

# CS231: Nhập môn Thị giác máy tính

## Đồ án Cuối kỳ

### Phân loại lỗi bề mặt ốc vít

Đoàn Thái Hoàng, Phạm Ngô Quốc Anh  
23520514, 23520070

Trường Đại học Công nghệ Thông tin  
Đại học Quốc gia Thành phố Hồ Chí Minh

## 1 Thành viên và Phân công nhiệm vụ

Bảng 1: Bảng phân chia công việc giữa các thành viên.

Tên và MSSV	Nhiệm vụ	Mức độ hoàn thành
Đoàn Thái Hoàng 23520514	Thu thập và xử lý dữ liệu Phát triển mô hình EfficientNetB0 Viết báo cáo	100%
Phạm Ngô Quốc Anh 23520070	Trích xuất đặc trưng HOG Phát triển mô hình SVM Tối ưu tham số bằng GridSearch Chuẩn bị slide thuyết trình	100%

## 2 Giới thiệu

Ốc vít là thành phần thiết yếu trong nhiều hệ thống cơ khí, đảm bảo sự kết nối chắc chắn giữa các chi tiết. Chất lượng bề mặt của ốc vít ảnh hưởng trực tiếp đến hiệu suất của sản phẩm. Do đó, việc kiểm tra và phân loại các lỗi trên bề mặt ốc vít đóng vai trò quan trọng trong quá trình sản xuất.

Truyền thống, quá trình kiểm tra được thực hiện thủ công bằng mắt thường hoặc các thiết bị cơ bản, tuy nhiên những phương pháp này thường chậm, tốn kém và dễ dẫn đến sai sót do phụ thuộc nhiều vào kinh nghiệm của người kiểm tra. Sự phát triển của công nghệ thị giác máy tính và các thuật toán học máy đã mở ra cơ hội xây dựng các hệ thống tự động, giúp tăng hiệu quả và chính xác trong việc phát hiện, phân loại các lỗi trên bề mặt ốc vít.

**Lý do chọn đề tài:** Việc tự động hóa quá trình phân loại lỗi trên bề mặt ốc vít là một nhu cầu thực tế trong công nghiệp nhằm giải quyết những hạn chế của kiểm tra thủ công như chi phí cao, thời gian dài và độ chính xác không ổn định. Ngoài ra, với sự đa dạng về loại lỗi và mức độ phức tạp của bề mặt ốc vít, việc phát triển một hệ thống phân loại hiệu quả đòi hỏi phải kết hợp các kỹ thuật xử lý ảnh và học máy phù hợp.

Đề tài này nhằm mục đích nghiên cứu và xây dựng một hệ thống tự động phân loại tình trạng lỗi bề mặt ốc vít dựa trên ảnh chụp, giúp nâng cao hiệu quả kiểm soát chất lượng trong sản xuất. Đây cũng là cơ hội để ứng dụng và phát triển các phương pháp trích xuất đặc trưng và mô hình học máy trong lĩnh vực kiểm tra sản phẩm cơ khí, hướng tới việc giảm thiểu sự can thiệp của con người và tăng tính tự động trong quy trình sản xuất.

### 3 Phát biểu bài toán

**Phân loại tình trạng lỗi bề mặt ốc vít** được định nghĩa là một bài toán phân loại ảnh có giám sát, trong đó mục tiêu là xác định tình trạng bề mặt của một ốc vít dựa trên đặc trưng thị giác trích xuất từ ảnh chụp. Bài toán yêu cầu gán một nhãn tương ứng với tình trạng cụ thể cho mỗi ảnh đầu vào.

Bài toán được mô hình hóa như sau:

$$y = f(x) \tag{1}$$

trong đó:

- $x$ : là ảnh đầu vào chứa một ốc vít duy nhất, đã được tiền xử lý.
- $f(x)$ : là hàm phân loại ánh xạ đặc trưng từ ảnh  $x$  sang một nhãn  $y$  biểu thị tình trạng bề mặt của ốc vít.

## Dữ liệu đầu vào (Input)

Hệ thống yêu cầu hai dạng dữ liệu đầu vào:

### 1. Tập dữ liệu huấn luyện (training set):

- Gồm nhiều ảnh chứa ốc vít (mỗi ảnh chứa một ốc vít duy nhất).
- Mỗi ảnh đã được gán nhãn tương ứng với tình trạng bề mặt thực tế.
- Tập dữ liệu này được sử dụng để huấn luyện mô hình học máy.

### 2. Ảnh đầu vào cần phân loại:

- Là một ảnh chưa biết nhãn, cần hệ thống dự đoán tình trạng, thỏa mãn các điều kiện sau:
  - Ảnh ở định dạng .jpg hoặc .png.
  - Ảnh chỉ chứa **một ốc vít duy nhất**, chiếm phần lớn khung hình.
  - Ốc vít hiển thị rõ ràng, không bị che khuất, đủ độ phân giải để phân biệt chi tiết bề mặt.
  - Ảnh có thể là ảnh xám (grayscale) hoặc ảnh màu (RGB), tùy theo điều kiện thu nhận dữ liệu.
  - Ảnh đã được cắt sẵn vùng chứa ốc vít, không cần thực hiện thêm bước phát hiện đối tượng.

## Dữ liệu đầu ra (Output)

Đầu ra của hệ thống là một **nhãn phân loại** biểu diễn tình trạng bề mặt của ốc vít. Nhãn này thuộc tập hữu hạn các lớp được xác định trước dựa trên yêu cầu ứng dụng.

## Yêu cầu của hệ thống

- Hệ thống phải có khả năng **phân loại chính xác** các tình trạng bề mặt khác nhau, bao gồm cả những khác biệt nhỏ, khó phát hiện bằng mắt thường.
- Độ chính xác tổng thể cần đạt mức cao, ổn định qua các tập kiểm thử.
- Hệ thống cần có khả năng **mở rộng**, dễ dàng cập nhật khi có thêm dữ liệu hoặc lớp lỗi mới.

## Ràng buộc và giả định

- Mỗi ảnh đầu vào chỉ chứa một ốc vít duy nhất, nằm trọn trong khung hình.
- Không xét đến trường hợp ảnh chứa nhiều ốc vít hoặc ảnh trong môi trường phức tạp (nhiều nhiễu nền).
- Không thực hiện phát hiện đối tượng (detection) hoặc phân vùng ảnh (segmentation). Chỉ tập trung vào **bài toán phân loại toàn ảnh**.

## 4 Phương pháp giải quyết

### 4.1 Kiến trúc tổng quan hệ thống

Hệ thống phân loại tình trạng lỗi bề mặt ốc vít được thiết kế theo một quy trình xử lý tuần tự, bao gồm các giai đoạn chính: tiền xử lý dữ liệu, trích xuất đặc trưng, huấn luyện mô hình, dự đoán và đánh giá hiệu năng. Đây là một kiến trúc tổng quát, có thể áp dụng cho nhiều hướng tiếp cận khác nhau trong lĩnh vực thị giác máy tính.

Trước tiên, dữ liệu hình ảnh đầu vào sẽ trải qua bước tiền xử lý nhằm đảm bảo tính nhất quán và chất lượng phù hợp cho các bước tiếp theo. Các thao tác phổ biến ở giai đoạn này bao gồm chuẩn hóa kích thước ảnh, chuyển đổi định dạng màu, loại bỏ nhiễu và chuẩn hóa giá trị điểm ảnh. Mục tiêu là đưa dữ liệu về một định dạng chuẩn, dễ dàng khai thác thông tin.

Sau khi được tiền xử lý, ảnh sẽ được chuyển sang bước trích xuất đặc trưng. Giai đoạn này đóng vai trò quan trọng trong việc biến đổi dữ liệu hình ảnh sang dạng biểu diễn có ý nghĩa – tức là làm nổi bật những đặc điểm liên quan đến hình dạng, kết cấu hoặc các dấu hiệu nhận diện lỗi. Các đặc trưng thu được sẽ là cơ sở để huấn luyện mô hình phân loại trong bước tiếp theo.

Tiếp theo, hệ thống tiến hành xây dựng mô hình phân loại dựa trên tập đặc trưng và nhãn đã biết. Mô hình được huấn luyện để học cách phân biệt giữa các tình trạng lỗi khác nhau của ốc vít. Quá trình huấn luyện có thể bao gồm các thao tác như tối ưu tham số, đánh giá hiệu năng tạm thời trên tập kiểm tra, và điều chỉnh để cải thiện độ chính xác tổng thể.

Khi mô hình đã được huấn luyện đầy đủ, hệ thống có thể tiếp nhận các ảnh mới và thực hiện dự đoán tình trạng lỗi. Mỗi ảnh đầu vào sẽ được xử lý

tương tự: tiền xử lý  $\rightarrow$  trích xuất đặc trưng  $\rightarrow$  đưa vào mô hình để suy luận và đưa ra nhãn đầu ra tương ứng.

Cuối cùng, để đánh giá hiệu quả hoạt động của hệ thống, các chỉ số định lượng như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (specificity), điểm F1 (F1-score) và ma trận nhầm lẫn (confusion matrix) được sử dụng. Việc đánh giá này không chỉ giúp so sánh các phương pháp mà còn cung cấp cơ sở để cải tiến hệ thống trong tương lai.

## 4.2 Trích xuất đặc trưng HOG và mô hình phân loại SVM

Histogram of Oriented Gradients (HOG) là một phương pháp trích xuất đặc trưng hiệu quả, được sử dụng rộng rãi trong các bài toán nhận dạng ảnh. Phương pháp này dựa trên việc mô tả cấu trúc hình học cục bộ của đối tượng bằng cách tính toán hướng và cường độ của gradient trong các vùng nhỏ của ảnh (cell), từ đó tạo thành một vector đặc trưng có khả năng mô tả hình dáng tổng quát mà không bị ảnh hưởng nhiều bởi nhiễu hay thay đổi ánh sáng.

Trong đề tài này, sau khi ảnh được tiền xử lý, đặc trưng HOG được trích xuất từ mỗi ảnh đầu vào với các tham số như kích thước cell, block và số hướng gradient được lựa chọn phù hợp. Vector đặc trưng thu được có kích thước cố định và được sử dụng làm đầu vào cho mô hình phân loại SVM (Support Vector Machine).

SVM là một mô hình học máy mạnh mẽ trong phân loại nhị phân và đa lớp, hoạt động dựa trên nguyên lý tìm siêu mặt phẳng tối ưu phân tách các lớp với biên rộng nhất. Để tối ưu hiệu suất mô hình, kỹ thuật tìm kiếm có hệ thống Grid Search được áp dụng để xác định bộ siêu tham số tốt nhất, bao gồm hệ số phạt  $C$  và loại kernel (ví dụ: tuyến tính hoặc RBF). Quá trình này đảm bảo rằng mô hình đạt được hiệu suất phân loại cao nhất trên dữ liệu đã trích xuất đặc trưng.

## 4.3 Fine-tuning mô hình EfficientNet-B0

**EfficientNet-B0** là một mô hình mạng nơ-ron tích chập (Convolutional Neural Network - CNN) thuộc họ EfficientNet, được phát triển bởi Tan và Le (2019). Mô hình này được thiết kế dựa trên nguyên tắc mở rộng hợp nhất (compound scaling), tối ưu hóa đồng thời độ sâu (depth), chiều rộng (width) và độ phân giải (resolution) của mạng để đạt hiệu suất cao với chi phí tính toán tối thiểu. EfficientNetB0 là biến thể cơ bản nhất trong họ EfficientNet,

sử dụng hệ số mở rộng nhỏ nhưng vẫn duy trì hiệu quả vượt trội so với các mô hình truyền thống như ResNet hay VGG nhờ vào việc cân bằng các tham số mạng một cách khoa học. Mô hình tận dụng các khối tích chập di động nghịch đảo (Mobile Inverted Bottleneck Convolution - MBConv) và cơ chế chú ý (squeeze-and-excitation) để tăng cường khả năng học biểu diễn đặc trưng. EfficientNetB0 được huấn luyện trước trên tập dữ liệu ImageNet, cung cấp nền tảng mạnh mẽ cho các nhiệm vụ phân loại hình ảnh, và được tinh chỉnh (fine-tuned) trên tập dữ liệu của dự án để đảm bảo độ chính xác và khả năng thích nghi với yêu cầu cụ thể.

Bảng 2: **Mạng cơ sở EfficientNet-B0** – Mỗi hàng mô tả một giai đoạn  $i$  với  $\hat{L}_i$  tầng, độ phân giải đầu vào là  $\langle \hat{H}_i, \hat{W}_i \rangle$  và số kênh đầu ra là  $\hat{C}_i$ .

Stage $i$	Operator $\hat{F}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	$224 \times 224$	32	1
2	MBConv1, k3x3	$112 \times 112$	16	1
3	MBConv6, k3x3	$112 \times 112$	24	2
4	MBConv6, k5x5	$56 \times 56$	40	2
5	MBConv6, k3x3	$28 \times 28$	80	3
6	MBConv6, k5x5	$14 \times 14$	112	3
7	MBConv6, k5x5	$14 \times 14$	192	4
8	MBConv6, k3x3	$7 \times 7$	320	1
9	Conv1x1 & Pooling & FC	$7 \times 7$	1280	1

**Chiến lược fine-tune** được áp dụng trong đề tài là giữ nguyên toàn bộ phần thân mạng Convolution của EfficientNetB0 và chỉ thay thế tầng phân loại cuối cùng bằng một tầng Fully Connected mới, phù hợp với số lớp của bài toán phân loại lỗi ốc vít. Toàn bộ phần thân mạng được đóng băng, tức là các trọng số convolution không được cập nhật trong quá trình huấn luyện. Việc này nhằm bảo toàn khả năng trích xuất đặc trưng đã được học trước đó và hạn chế hiện tượng overfitting khi dữ liệu huấn luyện không quá lớn hoặc đa dạng.

Phần đầu ra mới sẽ được khởi tạo và huấn luyện lại hoàn toàn để mô hình có thể học được mối quan hệ giữa các đặc trưng tổng quát được trích xuất bởi phần thân mạng với các nhãn lỗi cụ thể của ốc vít trong tập dữ liệu hiện có. Chiến lược này cho phép mô hình tận dụng hiệu quả các kiến thức đã học, đồng thời tập trung vào việc thích nghi với đặc điểm riêng của bài toán phân loại lỗi bề mặt ốc vít, giúp tăng cường độ chính xác mà vẫn tiết kiệm được tài nguyên và thời gian huấn luyện.

## 5 Thực nghiệm

### 5.1 Dataset

Thực nghiệm cho đề tài sử dụng một tập dữ liệu chuyên biệt phục vụ cho bài toán phân loại lỗi bề mặt ốc vít, bao gồm nhiều ảnh ốc vít với các trạng thái và loại lỗi khác nhau được gán nhãn sẵn. Đây là tập dữ liệu nhóm tự phát triển từ tập dữ liệu có sẵn MVTec AD được lấy từ nền tảng Kaggle. Do dữ liệu gốc có số lượng tương đối ít và số lượng ảnh ở các lớp không đồng đều nên nhóm đã áp dụng kỹ thuật tăng cường dữ liệu (Data Augmentation) với tỉ lệ phù hợp cho từng lớp để nhận được một tập dữ liệu với kích thước đủ lớn và đảm bảo mức độ cân bằng giữa các lớp.

Quy trình tăng cường được xây dựng dựa trên một pipeline được định nghĩa bằng Albumentations, bao gồm các phép biến đổi sau:

- Lật ngang và lật dọc (HorizontalFlip và VerticalFlip với xác suất 0.5): Tạo ảnh gương ngược theo chiều ngang và dọc để mô hình học được các đặc trưng không phụ thuộc vào hướng của vít.
- Xoay ngẫu nhiên (Rotate với giới hạn 20 độ và xác suất 0.8): Thay đổi góc quay của ảnh trong khoảng  $\pm 20$  độ, giúp mô hình thích nghi với các góc chụp khác nhau.
- Thay đổi độ sáng và tương phản (RandomBrightnessContrast với xác suất 0.7): Điều chỉnh ngẫu nhiên độ sáng và tương phản để mô hình xử lý tốt hơn trong các điều kiện ánh sáng khác nhau.
- Làm mờ Gaussian (GaussianBlur với xác suất 0.3): Thêm nhiễu mờ nhẹ để mô hình học được cách xử lý ảnh có chất lượng không hoàn hảo.
- Thay đổi kích thước (Resize về 224x224 pixel): Đảm bảo tất cả ảnh đầu ra có kích thước đồng nhất.

Hàm tăng cường được sử dụng để áp dụng quy trình cho toàn bộ các lớp trong thư mục gốc. Tập dữ liệu bao gồm các lớp như good và các lớp lỗi (defect), với số lượng ảnh tăng cường được điều chỉnh linh hoạt tùy thuộc vào số lượng ảnh gốc:

- Đối với lớp good, số lượng ảnh tăng cường được đặt là 12 ảnh cho mỗi ảnh gốc.
- Đối với các lớp lỗi (defect), số lượng ảnh tăng cường được đặt là 20 ảnh cho mỗi ảnh gốc.

Bộ dữ liệu bao gồm các ảnh ốc vít đơn lẻ đã được cắt sẵn, mỗi ảnh thể hiện một con ốc vít duy nhất với một tình trạng bề mặt cụ thể. Kích thước ảnh không cố định hoàn toàn nhưng đều đảm bảo đủ độ phân giải để mô hình có thể trích xuất các đặc trưng cần thiết cho việc phân loại. Các ảnh được lưu dưới định dạng phổ biến như .jpg hoặc .png, phần lớn là ảnh màu (RGB), tuy nhiên trong một số trường hợp có thể được xử lý chuyển sang ảnh xám (grayscale) để đơn giản hóa quá trình trích xuất đặc trưng.

Các trạng thái bề mặt ốc vít trong tập dữ liệu:

**Good:** Lớp này bao gồm các hình ảnh ốc vít không có bất kỳ lỗi bề mặt hay sai sót hình học nào. Các ốc vít có hình dạng đồng nhất, không trầy xước, biến dạng hay bị thao túng. Đây là lớp đại diện cho sản phẩm đạt tiêu chuẩn chất lượng.

**Manipulated Front:** Lớp này chứa các hình ảnh ốc vít bị thao tác hoặc biến dạng ở phần mặt trước, có thể do sai lệch trong quá trình sản xuất hoặc lỗi cơ khí. Biểu hiện có thể là phần đầu ốc bị móp, méo, không còn hình dạng ban đầu.

**Scratch Head:** Lớp này gồm các ảnh ốc vít có vết trầy xước rõ ràng trên phần đầu (đỉnh ốc). Các vết trầy này có thể do va chạm trong khâu lắp ráp, kiểm tra, hoặc đóng gói. Vết xước có thể làm ảnh hưởng đến khả năng bắt vít hoặc tính thẩm mỹ của sản phẩm.

**Scratch Neck:** Lớp này mô tả các ốc vít có vết trầy xước ở phần cổ vít, tức là phần giữa đầu và thân vít. Vết xước ở vùng này có thể ảnh hưởng đến độ bền cơ học của sản phẩm hoặc gây hư hỏng trong quá trình sử dụng.

**Thread Side:** Lớp này bao gồm những ốc vít bị lỗi tại phần rãnh xoắn bên hông thân vít, có thể là do khuyết tật, biến dạng, hoặc thiếu rãnh. Các lỗi này làm giảm khả năng giữ chặt khi sử dụng. Đây là lỗi thường gặp trong quá trình tạo ren.

**Thread Top:** Lớp này chứa các ảnh ốc vít có lỗi tại phần rãnh xoắn phía đầu vít. Các lỗi có thể bao gồm ren không đều, bị mẻ hoặc đứt đoạn, ảnh hưởng đến việc bắt vít chính xác.

Tập dữ liệu có 2872 ảnh với mỗi lớp có khoảng 460-500 ảnh. Được chia thành một tập huấn luyện gồm 1723 ảnh, một tập validation gồm 574 ảnh và một tập đánh giá gồm 575 ảnh duy trì tỉ lệ chia xấp xỉ 60:20:20.



Bảng 3: Thống kê số lượng ảnh trong mỗi lớp

	good	manipulated front	scratch head	scratch neck	thread side	thread top	Total
<b>Train</b>	295	288	288	300	276	276	1723
<b>Validation</b>	98	96	96	100	92	92	574
<b>Test</b>	99	96	96	100	92	92	575
<b>Total</b>	492	480	480	500	460	460	2872

## 5.2 Độ đo đánh giá

Để đánh giá hiệu quả của mô hình phân loại lỗi bề mặt ốc vít, các độ đo đánh giá tiêu chuẩn trong bài toán phân loại được sử dụng. Các độ đo này được tính toán dựa trên các giá trị sau:

- **TP (True Positive):** Số mẫu thực sự thuộc một lớp và được mô hình dự đoán đúng là lớp đó.
- **TN (True Negative):** Số mẫu không thuộc lớp đang xét và được mô hình dự đoán đúng là không thuộc lớp đó.
- **FP (False Positive):** Số mẫu không thuộc lớp đang xét nhưng bị mô hình dự đoán nhầm là thuộc lớp đó.
- **FN (False Negative):** Số mẫu thực sự thuộc lớp đang xét nhưng bị mô hình dự đoán nhầm sang lớp khác.

Từ đó, các độ đo được định nghĩa như sau:

- **Độ chính xác (Accuracy):**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Độ chính xác theo lớp (Precision):**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Độ nhạy (Recall):**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-score:**

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Đối với bài toán phân loại nhiều lớp như trong đề tài này, các độ đo trên được tính cho từng lớp (macro) hoặc trung bình theo số lượng mẫu của từng lớp (weighted). Ngoài ra, **ma trận nhầm lẫn (confusion matrix)** cũng được sử dụng để phân tích trực quan lỗi mà mô hình hay mắc phải giữa các lớp.

## 5.3 Thiết lập mô hình cho thực nghiệm

### 5.3.1 HOG và SVM

Trong thực nghiệm này, phương pháp kết hợp giữa trích xuất đặc trưng HOG và mô hình phân loại SVM được triển khai nhằm xây dựng hệ thống phân loại lỗi bề mặt ốc vít. Các bước cụ thể như sau:

- **Tiền xử lý ảnh:** Tất cả ảnh đầu vào được chuyển sang ảnh xám và thay đổi kích thước về cùng một kích thước chuẩn ( $128 \times 128$  pixel).
- **Trích xuất đặc trưng HOG:** Sử dụng thuật toán Histogram of Oriented Gradients (HOG) để mô tả đặc trưng hình học của đối tượng. Các tham số HOG như kích thước cell (12,12), kích thước block (2,2), được chọn do  $12 \times 12$  pixels per cell tạo ra đặc trưng tổng quát hơn, mô tả hướng cạnh trên các vùng lớn hơn. Trong bài toán phân loại lỗi, lỗi thường là các vùng bất thường lớn hoặc rõ ràng (ví dụ: cong, gãy, nứt), nên mô tả tổng quát lại mang tính phân biệt cao hơn, ốc vít có hình dạng thuôn dài, với các đặc điểm định hướng rõ (như rãnh xoắn) những đặc điểm này vẫn được giữ nguyên với cell lớn hơn, trong khi các chi tiết nhỏ bị bỏ qua, với (12,12), số lượng đặc trưng HOG giảm giúp tránh overfitting, đặc biệt khi tập dữ liệu đang sử dụng nhỏ và mô hình đơn giản như SVM. Kết quả của bước này là một vector đặc trưng đại diện cho mỗi ảnh.
- **Huấn luyện mô hình SVM:** Các vector đặc trưng từ HOG được đưa vào mô hình phân loại Support Vector Machine (SVM). Để lựa chọn siêu tham số tối ưu cho SVM, quá trình **Grid Search** được áp dụng, trong đó thử nghiệm nhiều giá trị khác nhau của các tham số như hệ số phạt  $C$  và loại kernel (linear, RBF, v.v.). Quá trình này được thực hiện kết hợp với đánh giá chéo (cross-validation) nhằm đảm bảo mô hình không bị overfitting và có khả năng tổng quát tốt.
- **Đánh giá:** Sau khi huấn luyện, mô hình được đánh giá trên tập kiểm tra bằng độ chính xác (accuracy).

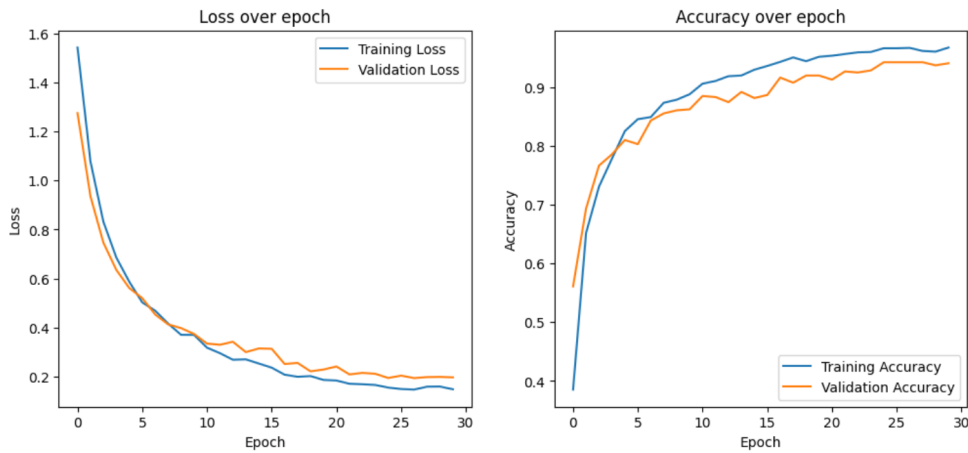
Phương pháp này có ưu điểm là đơn giản, dễ triển khai và hiệu quả trên các tập dữ liệu nhỏ hoặc trung bình. Tuy nhiên, hạn chế của nó là độ chính xác có thể bị ảnh hưởng khi gặp các mẫu có nhiều nhiễu, biến dạng hình học hoặc ánh sáng không đồng đều.

### 5.3.2 EfficientNet-B0

Chúng tôi bắt đầu bằng việc tải EfficientNet-B0 với trọng số từ ImageNet, bỏ đi tầng đầu ra gốc và thiết lập kích thước đầu vào là 224x224 pixel với 3 kênh màu. Để giữ nguyên các đặc trưng đã học, chúng tôi đóng băng toàn bộ base model, sau đó thêm vào một số tầng mới: tầng GlobalAveragePooling2D để thu gọn dữ liệu từ các feature map thành một vector đơn giản, tiếp theo là một tầng Dense với 128 đơn vị và hàm kích hoạt ReLU nhằm học thêm các đặc trưng phù hợp với bài toán, và cuối cùng là một tầng Dense với 6 đơn vị cùng hàm kích hoạt softmax để dự đoán xác suất cho từng lớp. Mô hình được biên dịch với optimizer Adam, đặt learning rate ban đầu là 1e-3, sử dụng hàm mất mát `sparse_categorical_crossentropy` phù hợp cho bài toán phân loại đa lớp, và theo dõi độ chính xác qua chỉ số accuracy.

Quá trình huấn luyện diễn ra trong 100 epoch, sử dụng tập huấn luyện để học và tập validation để theo dõi hiệu suất. Chúng tôi thiết lập các cơ chế kiểm soát bao gồm ModelCheckpoint để lưu mô hình tốt nhất, EarlyStopping để dừng sớm nếu hiệu suất không cải thiện, và ReduceLROnPlateau để điều chỉnh learning rate. Sau 30 epoch, mô hình đạt đỉnh cao với độ chính xác validation cao nhất tại epoch 27, trước khi dừng sớm và khôi phục trọng số tốt nhất. Khi đánh giá, mô hình đạt độ chính xác 94%, với F1-score trung bình là 0.94, cho thấy khả năng phân loại đồng đều và chính xác.

Trong quá trình huấn luyện, chúng tôi sử dụng các callback của Keras để tối ưu hóa hiệu suất. ModelCheckpoint đảm bảo rằng mô hình tốt nhất được lưu lại dựa trên giá trị `val_loss` thấp nhất, trong khi EarlyStopping giúp dừng huấn luyện khi `val_loss` không cải thiện sau 3 epoch, đồng thời khôi phục trọng số tốt nhất từ epoch 27. ReduceLROnPlateau đóng vai trò điều chỉnh learning rate linh hoạt, giảm từ 1e-3 xuống 6.25e-5 qua các epoch, giúp mô hình hội tụ tốt hơn và tránh vượt qua điểm tối ưu.



## 5.4 Kết quả thực nghiệm

Bảng 4: Kết quả phân loại theo từng lớp của mô hình HOG SVM

	precision	recall	f1-score	support
good	0.55	0.64	0.59	99
manipulated front	0.84	0.76	0.80	96
scratch head	0.57	0.58	0.58	96
scratch neck	0.69	0.66	0.68	100
thread side	0.59	0.53	0.56	92
thread top	0.54	0.58	0.56	92
accuracy	-	-	0.63	575
macro avg	0.63	0.62	0.63	575
weighted avg	0.63	0.63	0.63	575

Hiệu suất của mô hình SVM dựa trên HOG đạt accuracy trung bình 69%, cho thấy khả năng phân loại khá tốt nhưng chưa tối ưu.

Lớp manipulated front có hiệu suất cao nhất với F1-score 0.80, nhờ precision và recall cân bằng tốt (0.84 và 0.76).

Lớp thread side có hiệu suất thấp nhất với F1-score 0.56, do recall chỉ đạt 0.59, cho thấy mô hình bỏ sót nhiều mẫu của lớp này.

Các chỉ số macro avg và weighted avg gần bằng nhau (khoảng 0.62-0.63), phản ánh phân phối lớp tương đối cân bằng trong tập kiểm tra (support dao động từ 92 đến 100).

Bảng 5: Kết quả phân loại theo từng lớp của mô hình EfficientNet-B0

	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
good	0.93	0.99	0.96	99
manipulated front	0.97	0.97	0.97	96
scratch head	0.94	0.98	0.96	96
scratch neck	1.00	0.94	0.97	100
thread side	0.92	0.83	0.87	92
thread top	0.90	0.95	0.92	92
accuracy	-	-	0.94	575
macro avg	0.94	0.94	0.94	575
weighted avg	0.94	0.94	0.94	575

Mô hình EfficientNet-B0 đạt hiệu suất vượt trội với accuracy 94%, cao hơn đáng kể so với SVM.

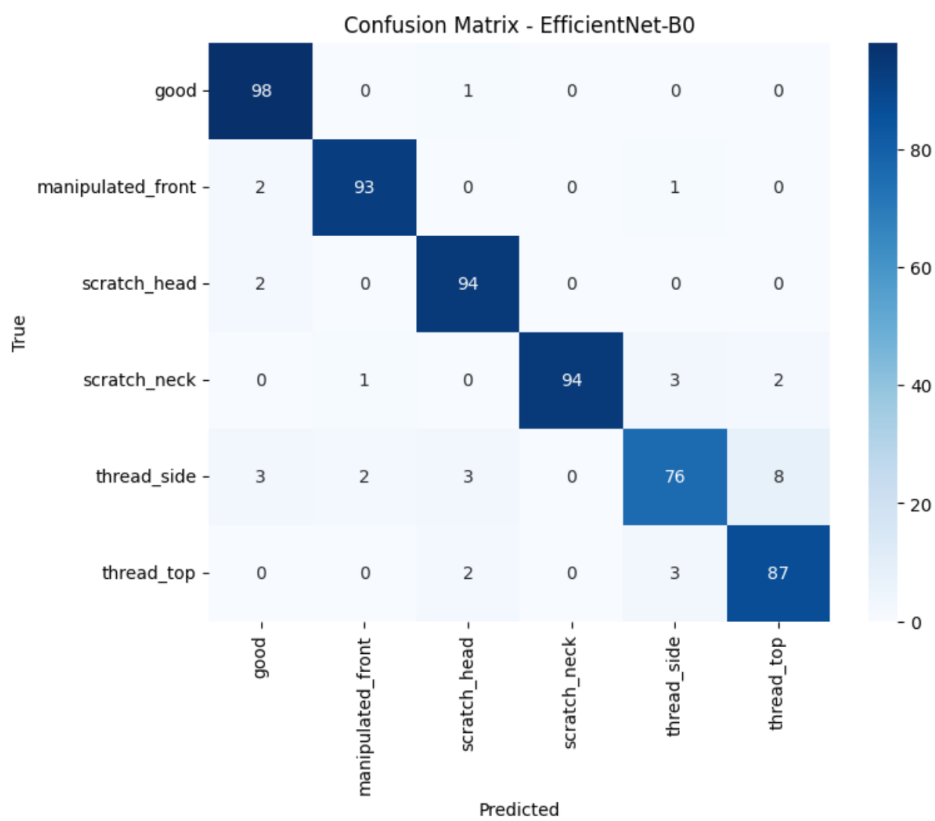
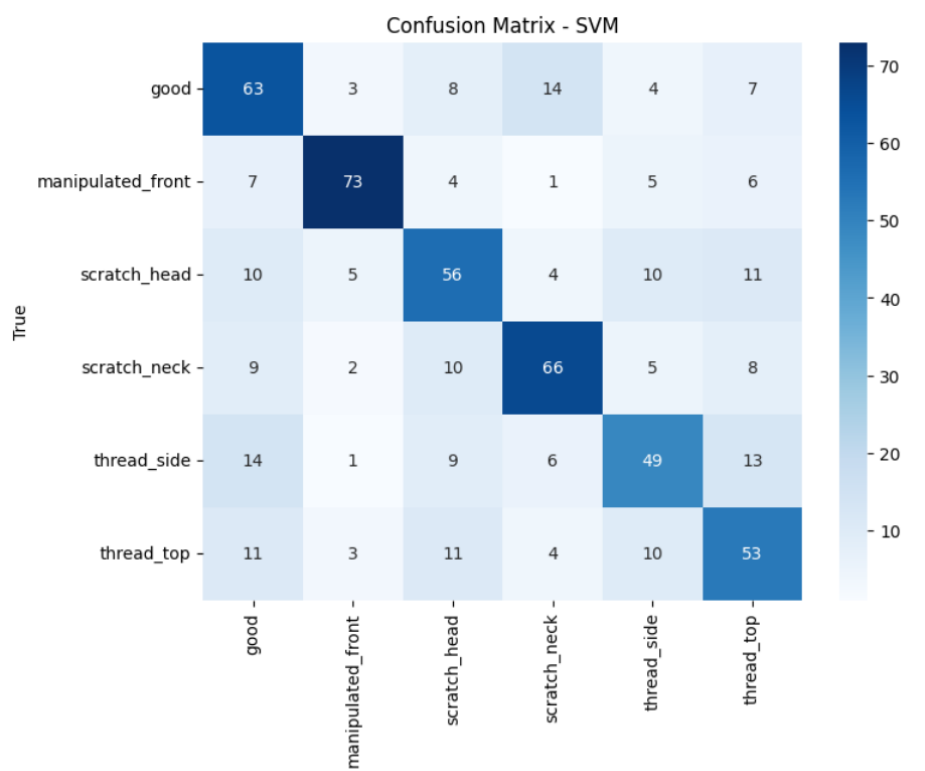
Tất cả các lớp đều có F1-score trên 0.86, với lớp good, manipulated\_front, và scratch\_head đạt F1-score 0.96–0.97, cho thấy khả năng phân loại rất tốt.

Lớp thread\_side có hiệu suất thấp nhất với F1-score 0.87, do precision và recall thấp hơn một chút (0.92 và 0.83), nhưng vẫn vượt trội so với SVM.

Các chỉ số macro avg và weighted avg đều đạt 0.94, phản ánh mô hình hoạt động nhất quán trên tất cả các lớp, nhờ vào việc học sâu từ đặc trưng hình ảnh.

## 5.5 Ma trận nhầm lẫn(Confusion matrix)

Để đánh giá hiệu suất của các mô hình HOG SVM và EfficientNet-B0 chi tiết hơn, chúng tôi trình bày các ma trận nhầm lẫn cho các cấu hình có hiệu suất tốt nhất của chúng.



**HOG SVM:** Ma trận nhầm lẫn cho thấy mô hình gặp khó khăn trong việc phân biệt một số lớp. Lớp thread side có recall thấp nhất (0.53), nghĩa là chỉ 53% mẫu của lớp này được dự đoán đúng (khoảng 49/92 mẫu), với 43 mẫu bị nhầm sang các lớp khác, có thể là thread top (do đặc trưng tương đồng). Lớp manipulated front và scratch neck có recall khá hơn (0.76 và 0.66), lần lượt đúng khoảng 73/96 và 66/100 mẫu, nhưng vẫn có nhầm lẫn đáng kể.

**EfficientNet-B0:** Ma trận nhầm lẫn thể hiện hiệu suất vượt trội, với các lớp đều có recall cao. Lớp good đạt recall 0.99 (98/99 mẫu đúng), scratch neck đạt 0.94 (94/100 mẫu đúng), nhưng lớp thread side có recall thấp nhất (0.83, khoảng 76/92 mẫu đúng), nghĩa là 16 mẫu bị nhầm sang lớp khác. Tuy nhiên, nhầm lẫn tổng thể giảm đáng kể so với SVM.

**So sánh:** EfficientNet-B0 phân loại chính xác hơn, với ít nhầm lẫn hơn giữa các lớp (đặc biệt ở thread side), trong khi HOG SVM có nhầm lẫn rõ rệt, đặc biệt ở các lớp khó như thread side và thread top.

## 5.6 Phân tích lỗi

Từ ma trận nhầm lẫn ta có thể thấy lớp bị dự đoán sai nhiều ở cả hai mô hình là thread side. Nguyên nhân có thể đến từ:

Dataset có thể không đủ đa dạng, đặc biệt ở các lớp khó như thread side (460 ảnh, ít hơn scratch neck với 500 ảnh).

Một số ảnh có thể bị nhiễu (mờ, ánh sáng kém, góc nhìn), ảnh hưởng đến khả năng học đặc trưng của mô hình.

Nguyên nhân từ mô hình: Mặc dù EfficientNet-B0 đạt hiệu suất cao (94%), base model bị đóng băng (trainable = False) có thể hạn chế khả năng học các đặc trưng cụ thể của dataset, đặc biệt với các lớp tương đồng như thread side và thread top. Còn với HOG SVM, có thể do HOG chưa đủ tối ưu cho những đặc trưng của lớp ảnh này dẫn đến dự đoán kém hiệu quả.

True: thread\_side  
Pred: thread\_top



Góc chụp trong ảnh này làm nổi bật phần đầu vít và ren ở đỉnh, giống với đặc trưng của thread top. Phần cạnh bên (đặc trưng của thread side) bị ẩn hoặc không đủ rõ để mô hình phân biệt. Góc chụp từ trên xuống làm mất đi đặc trưng cạnh bên, vốn là yếu tố quan trọng để phân biệt thread side. Điều này khiến mô hình nhầm sang thread top, vốn dựa vào phần đầu vít. Mô hình EfficientNet-B0, với base model đóng băng (trainable = False), có thể đã dựa nhiều vào trọng số pre-trained từ ImageNet, vốn ưu tiên các đặc trưng tổng quát (như hình dạng đầu vít) hơn là chi tiết cụ thể ở cạnh bên. Dataset có thể thiếu các ảnh thread side với góc chụp đa dạng, dẫn đến mô hình không học được đầy đủ các biến thể của lớp này.

## 6 Kết luận

Nghiên cứu này đã thực hiện phân tích toàn diện và so sánh hai phương pháp phân loại lỗi bề mặt ốc vít: phương pháp truyền thống HOG kết hợp SVM và phương pháp học sâu với mô hình EfficientNet-B0 fine-tune. Các thí nghiệm được triển khai trên tập dữ liệu ốc vít với 6 lớp lỗi, được chia tỉ lệ hợp lý cho train/validation/test, và sử dụng các độ đo đánh giá phổ biến (accuracy, precision, recall, F1-score) để đảm bảo tính khách quan.

Kết quả cho thấy mô hình học sâu (EfficientNetB0 fine-tune) có độ chính xác và độ ổn định cao hơn so với phương pháp truyền thống, khẳng định sức mạnh của đặc trưng học sâu đã được huấn luyện trước. Tuy vậy, phương pháp HOG kết hợp SVM vẫn đạt kết quả tương đối tốt, minh chứng cho tính khả thi của các phương pháp truyền thống trong trường hợp không có điều kiện sử dụng mô hình phức tạp.



**Hạn chế** Mặc dù đạt được kết quả tương đối tốt, nghiên cứu vẫn tồn tại một số hạn chế quan trọng. Đặc biệt, một hạn chế lớn đến từ chính bản chất của tập dữ liệu: toàn bộ dataset chỉ bao gồm hình ảnh của một loại ốc vít duy nhất, được chụp trong điều kiện kiểm soát với các tình trạng lỗi khác nhau. Điều này dẫn đến mô hình học được các đặc trưng rất đặc thù của loại ốc vít đó như hình dạng, kết cấu bề mặt, tỉ lệ giữa các bộ phận... Do đó, khi áp dụng mô hình đã huấn luyện này lên hình ảnh của các loại ốc vít có hình dạng hoặc thiết kế khác (ví dụ: ốc vít kích thước khác, loại ren khác, hoặc loại ốc vít từ nhà cung cấp khác), mô hình dễ bị nhận nhầm hoặc dự đoán sai. Hiện tượng này phản ánh rõ nét sự phụ thuộc của mô hình vào “domain” ban đầu, mô hình không đủ tổng quát hóa khi gặp những đặc trưng hình học khác lạ so với tập dữ liệu huấn luyện.

Ngoài ra, do giới hạn về thời gian và tài nguyên, việc tinh chỉnh tham số tối ưu chưa được thực hiện triệt để, có thể bỏ lỡ các cấu hình tham số tốt hơn. Thêm nữa, thí nghiệm chỉ được thực hiện trong môi trường giả lập, chưa mô phỏng đủ các yếu tố biến thiên trong thực tế như ánh sáng phức tạp, góc chụp thay đổi, hoặc sự che khuất bề mặt ốc vít.

**Hướng nghiên cứu tương lai** Để khắc phục những hạn chế này, nghiên cứu tương lai nên tập trung mở rộng tập dữ liệu, bổ sung hình ảnh của nhiều loại ốc vít khác nhau với các hình dạng, kích thước, và kiểu ren đa dạng. Điều này giúp mô hình học được các đặc trưng tổng quát và giảm hiện tượng nhận nhầm khi gặp dữ liệu thực tế. Bên cạnh đó, các phương pháp cân bằng dữ liệu như oversampling/undersampling hoặc kỹ thuật tổng hợp dữ liệu cần được áp dụng để cải thiện khả năng nhận diện các lớp hiếm gặp. Về mô hình, việc kết hợp các mô hình học sâu tiên tiến hơn (như transformers hoặc mô hình CNN phức tạp hơn) có thể khai thác tốt hơn các đặc trưng hình ảnh. Cuối cùng, triển khai và kiểm thử mô hình trong môi trường thực tế, với các biến thiên ánh sáng, góc chụp và sự biến dạng của ốc vít sẽ giúp đánh giá độ bền vững và khả năng ứng dụng thực tế của hệ thống.

**Tóm lại** nghiên cứu này đã phân tích và so sánh hai phương pháp chính SVM kết hợp HOG và mô hình học sâu EfficientNetB0 fine-tune để phân loại lỗi bề mặt ốc vít. Mặc dù còn tồn tại hạn chế, đặc biệt do tập dữ liệu chỉ chứa ảnh của một loại ốc vít nên chưa tổng quát hóa tốt, nghiên cứu đã đặt nền móng vững chắc để tiếp tục phát triển các mô hình kiểm tra lỗi tự động trong công nghiệp. Các kết quả này không chỉ cung cấp cơ sở so sánh quan trọng giữa mô hình học sâu và phương pháp truyền thống, mà còn mở đường cho việc áp dụng các mô hình tiên tiến hơn, khai thác tập dữ liệu đa dạng hơn nhằm nâng cao độ chính xác và khả năng tổng quát của hệ thống trong thực tế.

## Tài liệu tham khảo

- [1] Connor Shorten and Taghi M. Khoshgoftaar, *A survey on image data augmentation for deep learning*, Journal of Big Data, vol. 6, no. 1, pp. 1–48, 2019.
- [2] N. Dalal and B. Triggs, *Histograms of Oriented Gradients for Human Detection*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 886–893.
- [3] C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
- [4] Z. Zhang, *Feature-based image classification using HOG and SVM*, Proc. of the 2012 Int. Conf. on Computer Application and System Modeling (ICCASM), pp. V5-180–V5-183.
- [5] Mingxing Tan and Quoc V. Le, *EfficientNet: Rethinking model scaling for convolutional neural networks*, arXiv preprint arXiv:1905.11946, 2019.