

ỨNG DỤNG THUẬT TOÁN YOLO KẾT HỢP PHƯƠNG PHÁP NHẬN DẠNG HÌNH ẢNH TRONG VIỆC KIỂM TRA LINH KIỆN ĐIỆN TỬ TRÊN BẢNG MẠCH IN

APPLICATION OF YOLO ALGORITHM AND IMAGE RECOGNITION TECHNIQUES FOR ELECTRONIC COMPONENT INSPECTION ON PRINTED CIRCUIT BOARDS

Lê Hà An, Nguyễn Hồng Linh, Quách Hùng, Phạm Xuân Mỹ

Khoa Cơ khí – Ô tô và Xây dựng, Trường Đại học Điện Lực

Email: hungq@epu.edu.vn

TÓM TẮT

Bài báo này trình bày việc thiết kế và chế tạo thiết bị kiểm tra mạch Buck tự động dựa trên công nghệ thị giác máy tính, sử dụng camera và thuật toán YOLO kết hợp với phương pháp căn chỉnh ảnh SIFT và đối sánh mẫu nhằm nhận diện và phát hiện các lỗi thiếu hoặc sai lệch linh kiện trên bo mạch in (PCB), thay thế phương pháp kiểm tra thủ công. Hệ thống cho phép phân loại sản phẩm theo thời gian thực (OK/NG) với thời gian xử lý từ 200-500 ms cho mỗi bo mạch, đáp ứng yêu cầu vận hành trong thực tế. Giải pháp đề xuất góp phần nâng cao độ chính xác, tính nhất quán trong kiểm tra và giảm sự phụ thuộc vào nhân công.

Từ khoá: Mạch Buck; Kiểm tra PCB; Thị giác máy tính; YOLO.

ABSTRACT

This paper presents the design and development of an automated Buck converter inspection system based on computer vision technology, utilizing a camera and the YOLO algorithm combined with SIFT-based image alignment and template matching to detect missing or misaligned components on printed circuit boards (PCBs), replacing manual inspection methods. The system enables real-time classification of products (OK/NG) with a processing time of 200-500 ms per board, meeting practical operational requirements. The proposed solution improves inspection accuracy and consistency while reducing dependence on human labor.

Keywords: Buck converter; PCB inspection; Computer vision; YOLO.

1. ĐẶT VẤN ĐỀ

Việc áp dụng tự động hóa trong kiểm tra bảng mạch in (PCB) đóng vai trò quan trọng trong việc nâng cao năng suất và đảm bảo chất lượng sản phẩm. Tuy nhiên, các hệ thống kiểm tra quang học tự động (AOI) hiện nay có chi phí cao và yêu cầu vận hành phức tạp, gây khó khăn cho việc triển khai ở quy mô nhỏ hoặc trong nghiên cứu – thử nghiệm.

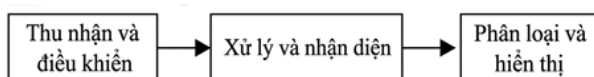
Trong thực tế, các mạch nguồn sử dụng IC LM2596 thường được kiểm tra thủ công, dễ xảy ra sai sót do yếu tố chủ quan và điều kiện thu nhận ảnh không ổn định. Với sự phát triển của thị giác máy tính, các mô hình như YOLO [1] cùng thư viện OpenCV [2, 5] và các công cụ hỗ trợ như Roboflow [3], Google Colab [4] đã mở ra khả năng xây dựng hệ thống kiểm tra tự động với chi phí thấp.

Do đó, nghiên cứu này đề xuất hệ thống kiểm tra PCB tự động dựa trên YOLO [1] kết hợp căn chỉnh ảnh và đối sánh mẫu, nhằm phát hiện lỗi nhanh, chính xác và giảm phụ thuộc vào nhân công.

2. PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Quy trình hoạt động của hệ thống

Quy trình kiểm tra được vận hành theo các bước sau:



Hình 1. Các bước của quy trình kiểm tra

- Bước 1 – Thu nhận và Điều khiển: PCB chạy trên băng tải; khi Cảm biến 1 phát hiện mạch, Arduino dừng băng tải và gửi tín hiệu kích hoạt cho Python.

- Bước 2 – Xử lý và Nhận diện: Camera chụp ảnh; Python dùng mô hình YOLOv8 để phát hiện linh kiện và đối sánh với template mẫu (Profile “OK”).

Kết luận OK: Đủ linh kiện, đúng vị trí.

Kết luận NG: Thiếu hoặc sai lệch linh kiện.

Bước 3 – Phân loại và Hiển thị: Arduino điều khiển băng tải chạy tiếp; nếu NG, Servo tại Cảm biến 2 sẽ gạt mạch lỗi. Đồng thời, kết quả và ảnh nhận diện được cập nhật thời gian thực lên giao diện Flutter.

Bảng 1. Thông số kỹ thuật của thiết bị

STT	Tên thông số	Thứ nguyên	Giá trị
1	Số linh kiện nhận diện	–	6
2	Tốc độ băng tải	mm/s	15
3	Kích thước băng tải	mm	640×99
4	Kích thước tổng thể	mm	680×142×122
5	Camera	–	Sony IMX503
6	Tốc độ khung hình	fps	25 – 30
7	Thời gian xử lý/1 ảnh	ms	200 – 500
8	Giao diện hiển thị	–	Flutter Desktop
9	Nguồn cấp hệ thống	V – A	12 – 3

2.2. Thiết kế các bộ phận cơ cấu chấp hành

Khởi vận chuyển và Định vị: Băng tải kết hợp gá đỡ giúp di chuyển và cố định PCB ổn định tại vùng kiểm tra.

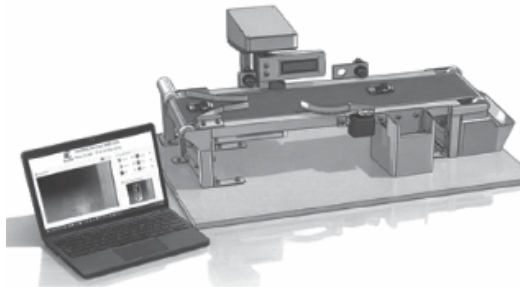
Khởi kích hoạt (Trigger): Cảm biến quang xác định vị trí PCB để kích hoạt lệnh chụp ảnh tự động.

Khởi thu nhận hình ảnh: Camera độ phân giải cao, đảm bảo ảnh chụp rõ nét, chống nhiễu sáng và phản quang.

Khởi xử lý và Phân loại (Trọng tâm): Máy tính chạy mô hình YOLOv8 và thuật toán Template Matching để nhận diện linh kiện và kết luận OK/NG.

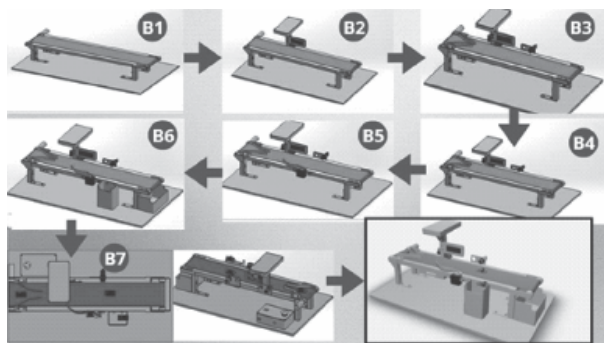
Khởi chấp hành: Cơ cấu Servo/tay gạt thực hiện phân loại sản phẩm vào nhánh đạt hoặc lỗi.

Khởi giao diện và Lưu trữ: Hiển thị kết quả thời gian thực và lưu trữ hồ sơ (log/ảnh) phục vụ thống kê, truy xuất.



Hình 2. Mô hình 3D của thiết bị

2.3. Quy trình lắp ráp thiết bị



Hình 3. Quy trình lắp ráp thiết bị trên phần mềm

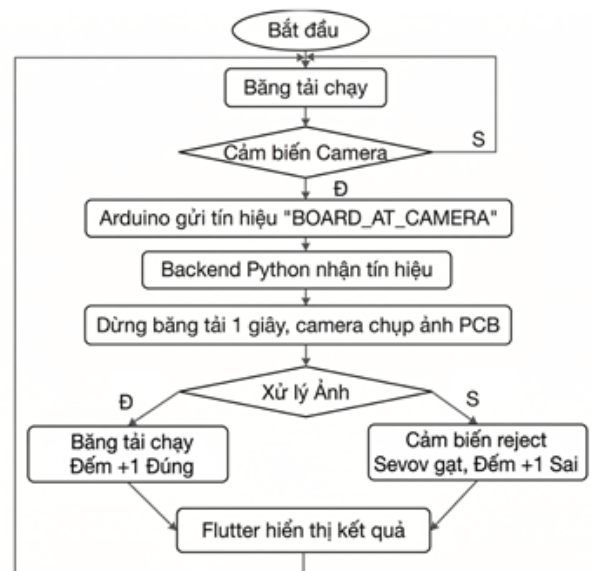
Sau khi hoàn thành từng chi tiết, nhóm lắp ráp trong môi trường Assembly và sử dụng Mate để cố định quan hệ hình học. Kết quả lắp ráp giúp:

- Phát hiện và loại bỏ các điểm va chạm cơ khí.

- Kiểm tra tính hợp lý về kích thước và bố trí cụm.

- Đánh giá khả năng lắp ráp, tháo lắp và bảo trì.

2.4. Lưu đồ thuật toán điều khiển hệ thống



Hình 4. Lưu đồ thuật toán điều khiển hệ thống

Mục tiêu và phạm vi chương trình:

- Khởi tạo: Thiết lập kết nối Serial (Arduino - Python), camera và nạp mô hình YOLOv8.

- Chờ tín hiệu: Băng tải chạy. Arduino kiểm tra Cảm biến 1.

- Kích hoạt (Trigger): Nếu phát hiện PCB, Arduino dừng băng tải và gửi tín hiệu "Chụp ảnh" sang Python.

- Xử lý AI (Python): Chụp ảnh và căn chỉnh; Chạy YOLOv8 nhận diện linh kiện; Đối sánh với Template Profile (Mẫu OK).

- Phân loại:

+ Nếu Đủ (OK): Gửi lệnh cho Arduino tiếp tục chạy băng tải + đếm sản phẩm OK.

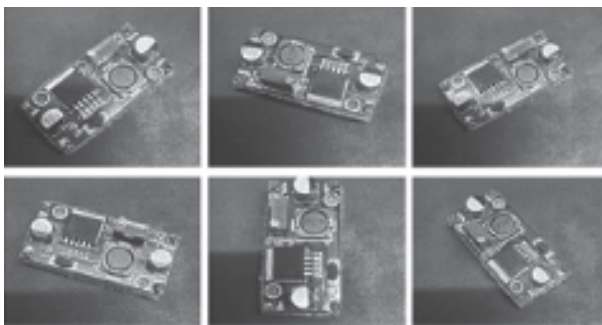
+ Nếu Thiếu (NG): Gửi lệnh cho Arduino chạy đến vị trí Cảm biến 2, kích hoạt Servo gạt mạch lỗi + đếm sản phẩm NG.

- Hiển thị: Cập nhật ảnh có Bounding Box và trạng thái lên giao diện Flutter.

- Lặp lại: Quay về bước chờ tín hiệu cho sản phẩm tiếp theo.

2.5. Căn chỉnh và thu thập hình ảnh bo mạch (PCB)

Bài toán sai lệch hình học trong ảnh PCB:



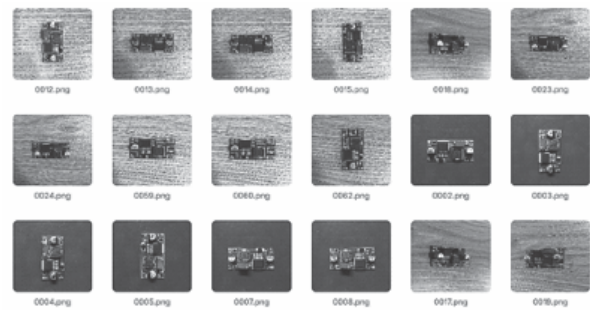
Hình 5. Bo mạch bị lệch vị trí

Khi chụp PCB trong thực tế, bo mạch có thể bị lệch vị trí, xoay nhẹ hoặc thay đổi khoảng cách so với camera. Các sai lệch này làm cho vị trí linh kiện trong ảnh không trùng nhau giữa các lần chụp, gây khó khăn cho việc so sánh với mẫu chuẩn.

Vì vậy, căn chỉnh ảnh (alignment/registration) là bước đưa ảnh kiểm tra về cùng

hệ tọa độ với ảnh chuẩn (reference). Khi đó các linh kiện sẽ nằm "đúng chỗ" và thuật toán kiểm tra thiếu linh kiện sẽ chính xác hơn.

2.5.1. Thu thập ảnh PCB gốc và nguyên tắc chụp ảnh phục vụ training



Hình 6. Thu thập ảnh bo mạch

Trước khi tiến hành xử lý ảnh trong hệ thống, bước chuẩn bị dữ liệu là rất quan trọng. Dữ liệu tốt chất lượng là nền tảng cho một mô hình học máy hiệu quả.

2.5.2. Quy trình Annotation bounding box trên Roboflow



Hình 7. Gắn nhãn cho từng linh kiện

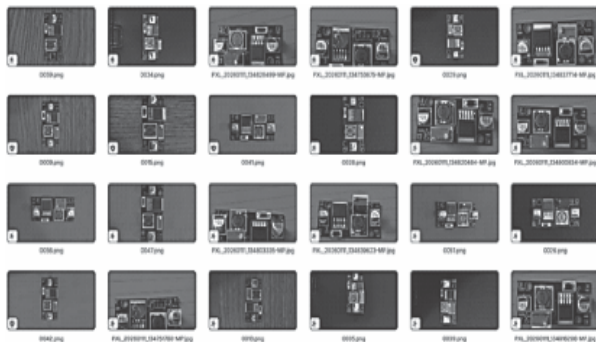
Annotation là bước vẽ bounding box cho từng linh kiện trong ảnh và gán đúng class với quy trình thực hiện:

- Mở ảnh, phóng to các vùng có linh kiện nhỏ.



- Vẽ bbox bao trọn linh kiện, không cắt mất phần quan trọng.
- Chọn class tương ứng.
- Lặp lại cho tất cả linh kiện cần nhận diện.

2.5.3. Kiểm tra chất lượng nhãn và chuẩn hóa nhãn



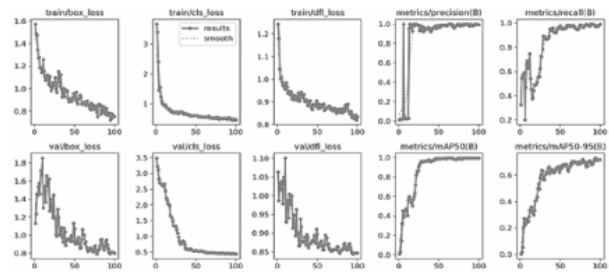
Hình 8. Kiểm tra lại các ảnh đã gán nhãn

Trước khi xuất dataset, cần kiểm tra chất lượng nhãn theo các tiêu chí:

- Đúng class: Linh kiện thuộc loại nào thì nhãn đúng loại đó.
- Đúng vị trí: bbox bao quanh đúng đối tượng, không lệch sang linh kiện khác.
- Tính nhất quán: Cùng một loại linh kiện thì cách vẽ box tương tự nhau.

Nếu phát hiện nhãn sai, nên sửa ngay trên Roboflow. Đây là bước quan trọng vì nhãn sai sẽ kéo chất lượng mô hình xuống rất nhanh. Đặc biệt trong PCB, chỉ cần một số lượng nhỏ nhãn sai cũng gây nhầm lẫn giữa các lớp giống nhau.

2.5.4. Đánh giá kết quả training



Hình 9. Biểu đồ mức độ sai số

Sau Training, Ultralytics tạo ra nhiều biểu đồ:

- Results.png: Thể hiện loss và metric theo epoch.
- Confusion_matrix.png: Mức độ nhầm lẫn giữa các lớp.
- PR_curve.png: Đường cong Precision-Recall.
- Labels.jpg hoặc samples: Thống kê kích thước bbox.

Biểu đồ này thể hiện “loss” giảm nghĩa là mô hình học tốt hơn; “mAP” tăng nghĩa là mô hình dự đoán tốt hơn.



Hình 10. Giao diện hiển thị

Giao diện được thiết kế theo tiêu chí:

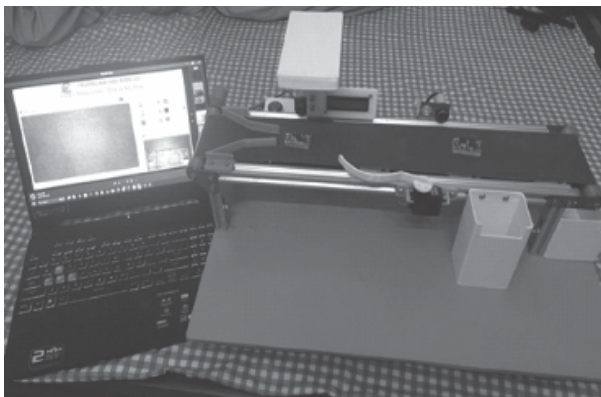
- Đơn giản, dễ quan sát, ưu tiên thông tin quan trọng.

- Có khu vực trạng thái thiết bị, cảnh báo, và nút điều khiển cơ bản.

- Hiển thị kết quả theo thời gian thực, giúp người vận hành theo dõi ngay.

2.6. Mô hình thực tế

Khung băng tải, con lăn, dây đai, ray dẫn hướng, cơ cấu gạt sử dụng servo được chế tạo và lắp ráp đúng theo mô hình thiết kế.



Hình 11. Hình ảnh thiết bị trên thực tế

Phần điện – điều khiển: Mạch nguồn, cảm biến, động cơ băng tải, servo và vi điều khiển được đấu nối đầy đủ, đảm bảo thiết bị hoạt động ổn định theo đúng trình tự.

Phần mềm: Chương trình xử lý ảnh nhận diện linh kiện, kết hợp với giao diện hiển thị/điều khiển giúp người dùng theo dõi trạng thái và kết quả kiểm tra trực quan.

3. KẾT LUẬN

Nghiên cứu đã hoàn thiện một hệ thống kiểm tra PCB tự động tích hợp đồng bộ giữa phần cứng và phần mềm. Về phần cứng, mô hình băng tải kết hợp cảm biến, cơ cấu servo và hệ thống camera đảm bảo khả năng thu nhận và phân loại sản phẩm ổn định. Về phần mềm,

quy trình xử lý ảnh ứng dụng thuật toán YOLO đã nâng cao độ tin cậy trong phát hiện lỗi linh kiện.

Kết quả thực nghiệm cho thấy hệ thống vận hành ổn định và có độ tin cậy cao, đồng thời giảm đáng kể sự phụ thuộc vào thao tác thủ công. Giải pháp đề xuất có tính khả thi cao và tiềm năng ứng dụng trong các dây chuyền sản xuất quy mô nhỏ và trung bình. ❖

Ngày nhận bài: 24/3/2026

Ngày phản biện: 09/4/2026

Tài liệu tham khảo:

- [1]. Ultralytics. (2024), “*Ultralytics YOLOv8 documentation*”. Retrieved January 26, 2026, from <https://docs.ultralytics.com>.
- [2]. OpenCV Team (2024), “*OpenCV documentation*”. Retrieved January 26, 2026, from <https://docs.opencv.org>.
- [3]. Roboflow (2024), “*Roboflow documentation (dataset management & annotation)*”. Retrieved January 26, 2026, from <https://docs.roboflow.com>.
- [4]. Google. (2024), “*Google Colab documentation*”. Retrieved January 26, 2026, from <https://colab.research.google.com>.
- [5]. Bradski, G., & Kaehler, A. (2008), “*Learning OpenCV: Computer vision with the OpenCV library*”. O’Reilly Media.