

THÀNH PHỐ HỒ CHÍ MINH **TRƯỜNG ĐẠI HỌC SÀI GÒN**

**BÁO CÁO TỔNG KẾT MÔN HỌC**

**KHAI PHÁ DỮ LIỆU**

**MÃ MÔN HỌC: 841447**

**PHÂN LOẠI VẢI DỆT THÔNG QUA KẾT CẤU**

**TRONG CÔNG NGHIỆP BẰNG MẠNG THẦN KINH TÍNH CHẬP**

Sinh viên thực hiện đề tài: Nguyễn Quốc Huy

Giảng viên hướng dẫn: Th.S Nguyễn Thanh Phước

**Thành phố Hồ Chí Minh, 5/2025**

# LỜI CẢM ƠN

Để hoàn thành bài tập này, em xin tỏ lòng biết ơn sâu sắc đến Thầy Nguyễn Thanh Phước đã tận tình hướng dẫn, chỉ dạy trong suốt quá trình thực hiện đề tài. Suốt khoảng thời gian qua, dù đã rất cố gắng, một phần cũng do kiến thức còn hạn chế nên bài báo cáo này khó tránh những sai sót. Em rất mong được nhận ý kiến đóng góp của Thầy để bài báo cáo này được hoàn hảo hơn. Em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc198711891)

[DANH MỤC HÌNH i](#_Toc198711892)

[DANH MỤC CÁC CHỮ VIẾT TẮT i](#_Toc198711893)

[TÓM TẮT ĐỀ TÀI 2](#_Toc198711894)

[1. PHẦN MỞ ĐẦU 7](#_Toc198711895)

[1.1. Lý do chọn đề tài 7](#_Toc198711896)

[1.2. Lịch sử nghiên cứu vấn đề 7](#_Toc198711897)

[1.3. Mục đích và nhiệm vụ nghiên cứu 8](#_Toc198711898)

[1.4. Đối tượng và phạm vi nghiên cứu 9](#_Toc198711899)

[1.5. Phương pháp nghiên cứu 9](#_Toc198711900)

[1.5.1. CNN (Tensorflow) 9](#_Toc198711901)

[1.5.2. CNN(Tensorflow) kết hợp SVM (sử dụng trích xuất HOG) 9](#_Toc198711902)

[1.5.3. CNN(Tensor) 10](#_Toc198711903)

[1.6. Cấu trúc đề tài 11](#_Toc198711904)

[2. PHẦN NỘI DUNG 12](#_Toc198711905)

[2.1. Thực nghiệm và đánh giá 13](#_Toc198711906)

[2.1.1. Miêu tả bộ dữ liệu. 13](#_Toc198711907)

[2.1.2. Lấy dữ liệu từ Kaggle 13](#_Toc198711908)

[2.1.3. Kết quả của bài toán theo cách hướng phát triển CNN (Tensorflow) 14](#_Toc198711909)

[2.1.4. Kết quả của bài toán theo cách hướng phát triển CNN(Tensorflow) kết hợp SVM (sử dụng trích xuất HOG) 18](#_Toc198711910)

[2.1.5. Kết quả của bài toán theo cách hướng phát triển CNN(Tensor) 21](#_Toc198711911)

[2.2. Kết luận 24](#_Toc198711912)

[2.2.1. Tổng kết kết quả đạt được 24](#_Toc198711913)

[2.2.2. Hạn chế và khó khăn 24](#_Toc198711914)

[TÀI LIỆU THAM KHẢO 27](#_Toc198711915)

# DANH MỤC HÌNH

[Hình 1. Sơ đồ khối Hướng phát triển CNN(Tensorflow) 3](#_Toc198647267)

[Hình 2. Sơ đồ khối phương pháp nghiên cứu bài toán CNN(Tensorflow) kết hợp SVM (sử dụng trích xuất HOG) 4](#_Toc198647268)

[Hình 3. Sơ đồ khối phương pháp nghiên cứu bài toán CNN (Tensor) 5](#_Toc198647269)

[Hình 4. Ảnh minh họa cho các loại vải 8](#_Toc198647270)

[Hình 5. Cấu trúc của mạng CNN(Tensorflow) 10](#_Toc198647271)

[Hình 6. Quá trình huấn luyện của mạng CNN (Tensorflow) 12](#_Toc198647272)

[Hình 7. Biểu đồ độ chính xác của mô hình CNN (Tensorflow) 12](#_Toc198647273)

[Hình 8. Biểu đồ Loss của mô hình CNN (Tensorflow) 13](#_Toc198647274)

[Hình 9. Biều đồ Loss và Độ chính xác của mô hình CNN (Tensorflow) của Loại Good vs All 13](#_Toc198647275)

[Hình 10. Bảng báo cáo của mô hình CNN (Tensorflow) Loại Good vs All 14](#_Toc198647276)

[Hình 11. Biểu đồ Loss và Độ chính xác của mô hình CNN (Tensorflow) Loại Metal vs All 14](#_Toc198647277)

[Hình 12. Bảng báo cáo của mô hình CNN (Tensorflow) Loại Metal vs All 15](#_Toc198647278)

[Hình 13. Hính ảnh minh họa cho trích xuất HOG cho 3 loại Cut, Hole, Thread 15](#_Toc198647279)

[Hình 14. Bảng báo cáo của mô hình SVM 16](#_Toc198647280)

[Hình 15. Sơ đồ hoạt động của Deep Learning 17](#_Toc198647281)

[Hình 16. Kết quả huấn luyện của mô hình CNN(Tensor) phân loại Good vs All 18](#_Toc198647282)

[Hình 17. Kết quả huấn luyện của mô hình CNN(Tensor) phân loại Metal vs All 18](#_Toc198647283)

[Hình 18. Kết quả huấn luyện của mô hình CNN(Tensor) phân loại Cut vs All 19](#_Toc198647284)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Diễn giải đầy đủ (Tiếng Anh)** | **Diễn giải/Mô tả (Tiếng Việt)** |
| CNN | Convolutional Neural Network | Mạng nơ ron tính chập |
| R-CNN | Regions with Convolutional Neural Network features | R-CNN là một mô hình phát hiện đối tượng sử dụng mạng nơ-ron tích chập (CNN) để trích xuất đặc trưng từ các vùng được đề xuất trong ảnh, sau đó phân loại từng vùng bằng SVM. |
| SSD | |  | | --- | | Single Shot MultiBox Detector |  |  | | --- | |  | | SSD là mô hình phát hiện đối tượng thực hiện nhận dạng và định vị đối tượng trong ảnh chỉ trong một lần lan truyền (single shot) qua mạng, cho tốc độ nhanh hơn R-CNN mà vẫn duy trì độ chính xác cao. |
| YOLO | |  | | --- | | You Only Look Once | | YOLO là một mô hình phát hiện đối tượng thời gian thực. Thay vì quét từng vùng như R-CNN, YOLO chia ảnh đầu vào thành các ô lưới và dự đoán trực tiếp bounding box và nhãn lớp trong một lần duy nhất, giúp tăng tốc độ rất nhiều so với các phương pháp truyền thống. |
| SVM | |  | | --- | | Support Vector Machine |  |  | | --- | |  | | SVM là một thuật toán học máy có giám sát, thường được sử dụng trong các bài toán phân loại. SVM tìm siêu phẳng (hyperplane) tốt nhất để phân tách các lớp dữ liệu, hoạt động hiệu quả với dữ liệu có số chiều lớn và trong trường hợp dữ liệu không tuyến tính. |
| HOG | |  | | --- | | Histogram of Oriented Gradients |  |  | | --- | |  | | HOG là một kỹ thuật trích xuất đặc trưng hình ảnh, hoạt động bằng cách đếm số lượng các hướng gradient xuất hiện trong các vùng nhỏ của ảnh. HOG thường được sử dụng trong phát hiện đối tượng (đặc biệt là người đi bộ) nhờ khả năng nắm bắt được hình dạng và cấu trúc của đối tượng. |

# TÓM TẮT ĐỀ TÀI

Đề tài "Phân loại vải dệt thông qua kết cấu trong công nghiệp bằng mạng thần kinh tính chập" tập trung vào việc ứng dụng mạng nơ-ron tích chập (Convolutional Neural Network - CNN) để phân tích và phân loại các loại vải dệt dựa trên đặc điểm kết cấu bề mặt. Bằng cách sử dụng các thuật toán học sâu, đề tài tận dụng dữ liệu hình ảnh chất lượng cao để xây dựng mô hình nhận diện tự động, giúp giảm thiểu sự phụ thuộc vào phương pháp kiểm tra thủ công vốn tốn thời gian và dễ sai sót. Mục tiêu chính của nghiên cứu là nâng cao độ chính xác trong phân loại vải, đạt được tỷ lệ chính xác cao hơn so với các phương pháp truyền thống, đồng thời phát triển các ứng dụng thực tiễn như hệ thống kiểm định chất lượng tự động, phát hiện vật thể, hỗ trợ phân loại vải trong sản xuất hàng loạt, và tích hợp vào các dây chuyền sản xuất thông minh trong ngành công nghiệp dệt may. Kết quả từ nghiên cứu này không chỉ cải thiện chất lượng phân loại vải mà còn mang lại một hệ thống tự động hóa hiệu quả, giúp các doanh nghiệp tối ưu hóa quy trình sản xuất, giảm chi phí vận hành, nâng cao chất lượng sản phẩm cuối cùng, và mở ra tiềm năng ứng dụng trong các lĩnh vực như thời trang, nội thất, và sản xuất vải kỹ thuật.

# PHẦN MỞ ĐẦU

## Lý do chọn đề tài

Trong Trong ngành dệt may, lỗi trên vải (như lỗ thủng, rách, đốm, đan sợi không đều, nếp gấp) ảnh hưởng nghiêm trọng đến chất lượng sản phẩm, chi phí sản xuất, và hình ảnh thương hiệu. Theo báo cáo từ TextileWorld, lỗi vải có thể làm giảm giá trị thương mại của sản phẩm từ 45% đến 65%, gây ra lãng phí nguyên liệu, thiệt hại tài chính, và thậm chí dẫn đến việc trả hàng từ đối tác quốc tế.

Về mặt kinh tế, ngành dệt may là một trong những ngành sản xuất xuất khẩu mũi nhọn của nhiều quốc gia, bao gồm cả Việt Nam. Việc phát hiện và loại bỏ lỗi vải sớm, ngay trong dây chuyền kiểm tra chất lượng, giúp: Giảm thiểu tổn thất tài chính do sản phẩm bị trả lại, sửa lỗi hoặc tái chế, tăng hiệu suất lao động, vì không cần phải kiểm tra lại bằng tay nhiều lần, tối ưu hóa quy trình tự động hóa, hướng tới nhà máy thông minh (smart factory), tăng độ tin cậy của sản phẩm đầu ra, giúp giữ chân khách hàng và mở rộng thị trường.

Về mặt học thuật, đề tài phát hiện lỗi vải đặt ra nhiều thách thức kỹ thuật và có ý nghĩa lớn trong lĩnh vực thị giác máy tính (Computer Vision) và học sâu (Deep Learning) vấn đề kỹ thuật cốt lõi: Các lỗi vải có hình dạng, màu sắc, kích thước, kết cấu rất đa dạng và thường xuất hiện mờ nhạt, khiến cho các mô hình đơn giản rất khó phát hiện, khó khăn trong việc phân loại lỗi: Một số lỗi có hình thái tương tự (ví dụ: vết đốm nhỏ vs. vết bẩn), yêu cầu mô hình phải học đặc trưng rất sâu, đặt yêu cầu cao về dữ liệu và kiến trúc mạng, thiếu hụt dữ liệu chất lượng cao: Hầu hết các doanh nghiệp giữ riêng dữ liệu ảnh lỗi vải do tính chất cạnh tranh, dẫn đến khó khăn trong huấn luyện mô hình tổng quát, cần tối ưu hóa hiệu suất mô hình: Đề tài này tạo điều kiện để áp dụng, cải tiến và đánh giá các mô hình tiên tiến như YOLOv5, EfficientDet, hoặc Faster R-CNN, kết hợp với kỹ thuật augmentation, transfer learning nhằm nâng cao hiệu quả nhận diện trong điều kiện hạn chế tài nguyên, từ góc nhìn học thuật, nghiên cứu này không chỉ có giá trị trong lĩnh vực dệt may mà còn là bài toán điển hình về nhận dạng lỗi bề mặt, có thể mở rộng sang các ngành công nghiệp khác như: kiểm tra bảng mạch điện tử, bề mặt kim loại, vật liệu xây dựng, v.v.

## Lịch sử nghiên cứu vấn đề

Trong giai đoạn đầu, việc phát hiện lỗi trên bề mặt vải chủ yếu dựa vào xử lý ảnh cổ điển và mô hình thống kê thủ công. Các kỹ thuật tiêu biểu bao gồm: Bộ lọc Gabor: Dùng để phát hiện các đặc trưng kết cấu hướng tính của vải, hữu ích với các lỗi như đan sai hoặc đứt sợi theo chiều dọc/ngang. Biến đổi wavelet (Wavelet Transform): Giúp phân tích ảnh ở nhiều tần số và độ phân giải khác nhau để phát hiện bất thường về kết cấu. Ngưỡng hoá (Thresholding) và phân tích histogram: Phát hiện vùng có cường độ pixel khác biệt so với nền. Phân đoạn theo vùng kết cấu (Texture Segmentation) và đo lường thống kê (mean, variance, entropy): Dùng cho các mẫu vải có nền đều hoặc họa tiết lặp lại.

Tuy nhiên, các phương pháp này có hạn chế rõ rệt phụ thuộc nhiều vào thiết kế đặc trưng thủ công (hand-crafted features), chỉ hiệu quả với vải có mẫu nền đơn giản, ánh sáng ổn định, và lỗi rõ ràng về màu hoặc hình dạng., Khó tổng quát sang các kiểu vải có họa tiết phức tạp hoặc nền không đồng nhất.

Trong vòng 10 năm trở lại đây, đặc biệt từ sau 2014 khi các mô hình deep learning bắt đầu chứng minh hiệu quả vượt trội trong thị giác máy tính, các nghiên cứu trong phát hiện lỗi vải đã chuyển hướng sang Mạng nơ-ron tích chập (CNN): Có khả năng tự học đặc trưng hình ảnh mà không cần lập trình thủ công. CNN giúp mô hình hóa tốt các loại lỗi có dạng phức tạp hoặc có kích thước rất nhỏ. Faster R-CNN, SSD, YOLO: Đây là các mô hình phát hiện vật thể (object detection), không chỉ nhận diện ảnh có lỗi mà còn xác định vị trí lỗi trong ảnh. Ví dụ: Faster R-CNN: Hiệu quả cao, chính xác nhưng tốc độ chậm, phù hợp với kiểm tra offline. YOLOv5/v8: Cân bằng giữa tốc độ và độ chính xác, thích hợp với ứng dụng thời gian thực trong dây chuyền công nghiệp. Transfer Learning: Nhiều nghiên cứu dùng mô hình tiền huấn luyện như ResNet, MobileNet để giảm thời gian huấn luyện và cải thiện hiệu suất trên dataset nhỏ. Các kết quả thực nghiệm cho thấy: Mô hình deep learning đạt độ chính xác từ 85–95%, vượt xa các kỹ thuật truyền thống. Có thể xử lý các mẫu vải phức tạp, đa dạng về màu sắc, kết cấu, kể cả với độ tương phản thấp.

## Mục đích và nhiệm vụ nghiên cứu

Mục tiêu chính của nghiên cứu này là xây dựng một hoặc nhiều mô hình học máy và học sâu, mạng nơ-ron tích chập (CNN), Support Vector Meaching (SVM) để tự động phát hiện và phân loại lỗi trên ảnh bề mặt vải công nghiệp. Mô hình cần có khả năng phân biệt rõ ràng giữa các vùng vải bình thường và vùng có lỗi, bao gồm các loại lỗi phổ biến như đứt, lỗ thủng, lỗi dệt, lẫn tạp chất. Việc phát hiện này không chỉ dừng lại ở nhận diện có lỗi hay không mà còn phải xác định được loại lỗi cụ thể để hỗ trợ tốt cho công đoạn xử lý sau sản xuất. Bên cạnh việc xây dựng mô hình, nghiên cứu đặt mục tiêu cải thiện hiệu năng bằng cách tối ưu hóa kiến trúc mạng và tinh chỉnh các siêu tham số huấn luyện như learning rate, batch size, số epoch nhằm giúp mô hình học được tốt hơn. Phương pháp tiền xử lý ảnh HOG dược dùng nhằm tăng độ ổn định và khả năng khái quát của mô hình trong môi trường thực tế. Cuối cùng, nghiên cứu hướng đến việc đánh giá toàn diện hiệu quả mô hình trên các thước đo chuẩn trong lĩnh vực học máy, bao gồm độ chính xác (accuracy), độ nhạy (recall), độ chính xác (precision) và điểm F1 (F1-score). Kết quả thu được sẽ được so sánh với các mô hình tham khảo (baseline) hoặc các nghiên cứu hiện có để phân tích ưu và nhược điểm của mô hình đề xuất, từ đó rút ra những kết luận về tính khả thi, mức độ ứng dụng thực tiễn cũng như tiềm năng cải tiến trong các nghiên cứu tiếp theo.

.

## Đối tượng và phạm vi nghiên cứu

* **Đối tượng:** Các ảnh vải trong bộ dữ liệu “Textile Defect Detection” từ Kaggle, bao gồm ảnh miếng vải có thể có lỗi và ảnh vải bình thường. Các loại lỗi quan tâm thường là đứt sợi, lỗ thủng, lỗi chỉ, lẫn tạp chất.
* **Phạm vi:** Nghiên cứu chỉ sử dụng bộ dữ liệu đã cho, tập trung vào thiết kế, huấn luyện và đánh giá mô hình AI/ML để phát hiện/phân loại lỗi. Nghiên cứu giới hạn ở ảnh tĩnh (2D) và không xem xét các điều kiện môi trường như thay đổi ánh sáng hay video.
* **Giới hạn:** Dự án thực hiện trong khuôn khổ học phần nên có giới hạn về thời gian và tài nguyên tính toán Bộ dữ liệu có kích thước và độ đa dạng giới hạn, có thể không phản ánh đủ mọi tình huống thực tế, do đó kết quả thu được có thể không tổng quát cho mọi điều kiện sản xuất.

## Phương pháp nghiên cứu

### CNN (Tensorflow)

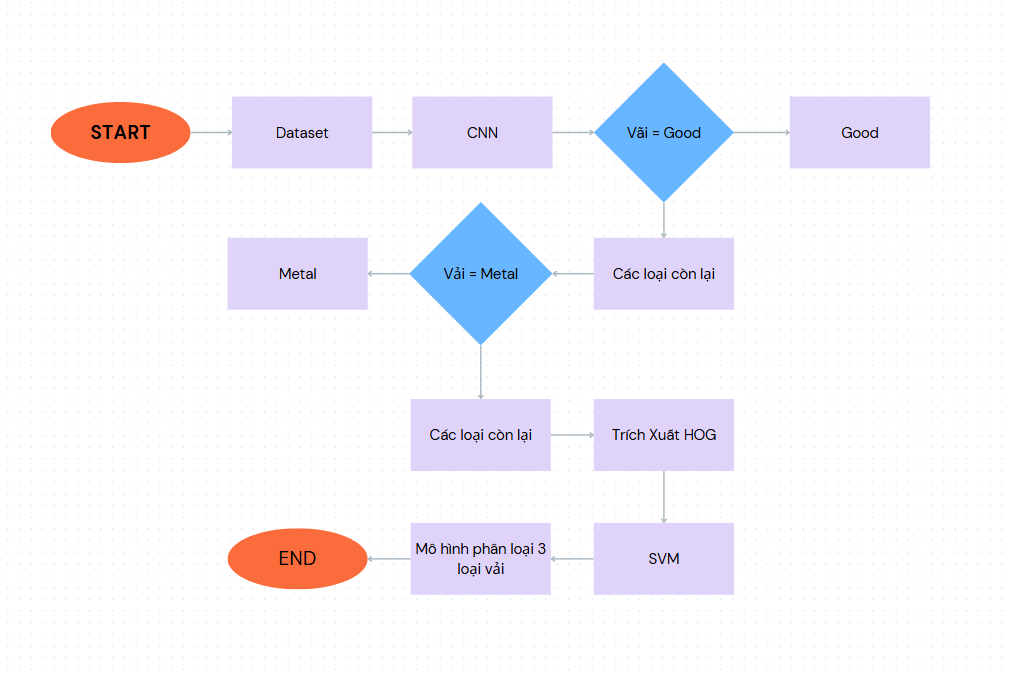
A diagram of a process

AI-generated content may be incorrect.

Hình 1. Sơ đồ khối Hướng phát triển CNN(Tensorflow)

Bắt đầu bằng từ dữ liệu sử dụng thuật toán CNN để phân loại ra các loại vải mà không cần sử lý thêm. Việc này để đánh giá sơ bộ bộ dữ liệu xem những khó khăn trong quá trình thực hiện bộ dữ liệu có bị nhiễu hay khó khăn gì không.

### CNN(Tensorflow) kết hợp SVM (sử dụng trích xuất HOG)



Hình 2. Sơ đồ khối phương pháp nghiên cứu bài toán CNN(Tensorflow) kết hợp SVM (sử dụng trích xuất HOG)

Quá trình bắt đầu bằng bước từ dữ liệu sử dụng thuật toán CNN để huần luyện mô hình để tách ra tưng loại vải

Đầu tiên tách loại vải Good với các loại vải còn lại do độ quan trọng về mặt kinh tế và độ phức tạp. Vải good có đặc điểm nhận dạng dễ nhận biết nhất so với các loại còn lại các đường chỉ đều và không có vết cắt hay lẫn tạp chất bên trong.

Sau khi tách loại Good với 4 loại còn lại, tiếp theo đến với loại Có lẫn tạp chất (Kim loại). Do xuất hiện một vật thể khác so với các picxel còn lại nên đây sẽ là loại được nhận dạng tiếp theo.

Loại tiếp theo là Hole, Cut và Thread do độ khó và độ nhiễu của dữ liệu nên 3 loại này sẽ được sử dụng phương pháp HOG để trích xuất kèm theo đó là sử dụng thuật toán SVM để huấn luyện mô hình. Để cho ra mô hình phân loại.

### CNN(Tensor)

A diagram of a process

AI-generated content may be incorrect.

Hình 3. Sơ đồ khối phương pháp nghiên cứu bài toán CNN (Tensor)

Cũng giống như ở mục a nhưng lần này dùng Tensor và các tham số khác so với sử dụng Tensorflow

## Cấu trúc đề tài

2.1. Thực nghiệm và đánh giá

2.2 Kết luận

# PHẦN NỘI DUNG

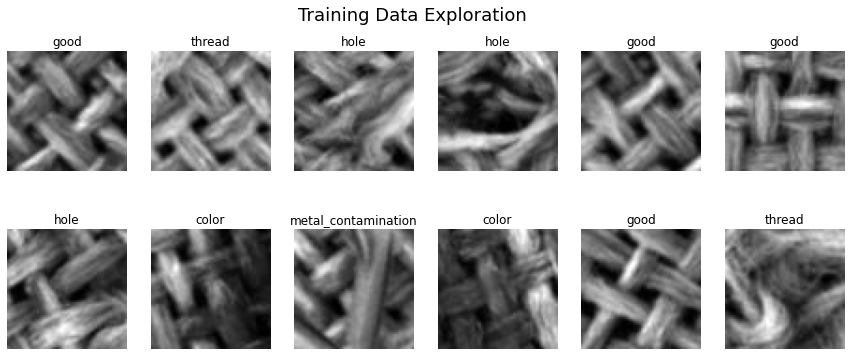
## Thực nghiệm và đánh giá

### Miêu tả bộ dữ liệu.

Bộ Bộ dữ liệu này được thiết kế cho mục đích giáo dục, tập trung vào việc phát hiện các khuyết tật hiếm gặp trong vải dệt, vốn có thể ảnh hưởng đến chất lượng của sản phẩm. Dữ liệu bao gồm các hình ảnh có kích thước 32x32 hoặc 64x64, được chia thành các lớp sau: ['tốt', 'màu sắc', 'cắt', 'lỗ', 'chỉ', 'nhiễm kim loại']. Mỗi hình ảnh được xoay ở 8 góc độ khác nhau: [0, 20, 40, 60, 80, 100, 120, 140]. Bộ dữ liệu bao gồm tập huấn luyện và tập kiểm tra, với các mảng hình ảnh được tạo ngẫu nhiên, đảm bảo rằng các hình ảnh nguồn từ hai tập này không chồng lấn nhau.

Nhưng trong dự án này chỉ thực hiện với loại ảnh 64x64 và bỏ đi lớp “màu”.

Bộ dữ liệu hỗ trợ nhiều nhiệm vụ khác nhau, bao gồm: phân loại các loại khuyết tật, phân loại góc xoay chỉ sử dụng hình ảnh "tốt" và kiểm tra trên các lớp khác, cũng như học biểu diễn kết cấu thông qua học tự giám sát. Bộ dữ liệu này được xây dựng dựa trên tập dữ liệu công khai của công ty MVTec, nhằm mục đích khám phá học tự giám sát trên hình ảnh kết cấu để giải quyết vấn đề phát hiện khuyết tật và tạo ra biểu diễn kết cấu mạnh mẽ, thay thế cho các đặc trưng xử lý hình ảnh truyền thống (như GLCM, Gabor,…)



Hình 4. Ảnh minh họa cho các loại vải

### Lấy dữ liệu từ Kaggle

Mục tiêu của nhiệm vụ là chuyển đổi dữ liệu ảnh từ định dạng file HDF5, chứa thông tin về các lớp và góc xoay của ảnh, sang định dạng JPEG. Các ảnh sau khi chuyển đổi sẽ được lưu trữ vào các thư mục riêng biệt, với mỗi thư mục tương ứng với một lớp cụ thể. Đồng thời, thông tin metadata về góc xoay của từng ảnh sẽ được lưu lại để phục vụ cho việc quản lý và sử dụng dữ liệu sau này. Đầu tiên, để đọc dữ liệu ảnh từ file HDF5, cần mở file này bằng một thư viện phù hợp như h5py trong Python. File HDF5 được giả định có cấu trúc gồm các nhóm (group), mỗi nhóm đại diện cho một lớp dữ liệu như "good", "color", "cut", v.v. Trong mỗi nhóm, dữ liệu được tổ chức theo các góc xoay (ví dụ: 0°, 20°, 40°, …), mỗi góc xoay tương ứng với một tập hợp ảnh. Quá trình duyệt bắt đầu bằng việc truy cập từng nhóm trong file HDF5. Với mỗi nhóm, tiếp tục duyệt qua các góc xoay có trong nhóm đó và nạp dữ liệu ảnh tương ứng, thường ở dạng mảng numpy, từ các dataset trong nhóm.

Tiếp theo, để xử lý dữ liệu một cách hiệu quả, một generator có tên generate\_img\_arr() sẽ được tạo. Generator này sinh ra các tuple chứa ba phần tử: mảng ảnh (img), tên lớp (class), và góc xoay (angle) cho từng ảnh. Trong quá trình này, góc xoay của mỗi ảnh được ghi nhận vào một danh sách để theo dõi, đồng thời được lưu vào một cấu trúc metadata (ví dụ: file CSV hoặc JSON) để sử dụng sau này. Metadata sẽ bao gồm thông tin như đường dẫn ảnh sau khi lưu và góc xoay tương ứng.

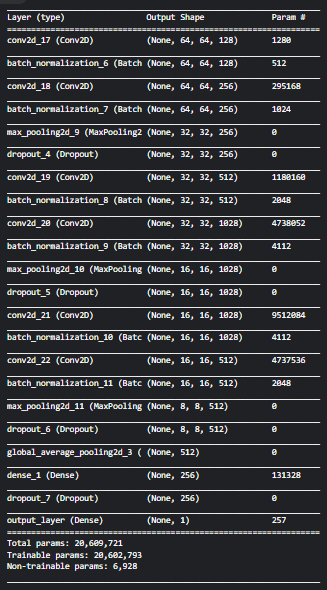
Về việc lưu ảnh, với mỗi lớp, một thư mục riêng sẽ được tạo trong một cấu trúc thư mục được xác định trước, ví dụ: ./dataset/train/good/, ./dataset/train/color/, v.v. Mỗi ảnh được chuyển đổi từ mảng dữ liệu sang định dạng JPEG và lưu vào thư mục tương ứng với lớp của nó. Tên file ảnh được đặt theo định dạng i\_angle.jpeg, trong đó i là chỉ số ảnh (bắt đầu từ 0) và angle là góc xoay (ví dụ: 0\_20.jpeg cho ảnh đầu tiên với góc xoay 20°). Quá trình lưu sử dụng các thư viện xử lý ảnh như PIL (Pillow) để đảm bảo chất lượng ảnh được duy trì.

Cuối cùng, metadata được lưu dưới dạng một file riêng, chẳng hạn như CSV hoặc JSON, chứa các cặp thông tin gồm đường dẫn đầy đủ của từng file ảnh (ví dụ: ./dataset/train/good/0\_20.jpeg) và góc xoay tương ứng (ví dụ: 20). File metadata này cho phép kiểm tra và truy xuất thông tin về ảnh một cách dễ dàng trong các bước xử lý dữ liệu tiếp theo, chẳng hạn như huấn luyện mô hình học máy hoặc phân tích dữ liệu.

Kết quả bộ dữ liệu 64x64 ban đầu được chia làm 2 tập train và test. Tập train bao gồm 5 lớp, mỗi lớp có 12000 bức hình. Tập test gồm 5 lớp và mỗi lớp có 6000 bức hình.

### Kết quả của bài toán theo cách hướng phát triển CNN (Tensorflow)

Cấu trúc mạng trong hình là một mô hình học sâu (deep learning model), cụ thể là một mạng nơ-ron tích chập (Convolutional Neural Network - CNN) kết hợp với các lớp fully connected (dense layers) để xử lý dữ liệu đầu vào, có khả năng là hình ảnh hoặc dữ liệu tương tự. Mạng có tổng cộng 28,609,721 tham số, trong đó 28,602,793 tham số là trainable (có thể huấn luyện) và 6,928 tham số là non-trainable (không thể huấn luyện). Kích thước đầu ra cuối cùng là (None, 1), cho thấy mạng được thiết kế để dự đoán một giá trị duy nhất, có thể là phân loại nhị phân hoặc hồi quy.



Hình 5. Cấu trúc của mạng CNN(Tensorflow)

Chi tiết cấu trúc mạng như sau:

Lớp tích chập (Convolutional Layers): Có 6 lớp conv2d (conv2d\_17, conv2d\_18, conv2d\_19, conv2d\_20, conv2d\_21, conv2d\_22) với các kích thước đầu ra lần lượt là (None, 64, 64, 128), (None, 64, 64, 256), (None, 32, 32, 512), (None, 32, 32, 1928), (None, 16, 16, 1928), và (None, 16, 16, 512).

Số tham số lần lượt là 1288, 295168, 1180166, 4738852, 9512884, và 4737536.

Lớp chuẩn hóa batch (Batch Normalization): Có 6 lớp batch\_normalization (batch\_normalization\_6 đến batch\_normalization\_11) tương ứng với các lớp conv2d, với đầu ra giống đầu vào của chúng và số tham số là 512 mỗi lớp.

Lớp Max Pooling: Có 2 lớp max\_pooling2d (max\_pooling2d\_9, max\_pooling2d\_11) giảm kích thước không gian, lần lượt từ (None, 32, 32, 256) xuống (None, 16, 16, 256) và từ (None, 16, 16, 512) xuống (None, 8, 8, 512).

Lớp dropout: Có 7 lớp dropout (dropout\_4 đến dropout\_7) được áp dụng sau các lớp tích chập, pooling, và dense để tránh overfitting, nhưng không có tham số nào (0 params).

Lớp gộp toàn cục (Global Average Pooling):Lớp global\_average\_pooling2d\_3 chuyển đổi đầu ra từ (None, 8, 8, 512) thành (None, 512), chuẩn bị cho các lớp dense.

Lớp fully connected (Dense Layers): Có 2 lớp dense (dense\_1, output\_layer) với đầu ra lần lượt là (None, 256) và (None, 1). Lớp dense\_1 có 131328 tham số, trong khi output\_layer có 257 tham số, cho thấy đây là lớp đầu ra với kích thước nhỏ.

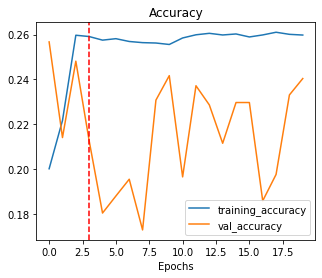
Tổng số tham số lớn (hơn 28 triệu) cho thấy mô hình rất phức tạp, phù hợp với các bài toán hình ảnh phức tạp như phân loại hình ảnh hoặc nhận diện đối tượng. Mạng sử dụng nhiều lớp tích chập sâu, kết hợp batch normalization và dropout để tăng cường khả năng tổng quát hóa, cùng với max pooling để giảm kích thước không gian và global average pooling để chuyển đổi sang không gian đặc trưng. Kích thước (None, 1) tại lớp output\_layer gợi ý đây là một bài toán phân loại nhị phân hoặc hồi quy đơn biến.

A screenshot of a computer program

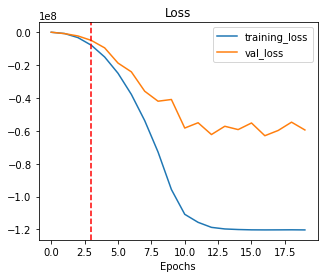
AI-generated content may be incorrect.

Hình 6. Quá trình huấn luyện của mạng CNN (Tensorflow)

Tuy nhiên do chưa nắm rõ được bộ dữ liệu nên kết quả huấn luyện mô hình cho thấy mô hình đã bị overfitting do sử dụng quá nhiều node:



Hình 7. Biểu đồ độ chính xác của mô hình CNN (Tensorflow)



Hình 8. Biểu đồ Loss của mô hình CNN (Tensorflow)

Mô hình mất 5 giờ để học nhưng không được kết quả như mong đợi, gây lãng phí tài nguyên.

### Kết quả của bài toán theo cách hướng phát triển CNN(Tensorflow) kết hợp SVM (sử dụng trích xuất HOG)

Ở lần huấn luyện mô hình lần đầu cho thấy mô hình học khá tốt cho loại Good và loại Metal nên ta sẽ huấn luyện 2 mô hình này riêng và huấn luyện các loại còn lại một mô hình khác.Đầu tiên tách các hình chia theo cặp One Vs All, good và còn lại, metal và còn lại. Sau đó sử dụng phương pháp trích xuất đặc trưng bằng HOG (Histogram of oriented gradients). Histogram of Oriented Gradients (HOG) là một kỹ thuật trích xuất đặc trưng phổ biến trong lĩnh vực xử lý ảnh và thị giác máy tính, được giới thiệu bởi Navneet Dalal và Bill Triggs vào năm 2005. HOG hoạt động bằng cách phân tích sự phân bố của các hướng gradient trong một hình ảnh, giúp mô tả hình dạng và cấu trúc của các đối tượng. Kỹ thuật này chia hình ảnh thành các ô nhỏ (cells), tính toán gradient theo hướng và độ lớn tại mỗi pixel, sau đó xây dựng biểu đồ (histogram) của các hướng gradient trong từng ô. Các biểu đồ này được chuẩn hóa theo khối (blocks) để giảm ảnh hưởng của ánh sáng và tăng tính ổn định.

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

AI-generated content may be incorrect.

Hình 9. Biều đồ Loss và Độ chính xác của mô hình CNN (Tensorflow) của Loại Good vs All

Như đã nói trước đó việc phân loại giữa loại Good và các loại còn lại khá dễ dàng. Mô hình hoạt động tốt độ chính xác tầm 94% và loss tầm 0.2.

A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 10. Bảng báo cáo của mô hình CNN (Tensorflow) Loại Good vs All

Báo cáo phân loại cho thấy mô hình đạt độ chính xác tổng thể là 89% trên tổng số 30.000 mẫu, cho thấy hiệu suất khá cao trong nhiệm vụ phân loại lỗi vải. Tuy nhiên, khi xem xét chi tiết theo từng lớp, có thể thấy sự chênh lệch đáng kể giữa hiệu quả phân loại hai lớp. Với lớp 0 (đại diện cho vải Good), mô hình chỉ đạt precision là 0.73, recall là 0.74 và f1-score là 0.73. Ngược lại, đối với lớp 1 (vải còn lại), mô hình thể hiện rất tốt với precision và recall đều đạt 0.93, kéo theo f1-score cũng đạt 0.93. Điều này phản ánh rõ sự thiên lệch trong hiệu năng mô hình, có thể bắt nguồn từ hiện tượng mất cân bằng dữ liệu khi số lượng mẫu lớp 1 (24.000) áp đảo so với lớp 0 (6.000). Trung bình cộng theo từng lớp (macro average) cho precision, recall và f1-score đều là 0.83, trong khi trung bình có trọng số (weighted average) đạt 0.89 do bị chi phối bởi lớp chiếm đa số.

A graph of a line

AI-generated content may be incorrect.

Hình 11. Biểu đồ Loss và Độ chính xác của mô hình CNN (Tensorflow) Loại Metal vs All

Kết quả của huấn luyện mô hình Metal cũng hoạt động khá tốt và dừng sớm do đã tìm được cực trị toàn cục.

A screenshot of a report

AI-generated content may be incorrect.

Hình 12. Bảng báo cáo của mô hình CNN (Tensorflow) Loại Metal vs All

Báo cáo phân loại cho mô hình phân biệt giữa "metal" và các loại lỗi khác cho thấy độ chính xác tổng thể đạt 90% trên tập dữ liệu gồm 24.000 mẫu, phản ánh khả năng phân loại khá tốt của mô hình. Khi phân tích chi tiết theo từng lớp, mô hình đạt hiệu năng khác biệt giữa hai nhóm. Đối với lớp 0 (metal) precision là 0.74 và recall đạt 0.89, dẫn đến f1-score là 0.81. Điều này cho thấy mô hình có xu hướng nhận diện tốt các mẫu thuộc lớp 0 (nhờ recall cao), nhưng precision còn thấp, nghĩa là vẫn tồn tại nhiều dự đoán sai nhầm từ lớp 1 sang lớp 0. Trong khi đó, lớp 1 (còn lại) đạt precision rất cao là 0.96, recall là 0.90 và f1-score là 0.93. Trung bình cộng theo từng lớp (macro average) đạt precision 0.85, recall 0.90 và f1-score 0.87, cho thấy mô hình vẫn giữ được hiệu suất khá đồng đều giữa hai lớp. Trung bình có trọng số (weighted average) cũng đạt các chỉ số cao và đồng nhất ở mức 0.

A close-up of several images of threads

AI-generated content may be incorrect.

Hình 13. Hính ảnh minh họa cho trích xuất HOG cho 3 loại Cut, Hole, Thread

Sau khi dùng phương pháp trích xuất HOG các đường chỉ sẽ trở thành các hướng của vector khiến mô hình dễ nhận biết hơn khi phân loại 3 loại khó như cut, hole và thread.

Sử dụng mô hình SVM với chỉ các tham số mặc định của mô hình nhưng lại cho ra kết quả khá tốt so với dự đoán.

A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 14. Bảng báo cáo của mô hình SVM

Báo cáo phân loại cho mô hình phát hiện ba loại lỗi vải gồm “cut”, “hole” và “thread” cho thấy độ chính xác tổng thể đạt mức trung bình 76% trên 7.200 mẫu, phản ánh hiệu suất tương đối ổn định. Khi đi vào chi tiết từng lớp, mô hình có khả năng phân biệt khá đều giữa ba loại lỗi. Cụ thể, lỗi “cut” có precision 0.75, recall 0.74 và f1-score là 0.74, cho thấy mức độ chính xác và khả năng bao phủ gần như tương đương. Tương tự, lỗi “hole” đạt precision 0.73 và recall 0.75, với f1-score cũng ở mức 0.74, phản ánh hiệu năng trung bình. Trong khi đó, lỗi “thread” được mô hình nhận diện tốt hơn một chút với precision và recall đều đạt 0.80, kéo theo f1-score cũng cao nhất trong ba lớp ở mức 0.80. Trung bình cộng theo từng lớp (macro average) và trung bình có trọng số (weighted average) đều bằng nhau ở mức 0.76, cho thấy mô hình hoạt động khá cân bằng giữa các lớp mà không bị ảnh hưởng đáng kể bởi sự chênh lệch về số lượng mẫu.

Việc nhìn nhận sâu hơn về bộ dữ liệu khiến em tìm ra được cách tối ưu hơn và mất ít thời gian hơn khi huấn luyện mô hình.

### Kết quả của bài toán theo cách hướng phát triển CNN(Tensor)

Sau khi nghiên cứu kĩ hơn về Deep Learning thì em đã xây dựng lại mô hình theo ý mình bằng cách dò từng tham số nhỏ như số lượng nơ ron, số lượng lớp chập. Sau nhiều lần thử với thì cũng cho ra kết quả như mong muốn. Các bước thực hiện gồm:

A diagram of a process

AI-generated content may be incorrect.

Hình 15. Sơ đồ hoạt động của Deep Learning

Với bài toán này, độ hỗn loạn của các đường chỉ và mức độ phức tạp của các loại khuyết tật trên bề mặt vải khiến việc nhận diện không thể chỉ dựa vào những đặc trưng đơn giản. Do đó, cần sử dụng một mô hình học sâu với kiến trúc đủ sâu và khả năng trích xuất đặc trưng mạnh mẽ. Cụ thể, mô hình được thiết kế gồm hai phần chính là các lớp tích chập (convolutional layers) và các lớp fully connected. Phần conv\_layers bao gồm bốn khối tích chập liên tiếp, mỗi khối gồm một lớp Conv2d với padding mở rộng để giữ lại thông tin biên, tiếp theo là MaxPool2d nhằm giảm kích thước đặc trưng và tăng tính khái quát, sau đó là ReLU giúp mô hình học phi tuyến và BatchNorm2d để ổn định quá trình huấn luyện. Bắt đầu từ ảnh đầu vào xám (1 kênh), mô hình lần lượt tăng số lượng kênh lên 8, 32, 64 và 128, đồng thời giảm dần kích thước đặc trưng từ 64×64 còn 5×5 sau 4 lần pooling.

Tiếp theo, phần fc\_layers là một chuỗi các lớp fully connected, bắt đầu bằng việc làm phẳng đầu ra từ phần tích chập (kích thước 128×5×5), đưa vào mạng neural có ba lớp ẩn với số lượng nút tăng dần (128 → 256 → 512), mỗi lớp đều có Dropout để giảm hiện tượng overfitting. Cuối cùng là một lớp Linear đưa đầu ra về số chiều bằng với số lớp cần phân loại (num\_classes). Kiến trúc này cho phép mô hình học được các đặc trưng hình học phức tạp của từng loại khuyết tật, đồng thời giữ được sự ổn định và khả năng tổng quát hóa khi áp dụng cho các mẫu ảnh chưa thấy.

Mô hình cho ra được kết quả như sau:

A graph of different colored lines

AI-generated content may be incorrect.

Hình 16. Kết quả huấn luyện của mô hình CNN(Tensor) phân loại Good vs All

Biểu đồ huấn luyện trên cho thấy sự mất cân đối rõ rệt giữa hiệu quả trên tập huấn luyện và tập kiểm tra, phản ánh vấn đề overfitting trong quá trình huấn luyện mô hình. Cụ thể, biểu đồ Training Loss (trái) thể hiện xu hướng giảm tổng thể, mặc dù có dao động giữa các epoch. Đặc biệt, loss giảm mạnh từ epoch 1 đến epoch 4, nhưng sau đó lại dao động khá nhiều và không ổn định, cho thấy mô hình không học một cách mượt mà. Ở biểu đồ Accuracy (phải), độ chính xác trên tập huấn luyện tăng đều qua các epoch, đạt gần 100%, cho thấy mô hình học rất tốt trên dữ liệu huấn luyện. Tuy nhiên, độ chính xác trên tập kiểm tra (Test Acc) lại dao động quanh mức 90–92%, không có xu hướng tăng ổn định và thậm chí có dấu hiệu giảm nhẹ ở những epoch cuối.

A graph of different colored lines

AI-generated content may be incorrect.

Hình 17. Kết quả huấn luyện của mô hình CNN(Tensor) phân loại Metal vs All

Ở biểu đồ Training Loss (bên trái), ta thấy loss có xu hướng giảm theo thời gian, mặc dù vẫn có vài dao động nhỏ, đặc biệt là ở epoch 3 và 6–7. Dù không giảm đều, nhưng tổng thể loss đã giảm từ khoảng 280 xuống còn gần 150, thể hiện rằng mô hình vẫn tiếp tục học được đặc trưng của dữ liệu huấn luyện.Biểu đồ Accuracy (bên phải) cho thấy độ chính xác trên tập huấn luyện tăng đều từ 96% ở epoch đầu lên gần 99.9% ở epoch 10, phản ánh khả năng học rất tốt của mô hình trên tập train. Tuy nhiên, độ chính xác trên tập kiểm tra (Test Acc) lại dao động mạnh và không có xu hướng tăng rõ rệt, chỉ nằm trong khoảng từ 90.8% đến 95.7%.

A graph of a graph

AI-generated content may be incorrect.

Hình 18. Kết quả huấn luyện của mô hình CNN(Tensor) phân loại Cut vs All

Dựa vào biểu đồ huấn luyện, ta có thể nhận thấy rằng giá trị loss giảm mạnh trong những epoch đầu, đặc biệt từ epoch 1 đến epoch 2, cho thấy mô hình học rất nhanh ở giai đoạn đầu. Tuy nhiên, sau đó giá trị loss dao động mạnh và không ổn định, có thời điểm tăng vọt (epoch 7) rồi lại giảm. Ở biểu đồ độ chính xác, độ chính xác trên tập huấn luyện tăng đều và đạt gần 100%, chứng tỏ mô hình học rất tốt trên dữ liệu huấn luyện. Tuy nhiên, độ chính xác trên tập kiểm tra lại dao động mạnh trong khoảng 47% đến 65%, thậm chí có thời điểm giảm sâu trong khi train accuracy vẫn rất cao, điều này cho thấy mô hình đang bị quá khớp (overfitting). Mô hình học thuộc dữ liệu huấn luyện nhưng không thể tổng quát hóa tốt trên dữ liệu mới.

Nhìn chung về mặc tổng thể các mô hình hoạt động khá tốt với các loại như Good, Metal, Cut và hạn chế ở Thread và Hole do bộ dữ liệu phức tạp.

## Kết luận

### Tổng kết kết quả đạt được

Đề tài đã thành công trong việc xây dựng một hệ thống phân loại vải dệt, với bộ dữ liệu 60000 bức ảnh train và 30000 bức ảnh test, cho ra kết quả tương đối.Với hướng phát triển đầu tiên Tensorflow: mô hình đạt độ chính xác trung bình giao động ở mức 30%. Ở hướng phát triển thứ 2: Việc thay đổi cách nhận xét về bộ dữ liệu và sử dụng thêm cách trích xuất khác dẫn đến mô hình hoạt động tốt hơn cho ra kết quả tối ưu hơn kết quả được 80%. Ở hướng phát triển cuối cùng cũng mang lại kết quả tương đối giao động trung bình ở mức 85%.

### Hạn chế và khó khăn

Mặc dù đạt được nhiều kết quả khả quan, đề tài vẫn còn tồn tại một số hạn chế. Thứ nhất, bộ dữ liệu sử dụng trong quá trình huấn luyện có số lượng lớn và được phân thành nhiều lớp, tuy nhiên các ảnh không được tổ chức sẵn theo từng thư mục tương ứng với nhãn, gây khó khăn trong quá trình tiền xử lý và phân loại. Ngoài ra, dữ liệu còn chứa nhiều nhiễu như ảnh mờ, ánh sáng không đồng đều hoặc vật thể không rõ ràng, ảnh hưởng đến khả năng trích xuất đặc trưng của mô hình. Kích thước dữ liệu lớn cũng khiến việc xử lý và huấn luyện mô hình tiêu tốn nhiều tài nguyên và thời gian, đặc biệt trong bối cảnh thiết bị phần cứng hạn chế. Thời gian huấn luyện kéo dài làm chậm tiến độ thử nghiệm và điều chỉnh mô hình, đồng thời gây khó khăn trong việc đánh giá hiệu quả của các cấu hình khác nhau trong một khoảng thời gian hợp lý. Những yếu tố này đã ảnh hưởng nhất định đến hiệu quả tổng thể và tính ứng dụng của hệ thống trong thực tế.

# TÀI LIỆU THAM KHẢO

1. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
2. R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580-587, doi: 10.1109/CVPR.2014.81.
3. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in European Conference on Computer Vision (ECCV), 2016, pp. 21–37.
4. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
5. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 2005, pp. 886-893, doi: 10.1109/CVPR.2005.177.