

IMPLEMENTACIÓN DE YOLOV8 PARA LA CLASIFICACIÓN DE CUBOS DE COLORES EN TIEMPO REAL

1st Jorge Abraham Gómez Blanco
Tecnológico Nacional de México
Tecnológico de Estudios Superiores de Coacalco
Coacalco de Berriozábal, México
jorge_abraham_gomez.mct@tesco.edu.mx

3rd José Antonio Valladares Farias
Tecnológico Nacional de México
Tecnológico de Estudios Superiores de Coacalco
Coacalco de Berriozábal, México
jose_valladares.ccai@tesco.edu.mx

2nd Julio César Saucedo Soto
Tecnológico Nacional de México
Tecnológico de Estudios Superiores de Coacalco
Coacalco de Berriozábal, México
201910291@tesco.edu.mx

4th Luis Edgar Alanís Carranza
Tecnológico Nacional de México
Tecnológico de Estudios Superiores de Coacalco
Coacalco de Berriozábal, México
luis.alanis@tesco.edu.mx

Abstract—Durante los últimos años, la utilización de sensores y elementos mecánicos en procesos industriales ha sido la solución más utilizada en la clasificación de objetos por color. Los sensores tienden a ser sensibles a condiciones ambientales, mientras que en el caso de los elementos mecánicos se requieren hacer adaptaciones a la línea de producción. El costo y la carga de trabajo en el mantenimiento aumentan debido a la cantidad de elementos necesarios en el sistema. En este artículo se propone implementar un modelo de visión por computadora basado en YOLO con el objetivo de clasificar cubos de colores (negro, rojo y rosa) en tiempo real. Los cubos de colores fueron elaborados mediante un software de diseño en línea y madera. Se capturaron 1,146 imágenes de los cubos fabricados en diversas iluminaciones, escenarios, ángulos y fondos con el fin de entrenar el modelo de YOLOv8. Los resultados obtenidos demostraron que el modelo es capaz de detectar y clasificar cubos de colores con un mAP50-95 del 0.914 %, así como de una velocidad de 0.4 ms por fotograma en tiempo real. Este avance tiene un gran potencial en diversas empresas y contribuye a mejorar la eficiencia y calidad en la producción mediante la detección de objetos por color en tiempo real.

Keywords—Yolov8, Visión por computadora, Clasificación de color, Detección de cubos.

I. INTRODUCCIÓN

A lo largo de los últimos años, el uso de la automatización en las diferentes etapas de producción se ha convertido en un estándar en la industria. Esto ha provocado que se desarrollen diferentes tecnologías para buscar una mayor agilidad en los procesos de producción y manufactura. Este acontecimiento ha propiciado la implementación de sensores y estructuras mecánicas. En la mayoría de las etapas del proceso, los sensores se sienten afectados por diferentes condiciones ambientales. Asimismo, los sensores como las estructuras mecánicas requieren una calibración y un mantenimiento periódico. La clasificación de cubos de colores en la industria no es una excepción, ya que, si se requiere modificar el tamaño y color, se requerirían cambiar varios componentes del sistema. Una forma de reducir la cantidad de cambios necesarios al incorporar otras características de cubos es la implementación de visión por computadora. [1] emplearon visión artificial y un brazo robótico con cuatro grados de libertad para clasificar cubos de tres colores: verde, rojo y azul. Llevaron a cabo un procedimiento de reconocimiento de color en función del modelo HSV, mediante un microcontrolador ATmega328. Una de las desventajas del trabajo fue el uso de un fondo oscuro para reconocer y detectar los colores de manera correcta. [2] construyeron un brazo

robótico con visión artificial controlada mediante una interfaz GUI interactiva. En el sistema se puede seleccionar el método de clasificación, ya sea por color, tamaño o forma. La precisión del modelo fue del 92.33 % en cuanto al color, 94.6 % para la forma y 98.6 % en el tamaño. [3] elaboraron un prototipo de clasificación de cubos de colores RGB, controlado por una FPGA. El sistema clasificador emplea sensores instalados en la banda transportadora. Asimismo, es capaz de reconocer y detectar únicamente tres colores, tales como rojo, azul y verde, mediante el empleo del sensor RGB. [4] emplearon un sistema de paletización para la clasificación de cajas a través de un entorno web. Los colores verde y rojo son reconocidos por el sistema de visión artificial, el cual utiliza dos redes neuronales. Este sistema se controla mediante un PLC S7-1200 y una Raspberry, lo que posibilita la manipulación del proceso a través de la pantalla HMI. [5] implementó YOLOv4 para detectar 4 cubos de diferente color (rojo, azul, verde y amarillo). La aplicación principal de este dispositivo es el control de un robot SCARA, el cual será capaz de detectar la posición del cubo en el entorno de trabajo. Esto permitirá al robot establecer la ubicación de trabajo en concordancia con las áreas de mayor a menor peligro. [6] elaboraron un sistema de visión artificial con el fin de efectuar la clasificación de latas de pintura, considerando el color de esta. Las técnicas de procesamiento digital de imágenes empleadas fueron la binarización y la operación morfológica para el reconocimiento de las latas. El sistema obtuvo un rendimiento del 98.33 %, teniendo en cuenta una proporción de 41.12 lúmenes. [7] utilizaron YOLOv3 con el fin de detectar la madurez de los tomates en función de su color: verde inmaduro, amarillo o naranja semi maduro, rojo brillante maduro y rojo oscuro sobre- maduro. Les llevo 2 años recolectar la información necesaria para la base de datos, y de esta forma, realización el entrenamiento del modelo. [8] aplicaron el modelo YOLOv2, el cual permite identificar las luces de semáforos peatonales, que solo presentan dos estados: verde para avanzar y rojo para detenerse. En la actualidad, existen sistemas de visión artificial para la detección y reconocimiento de cubos de colores. No obstante, no se ha implementado el modelo de última generación Yolov8 para este problema. La contribución de este proyecto es la implementación de este modelo para detectar y reconocer cubos de diferentes colores. Los cubos poseen una misma forma y tamaño, sin embargo, se distinguen por su color.

II. OBJETIVOS

- Diseñar y construir cubos de diferentes colores con el mismo tamaño.
- Crear y preparar una base de datos que contenga imágenes de los cubos en diversos escenarios e iluminaciones.
- Entrenar y evaluar el rendimiento de YOLOv8 mediante el conjunto de datos.
- Implementar el modelo entrenado en una cámara web para la clasificación de los cubos de colores en tiempo real.

III. METODOLOGÍA

A. Fabricación de cubos de colores

Se elaboraron cubos diseñados en el sitio web Maker Case, con dimensiones de 50 x 50 x 50 milímetros (mm) Fig. 1. Estos cubos se diseñaron con pestañas con el fin de facilitar el armado de cada uno de ellos, y se pintaron mediante pintura acrílica de color rojo, negro y rosa.

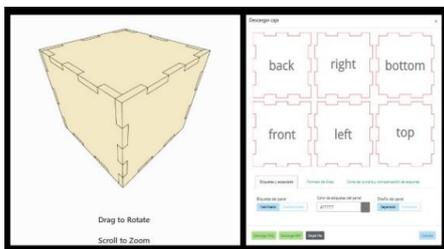


Figura 1. Diseño de los cubos

El material utilizado para la elaboración de los cubos de colores fue la madera, ya que se disponía de las herramientas necesarias para un manejo fácil de este material. Asimismo, este material tiene la facilidad de pintarse con facilidad. Los cubos, por último, fueron pintados de 3 colores diferentes: rojo, negro y rosa, como se muestra en la Fig. 2.

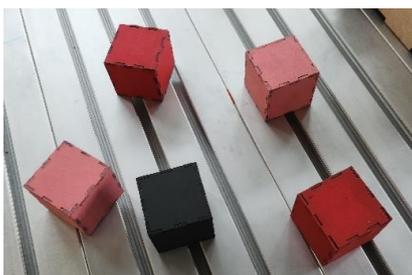


Figura 2. Cubos armados y pintados

B. Recolección y preparación de los datos

Para entrenar un modelo de visión artificial, que posibilite la detección y reconocimiento de objetos, es necesario elaborar una base de datos que contenga imágenes del objeto de interés. El número total de fotografías recolectadas fue de 382 por cada color y está conformada de tres categorías (negro, rojo y rosa). En algunas imágenes se modificó la posición y el ángulo del objeto al momento de la captura. Asimismo, se tomaron

fotografías del objeto en diferentes escenarios y condiciones de iluminación, para obtener una base de datos más completa y versátil, ejemplo Fig. 3.

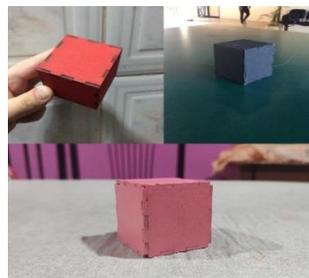


Figura 3. Ejemplos de escenarios.

La cámara que se usó para la captura de datos tiene una resolución de 1080 x 720 píxeles, lo cual es similar a la que se implementará en tiempo real como se muestra en la Fig. 4.

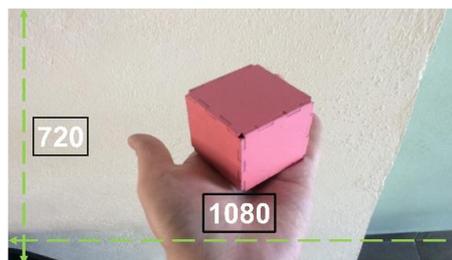


Figura 4. Tamaño de las fotografías

El entrenamiento del modelo requiere de las coordenadas y la categoría del objeto que se encuentra presente en la imagen. Para lograr tal objetivo, se empleó un algoritmo denominado Labeling, desde la consola de “Anaconda”, el cual permite indicar manualmente las coordenadas del objeto de interés en la imagen completa Fig. 5.

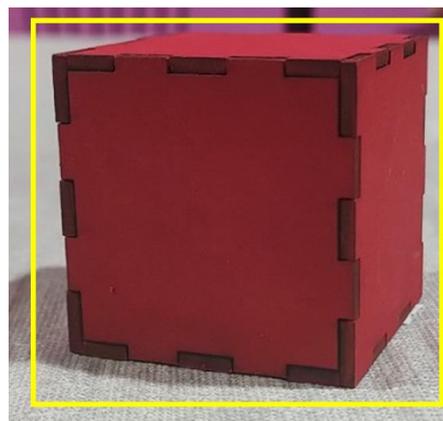


Figura 5. Etiquetado manual de coordenadas

La información se guarda en un archivo de tipo block de notas y contiene las coordenadas de los objetos que se delimitaron manualmente.

C. YOLOv8

YOLO (You Only Look Once) o, en su traducción al español (Solo Miras Una Vez), tiene 16 versiones desde 2016 [9]. La arquitectura de este modelo permite detectar y reconocer objetos en tiempo real y utiliza redes neuronales convolucionales. [10] El modelo de YOLO fue creado por Joseph Redmon, Santosh Divvala, Ross Girshick y Ali Farhadi en 2016. Asimismo, emplea la técnica clásica de aprendizaje automático denominada red neuronal, la cual simula cómo el cerebro humano procesa la información Fig. 6. Como su nombre lo indica, se basa en las neuronas, las cuales están interconectadas y buscan resolver un problema específico.

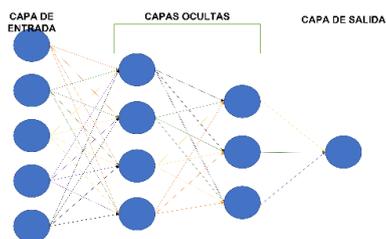


Figura 6: Red Neuronal

Este método permite clasificar objetos mediante la extracción de características mediante el uso de operaciones matemáticas en cada una de las capas [11]. El modelo YOLO, se distingue por su enfoque único en el análisis de objetos. La cual no divide las tareas en múltiples etapas, YOLO lleva a cabo un análisis completo de la imagen en una sola pasada, a través de una sola red neuronal. Este enfoque ejecuta la red neuronal en toda la imagen y descarta de manera automática la información no relevante, basándose en la selección de los objetos a identificar. La primera capa del modelo contiene una gran cantidad de información innecesaria. Al progresar las capas convolucionales, la cantidad de información irrelevante disminuye significativamente. Este proceso de reducción progresiva de información superflua es fundamental para determinar la categoría del objeto. La empresa Ultralytics, una compañía especializada en la investigación de tecnologías de visión por computadora, presentó una versión mejorada de YOLOv5 [12]. Esta versión es la ocho y ha tenido precisiones mayores que en sus versiones anteriores debido a su estructura de detección. El modelo de YOLOv8 tiene 5 modificaciones: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) y YOLOv8x (extra large). Estas modificaciones se distinguen por su precisión y recursos computacionales, y cuanto mayor sea el modelo, se obtiene una mayor precisión, sin embargo, se requiere de mayores recursos computacionales. Por el contrario, si se utiliza con un menor tamaño, se obtendrá menor precisión, pero un incremento en la velocidad de procesamiento.

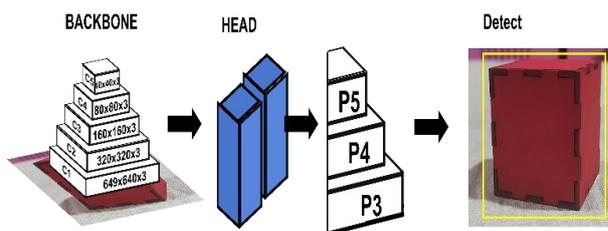


Figura 7 Red Neuronal YOLOv8

La red neuronal YOLOv8 se compone de múltiples fases para llevar a cabo detecciones precisas de objetos en imágenes de diversos tamaños, mediante diferentes operaciones matemáticas [13]. En la capa de Backbone (Cuerpo Principal), se procesa la imagen de entrada mediante múltiples capas con el fin de extraer características fundamentales. A medida que avanzan las capas, la imagen se reduce de tamaño y se capturan características a diversas escalas. La capa Head (cabeza) procesa la información recopilada de las capas anteriores. Se observa dos torres, que son representaciones de cómo se analiza y procesa la información como si fuera una computadora. Al utilizar tres bloques, se indica cómo se llevan a cabo las predicciones al interactuar con las imágenes de la base de datos y como esta va a analizar las imágenes. Esta subdivisión es importante para abordar objetos de diferentes tamaños, asegurando que YOLOv8 pueda detectar de manera correcta los objetos. Se utilizan los resultados de las predicciones para generar las salidas finales que indican la presencia y ubicación de objetos en la imagen durante la etapa de detección. Asimismo, este establecerá las categorías a las que pertenece ese objeto y otras características relevantes. El enfoque modular de YOLOv8 posibilita procesar y analizar adecuadamente las características de la imagen en cada fase, lo que contribuye a su elevado rendimiento en la detección de objetos.

D. Entrenamiento del modelo

Para el entrenamiento del modelo, la base de datos se dividió en 3 conjuntos: entrenamiento, validación y prueba. El conjunto de entrenamiento será el 80 % de la base total. En el caso del conjunto de validación, el 10 % será necesario para evaluar el rendimiento del modelo en cada una de las iteraciones durante el entrenamiento. Por último, el 10 % restante se usará para evaluar el rendimiento del modelo entrenado. Los parámetros empleados para el entrenamiento del modelo de YOLOv8 son mostrados en la siguiente tabla.

TABLA I. PARÁMETROS DE ENTRENAMIENTO

Entrenamiento	
epochs	800
imgsz	640
modelo	Yolo8n
batch	2
device	0
patience	150

De acuerdo con la *Tabla I*. La cantidad de épocas fue de 800, ya que después de 500 no había mejora en el rendimiento del modelo al probarlo con el conjunto de validación. *imgsz* modifica la imagen de entrada previo al ingreso a la primera

capa convolucional, con un tamaño de 640 x 640 píxeles, como se observa en la Fig. 7.

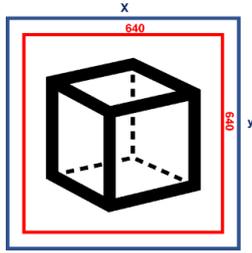


Figura 8. Primera capa convolucional

Se utilizó el modelo de YOLOv8n para reducir el proceso necesario para detectar en cada fotograma del video el objeto sin perder tanta precisión. A causa de las limitaciones del ordenador empleando en el entrenamiento, se usaron los parámetros bajos para el batch y el device. El *patience* determina si es necesario proseguir con el entrenamiento en función de si hay o no mejoras después de una determinada cantidad de épocas. Tras 150 épocas, el modelo dejó de presentar mejoras en el rendimiento y detuvo el entrenamiento.

IV. RESULTADOS

A. Conjunto de validación

Los resultados del entrenamiento se muestran en la gráfica 1.

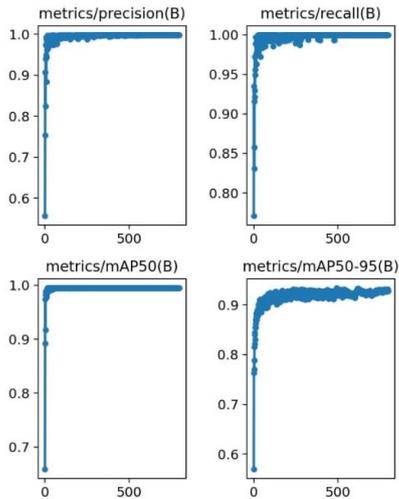


Figura 9. Entrenamiento del modelo

Las gráficas de la Fig.9 muestran el progreso del entrenamiento del modelo a lo largo de las épocas, con un total de 800 épocas consideradas en el eje de X. La primera gráfica, etiquetada como precisión, muestra cómo el modelo comienza su entrenamiento. En un principio, el modelo no puede reconocer las formas de los objetos, y su precisión es relativamente inferior al 0.8 %. No obstante, a medida que transcurren las épocas, aproximadamente en la época 10, el modelo comienza a comprender y señalar los objetos con una precisión mejorada. Llegando a valores superiores al 0.7 % pero aún por debajo del 0.8 %. A medida que las épocas transcurren, el modelo sigue mejorando su precisión.

Aproximadamente a la época 500, se observa un punto en el que la precisión deja de aumentar significativamente y se estabiliza en un 0.9 %. La segunda gráfica, denominada como recall muestra cómo el modelo retiene y recuerda las características de las imágenes durante el entrenamiento. Esta métrica demuestra la capacidad del modelo para reconocer y recordar los datos más relevantes. A medida que el modelo avanza, se puede observar un incremento constante en la métrica. En un principio, esta métrica comienza con un 0 %. No obstante, a medida que el modelo acumula conocimiento, este alcanzará un valor superior al 0.95 %. A partir de la época 500, la gráfica muestra una estabilización, lo que indica que el modelo ha aprendido de manera efectiva y ha capturado las características clave de las imágenes. A partir de ese punto, no se observan cambios significativos en la métrica de recuerdo, y permanece en un nivel alto y constante. La gráfica que tiene la métrica mAP50 se enfoca en evaluar el rendimiento del modelo mediante la comparación del cuadro delimitador encerrado manualmente contra el detectado por el modelo.

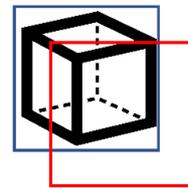


Figura 10: Map50

Al observar la Fig. 10 se señala que el cuadro azul se encuentra en posición marcada manualmente y el color rojo es la detección del modelo. Esto indica cómo el cuadro de detección se ajusta y puede captar una porción del objeto. En las primeras etapas del entrenamiento, el mAP50 alcanza un 0.90 % de la detección del objeto deseado. Sin embargo, a medida que el modelo acumula experiencia, especialmente después de aproximadamente 20 épocas, se observa un cambio significativo. La métrica mejora significativamente y supera el 0.95 %. A partir de este punto, la gráfica muestra una estabilización en la precisión de la detección, lo que indica que el modelo ha aprendido a reconocer y localizar los objetos de interés. La última gráfica con la métrica mAP50-95 se centra en evaluar el rendimiento del modelo en umbrales de 0.50 a 95 en pasos pequeños de 0.1 Fig. 11. Esto demuestra la capacidad del modelo de reconocer y detectar objetos, incluso cuando solo detecta una pequeña parte del objeto.

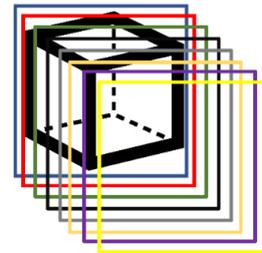


Figura 11. Map50 - 95

Estas gráficas indican que el modelo es capaz de reconocer y detectar objetos, incluso cuando solo se puede detectar una pequeña parte del objeto. A pesar de que esta métrica puede experimentar fluctuaciones, demuestra la capacidad del modelo para realizar detecciones precisas, incluso en situaciones de visibilidad reducida. Las métricas previamente mencionadas fueron obtenidas durante el entrenamiento mediante el uso del conjunto de validación. Los promedios de cada una se presentan en la siguiente tabla:

TABLA II. RESULTADOS DE ENTRENAMIENTO-VALIDACIÓN

Clases	Imágenes	Precisión	Recall	mAP50	mAP50-95
general	114	0.998	1	0.995	0.934
negro	114	0.998	1	0.995	0.948
rojo	114	0.998	1	0.995	0.938
rosa	114	0.999	1	0.995	0.917

Los resultados del entrenamiento con el conjunto de validación indican que el modelo mantiene una alta precisión, sin disminuir por debajo del 90 %. Esto indica que el modelo es robusto y fiable en la detección de cubos de colores. Los valores obtenidos en las métricas de precisión, recall y en las puntuaciones Box mAP50 y Box mAP50-95 respaldan la efectividad del modelo en su capacidad para clasificar los objetos.

B. Conjunto prueba

El modelo entrenado fue evaluado mediante el conjunto de prueba y los resultados se muestran en la siguiente tabla.

TABLA III. RESULTADOS DE ENTRENAMIENTO-PRUEBA

Clases	Instancias	Precisión	Recall	mAP50	mAP50-95
general	114	0.987	0.983	0.993	0.914
negro	39	0.999	1	0.995	0.914
rojo	36	0.988	0.972	0.988	0.906
rosa	39	0.974	0.976	0.994	0.922

La primera columna presenta las categorías que se están analizando y la segunda se refiere a la cantidad de imágenes que se evaluaron con el modelo entrenado por categoría. La precisión y el recall son mostradas en la tercera y cuarta columna, obteniendo una precisión total del 98 % y un recall del 98 %.

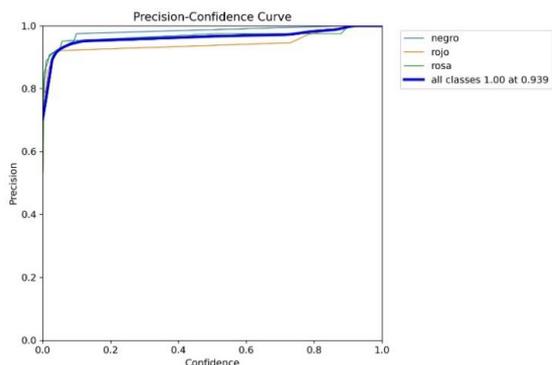


Figura 12. Gráfica curva de Precisión.

Como se muestra en la gráfica Fig. 12, los cubos de color negro no tienen dificultades para su detección y tienen una probabilidad del 0.999 %. Por el contrario, los cubos de color rojo y rosado presentan una probabilidad ligeramente menor debido a su tendencia a confundirse, especialmente en condiciones de iluminación desfavorable y tonalidades similares. Los cubos de color rojo alcanzan una probabilidad del 0.988 %, mientras que los cubos rosados obtienen una probabilidad del 0.974 %. La línea gruesa en la gráfica representa la probabilidad global de clasificación de todos los colores, con un 0.987 % de precisión general.

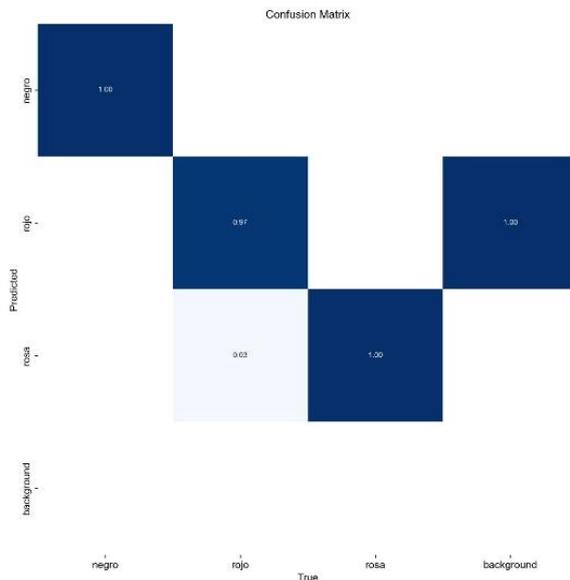


Figura 13. Matrix de confusión

La matriz de confusión Fig. 13 se distingue al tener 4 clases, dado que incluye el fondo. En el lado derecho, existe un porcentaje que, cuanto mayor probabilidad tenga, tendrá una tonalidad más oscura, llegando a ser un valor del 1 %. Por el contrario, en menor probabilidad, la tonalidad del color reducirá llegando a blanco. Los cubos de color negro se distinguen claramente de los otros cubos y fondo, alcanzando un rango del 1 %. No obstante, los cubos de color rojo pueden confundirse con el fondo, especialmente debido a la utilización de fondos rojos en el conjunto de prueba.

Finalmente, los cubos rosas también pueden confundirse con los cubos rojos, pero esta depende de la tonalidad y la confusión tiene un valor debajo del 0.03 %. En conjunto, estos resultados destacan el óptimo desempeño del modelo en la clasificación de cubos de colores.

C. Implementación en imágenes y en tiempo real

Con el fin de evaluar el desempeño del modelo entrenado, se implementaron dos tipos de códigos: uno destinado a la lectura de imágenes y otro diseñado para operar en tiempo real. Se emplearon nuevas fotografías que no se utilizaron para el entrenamiento del modelo. Para determinar si el modelo era capaz de detectar los cubos de colores y clasificarlos.

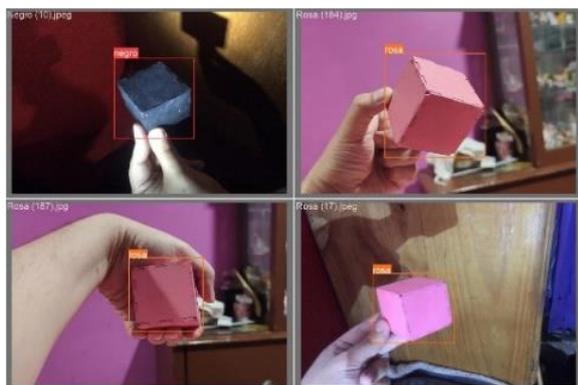


Figura 14. Lectura de fotografías nuevas

Se utilizó una cámara para implementarlo en tiempo real, y el modelo se encarga de clasificar los cubos de colores a una velocidad de 0.4 ms por fotograma. No obstante, la iluminación adecuada desempeña un papel fundamental en la precisión de la clasificación. En áreas con una buena iluminación, el modelo logra tener predicciones favorables, como se puede apreciar en las imágenes capturadas.



Figura 15. Prueba en tiempo real

Este resultado demuestra la versatilidad del modelo en la detección y clasificación de cubos de colores tanto en escenarios estáticos Fig. 14 como en tiempo real Fig. 15. Asimismo, recalcan la importancia de asegurar las condiciones de iluminación apropiadas para optimizar el rendimiento del modelo en aplicaciones prácticas.

V. CONCLUSIONES

La visión por computadora puede usarse de manera eficaz para controlar y monitorear áreas industriales sin la necesidad de realizar modificaciones costosas en las líneas de producción. Esta solución supone un importante paso hacia la eficacia y la mejora de la calidad en una variedad de entornos industriales. En este trabajo se utilizó el modelo YOLOv8, el cual es capaz de detectar y clasificar los cubos de colores de forma eficiente. Los cubos de colores fueron diseñados y construidos mediante madera para su pintado sencillo. La base de datos contiene un total aproximado de 400 imágenes por categoría de color recopiladas a través de una cámara de 1080 x 720 píxeles en diversos escenarios. El modelo fue entrenado con el 80 % del conjunto total, lo que dio como resultado una precisión superior al 90 %. La velocidad de procesamiento en tiempo real por fotograma fue de 0.4 ms mediante el uso del modelo entrenado. Este logro representa un avance significativo en la detección de colores en objetos tridimensionales y tiene aplicaciones potenciales en una amplia variedad de campos de la industria. En su instalación, no sería necesario efectuar modificaciones significativas en las áreas de producción existentes. No obstante, es fundamental asegurar una iluminación adecuada para evitar errores en la identificación de colores. La presente circunstancia podría ser debida a la similitud de la tonalidad en algunas situaciones, lo que podría generar confusión en el modelo.

REFERENCIAS

- [1] Buitrón Tandalia, I. A. (2023) . Análisis de rendimiento de you only look once, retinanet y single shot detector aplicado a la detección y conteo vehicular (Master's thesis, Quito: EPN, 2023.) .
- [2] Ramos Castro, R. (2023) . Conteo automatizado mediante visión artificial con herramientas de deep learning para el cálculo de la medida IMD.
- [3] Parés Marín, G. (2023) . Desarrollo de un sistema de detección de prendas de ropa en imágenes de moda mediante la implementación de redes YOLOv8 personalizadas (Doctoral dissertation, Universitat Politècnica de València) , pp.10-12.
- [4] Acebey, Mateo Rangel (2021) Presentación de documentos finales de taller de grado II
- [5] Blandón Quiroz, A., & Holguín Ramírez, K. (2022). Desarrollo de un sistema de control control basado en visión de máquina para la plataforma robótica SCARA de la Universidad EIA.
- [6] Acosta, L. A., Chango, E. F., & Moyon, J. F. (2020). Sistema de visión artificial para clasificación de latas de pintura por color considerando el espacio de color RGB. Espacios, 41, 18.
- [7] Liu, G., Nouaze, J. C., Touko Mbouembe, P. L., & Kim, J. H. (2020). YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. Sensors, 20(7), 2145.
- [8] Liu, G., Nouaze, J. C., Touko Mbouembe, P. L., & Kim, J. H. (2020). YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. Sensors, 20(7), 2145.
- [9] Zhang, X., Qiu, Z., Huang, P., Hu, J., & Luo, J. (2018, October). Application research of YOLO v2 combined with color identification. In 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 138-1383). IEEE.
- [10] Buitrón Tandalia, I. A. (2023). Análisis de rendimiento de you only look once, retinanet y single shot detector aplicado a la detección y conteo vehicular (Master's thesis, Quito: EPN, 2023.) .
- [11] Ramos Castro, R. (2023). Conteo automatizado mediante visión artificial con herramientas de deep learning para el cálculo de la medida IMD.
- [12] De, M. J. I. (2021). Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning. Universidad de Valladolid. <https://uvadoc.uva.es/handle/10324/45359>

- [13] Parés Marín, G. (2023). Desarrollo de un sistema de detección de prendas de ropa en imágenes de moda mediante la implementación de redes YOLOv8 personalizadas (Doctoral dissertation, Universitat Politècnica de València), pp.10-12.