

# NGHIÊN CỨU ỨNG DỤNG MẠNG NEURAL NETWORK CHO HỆ THỐNG KIỂM TRA LINH KIỆN ĐIỆN TỬ THỜI GIAN THỰC

A REAL-TIME SYSTEM FOR PCB AUTOMATED INSPECTION USING CONVOLUTIONAL NEURAL NETWORK

Nguyễn Văn Trường\*, Nguyễn Đức Linh

## TÓM TẮT

Bài báo trình bày việc ứng dụng công nghệ xử lý ảnh trong phân tích và kiểm tra linh kiện điện tử trên mạch in PCB trong miền thời gian thực. Trong bài báo này, mô hình mạng nơ-ron tích chập CNN 53 lớp được sử dụng để trích xuất đặc trưng của vật thể và dự đoán một vật thể trong một tấm ảnh mới. Mô hình thực nghiệm được tích hợp trên một trạm phân loại mạch in PCB có năng suất làm việc 120 sản phẩm/giờ. Quá trình thử nghiệm trong các môi trường làm việc khác nhau cho thấy hệ thống có độ chính xác lên tới 96%. Các ảnh hưởng nhiễu loạn khác nhau từ môi trường cũng được xét đến để đánh giá hiệu quả của hệ thống như chế độ sáng, kích thước ảnh và nền.

**Từ khóa:** PCB, CNN, công nghệ xử lý ảnh, AOI.

## ABSTRACT

The paper presents the application of image processing technology in checking real-time electronic components on the Printed Circuit Board (PCB). The convolutional neural network model is used to extract features and predict of objects in a new image. The PCB system has a working capacity of 120 products/hour. From the experiment, it is certain that the proposed system has an accuracy of up to 96% for PCB. The various noise effects from the environment are considered to evaluate the effectiveness of the system such as brightness, image size, and background.

**Keywords:** PCB, CNN, Image processing technology, AOI.

Khoa Cơ khí, Trường Đại học Công nghiệp Hà Nội

\*Email: nguyenvantruong@hau.edu.vn

Ngày nhận bài: 01/8/2020

Ngày nhận bài sửa sau phản biện: 25/11/2020

Ngày chấp nhận đăng: 23/12/2020

## CHỮ VIẾT TẮT

PCB	Printed Circuit Board
AOI	Automated Optical Inspection
CNN	Convolutional Neural Network
IoU	Interest of Union
NMS	Non-max suppression

## 1. GIỚI THIỆU

Bảng mạch in hay bo mạch in (PCB), đôi khi gọi tắt là mạch in, là bảng mạch điện dùng phương pháp in để tạo

hình các đường mạch dẫn điện và điểm nối linh kiện trên tấm nền cách điện [1]. Việc lắp ráp bảng mạch in là công đoạn quan trọng trong quá trình chế tạo bảng mạch điện tử. Trong quá trình sản xuất bản mạch vẫn còn một số mạch in PCB bị lỗi sau khi gia công vì một số nguyên nhân như: linh kiện dán lệch vị trí, cắm sai chiều, cắm sai linh kiện,... Để đảm bảo chất lượng của sản phẩm, các bản mạch được kiểm tra sau mỗi công đoạn gia công hay lắp ráp mạch in. Trong đó, người công nhân và người kiểm tra phát hiện bản mạch lỗi bằng mắt thường ở các dây chuyền sản xuất. Tuy nhiên, việc kiểm tra bằng mắt thường không mang lại hiệu quả và năng suất cao. Ngày nay, có nhiều phương pháp tiên tiến được nghiên cứu để cải thiện năng suất lắp ráp mạch dựa trên công nghệ xử lý ảnh. Công nghệ xử lý ảnh là một lĩnh vực thuộc trí tuệ nhân tạo hoạt động theo nguyên tắc trích xuất đặc trưng của đối tượng từ dữ liệu đầu vào, từ đó đưa ra quyết định đúng/sai với các dữ liệu đầu vào mới. Tích hợp công nghệ xử lý ảnh vào các hệ thống kiểm tra linh kiện điện tử mang lại hiệu quả cao, làm tăng năng suất và đảm bảo độ chính xác.

Dựa trên kỹ thuật xử lý ảnh, một số phương pháp kinh điển được ứng dụng để nhận diện đối tượng như là haar features [2], histogram of oriented gradients [3],... Haar features được dùng để phát hiện vật thể có trong ảnh. Histogram of oriented gradients được dùng để phát hiện vùng vật thể có trong ảnh. Hai phương pháp này dễ ứng dụng trong các lớp đối tượng đơn giản. Tuy nhiên, khi ứng dụng vào việc phát hiện đối tượng trên mạch in PCB thì có những hạn chế nhất định như: khó phát hiện vật nếu tấm ảnh chứa nhiều vật thể phức tạp và vùng ảnh gần giống nhau; chưa trích xuất được những đặc trưng cụ thể của một vật.

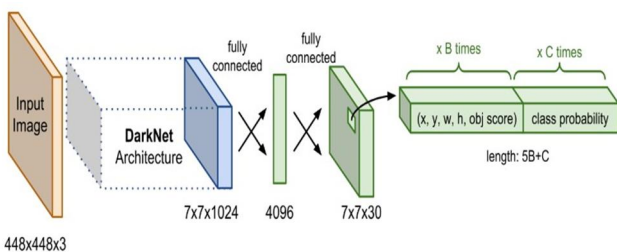
Trong kỷ nguyên công nghệ 4.0, cùng với sự phát triển của khoa học và công nghệ, lĩnh vực nghiên cứu chuyên sâu về giải thuật, thuật toán nhận diện cũng được phát triển theo. Trong đó, những thành tựu trong giải thuật học máy machine learning [4-8], giải thuật học sâu deep learning [9-12] được sử dụng phổ biến trong các ứng dụng: mở khóa bằng khuôn mặt, điểm danh bằng khuôn mặt, nhận diện giọng nói,... Trên thế giới đã có các thành tựu về công nghệ xử lý ảnh để kiểm tra linh kiện điện tử trên mạch in PCB hay hệ thống kiểm tra quang học tự động AOI [13, 14]. Hệ thống làm việc ổn định với kích thước linh kiện lớn hơn 1mm. Tuy nhiên, ở Việt Nam các nghiên cứu này

vẫn còn hạn chế, đặc biệt là ứng dụng vào hệ thống kiểm tra linh kiện điện tử.

Trong bài báo này, mô hình mạng nơ-ron tích chập CNN 53 lớp (Darknet-53) được sử dụng để trích xuất đặc trưng của vật thể và dự đoán một vật thể trong một tấm ảnh mới. Hệ thống thư viện ảnh do tác giả tự tạo được đưa vào huấn luyện trong mô hình mạng CNN. Model tạo ra được sử dụng để làm tín hiệu đầu vào cho hệ thống điều khiển trong trạm phân loại sản phẩm. Các công cụ hỗ trợ như OpenCV library [15], Python [16] được tích hợp trên hệ thống máy tính để tính toán và phân tích tín hiệu trước khi truyền đến cho vi điều khiển. Hệ thống phân loại cho phép làm việc trong miền thời gian thực với năng suất 120 mạch/giờ. Các yếu tố ảnh hưởng khác nhau từ môi trường dẫn tới kết quả không mong muốn trong quá trình xử lý ảnh như chế độ sáng, kích thước ảnh và nền được nghiên cứu tính toán dựa trên thực nghiệm để tìm ra vùng môi trường phù hợp nhất cho quá trình xử lý ảnh. Kết quả nghiên cứu được kiểm nghiệm dựa trên thực nghiệm cho thấy kết quả chính xác lên tới 96% cho các bản mạch in.

**2. MÔ HÌNH GIẢI THUẬT HỌC SÂU**

Theo tài liệu [19], kiến trúc mạng nơ-ron tích chập CNN (Darknet-53) được sử dụng trong bài báo được mô tả như hình 1, bao gồm: mạng cơ bản và các lớp trích xuất thông tin dự đoán vật thể (extra layers). Trong đó, mạng cơ bản là các mạng convolution làm nhiệm vụ trích xuất đặc trưng. Extra layers được áp dụng để phát hiện vật thể dựa trên các liên kết đặc trưng của vật thể (feature map). Mạng Darknet-53 có kích thước ảnh đầu vào chủ yếu là 448x448. Mỗi một đầu vào sẽ có một thiết kế lớp riêng phù hợp với hình dạng của ảnh đầu vào. Sau khi qua các lớp CNN, hình dạng của vật thể giảm dần và cuối cùng thu được một feature map có kích thước tương đối nhỏ được dùng để dự đoán.



Hình 1. Sơ đồ kiến trúc mạng Darknet-53 và extra layers [19]

Trong mô hình mạng nơ-ron tích chập CNN, để tìm các vùng bao quanh vật thể trong ảnh (bounding box), thuật toán cần các anchor boxes để làm cơ sở cho việc ước lượng tính toán độ sai lệch. Những anchor boxes này được xác định trước và bao quanh vật thể tương đối chính xác. Trong mạng CNN, mỗi vật thể trong hình ảnh huấn luyện được phân bổ về một cell trong feature map, có chứa điểm trung tâm của vật thể. Từ đó các anchor boxes bao quanh vật thể đó được xác định. Mỗi một vật thể trong hình ảnh huấn luyện được phân bổ về anchor boxes. Trong trường hợp có hai anchor boxes trở lên, anchor box được xác định khi có

giá trị IoU giao với Ground truth bounding box là cao nhất. Tuy nhiên, thuật toán không thể gán nhãn khi trường hợp hai vật thể bị trùng tâm như ví dụ được nêu trong hình 2. Trường hợp này rất khó có thể xảy ra và đây là điểm hạn chế duy nhất của thuật toán.



Hình 2. Hai vật thể bị trùng tâm [19]

Để tối ưu giá trị dữ liệu đầu vào, nhận diện và dự đoán tên vật thể, CNN sử dụng hàm mất mát loss. Theo tài liệu [17], hàm mất mát được chia làm hai phần: hàm tính toán sai số kích thước và hàm tính toán phân phối xác suất phân nhóm. Trong đó, hàm tính toán sai số kích thước  $L_{coord}$  tính toán độ sai lệch nhỏ nhất giữa giá trị ban đầu và giá trị dự đoán. Hàm tính toán phân phối xác suất phân nhóm  $L_{class}$  tính toán sai lệch giữa các nhóm dữ liệu trong khi huấn luyện mô hình thuật toán. Mục đích là trả về phân nhóm dự đoán đúng cho vùng chứa vật thể.

$$loss = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \tag{1}$$

$$+ \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (c_i - \hat{c}_i)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{noobj} (c_i - \hat{c}_i)^2 + \sum_{i=0}^{s^2} 1_i^{obj} \sum_{c \in classes} [p_i(c) - \hat{p}_i(c)]^2$$

$$L_{coord} = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \tag{2}$$

$$L_{class} = \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (c_i - \hat{c}_i)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{noobj} (c_i - \hat{c}_i)^2 + \sum_{i=0}^{s^2} 1_i^{obj} \sum_{c \in classes} [p_i(c) - \hat{p}_i(c)]^2 \tag{3}$$

Trong đó:

$[(x, y), (w, h)]$ : Kích thước của ô mốc;

$[(\hat{x}_i, \hat{y}_i), (\hat{w}_i, \hat{h}_i)]$ : Kích thước của ô dự đoán;

$1_i^{obj}$ : Hàm chỉ định có giá trị [0, 1], xét xem ô đó có vật hay không;

$1_{ij}^{obj}$ : Cho biết tạo vùng thứ j của ô thứ i có chứa vật thể hay không;

$C_{ij}$ : Điểm tin cậy của ô thứ i;

$\hat{C}_{ij}$ : Điểm dự đoán;

c: Tập hợp tất cả các nhóm;

$p_i(c)$ : Hàm xác suất có điều kiện để xác định ô chứa đối tượng thuộc nhóm c;

$\hat{p}_i(c)$ : Hàm xác suất có điều kiện dự đoán.

$\lambda_{coord}, \lambda_{noobj}$ : Hằng số điều chỉnh có nhiệm vụ làm giảm giá trị của hàm mất mát.

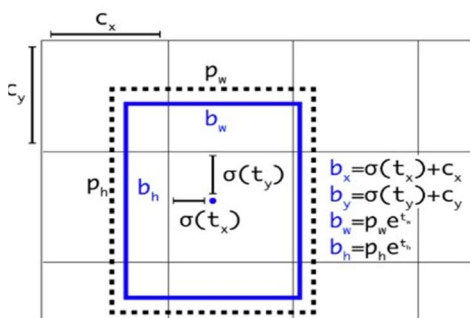
Để dự báo vùng bao quanh của một vật thể dựa trên một phép biến đổi từ anchor box và cell. Thuật toán dự đoán các bounding boxes sẽ tính toán sao cho không lệch khỏi vị trí trung tâm quá nhiều. Ví dụ: một anchor box như hình 7 có kích thước  $(p_w, p_h)$ , với  $(c_x, c_y)$  là kích thước của một ô (cell). Mỗi bounding box dự đoán chứa các thông số:  $(t_x, t_y, t_w, t_h)$  và độ chính xác. Các thông số này giúp xác định bounding box dự đoán có tâm  $b(b_x, b_y)$  thông qua hàm sigmoid [12] và hàm exponential như sau:

$$b_x = \sigma(t_x) + c_x \tag{4}$$

$$b_y = \sigma(t_y) + c_y \tag{5}$$

$$b_w = p_w \times e^{t_w} \tag{6}$$

$$b_h = p_h \times e^{t_h} \tag{7}$$



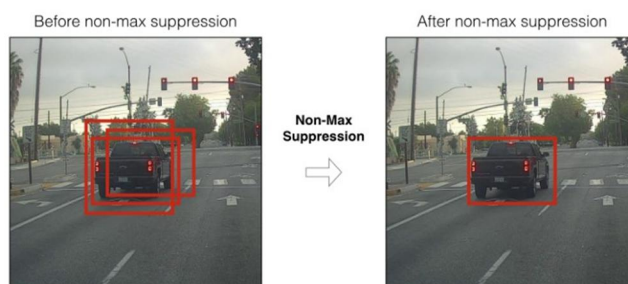
Hình 3. Ước lượng bounding box từ anchor box [19]

Ngoài ra, do các tọa độ đã được hiệu chỉnh theo chiều rộng và chiều cao của bức ảnh nên luôn có giá trị trong ngưỡng [0, 1]. Do đó khi áp dụng hàm sigmoid giúp giới hạn được tọa độ không vượt quá xa các ngưỡng này. Theo hình 3, hình chữ nhật nét đứt bên ngoài là anchor box có kích thước là  $(p_w, p_h)$ . Tọa độ của một bounding box được xác định dựa trên đồng thời cả anchor box và cell mà nó thuộc về. Điều này giúp kiểm soát vị trí của bounding box dự đoán quanh vị trí của cell và bounding box mà không vượt quá xa ra bên ngoài giới hạn này.

Kỹ thuật chọn lựa do thuật toán CNN dự báo cho ra rất nhiều bounding box bao quanh vật thể trong một bức ảnh nên đối với những cell có vị trí gần nhau, khả năng các vùng giao nhau là rất cao. Trong trường hợp đó, giải thuật non-max suppression (NMS) được sử dụng để giảm bớt số lượng các vùng bao được sinh ra. Theo hình 4, từ ba bounding box ban đầu cùng bao quanh chiếc xe đã giảm xuống còn một bounding box sau khi sử dụng phương pháp NMS. Giải thuật NMS được trình bày ngắn gọn như sau:

- Bước 1: Loại bỏ toàn bộ những bounding box có xác suất chứa vật thể lớn hơn một ngưỡng, thường là 0,5.

- Bước 2: Đối với các bounding box giao nhau, NMS lựa chọn ra một bounding box có xác suất chứa vật thể là lớn nhất. Sau đó tính toán chỉ số giao thoa IOU với các bounding box còn lại. Nếu có bất kỳ box nào có giá trị IOU lớn hơn giá trị ngưỡng thì loại box đó.

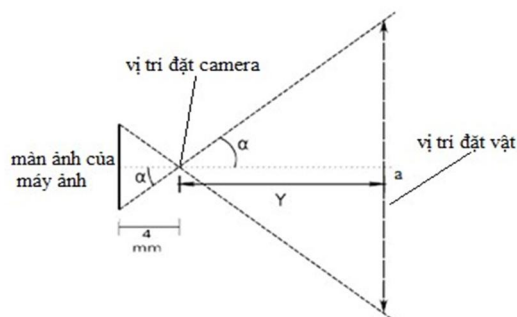


Hình 4. Non-max suppression [18]

### 3. THIẾT KẾ HỆ THỐNG KIỂM TRA LINH KIỆN ĐIỆN TỬ

#### 3.1. Tính toán vị trí đặt camera

Khi camera chụp hình đối tượng để xử lý thì có những vùng của hình ảnh nằm ngoài vùng mà camera có thể xử lý. Điều này làm ảnh hưởng đến độ chính xác của thuật toán. Vì vậy, cần phải xác định chính xác vị trí đặt camera sao cho toàn bộ hình ảnh nằm trong vùng mà camera có thể xử lý được.



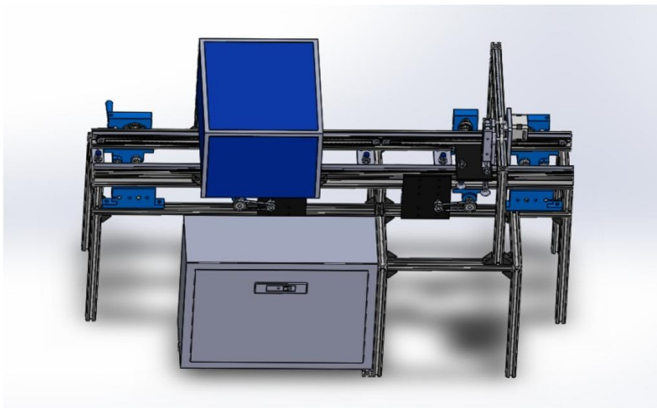
Hình 5. Vị trí đặt camera

Khoảng cách từ ống kính đến chip cảm biến là tiêu cự của thấu kính. Tiêu cự càng ngắn thì góc quét của camera càng lớn. Với mỗi giá trị tiêu cự cần đặt camera ở các vị trí khác nhau để ảnh thu được nằm trọn trong vùng mà camera có thể xử lý. Giả sử với camera có tiêu cự 4mm, góc quét  $\alpha$ , ta cần xác định khoảng cách Y là khoảng cách từ vị trí đặt camera đến vật. Như vậy, cần đặt camera tại vị trí cách vật một khoảng Y và vuông góc với mặt phẳng chứa vật.

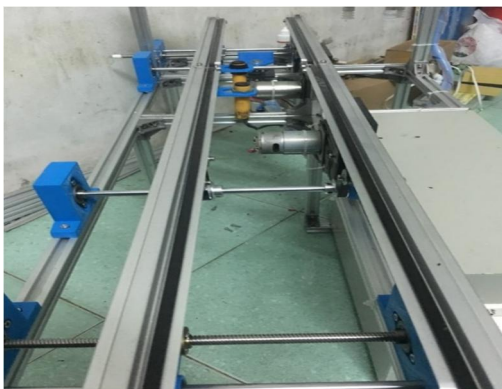
$$Y = \frac{1}{2} a \cdot \cot \alpha \quad (8)$$

### 3.2. Hệ thống kiểm tra linh kiện điện tử

Để kiểm chứng kết quả nghiên cứu từ việc phân tích xử lý ảnh trong kiểm tra linh kiện điện tử thời gian thực, nhóm tác giả đề xuất mô hình hệ thống kiểm tra linh kiện điện tử được thiết kế như hình 6. Trong đó, hệ thống truyền động (hình 7) gồm hai động cơ điện một chiều có giảm tốc cùng với hệ thống dây đai răng, pulli truyền chuyển động đến hệ thống băng tải làm nhiệm vụ di chuyển bản mạch. Cụm hệ thống tay hút bản mạch được thiết kế như hình 8. Hệ thống điều khiển sử dụng mạch Arduino Mega 2560R3 được trình bày như hình 9 bao gồm module nguồn 24V-3A, module điều khiển tốc độ động cơ giảm tốc, module điều khiển tốc độ động cơ bước, module relay 4 kênh 24V.



Hình 6. Mô hình hệ thống kiểm tra linh kiện điện tử



Hình 7. Hệ thống truyền động



Hình 8. Tay máy hút bản mạch



Hình 9. Hệ thống các thiết bị trong tủ điện

Hệ thống camera và điều chỉnh chiếu sáng được trình bày như hình 10. Trong đó khoảng cách và các lắp đặt camera được trình bày như trong mục 3.1. Hệ thống kiểm tra linh kiện điện tử được chế tạo như hình 11. Hệ thống có khối lượng 20kg, năng suất 120 mạch/h, độ chính xác lên đến 96%.



Hình 10. Hệ thống Camera và chiếu sáng



Hình 11. Hệ thống kiểm tra linh kiện điện tử hoàn chỉnh

### 4. ĐÁNH GIÁ KẾT QUẢ THỰC NGHIỆM

Quá trình xử lý ảnh được thực hiện trên CPU máy tính có cấu hình vi xử lý Core i7-3250M bộ nhớ RAM 4GB; sử dụng camera Aoni độ phân giải 1080P; cùng với chip ATmega2560 có bộ nhớ 256KB, SRAM 8KB, EEPROM 4KB có nhiệm vụ điều khiển trạm phân loại băng mạch (kích thước 12x16cm) và

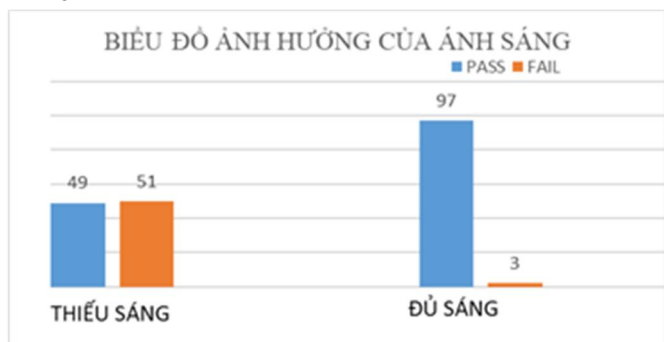
giao tiếp với máy tính. Quá trình huấn luyện trong khoảng thời gian 9 đến 10 tiếng, sử dụng GPU của Google Colab làm môi trường huấn luyện. Trong đó, thông số của GPU của Google là Tesla 80k, với bộ nhớ RAM 12GB.

Trong quá trình thực nghiệm, thời gian xử lý ảnh sử dụng ngôn ngữ Python trong môi trường Window 10 cho kết quả xử lý trên mỗi ảnh 0,4s, thời gian thực hiện khoảng 2 khung hình/giây (2FPS). Để tìm ra vùng làm việc hiệu quả do ảnh hưởng của môi trường, các kết quả thực nghiệm được trình bày trong các điều kiện khác nhau như chế độ sáng, nền và kích cỡ của ảnh đầu vào.

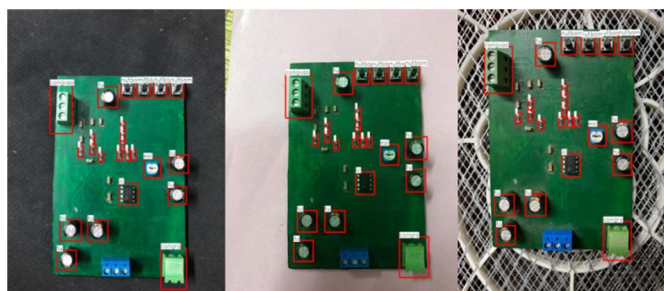
Số lượng ảnh sử dụng cho bộ thử là 100 ảnh được chụp ngẫu nhiên. Kết quả thử nghiệm được trình bày như hình 12-17. Trong đó, hình 12, 13 trình bày kết quả đạt được khi xét đến ảnh hưởng của chế độ sáng. Hình 14, 15 là kết quả trong điều kiện nền khác nhau. Hình 16, 17 là kết quả khi xét đến ảnh hưởng của kích thước ảnh. Như vậy, trong điều kiện đủ sáng, nền màu trắng và kích thước ảnh là 960x1280, độ chính xác xử lý lên tới 96%. Có thể kết luận rằng, hệ thống có khả năng ứng dụng thực tiễn vào quá trình sản xuất công nghiệp.



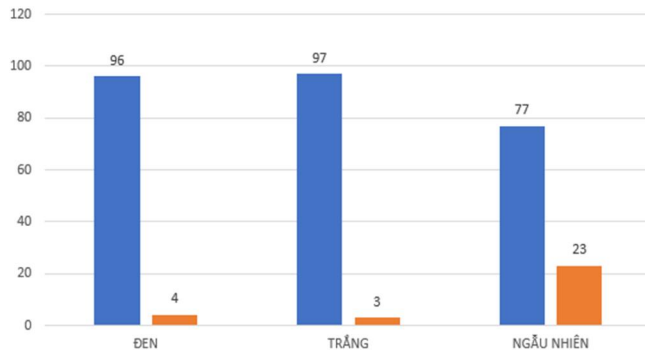
Hình 12. Ảnh hưởng của ánh sáng đến khả năng nhận diện: thiếu sáng, đủ sáng



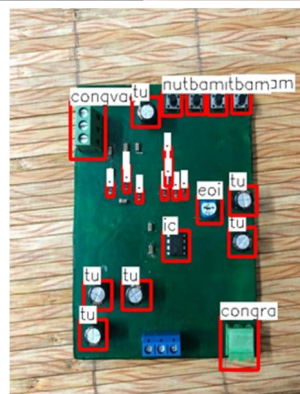
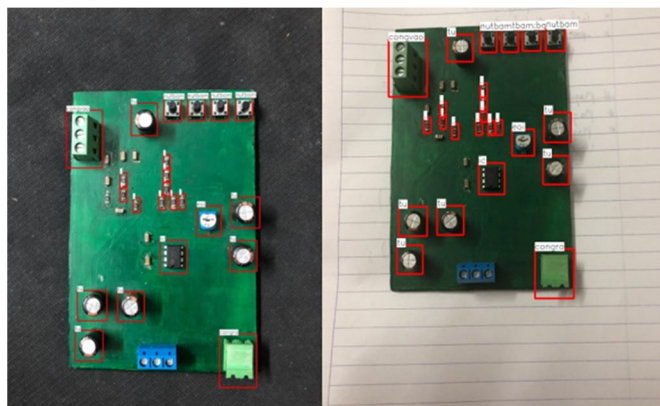
Hình 13. Biểu đồ ảnh hưởng của chế độ sáng đến khả năng nhận diện



Hình 14. Ảnh hưởng của nền: nền đen, nền trắng, nền ngẫu nhiên



Hình 15. Biểu đồ ảnh hưởng của nền đến khả năng nhận diện



Hình 16. Ảnh hưởng của kích thước ảnh đến khả năng nhận diện: 1512x2016, 960x1280, 480x640



Hình 17. Biểu đồ ảnh hưởng của kích thước ảnh đến khả năng nhận diện

**5. KẾT LUẬN**

Bài báo ứng dụng mô hình mạng nơ-ron tích chập CNN 53 lớp để trích xuất đặc trưng trong kiểm tra linh kiện điện tử trên mạch in PCB. Hệ thống cho phép xử lý đồng thời

nhiều loại linh kiện điện tử khác nhau và trên các loại mạch in PCB khác nhau. Kết quả thử nghiệm cho thấy, hệ thống làm việc với độ chính xác ổn định 96% trong điều kiện đủ sáng, nền màu trắng và kích thước ảnh là 960x1280. Có thể kết luận rằng, hệ thống có khả năng ứng dụng thực tiễn vào quá trình sản xuất công nghiệp.

### LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Quỹ Phát triển khoa học và công nghệ Quốc gia (NAFOSTED) trong đề tài mã số 107.01-2019.311.

### TÀI LIỆU THAM KHẢO

- [1]. Clyde F. Coombs, 2007. *Printed circuits handbook*. McGraw-Hill Professional.
- [2]. X. Luo, H. Hu, 2020. *Selected and refined local attention module for object detection*. electronics letters, vol. 56, no. 14, pp. 712-714.
- [3]. S. K. G. Manikonda, D. N. Gaonkar, 2020. *Islanding detection method based on image classification technique using histogram of oriented gradient features*. IET Generation, Transmission & Distribution, vol. 14, no. 14, pp. 2790-2799.
- [4]. G. A. Tahir, C. K. Loo, 2020. *An Open-Ended Continual Learning for Food Recognition Using Class Incremental Extreme Learning Machines*. IEEE Access, vol. 8, pp. 82328-82346.
- [5]. K. Huang, H. Yang, I. King, M. R. Lyu, 2008. *Maxi-Min margin machine: Learning large margin classifiers locally and globally*. IEEE Transactions on Neural Networks, vol. 19, no. 2, pp. 260-272.
- [6]. D. Martens, B. B. Baesens, T. Van Gestel, 2009. *Decompositional rule extraction from support vector machines by active learning*. IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 2, pp. 178-191.
- [7]. E. Pasolli, F. Melgani, Y. Bazi, 2011. *Support vector machine active learning through significance space construction*. IEEE Geoscience and Remote Sensing Letters, vol. 8, no. 3, pp. 431-435.
- [8]. G. Krummenacher, C. S. Ong, S. Koller, S. Kobayashi, J. M. Buhmann, 2018. *Wheel defect detection with machine learning*. IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 4, pp. 1176-1187.
- [9]. P. Kumar Shukla et al., 2020. *Efficient prediction of drug-drug interaction using deep learning models*. IET Systems Biology, vol. 14, no. 4, pp. 211-216.
- [10]. J. S. Shemona, A. K. Chellappan, 2020. *Segmentation techniques for early cancer detection in red blood cells with deep learning-based classifier - a comparative approach*. IET Image Processing, vol. 14, no. 9, pp. 1726-1732.
- [11]. A. Alamri, et al., 2020. *An effective bio-signal-based driver behavior monitoring system using a generalized deep learning approach*. IEEE Access, vol. 8, pp. 135037-135049.
- [12]. M. Zhu, Q. Chen, 2020. *Big data image classification based on distributed deep representation learning model*. IEEE Access, vol. 8, pp. 133890-133904.
- [13]. J. Jiang, J. Cheng, D. Tao, 2012. *Color biological features-based solder paste defects detection and classification on printed circuit boards*. IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 2, no. 9, pp. 1536-1544.

[14]. Y. Hara, H. Doi, K. Karasaki, T. Iida, 1988. *A system for PCB automated inspection using fluorescent light*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 1, pp. 69-78.

[15]. X. Farhodov, O. Kwon, K. W. Kang, S. Lee, K. Kwon, 2019. *Faster RCNN detection based opencv csrt tracker using drone data*. International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, pp. 1-3.

[16]. A. Kumar, S. P. Panda, 2019. *A survey: How Python pitches in IT-World*. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, pp. 248-251.

[17]. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 2016. *You Only Look Once: Unified, Real-Time Object*. Computer Vision and Pattern Recognition, pp. 1-10.

[18]. Jan Hosang, Rodrigo Benenson, Bernt Schiele, 2017. *Learning non-maximum suppression*. Computer Vision and Pattern Recognition, pp. 1-16.

[19]. Joseph Redmon, Ali Farhadi, 2018. *YOLOv3: An Incremental Improvement*. Computer Vision and Pattern Recognition, pp. 1-6.

### AUTHORS INFORMATION

**Nguyen Van Truong, Nguyen Duc Linh**

Faculty of Mechanical Engineering, Hanoi University of Industry