

Software De Generación De Esquemáticos Eléctricos A Partir De Imágenes De Circuitos Montados En Protoboard

➤ Autores

Enrique Antonio Calderón Flores

Universidad Mayor, Real Y Pontificia De San Francisco Xavier De Chuquisaca

Correo: antoniocfbb17@gmail.com

➤ Resumen

Este estudio presenta el desarrollo de un software basado en inteligencia artificial para generar esquemáticos eléctricos a partir de imágenes de circuitos montados en protoboards, con un enfoque en entornos educativos. Utilizando la arquitectura YOLOv8, se diseñó un pipeline modular que combina detección de objetos y segmentación por instancias para identificar componentes electrónicos, cables, pines y carriles, logrando una precisión promedio de 90%. La metodología incluyó la recolección de datasets personalizados con imágenes de alta resolución (4K), preprocesamiento con OpenCV y entrenamiento de modelos específicos para cada etapa del análisis. Los resultados muestran métricas de precisión (mAP@0.5) de hasta 0.995 para detección de zonas clave y 0.692-0.97 para otras tareas. Este software reduce errores en la verificación de circuitos, facilita el aprendizaje práctico y mejora la seguridad en entornos educativos. Las conclusiones destacan su utilidad para estudiantes de electrónica y recomiendan mejoras en datasets y robustez para circuitos complejos.

➤ Palabras clave

Inteligencia Artificial, Visión por Computadora, YOLOv8, Protoboards, Esquemáticos Eléctricos, Educación en Electrónica, Segmentación por Instancias, Detección de Objetos.

➤ Introducción

El armado de circuitos electrónicos en protoboards es una práctica fundamental en la enseñanza y prototipado de proyectos electrónicos, permitiendo validar diseños antes de fabricar placas de circuito impreso (PCBs). Sin embargo, la verificación manual de estos montajes es propensa a errores como conexiones incorrectas, cortocircuitos o componentes mal ubicados, lo que puede dañar elementos sensibles, aumentar costos y dificultar el aprendizaje, especialmente para principiantes.

En los últimos años, la inteligencia artificial (IA), en particular la visión por computadora, ha revolucionado la inspección de PCBs y la segmentación de cables mediante técnicas avanzadas. Por ejemplo, Hu et al. (2023) aplicaron modelos de aprendizaje profundo para identificar componentes en placas PCB, mientras que Zhang et al. (2021) demostraron la viabilidad de segmentar trayectorias de cables en imágenes de circuitos electrónicos mediante segmentación semántica, aunque sin distinguir instancias individuales de cables. En un contexto más amplio, Chiu et al. (2023) desarrollaron un sistema para segmentar y eliminar cables en imágenes de alta resolución, como cables eléctricos en escenas naturales, utilizando segmentación semántica, pero no enfocado en circuitos electrónicos. Por otro lado, el sistema CVASP intentó reconstruir conexiones en protoboards utilizando únicamente técnicas clásicas con OpenCV, pero limitado a cables rectos y sin integrar otros elementos del circuito como componentes, patillas o carriles (Rahnama & Tsai, 2020).

La relevancia de este trabajo radica en su potencial para reducir errores, mejorar la seguridad y apoyar la enseñanza práctica de la electrónica, fortaleciendo el vínculo entre teoría y práctica mediante una herramienta visual, precisa e interactiva que fomenta un aprendizaje autónomo y confiable.

Este estudio propone un software integral que combina detección y segmentación por instancias mediante modelos YOLOv8 para analizar circuitos montados en protoboards, con el objetivo de identificar cables, componentes, pines, carriles de alimentación y la geometría del protoboard a partir de imágenes reales, y reconstruir automáticamente un esquemático eléctrico con al menos un 90% de precisión.

➤ Metodología

Diseño del Estudio

El proyecto adoptó un enfoque experimental e iterativo, estructurado en un pipeline de visión por computadora basado en IA, desarrollado en el entorno de la Facultad de Ciencias y Tecnología de la Universidad Mayor, Real y Pontificia de San Francisco Xavier de Chuquisaca. Se diseñaron siete modelos YOLOv8 especializados para tareas de detección y segmentación, integrados con técnicas de análisis espacial para generar esquemáticos eléctricos precisos. El sistema opera en una arquitectura cliente-servidor, con una aplicación móvil Android para captura de imágenes y un servidor Linux para procesamiento.

Recolección de Datos

Se recolectaron aproximadamente 2100 imágenes únicas de protoboards con circuitos reales, capturadas en entornos controlados, incluyendo instalaciones de la Facultad de Ciencias y Tecnología de la Universidad Mayor, Real y Pontificia de San Francisco Xavier de Chuquisaca y un espacio de desarrollo personal. Estas imágenes forman la base para los conjuntos de entrenamiento, validación y prueba, con solapamientos limitados entre tareas, ya que algunas imágenes se utilizaron para múltiples propósitos (detección, segmentación o ambas) y otras fueron recortadas o anotadas específicamente para tareas particulares. Los datasets con dimensiones fijas corresponden a imágenes completas de protoboards, mientras que aquellos sin dimensiones especificadas utilizan recortes de tamaños variados centrados en componentes o regiones específicas. La conformación de cada dataset se basó en criterios específicos según el tipo de anotación y el propósito del análisis, empleando distintas resoluciones, técnicas de recorte, mejora de contraste y enfoques de segmentación.

Los conjuntos se detallan a continuación:

- **Segmentación de cables:** 947 imágenes (1024×1024), con entre 5 y 15 cables por imagen. Se aplicó segmentación por instancia con énfasis en cruces y trayectorias complejas.
- **Detección de pines en DIPs y sensores:** 260 imágenes con entre 8 y 40 pines por imagen, incluyendo condiciones de baja iluminación y resolución. Anotadas mediante *bounding boxes* individuales por pin.
- **Segmentación de zonas clave del protoboard:** 233 imágenes etiquetadas con *bounding boxes* para identificar zonas de alimentación y la zona central.
- **Segmentación de patillas delgadas:** 740 imágenes obtenidas por recortes de componentes tipo LED y resistencia. Se aplicó un aumento de contraste agresivo (+60) para facilitar la anotación por polígonos en elementos de bajo grosor.
- **Detección de componentes electrónicos:** 422 imágenes (800×800), con *bounding boxes* para resistencias, DIPs, LEDs, botones y sensores.
- **Segmentación de extremos de cables:** 227 imágenes enfocadas en puntas de cables, con énfasis en conexiones individuales o múltiples.
- **Detección de carriles conductores:** 127 imágenes, con anotaciones de *bounding boxes* para los carriles horizontales de alimentación en el protoboard.

Las imágenes se preprocesaron con OpenCV (redimensionamiento a 3000×3000 , suavizado, conversión BGR a RGB) y se anotaron en Roboflow para segmentación por instancias y detección, exportándose en

formato YOLO compatible con YOLOv8 para ambas tareas. Se aplicó aumento de datos con Albumentations (rotaciones, cambios de brillo, perspectiva) y una división de 75% para entrenamiento, 15% para validación y 10% para pruebas.

Instrumentos y Herramientas

- **Cliente:** Aplicación Android desarrollada en Android Studio con Kotlin 1.9.24, utilizando OpenCV 4.10 para captura y preprocesamiento, y Retrofit para comunicación HTTP.
- **Servidor:** Implementado en Python con Flask, ejecutando modelos YOLOv8 en un sistema Ubuntu con AMD Ryzen 7 5800H y GPU Radeon RX 6600M.
- **Modelos YOLOv8:** Entrenados con hiperparámetros específicos (épocas: 30-300, batch: 2-16, resoluciones: 320x320 a 1024x1024, optimizador AdamW).

Procedimientos

1. **Captura de Imagen:** La app móvil captura imágenes de protoboards en resolución 4K (3840×2160 píxeles), codificándolas en Base64. Una guía visual, basada en OpenCV con el algoritmo Canny, detecta bordes usando el contraste entre el protoboard y el fondo, permitiendo la fotografía solo si el porcentaje de bordes detectados en las líneas izquierda y derecha supera un umbral mínimo (minEdgePercentage).
2. **Procesamiento en Servidor:** El servidor recibe la imagen y ejecuta el pipeline en etapas secuenciales:
 - **Detección de componentes:** Detecta los componentes electrónicos en el circuito armado del protoboard.
 - **Segmentación de cables:** Segmenta por instancias todos los cables del circuito.
 - **Segmentación de extremos de cables:** Segmenta los extremos de los cables y los enlaza a la máscara correspondiente obtenida de la segmentación de cables para identificar sus extremos.
 - **Segmentación de patillas delgadas:** Filtra de la detección de componentes elementos como resistencias y LEDs, segmentando la posición de sus patillas para determinar dónde están conectadas.

- **Detección de pines DIPs y sensores:** Filtra de la detección de componentes los DIPs y sensores (como el ultrasónico), detectando pines y muescas regulares y simétricos como elementos clave.
 - **Detección de zonas clave del protoboard:** Detecta los carriles de alimentación y la división central del protoboard para ubicar los puntos de conexión de los componentes, distinguiendo si están en carriles centrales o de alimentación.
 - **Correcciones espaciales y geométricas:** Incluye interpolación de pines DIPs faltantes, ajuste de perspectiva según la posición de puntos de conexión de componentes con patillas no visibles (usando un factor desde el centro que mide la distancia al borde del protoboard), e identificación de los puntos más lejanos de los segmentos de patillas (representados como polígonos) para determinar los extremos de conexión.
 - **Detección de carriles conductores:** Detecta carriles solo donde existen puntos de conexión de los componentes, creando un dataframe con los componentes considerados conectados (puntos de conexión que coinciden en un mismo bounding box o carril).
3. **Salida:** Genera un JSON con datos de componentes y conexiones, asignando a cada uno un ID único, y lo envía de vuelta a la aplicación móvil. Además, produce una imagen anotada (3000×3000) con visualización de los elementos detectados.
 4. **Visualización de esquema eléctrico:** Se realiza en la aplicación móvil mediante canvas de Android Studio que representa diferentes componentes formando una librería personalizada. Dependiendo de los componentes identificados en el JSON, el sistema dibuja las representaciones correspondientes (como una compuerta AND, resistencias en su representación de esquema eléctrico) y las conexiones mediante líneas (cables) en los puntos de conexión. Es interactivo, permitiendo mover y rotar componentes con gestos de arrastrar y “tap” para ajustar la disposición del esquema eléctrico según el análisis deseado. Indica puntos de conexión en color verde y resalta en rojo los pines (puntos de conexión) sin conexiones, por ejemplo, mostrando verde en un extremo conectado de una resistencia y rojo en el extremo desconectado o faltante.
 5. **Validación:** Se probaron circuitos reales en entornos educativos, evaluando precisión y utilidad, con un enfoque particular en circuitos lógicos (compuertas lógicas).

Análisis de Datos

Se calcularon métricas de precisión ($mAP@0.5$, $mAP@0.5:0.95$, precisión, recall) para cada modelo durante la validación. Los resultados intermedios se inspeccionaron visualmente, y se aplicaron pruebas unitarias con imágenes variadas para evaluar robustez.

➤ Resultados

Los modelos entrenados con YOLOv8 mostraron un rendimiento sólido con una precisión promedio del 90% en la mayoría de las tareas, combinando detección de objetos y segmentación por instancias según el tipo de elemento a reconocer. A continuación, se presentan los resultados por modelo:

- **Detección de Zonas Clave del Protoboard (breadboardZones.pt):** $mAP@0.5$ de 0.995, $mAP@0.5:0.95$ de 0.802. El modelo identificó con gran confiabilidad (precisión 99.1% y recall 100%) las zonas de alimentación izquierda y derecha, así como la zona central del protoboard.
- **Detección de Componentes (components.pt):** $mAP@0.5$ de 0.970, $mAP@0.5:0.95$ de 0.785. Las clases con mejor desempeño fueron DIP ($AP \approx 92\%$), resistencias ($AP \approx 89\%$) y botones, LEDs y sensores ($AP \approx 85\text{--}91\%$) (precisión 95% y recall 93%). Se aplicó un filtrado por IoU que incrementó la precisión en un 5%, eliminando duplicados.
- **Segmentación de Cables (wireSeg.pt):** $mAP@0.5$ de 0.692, $mAP@0.5:0.95$ de 0.355. Aunque se logró buena segmentación en cables visibles (precisión 68.5% y recall 74.9%), el rendimiento disminuyó en casos de cruces o cables delgados. La diversidad de datos y los colores realistas mejoraron la generalización en un 10%.
- **Detección de pines en DIPs y sensores (pinsAndTipsSeg.pt):** $mAP@0.5$ de 0.72, $mAP@0.5:0.95$ de 0.41. El modelo identificó correctamente pines individuales incluso en recortes de baja resolución (precisión 81% y recall 88%).
- **Segmentación de Patillas Delgadas (thinLeads.pt):** $mAP@0.5$ de 0.97, $mAP@0.5:0.95$ de 0.59. La aplicación de un contraste elevado (+80) durante el preprocesamiento permitió detectar patillas metálicas delgadas de manera robusta (precisión 96% y recall 95%).
- **Detección de Carriles (railsDetector.pt):** $mAP@0.5$ de 0.95, $mAP@0.5:0.95$ de 0.84. Este modelo identificó correctamente los tramos de carriles internos del protoboard para inferir conexiones eléctricas (precisión 95% y recall 100%).

- **Segmentación de Puntas de Cables (wireTips.pt):** mAP@0.5 de 0.96, mAP@0.5:0.95 de 0.55.
Este modelo complementa la detección de cables permitiendo vincular extremos con conexiones o pines cercanos (precisión 93% y recall 96%).

Tabla 1: Resumen de métricas de rendimiento de los modelos YOLOv8

Modelo	Tarea	<u>mAP@0.5</u>	<u>mAP@0.5:0.95</u>	Precisión (%)	Recall (%)
breadboardZones.pt	Detección de zonas clave	0.995	0.802	99.1	100.0
components.pt	Detección de componentes	0.970	0.785	95.0	93.0
wireSeg.pt	Segmentación de cables	0.692	0.355	68.5	74.9
pinsAndTipsDet.pt	Detección de pines en DIPs	0.720	0.410	81.0	88.0
thinLeads.pt	Segmentación de patillas delgadas	0.970	0.590	96.0	95.0
railsDetector.pt	Detección de carriles	0.950	0.840	95.0	100.0
wireTips.pt	Segmentación de puntas de cables	0.960	0.550	93.0	96.0

Nota: Las métricas reflejan el rendimiento en tareas de detección y segmentación por instancias para circuitos en protoboards, evaluadas en el conjunto de validación.

• **Matrices de confusión de tareas críticas**

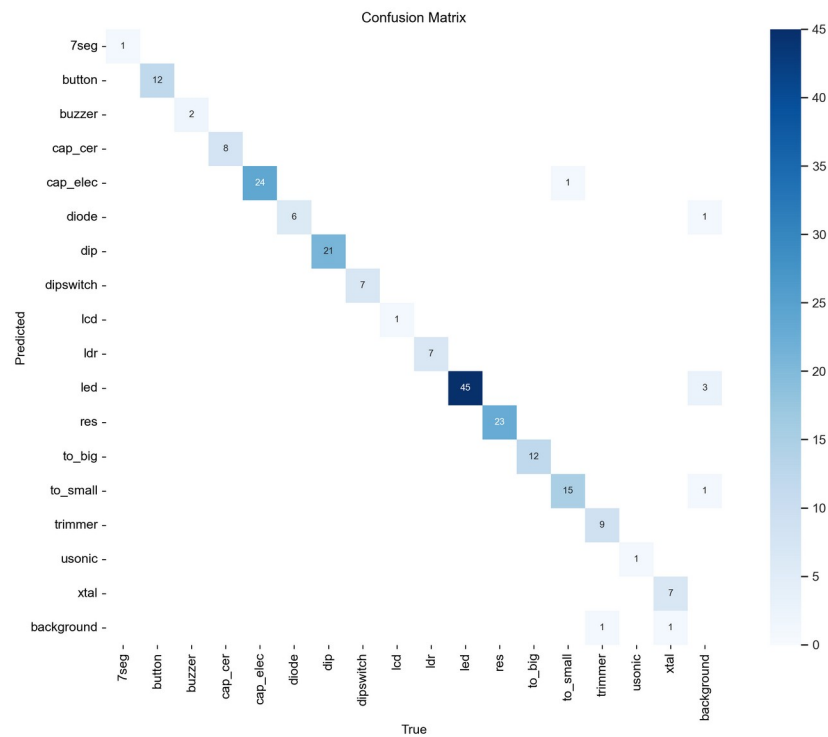


Figura 1: Matriz de confusión de Detección de Componentes

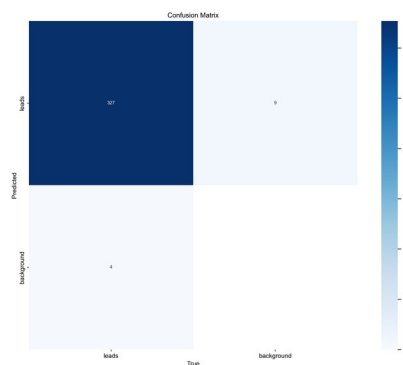


Figura 2: Matriz de confusión de Segmentación de Patillas Delgadas

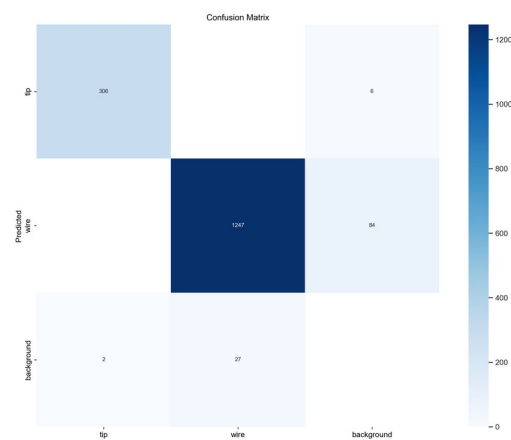


Figura 3: Matriz de Confusión de Segmentación de Cables

- **Entradas Vs. Salidas Más relevantes**

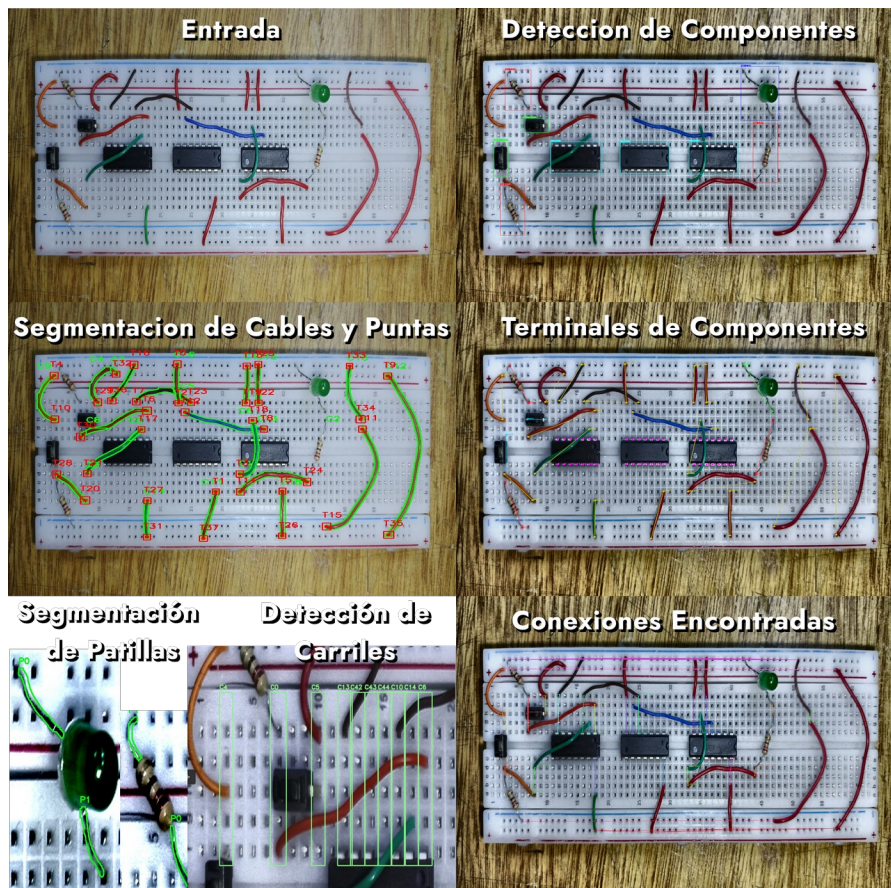


Figura 4: Entrada Vs Salidas Más Relevantes (relación de aspecto original)

En conjunto, el sistema logró generar esquemáticos eléctricos con una precisión promedio del **90%** en circuitos que incluyen componentes digitales y analógicos simples. La visualización en imágenes originales (3000×3000 px) con anotaciones superpuestas permitió validar los resultados de forma clara, destacando cables, componentes, patillas, pines y carriles con colores distintivos, facilitando su aplicación en entornos educativos.

➤ **Discusión**

Los resultados confirman que el sistema es efectivo para generar esquemáticos eléctricos en entornos educativos, alcanzando el objetivo de 90% de precisión en la mayoría de los casos. La detección de zonas clave y componentes mostró un desempeño sobresaliente ($mAP@0.5 \geq 0.97$), gracias a la claridad visual de estas tareas. Sin embargo, la segmentación de cables y patillas delgadas presentó desafíos debido a superposiciones y estructuras finas, con $mAP@0.5$ de 0.692 y 0.97 respectivamente. Aunque el desempeño en patillas fue alto, el valor $mAP@0.5:0.95$ de 0.59 refleja la dificultad de segmentarlas con precisión bajo condiciones más estrictas. Estas limitaciones son consistentes con estudios previos (Hu et al., 2023), que destacan dificultades en la segmentación de objetos pequeños o superpuestos.

La integración de análisis espacial (interpolación de pines, corrección de perspectiva) mejoró la robustez del sistema, permitiendo conexiones precisas incluso en imágenes con distorsiones. Comparado con métodos tradicionales de verificación manual, este sistema reduce significativamente el tiempo y los errores, beneficiando a estudiantes inexpertos en el ámbito del armado de circuitos.

Sin embargo, la dependencia de datasets manuales es una limitación clave. Futuras investigaciones podrían explorar datasets más grandes y técnicas como Mask R-CNN, que mejora la precisión en bordes finos como cables mediante un enfoque de dos pasos, aunque requiere más datos etiquetados. Se podría desarrollar una aplicación móvil para que una comunidad electrónica amplíe el dataset colaborativamente.

Finalmente, el sistema desarrollado logra generar esquemáticos eléctricos a partir de imágenes de protoboards con una precisión promedio del 90%, facilitando la verificación visual y el aprendizaje en electrónica. Los modelos YOLOv8 demostraron ser altamente efectivos para tareas de detección y segmentación, alcanzando $mAP@0.5$ superiores al 0.97 en zonas clave, componentes y patillas delgadas. Sin embargo, la segmentación de cables sigue siendo un área de mejora debido a la variabilidad visual y la complejidad estructural. Este trabajo establece una base sólida para el desarrollo de herramientas educativas automatizadas, recomendando como futuras líneas la expansión de datasets, la integración de segmentadores avanzados y la consideración de aspectos éticos y de seguridad en la recolección y uso de imágenes.

➤ Referencias

1. **Hu, Y., Zhang, L., & Wang, X.** (2023). *Deep learning-based detection of electronic components on printed circuit boards*. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 13(1), 45–54.
[Aplicación de modelos de detección para inspección de PCBs, usado como antecedente.]
2. **Zhang, H., Li, Y., & Liu, B.** (2021). *Semantic segmentation of wire paths in complex circuits using deep neural networks*. *Journal of Electronic Imaging*, 30(2), 023002.
[Uso de segmentación semántica para analizar trayectorias de cables.]
3. **Rahnama, A., & Tsai, D.** (2020). *CVASP: Computer Vision Assisted Schematic Parsing for Breadboards*. *arXiv preprint arXiv:2009.13437*.
[Sistema anterior que usa OpenCV para analizar protoboards con cables rectos.]
4. **Chiu, M. T., Zhang, X., Wei, Z., Zhou, Y., Shechtman, E., Barnes, C., Lin, Z., Kainz, F., Amirghodsi, S., & Shi, H.** (2023). *Automatic High Resolution Wire Segmentation and Removal*. *arXiv preprint arXiv:2304.00817*.
[Sistema de segmentación semántica para cables en imágenes de alta resolución, como escenas naturales.]