            xFinal = posS1x;
            yFinal = posS1y;
            contadorS++; // Aumentar el contador de S
            Serial.println("Posición S1 asignada.");
        }
        else if (contadorS == 1) {
            xFinal = posS2x;
            yFinal = posS2y;
            contadorS = 0; // Reiniciar el contador de S para que vuelva a posS1
            Serial.println("Posición S2 asignada.");
        }
    }
    else if (letra == 'C') {
        // Ciclo entre posC1, posC2
        if (contadorC == 0) {
            xFinal = posC1x;
            yFinal = posC1y;
            contadorC++; // Aumentar el contador de C
            Serial.println("Posición C1 asignada.");
        }
        else if (contadorC == 1) {
            xFinal = posC2x;
            yFinal = posC2y;
            contadorC = 0; // Reiniciar el contador de C para que vuelva a posC1
            Serial.println("Posición C2 asignada.");
        }
    }
    else {
        Serial.println("Figura no reconocida. Use 'T', 'S' o 'C'.");
    }
}
```

Fig. 10 Código para asignar posición.

```

void realizarTrayectoria(float xInicio, float yInicio, float xFinal, float yFinal) {
    float q1, q2;
    // Ir a los puntos iniciales (xInicio, yInicio)
    Prototipo_MCI(xInicio, yInicio, q1, q2);
    moverServos(q1, q2);
    delay(1000);
    // Tercer GDL: Establecer setpoint de la distancia linea
    controlMotor(85);
    // Activar el pin del imán
    digitalWrite(iman, HIGH);
    delay(500);
    // Cambiar setpoint de la distancia lineal a 60 mm
    controlMotor(60);
    delay(1000);
    // Ir a los puntos finales (xFinal, yFinal)
    Prototipo_MCI(xFinal, yFinal, q1, q2);
    moverServos(q1, q2);
    delay(1500);
    // Mover el servo de la faja a 95 grados
    myServo.write(95); // Posiciona el servomotor en el ángulo inicial (95°)
    // Establecer setpoint de la distancia lineal a 85 mm
    controlMotor(85);
    delay(1000);
    // Desactivar el pin del imán
    digitalWrite(iman, LOW);
    delay(500);
    // Asegurarse de que el setpoint de q3 esté en 60 mm al final
    controlMotor(60);
    delay(1500);
    Serial.println("Trayectoria completada.");
}

```

Fig. 11 Código para asignar posición.

Cuando la trayectoria termina el robot se queda en la última posición hasta que se detecte otra figura y comience nuevamente el proceso.

III. RESULTADOS

Pruebas de Identificación de Objetos

El sistema fue probado con formas geométricas comunes (círculo, cuadrado y triángulo). Se identificó tres tipos de objetos de diversos tamaños y composiciones, tomando en cuenta la cantidad de lados que tiene con un color naranja. La tasa de precisión fue superior al 95%, demostrando la robustez de los algoritmos de visión artificial implementados, esta tasa de precisión se puede ver influenciada por la cantidad de luz que existe en ese momento y las sombras que se forman en las figuras, la aplicación de filtros ayuda a el procesamiento de la imagen de mejor forma y más eficiente además de proporcionar mejores resultados a la hora de la identificación de su centro.

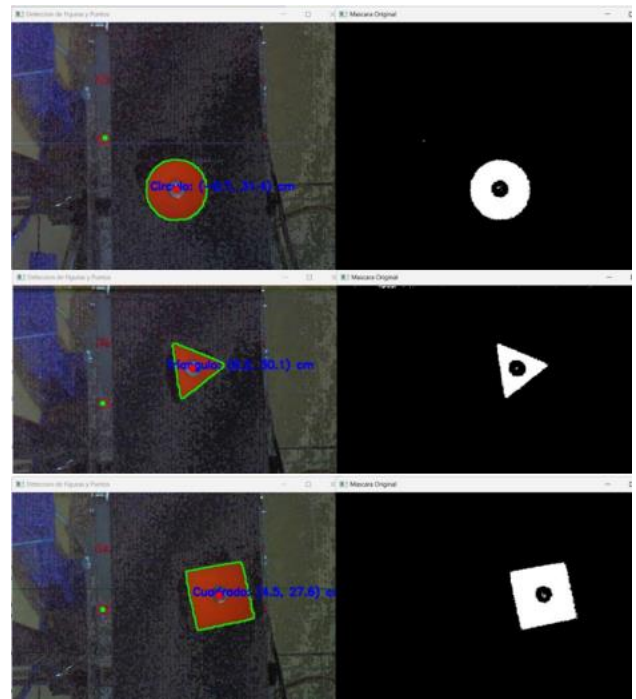


Fig. 12 Identificación de las distintas formas.

Pruebas de Clasificación

Se logró una clasificación exitosa de los objetos hacia sus compartimentos designados teniendo una ligera desviación del efector final al culminar una serie de diversas piezas en funcionamiento, la cual desvía el punto de entrega final un poco pero no demasiado como para colocarlo en otro compartimento. Los errores detectados fueron atribuidos principalmente a condiciones de iluminación poco óptimas y a algunos efectos de la utilización de servomotores MG955. Las posiciones también se reinician cuando el palet se llena, así creando un ciclo infinito de colocación de las piezas.

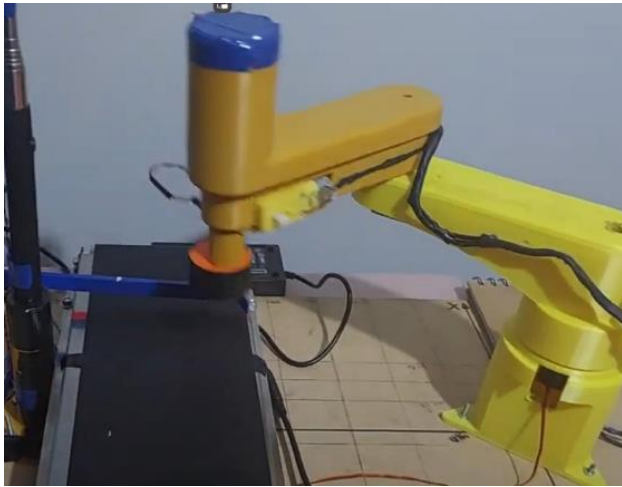


Fig. 13 Recogida de la pieza a recoger.



Fig. 14 Pieza en su punto Final.

Evaluación del Desempeño

El sistema procesó hasta 20 objetos en condiciones controladas. Teniendo en cuenta las velocidades que se manejó (10 segundos por pieza), el proceso completo es efectivo y con mucho futuro para ser aplicado en los distintos rubros de automatización como en la industria alimenticia o de elementos pequeños que necesiten ser empaquetados.

IV. DISCUSIÓN

El desarrollo del proyecto "Plataforma de Clasificación Automatizada con SCARA y Visión Artificial" permitió integrar tecnologías de visión artificial y robótica para resolver el problema de clasificación y manipulación automatizada de figuras geométricas. Los resultados obtenidos evidencian la capacidad del sistema para identificar y procesar figuras en tiempo real, pero también destacan aspectos que pueden ser mejorados para optimizar su desempeño.

En la etapa de visión artificial, la elección del modelo de color HSV demostró ser efectiva para segmentar figuras de colores específicos bajo condiciones controladas de iluminación. Sin embargo, se observó que el sistema es sensible a cambios drásticos en las condiciones de luz, lo que podría generar ruido en las máscaras generadas. Aunque las operaciones morfológicas y el suavizado de bordes ayudaron a reducir este efecto, sería beneficioso implementar técnicas más avanzadas, como algoritmos de balance de blancos dinámico o calibración adaptativa, para mejorar la robustez del sistema.

El cálculo de coordenadas reales mediante homografía fue preciso en entornos controlados, con desviaciones mínimas al comparar las posiciones estimadas con las reales. No obstante, el proceso de calibración inicial y la definición de puntos de referencia requieren una ejecución cuidadosa, ya que errores en esta etapa pueden propagarse y afectar la precisión global. Para aplicaciones más exigentes, podría considerarse el uso de cámaras de mayor resolución o algoritmos de calibración automática.

En cuanto a la transmisión de datos al ESP32, el uso de solicitudes HTTP POST permitió una comunicación efectiva entre Python y el SoC. Sin embargo, se identificó una limitación en el tiempo de respuesta, especialmente cuando se procesaban múltiples figuras de manera consecutiva. Esto resalta la necesidad de optimizar el manejo de datos, posiblemente mediante técnicas de transmisión asíncrona o sistemas de priorización de mensajes.

Por otro lado, el sistema de validación de estabilidad implementado para evitar detecciones erráticas fue efectivo al reducir errores en la transmisión de datos, pero agregó un retraso inherente al proceso de decisión. Este balance entre precisión y rapidez podría ajustarse mediante pruebas iterativas que definan umbrales óptimos de estabilidad para cada escenario.

Finalmente, desde una perspectiva operativa, la integración de los componentes físicos, como el robot SCARA y el electroimán, funcionó de manera fluida con el sistema de visión artificial. Sin embargo, el desempeño global podría beneficiarse de mejoras en la planificación de trayectorias del robot para minimizar tiempos de operación.

V. CONCLUSIONES

El proyecto "Plataforma de Clasificación Automatizada con SCARA y Visión Artificial" logró integrar visión artificial y control robótico para realizar tareas de clasificación automatizada de figuras geométricas de manera precisa y eficiente.

La implementación del sistema de visión artificial basado en la ESP32-CAM y OpenCV permitió identificar figuras geométricas en tiempo real mediante técnicas de segmentación por color, operaciones morfológicas y detección de contornos. El cálculo de coordenadas reales mediante una matriz de homografía garantizó una precisión adecuada en la localización de las figuras.

El uso de solicitudes HTTP POST para transmitir datos desde Python al SoC ESP32 demostró ser eficiente, aunque podría optimizarse para escenarios de alta demanda mediante técnicas de transmisión asíncrona.

La lógica de validación de estabilidad fue efectiva para garantizar que los datos enviados al robot SCARA fueran confiables, minimizando errores causados por fluctuaciones en las detecciones. Sin embargo, se identificaron oportunidades para equilibrar la precisión y el tiempo de respuesta.

La integración del robot SCARA y el sistema de visión artificial permitió ejecutar tareas de manipulación automatizada con una precisión destacable. La lógica optimizada para evitar movimientos innecesarios mejoró la eficiencia operativa del robot.

A pesar de los resultados positivos, el sistema puede ser mejorado trabajando sobre su sensibilidad a cambios en la iluminación, la optimización de tiempos de procesamiento y la robustez en la comunicación de datos. Estas oportunidades de mejora abren el camino para investigaciones futuras que busquen implementar técnicas más avanzadas de visión artificial y control robótico, fortaleciendo aún más las aplicaciones de este tipo de plataformas en entornos industriales y educativos.

REFERENCIAS

- [1] Y. Huang and S. Lian, "Vision-Based Control of SCARA Robots for Precision Assembly Tasks," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 876-886, 2015.
- [2] D. M. Tsai and W. Y. Chiu, "An Adaptive Vision System for Automatic Assembly by SCARA Robot," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 11, pp. 3920-3931, 2008.
- [3] D. Kragic and H. I. Christensen, "Survey on Visual Servoing for Manipulation," *Computational Vision and Active Perception Laboratory, Royal Institute of Technology*, Stockholm, Sweden, 2002.
- [4] E. J. Astudillo Flores and V. A. Dueñas López, "Diseño e implementación de un sistema de escaneo basado en visión artificial para la imitación del movimiento del brazo humano en el robot KUKA KR5-2 ARC HW," *Universidad Politécnica Salesiana*, 2024. [Online]. Available: <https://dspace.ups.edu.ec/handle/123456789/28515>
- [5] Universal Robots, "Visión Artificial en Robots," *Universal Robots Blog*, 2023. [Online]. Available: <https://www.universal-robots.com/es/blog/vision-artificial-en-robots>
- [6] J. Ghofrani, R. Kirschne, D. Rossburg, and D. Reichelt, "Machine Vision in the Context of Robotics: A Systematic Literature Review," *arXiv preprint arXiv:1905.03708*, 2019. [Online]. Available: <https://arxiv.org/abs/1905.03708>
- [7] H. Zhang and Y. Wang, "Development of a SCARA Robot for Precision Assembly Using Machine Vision," *International Journal of Advanced Robotic Systems*, vol. 18, no. 2, pp. 1-10, 2021.
- [8] X. Li and Z. Chen, "Integration of Vision Systems in SCARA Robots for Enhanced Pick-and-Place Operations," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101972, 2020.
- [9] P. Martínez and L. Gómez, "Aplicación de visión artificial en robots SCARA para la industria alimentaria," *Revista Iberoamericana de Automática e Informática Industrial*, vol. 19, no. 3, pp. 287-298, 2022.
- [10] S. Kumar and R. Verma, "Optimización de trayectorias en robots SCARA utilizando algoritmos de visión artificial," *Journal of Robotics and Automation*, vol. 35, no. 1, pp. 45-56, 2023.
- [11] M. López and J. Hernández, "Implementación de sistemas de visión artificial en robots SCARA para

ensamblaje electrónico," *Tecnología y Desarrollo*, vol. 18, no. 2, pp. 112-125, 2020.

[12] L. Wang and Q. Zhou, "Real-time Object Recognition for SCARA Robots Using Deep Learning," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3201-3209, 2021.

[13] A. Fernández and P. Ruiz, "Visión artificial aplicada al control de calidad en líneas de producción con robots SCARA," *Ingeniería y Competitividad*, vol. 21, no. 1, pp. 89-99, 2019.

[14] T. Nguyen and H. Le, "Enhancing SCARA Robot Accuracy with Machine Vision Feedback," *Journal of Manufacturing Processes*, vol. 60, pp. 765-773, 2022.

[15] R. Patel and A. Singh, "Machine Vision Integration in SCARA Robots for Pharmaceutical Packaging," *International Journal of Automation and Computing*, vol. 20, no. 5, pp. 657-669, 2023.

[16] L. García and M. Torres, "Desarrollo de un sistema de visión artificial para la calibración automática de robots SCARA," *Revista Latinoamericana de Robótica*, vol. 12, no. 4, pp. 345-358, 2021.