



Detecção e localização de defeitos em superfícies de couro natural

Y. S. Gan¹ · Sze-Teng Liong² · Danna Zheng³ · Yiyang Xia³ · Shuli Wu³ · Mengchen Lin³ · Yen-Chang Huang⁴

Recebido: 13 de janeiro de 2021 / Aceito: 8 de julho de 2021

© O(s) autor(es), sob licença exclusiva da Springer-Verlag GmbH Alemanha, parte da Springer Nature 2021

Abstrato

Defeitos que aparecem na superfície do couro podem ser resultado de variações naturais ou mau manuseio durante o processo de fabricação. A inspeção visual na fábrica é uma das etapas essenciais no processo de garantia de qualidade. Isso deve ser feito antes que os produtos acabados sejam despachados para o cliente. Até agora, a detecção dos defeitos do couro ainda é realizada manualmente, o que é trabalhoso, tedioso e pode ser passível de erro humano. Portanto, neste artigo, propomos um sistema automático de localização e detecção de defeitos de couro empregando uma série de métodos de processamento de imagens digitais baseados em aprendizado profundo. De forma sucinta, uma rede neural convolucional (CNN) é utilizada para realizar a tarefa de detecção, que é determinar a presença do defeito em um remendo de couro. Em seguida, o remendo de couro defeituoso detectado é processado para a operação de localização, que consiste em identificar as coordenadas do limite em nível de pixel. Para a tarefa de detecção, o resultado obtido usando AlexNet como descritor de recursos e SVM como classificador é de 100%. Para a etapa de localização, demonstramos que a técnica de segmentação de instâncias, Faster R-CNN supera a YOLOv2 ao obter a Intersecção sobre União (IoU) de 73%. Além disso, extensos experimentos e comparações das abordagens do estado da arte são apresentados para verificar a eficácia dos algoritmos propostos.

Palavras-chave Couro · Defeito · Detecção · Localização · R-CNN mais rápido · YOLOv2

1. Introdução

O couro é feito de couro e pele de animais, incluindo vacas, porcos, ovelhas, cabras e crocodilos. É um material natural resistente à água, flexível e durável. Devido a muitos fatores externos ou humanos inesperados, haverá algumas manchas e manchas irregulares aparecendo nas superfícies das peles dos animais. Por exemplo, os defeitos visíveis comuns são rugas, cortes, picadas de insetos, sarna e marcas em relevo deixadas

pela ferida. Assim, as peças de couro serão geralmente desnaturadas através de diversos refinamentos químicos e físicos, como as etapas de preparação, curtimento, crostas e revestimentos superficiais. No entanto, haverá novos tipos de defeitos criados pelo homem durante a fabricação do couro devido ao processamento inadequado. Este tipo de problemas contribui para diminuir a qualidade do couro e, assim, afetar o preço de venda.

O couro pode produzir grande variedade de artigos de luxo, como móveis, malas, vestuário, decoração e outros. A Figura 1 mostra o crescimento da receita global de artigos de couro de luxo de 2010 a 2019 (Statista 2019). Isso também reflete que o padrão de vida das pessoas aumentou.

As empresas de fabricação de couro incluem marcas como Louis Vuitton, Rimowa, Gucci, Coach, Michael Kors, Tumi e assim por diante.

Para acelerar a velocidade de processamento do couro na indústria, este trabalho visa focar na etapa de inspeção visual. Concretamente, é o processo de monitorar a qualidade do couro, identificando a área defeituosa na superfície. Até o momento, existem relativamente poucos trabalhos, como Liong et al. (2019a), Villar et al. (2011), Jian et al. (2010), Pistori et al. (2018), Kasi et al. (2014) foi realizado para realizar a identificação de

* Sze-Teng Liong
stliong@fcu.edu.tw

* Yen-Chang Huang
ychuang@mail.nutn.edu.tw

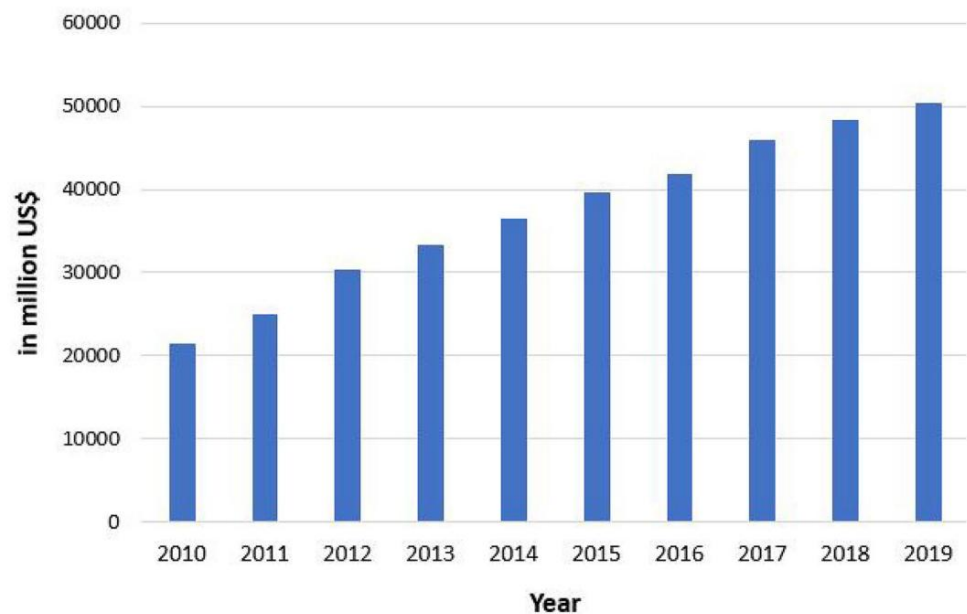
¹ Escola de Arquitetura, Universidade Feng Chia, Taichung 40724, Taiwan

² Departamento de Engenharia Eletrônica, Universidade Feng Chia, Taichung, Taiwan

³ Escola de Engenharia Elétrica e de Computação, Xiamen Universidade Malásia, Jalan Sunsuria, Sepang, Selangor, Malásia

⁴ Departamento de Matemática Aplicada, Universidade Nacional de Tainan, Tainan, Taiwan

Fig. 1 Aumento do crescimento da receita de artigos de couro de luxo de 2010 a 2019.
Fonte: Couro de luxo (Statista 2019)



defeito de couro usando técnicas de processamento automático de imagem. Hoje em dia, a maioria das indústrias ainda emprega a forma manual, que exige recursos humanos para separar a parte defeituosa das peças de couro. Esse sortimento manual é demorado, altamente subjetivo, pode ser propenso a erros e, portanto, levar a disputas durante as transações. Portanto, é fundamental digitalizar e vetorizar o couro para permitir os processos automáticos de classificação de grau e localização de defeitos/segmentação.

Há três objetivos principais neste artigo:

1. Os remendos de couro são categorizados como defeituosos ou não defeituosos com base em sua cor de superfície, empregando rede neural convolucional e classificadores populares.
2. Os limites das regiões de defeito com coordenadas XY precisas são identificados adotando-se dois tipos de redes de segmentação pixel a pixel.
3. Comparações quantitativas e qualitativas abrangentes são apresentadas para métodos de última geração. Além disso, a análise multivariada é conduzida e investigações adicionais são derivadas para avaliação de desempenho.

A organização do trabalho é a seguinte: Sec. 2 apresenta uma revisão detalhada da literatura sobre detecção e localização automática em couro.

A Seção 3 elabora o sistema de identificação de defeitos de couro proposto com descrição detalhada da implementação. O banco de dados usado e as configurações experimentais são discutidos na Seção. 4. Os resultados de desempenho estão resumidos na Seção. 5. Finalmente, Sec. 6 destaca as considerações finais.

2. Trabalho relacionado

Existem duas tarefas principais em um sistema automatizado de identificação de defeitos de couro, ou seja, detecção e localização.

A primeira classifica se o remendo de couro contém defeitos ou não.

Enquanto isso, este último localiza as posições das regiões defeituosas.

Deve-se notar que, até agora, poucos pesquisadores trabalham na localização de defeitos de couro. Esta seção discute importantes estudos existentes relacionados ao sistema automatizado de inspeção de couro.

2.1 Detecção

Um dos primeiros trabalhos que emprega rede neural na detecção de defeitos de couro é Jian et al. (2010). Eles combinam a Rede Neural Feed-forward (FNN) para selecionar atributos ideais que podem descrever de forma eficaz as características dos defeitos na superfície do couro. Em seguida, adota-se uma árvore de decisão para classificar o tipo de defeito. Os tipos de defeitos incluem furos de faca, cortes, lesões por impressão e aberrações.

A precisão geral da detecção é superior a 90%. No entanto, o número total de amostras é considerado pequeno (ou seja, 200 imagens). Assim, a evidência experimental não é suficientemente conclusiva.

Tarefa de detecção semelhante é realizada por Villar et al. (2011). Eles propõem um sistema automatizado para classificar os tipos de defeitos (ou seja, corte aberto, corte fechado e mordida aberta) no couro wet blue. Em resumo, eles utilizam Sequential Forward Selection (Jain et al. 2000) para extrair os recursos das imagens em tons de cinza e do modelo de cores. Os resultados gerais de detecção obtidos foram superiores a 95%.

Jawahar et al. (2014) classificam as características do couro com base nos descritores de características Wavelet Statistical Feature (WFS) e Wavelet Co-occurrence Feature (WCF). Em seguida, a detecção é realizada adotando-se o Support Vector Machine (SVM). Entre 700 remendos de couro, 500 estão com defeito e 200 sem defeito. O conjunto de dados é particionado na proporção de 7:3 para treinamento e teste, respectivamente. Os resultados experimentais relatados mostram que o algoritmo proposto pode alcançar alta precisão de detecção (ou seja, 99%).

Um método de inspeção de defeitos em 6 etapas é proposto por Bong et al. (2018) para categorizar o tipo de defeitos do couro (ou seja, cicatrizes, arranhões e furos). Propõe-se um sistema de captura de imagens, composto pela câmera e uma fonte de luz. Existem 2.500 imagens sendo extraídas da configuração. Entre eles, 2000 são os dados de treinamento e 500 dados de teste. Cada imagem passará primeiramente por uma série de métodos de pré-processamento, como detecção de limites, operações morfológicas e redimensionamento. O SVM é adotado como classificador para reconhecer as características extraídas, como os momentos de cor, correlogramas de cor, momentos de Zernike e textura. Como resultado, obtém-se um resultado médio de detecção de 98,8%.

Diferente do detector de couro automático convencional, Pistori et al. (2018) consideram os couros wet blue e raw hide em seu experimento. Particularmente, eles avaliam a abordagem proposta em quatro tipos de defeitos comuns: marcas de carrapato, marcas de ferro quente, cortes e sarna. Os recursos do GLCM são codificados a partir de diferentes tamanhos de entrada de remendos de couro, então são examinados em vários classificadores de aprendizado supervisionado [(SVM, Radial Basis Functions Networks (RBF) e Nearest Neighbors (KNN)]. × 40 resolução espacial atinge uma taxa de detecção perfeita de 100%.

No entanto, o conjunto de dados possui apenas um total de 16 imagens, o que pode não fornecer evidências convincentes para verificar a robustez do método proposto.

Um trabalho de classificação semelhante ao de Pistori et al. (2018) é apresentado recentemente por outro grupo de pesquisa (Jawahar et al. 2021), em que a abordagem GCLM é explorada para extrair as características dos padrões de defeitos do couro. De forma sucinta, um algoritmo de otimização Fast Convergence Particle Swarm (FCPSO) é introduzido para classificar eficazmente os remendos de couro nos grupos defeituosos ou não defeituosos.

Em seguida, são empregados diversos tipos de classificadores como rede neural, Árvore de Decisão, SVM, Naïve Bayes, KNN e Floresta Aleatória (RF). Como resultado, a tarefa de classificação binária atinge uma precisão de 88,64% quando avaliada em um total de 200 imagens. Embora a amostra de dados tenha sido aumentada em 12 vezes em relação a Pistori et al. (2018), o algoritmo pode permanecer duvidoso em uma amostra tão pequena de estudos.

Por outro lado, uma das primeiras equipes de pesquisa que adotou a rede neural no defeito automático do tecido de couro

a inspeção é conduzida por Kwak et al. (2000). Uma rede neural artificial (RNA) simples feedforward (modelos multicamadas por ceptron) com uma e duas camadas ocultas (Ventura e Chen 1996) são construídas para a classificação de defeitos. Experimentos mostram que perceptrons de duas camadas são mais promissores em comparação com os de três camadas quando testados em 140 amostras. Além disso, eles também demonstram que o método proposto superou o classificador da árvore de decisão.

Recentemente, Liong et al. (2019a) pretendem identificar as imagens de couro que contêm defeitos de picada de carrapato. De forma sucinta, a detecção de bordas Canny é empregada para melhorar a qualidade da imagem. Em seguida, a rede neural artificial (RNA) é adotada como extrator e classificador de características. Entre 2.000 imagens, o maior resultado alcançado é de 80%. Como o conjunto de dados tem um problema de desequilíbrio (ou seja, mais imagens não defeituosas), os autores testaram o algoritmo reconstruindo o conjunto de dados para formar distribuições de dados 1:1, 1:2, 1:3 para imagens defeituosas: não defeituosas. Eles mostraram que os resultados de detecção para o subconjunto de dados 1:3 são maiores em comparação aos 1:1 e 1:2.

Por outro lado, Winiarti et al. (2018) tentaram distinguir entre cinco tipos de couro, a saber, lagarto monitor, crocodilo, ovelha, cabra e pele de vaca. O descritor de característica utilizado é o método CNN pré-treinado, ou seja, AlexNet (Krizhevsky et al. 2012). Este papel é capaz de gerar uma precisão geral de detecção de 99,9%. Os resultados experimentais também mostram que a rede pré-treinada pode extrair melhor os recursos discriminativos em comparação com os recursos artesanais (ou seja, medição estatística de textura de GLCM, momentos de cor e características de coocorrência do nível de cinza). Observe que eles não realizam nenhuma detecção de defeitos de couro.

Uma análise mais abrangente sobre a eficácia de redes de aprendizagem profunda distintas para diferentes tipos de defeito de couro (ou seja, arranhões, superfícies podres, buracos e olhos de agulha) é relatada em Deng et al. (2020). Particularmente, as arquiteturas de rede neural investigadas incluem LeNet5 (LeCun et al. 1998), ResNet50 (He et al. 2016), Faster R-CNN (Ren et al. 2016), etc. ou seja, a precisão média é de 95%, o algoritmo é projetado de forma que a rede seja designada para lidar com um tipo de defeito de cada vez. Por outro lado, Aslam et al. (2020) sugerem realizar a técnica de aprendizagem por transferência após extrair os recursos dos conjuntos de dados como HAM10000 (Tschandl et al. 2018) (contém 10.000 imagens de lesões de pele pigmentadas que compõem 7 classes), CODEBRIM (Mundt et al. 2019) (contém 8.300 imagens de defeitos de concreto compostas de 5 classes) e ImageNet (Deng et al. 2009) (contém 1,2 milhão de imagens compostas de 1.000 classes) conjuntos de dados. No entanto, a maior precisão obtida na tarefa de classificação binária é de 83%. Isso pode ser atribuído ao problema de desequilíbrio de classes, pois o número total de imagens defeituosas é cerca de 2,5 vezes maior do que as não defeituosas.

2.2 Localização

Serafim (1992) utiliza a análise das pirâmides Laplacianas a partir dos modelos auto-regressivos. Ambos os resultados qualitativos de nível 0 e de nível 1 são relatados para demonstrar a localização de defeitos de rugas. Como este é um dos trabalhos anteriores (em 1992) realizando localização em couro, não está claro que o método proposto avaliou quantas imagens de couro. Além disso, não há métrica de medição para quantificar o desempenho da localização.

Semelhante ao trabalho de pesquisa realizado em Serafim (1992), Kasi et al. (2014) sugerem um algoritmo automático de detecção adaptativa de bordas para identificar os defeitos na superfície do couro. Primeiro, o couro da imagem é passado por um filtro Gaussiano para eliminar o ruído. Em seguida, um operador de Sobel é aplicado para detectar as bordas do defeito. Uma técnica de limiarização é realizada para normalizar a intensidade da imagem. No entanto, o resultado obtido por este método não foi quantificado e, portanto, pode não ser comparável a outros métodos.

Liong et al. (2019b) propõem um sistema totalmente automatizado de detecção e marcação de defeitos em couro de bezerro. Eles primeiro rotularam a verdade do solo (defeitos) manualmente em cada uma das imagens coletadas usando um braço robótico. Em seguida, as imagens anotadas são servidas como dados de entrada para treinar a arquitetura de segmentação de instância Mask R-CNN. Essa arquitetura treinada é usada para realizar a identificação de defeitos nas imagens de teste. Além disso, eles modificam o método LBP para separar alguns pontos de fronteira nas áreas defeituosas. As precisões de segmentação obtidas são 91,50% e 70,35% nos conjuntos de dados de trem e teste, respectivamente.

Com base na revisão descrita acima, pode-se resumir que a maioria das análises utilizou métodos artesanais. Neste artigo, pretendemos distinguir nosso trabalho de outros, aplicando CNNs de última geração para estabelecer um sistema completo e automatizado de detecção e localização de defeitos de couro. Até onde sabemos, esta é a primeira tentativa que compreende o modelo ponta a ponta de dois estágios.

3 Método proposto

Os processos gerais de detecção e localização são ilustrados na Fig. 2. Especificamente, o sistema de processamento de imagem proposto incorpora dois estágios: (a) detecção - para categorizar as imagens em classes defeituosas ou não defeituosas. O extrator de recursos adotado é o AlexNet. Em seguida, duas ferramentas classificadoras são empregadas, a saber, SVM e softmax; (b) localização - as imagens defeituosas detectadas são processadas posteriormente para identificar o limite da região do defeito. Duas redes eficientes de segmentação em nível de instância são utilizadas para obter a localização do defeito, a saber, YOLOv2 e Faster R-CNN. A descrição técnica detalhada do sistema é elaborada nas Seções. 3.1 e 3.2 abaixo.

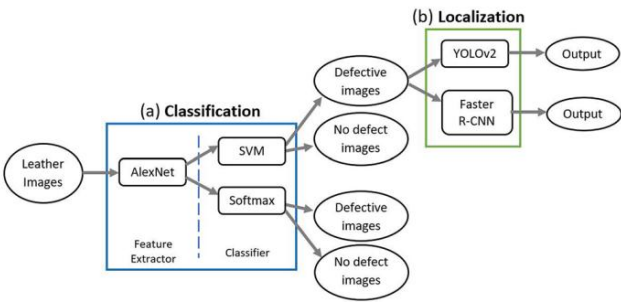


Fig. 2 Fluxograma dos processos de detecção e localização

Tabela 1 Configuração AlexNet modificada para as camadas convolucionais (Conv) , camadas de pool (Pool) , camada totalmente conectada (FC) e camada softmax de saída

Camada	Tamanho do filtro	# Filtros	Stride	Padding	Tamanho de saída
Entrada					227 × 227 × 3
Conv 1	11 × 11	3	96	[4, 4] [0, 0, 0, 0]	55 × 55 × 96
Grupo 1	3 × 3			[2, 2] [0, 0, 0, 0]	27 × 27 × 96
Conv 2	5 × 5	48	256	[1, 1] [2, 2, 2, 2]	27 × 27 × 256
Piscina 2	3 × 3			[2, 2] [0, 0, 0, 0]	13 × 13 × 256
Conv 3	3 × 3	256	384	[1, 1] [1, 1, 1, 1]	13 × 13 × 384
Conv 4	3 × 3	192	384	[1, 1] [1, 1, 1, 1]	13 × 13 × 384
Conv 5	3 × 3	192	256	[1, 1] [1, 1, 1, 1]	13 × 13 × 256
Piscina 5	3 × 3			[2, 2] [0, 0, 0, 0]	6 × 6 × 256
FC 6	-	-	-	-	1 × 1 × 4096
FC 7	-	-	-	-	1 × 1 × 4096
FC 8	-	-	-	-	1 × 1 × 4096
Softmax -	-	-	-	-	2 × 1

3.1 Detecção de defeitos

Entre muitas opções de CNN pré-treinadas, como AlexNet (Krizhevsky et al. 2012), GoogLeNet (Szegedy et al. 2015), SqueezeNet (Iandola et al. 2016), ShuffleNet (Zhang et al. 2018), DenseNet-201 (Huang et al. 2017) e VGG-16 (Simonyan e Zisserman 2014), optamos pelo AlexNet. Isso se deve à sua arquitetura simples e eficaz, que contém todos os estágios fundamentais em cascata, como camadas de convolução, camadas de pool, camadas de unidade linear recti-alimentada (ReLU) e camadas totalmente conectadas. A estrutura básica do AlexNet é mostrada na Tabela 1, que consiste em cinco camadas de convolução, três camadas de pooling e três camadas totalmente conectadas. AlexNet é um modelo de classificação que foi inicialmente treinado em mais de um milhão de imagens do conjunto de dados ImageNet (Deng et al. 2009) , que contém 1.000 categorias. A técnica de aprendizado de transferência adotada aqui é para superar o problema do regime de dados pequeno enfrentado no experimento.

Para realizar o aprendizado de transferência dessa rede pré-treinada em nosso experimento, realizamos pequenas modificações em alguns dos parâmetros, particularmente na última camada totalmente conectada. Observe que o tipo de classificador do AlexNet original é softmax. Adicionalmente, empregamos o classificador SVM para avaliar as características extraídas da rede. Portanto, são obtidos dois resultados de detecção: (a) AlexNet + softmax, e; (b) AlexNet + SVM.

O classificador softmax que usa a função de perda de entropia cruzada ção no AlexNet com regularização L2, é definida como:

$$L = \sum_{i=1}^N \sum_{j=1}^C y_i \log \frac{e^{z_{ij}}}{\sum_{j=1}^C e^{z_{ij}}} + \frac{1}{2} \sum_{k=1}^M \sum_{j=1}^M w[k]^2, \quad (1)$$

onde y_i é o rótulo one-hot das classes de entrada N , N é o número de amostras de entrada, C é o número de classes, $w[k]$ é o vetor de pesos na camada k , m é o número de camadas e z_i são os vetores de saída do modelo em i amostra definida como

$$z_i = \text{fsigmoid}(\text{fReLU}(X_m; C)), \quad (2)$$

de tal modo que

$$\text{fReLU}(x; W) = \max(0, y_n - \sum_j X_{mj}^{w_{j1}} + b_{m1}), \quad (3)$$

$$x_j^m = \text{fReLU}(X_m; \text{Varinha}) \quad (4)$$

$$X_j^1 = y_{w0} + b0_j. \quad (5)$$

O classificador SVM para AlexNet com regularização L2 é moldado para:

$$L = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N [\max(0, \text{fsigmoid}(\text{fReLU}(X_m; W)) - y_j) + y_j] + \frac{1}{2} \sum_{k=1}^M \sum_{j=1}^M w[k]^2, \quad (6)$$

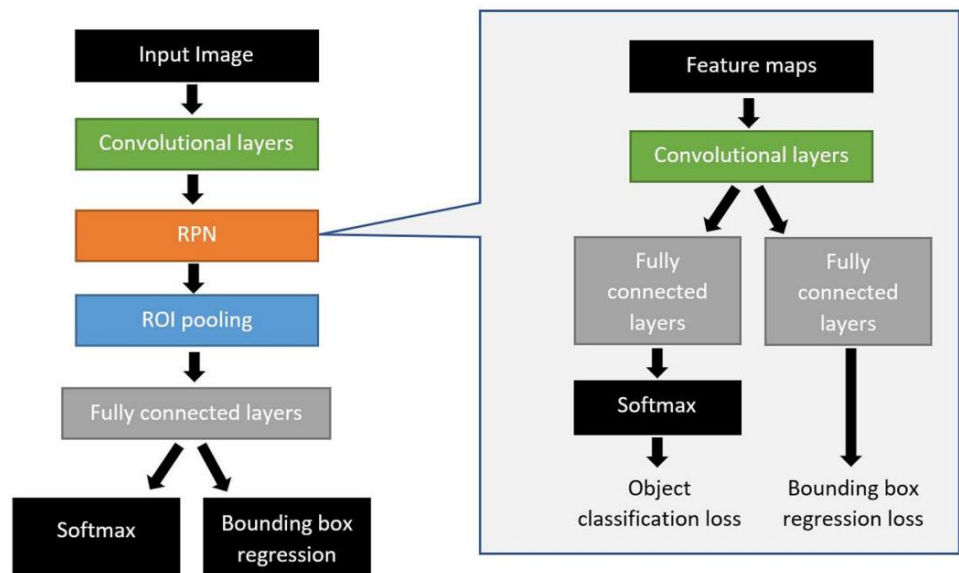
onde X_m é semelhante ao da Eq. (4).

3.2 Localização de defeitos

De acordo com o sistema projetado ilustrado na Fig. 2, as imagens defeituosas serão posteriormente processadas para localizar a região defeituosa. Os métodos envolvidos neste estágio de localização são Faster R-CNN e YOLOv2:

1. Faster R-CNN É a combinação de Fast R-CNN (Girshick 2015) e Region Proposta de Rede (RPN) (Ren et al. 2015). A estrutura resumida do Faster R-CNN é mostrada na Fig. 3. Basicamente, o Faster R-CNN usa camadas de convolução para extração de recursos e redução de redundância, então calcula boas caixas âncoras candidatas usando RPN. Esses recursos com caixas de ancoragem de tamanhos diferentes são passados para a camada de pool de ROI para criar mapas de recursos de propostas com tamanhos iguais. O RPN tem duas tarefas principais a serem executadas: regressão de caixa delimitadora e detecção de objetos. A regressão da caixa delimitadora ajusta as coordenadas das caixas de ancoragem enquanto a supressão não máxima (NMS) opera nas caixas de ancoragem de acordo com as pontuações de detecção de objetos para resolver para caixas precisas e precisas. O pseudocódigo do algoritmo NMS está listado no Algoritmo 1. Empregamos

Fig. 3 Estrutura do R-CNN mais rápido



ResNet50 (He et al. 2016) como a rede básica, pois seus subconjuntos de módulos residuais permitem um refinamento iterativo eficaz por meio de treinamento por meio de redes mais profundas. Em resumo, Faster R-CNN compreende duas redes que treinam em duas fases individualmente, ou seja, RPN e Object Detection Network (ODN). A função de perda do RPN é definida da seguinte forma:

$$L_{loc}(p_i, t_i) = \frac{1}{N_{class}} \sum_{i=1}^N L_{class}(p_i, p_{ij}^*) + \sum_{i=1}^N \frac{1}{N_{reg}} \sum_{j=1}^4 L_{reg}(t_i, t_{ij}^*), \quad (7)$$

onde p_i é a probabilidade prevista de ser um objeto para a âncora i . t_i refere-se às coordenadas da caixa delimitadora prevista para a âncora i . L_{cls} é a perda de log. p_{ij}^* é o rótulo de verdade, L_{reg} é a perda suave de L1 e t_{ij}^* é as coordenadas da caixa de verdade. Além disso, a regressão da caixa delimitadora explora a perda L1 na posição (x, y) do canto superior esquerdo, o logaritmo da altura e da largura da caixa. Em suma, a perda é dada da seguinte forma:

$$L_{RPN}(p, ct, bp, bt) = L_{cls}(p, ct) + \sum_{i=1}^N L_{loc}(bp, bt) \quad (8)$$

onde p as pontuações de classe previstas, ct a base da pontuação da classe, bp são as coordenadas da caixa previstas e bt são as coordenadas da caixa de verdade. Um algoritmo de treinamento alternado de 4 etapas é aplicado da seguinte forma:

- A rede é adotada para inicializar pesos e treinar o RPN;
- R-CNN mais rápido é treinado de acordo com a região proposta do passo (a) e os pesos são atualizados;
- RPN é treinado;
- R-CNN mais rápido é treinado de acordo com a região proposta do passo (c).

- YOLOv2 É um modelo aprimorado do YOLOv1, com menor minimização de erros de localização e maior precisão média. É o detector de objetos de última geração que foi realizado nas bases de dados PASCAL VOC (Everingham et al. 2010) e COCO (Lin et al. 2014). YOLOv2 pode se adaptar facilmente a diferentes entradas

tamanho da imagem e levam a bons resultados pelo treinamento multi-escala. O YOLOv2 supera o YOLOv1 porque a resolução espacial na rede de detecção é modificada de 224 × 224 para 448 × 448. Além disso, uma operação de agrupamento de média global é empregada para melhor previsão. Além disso, o YOLOv2 usa o método de agrupamento k-means para produzir caixas de ancoragem com maior resolução e caixas delimitadoras menores, calculando a função de distância:

$$d(caixa, centróide) = 1 - IoU(caixa, centróide). \quad (9)$$

A localização da previsão da caixa de ancoragem é implementada para produzir uma melhor localização de previsão tu . O localizador de caixa delimitadora para YOLOv2 é treinado usando a perda de localização definida na seguinte equação:

$$L_{loc} = \sum_{i=0}^{S_y-1} \sum_{j=0}^{S_x-1} \left(\sum_{b=1}^B \left(\frac{1}{w_i} \left(x_i - \hat{x}_i \right)^2 + \frac{1}{h_i} \left(y_i - \hat{y}_i \right)^2 \right) + \sum_{b=1}^B \left(\frac{1}{w_i} \left(x_i - \hat{x}_i \right)^2 + \frac{1}{h_i} \left(y_i - \hat{y}_i \right)^2 \right) \right) \quad (10)$$

onde \hat{y}_{obj} é a caixa delimitadora com a aparência u_i do objeto na célula i . (x_i, y_i, w_i, h_i) denotam as coordenadas previstas do canto superior esquerdo, largura e altura da caixa delimitadora. Em suma, propomos uma solução para solucionar a localização de defeito de couro. Usando o AlexNet + softmax, a localização prevista é expressa

Como:

$$local = \{ \text{tanh}(\frac{z_i}{\sigma}) \mid z_i = \frac{z_i - \mu}{\sigma} \} = 0 \quad (11)$$

onde $z_i = \text{fsigmoid}(\text{fReLU}(x_m, i, W))$ e tu são as caixas delimitadoras previstas por YOLOv2/ Faster R-CNN Por outro lado, usando AlexNet + SVM (Tabela 2), a localização prevista é formulada como:

$$local = \{ \text{tanh}(\frac{z_i}{\sigma}) \mid z_i = \frac{z_i - \mu}{\sigma} \} = 1 \quad (12)$$

Tabela 2 O YOLOv2 básico
Rede

Operação de camada		Filtro/tamanho do pool	Número de filtros	Canal/elemento de estofamento de passada			Tamanho da saída da âncora
Entrada -		-	-	-	-	-	128x128x3
1	Convolução 1 3x3x3	-	16	[1 1]	[1 1 1 1] -	-	128x128x16
	Normalização 1 -	-	-	-	-	16	128x128x16
	ReLU 1	-	-	-	-	-	128x128x16
	Agrupamento 1	2x2	-	[2 2]	[0 0 0 0] - [1 1	-	64x64x16
2	Convolução 2 3x3x16	-	32	[1 1]	1 1] -	-	64x64x32
	Normalização 2 -	-	-	-	-	32	64x64x32
	ReLU 2	-	-	-	-	-	64x64x32
	Agrupamento 2	2x2	-	[2 2]	[0 0 0 0] -	-	32x32x32
3	Convolução 3 3x3x32	-	64	[1 1]	[1 1 1 1] -	-	32x32x64
	Normalização 3 -	-	-	-	-	64	32x32x64
	ReLU 3	-	-	-	-	-	32x32x64
	Agrupamento 3	2x2	-	[2 2]	[0 0 0 0] -	-	16x16x64
4	Convolução 4 3x3x64	-	128	[1 1]	[1 1 1 1] -	-	16x16x128
	Normalização 4 -	-	-	-	-	128	16x16x128
	ReLU 4	-	-	-	-	-	16x16x128
	Convolução 5 3x3x128	-	128	[1 1]	Igual -	-	16x16x128
5	Normalização 5 -	-	-	-	-	128	16x16x128
	ReLU 5	-	-	-	-	-	16x16x128
	Convolução 6 3x3x128	-	128	[1 1]	Igual -	-	16x16x128
	Normalização 6 -	-	-	-	-	128	16x16x128
6	ReLU 6	-	-	-	-	-	16x16x128
	Convolução 7 1x1x128	-	24	[1 1]	[0 0 0 0] -	-	16x16x24
	Transformar	-	-	-	-	-	4
	Resultado	-	-	-	-	-	4

Algorithm 1 Non-Maximal Suppression (NMS)

```
1:  $B \leftarrow$  List of detection boxes
2:  $S \leftarrow$  Score of boxes
3:  $t \leftarrow$  Threshold
4:  $D \leftarrow \{\}$ 
5: repeat
6:   Find the detection box  $B^{(i)}$  in  $B$  with the largest score  $S^{(i)}$ 
7:   Assign  $B^{(i)}$  to  $D$ 
8:   repeat
9:     IF  $\text{IoU}(B^i, B^j)$  is greater than  $t$ 
10:      Remove  $B^{(j)}$  from  $B$ 
11:      Remove  $S^{(i)}$  from  $S$ 
12:   END
13: until All elements in  $B$  have been traversed
14: until  $B$  is empty
```

4 Configuração do experimento

4.1 Banco de dados

Nosso banco de dados contém 560 imagens de couro com resolução espacial de 140 x 140 x 3. Entre elas, 280 imagens

têm defeitos perceptíveis de corte aberto na superfície, enquanto 280 imagens não têm nenhum defeito. Para a tarefa de detecção (ou seja, para distinguir entre imagens defeituosas e não defeituosas), três experimentos foram realizados com a razão de dados de trem para dados de teste sendo 0,4, 0,6 e 0,8 respectivamente. Os detalhes do banco de dados são mostrados na Tabela 3.

Tabela 3 Configuração do banco de dados para tarefas de detecção e localização

Experimental	Conjunto de trem		Conjunto de teste		Total
	Defeito	Sem defeito	Defeito	Sem defeito	
Detecção—[0,4 trem/divisão de teste]	112	112	168	168	560
Detecção—[0,6 treino/divisão de teste]	168	168	112	112	560
Detecção—[0,8 treino/divisão de teste]	224	224	56	56	560
Localização	200	0	80	80	360

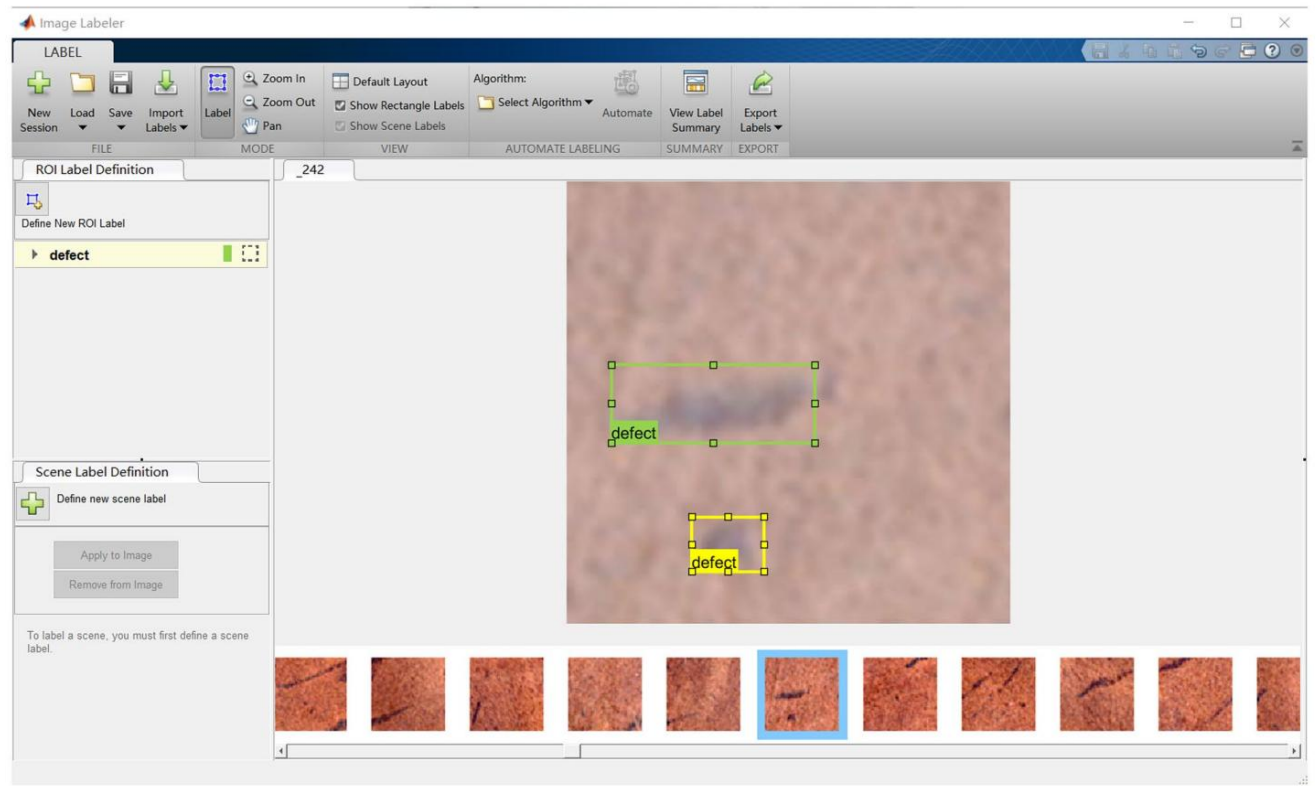


Fig. 4 Rotulador de imagem para anotar a verdade fundamental das regiões de defeito

Fig. 5 Amostra do conjunto de dados: sem defeito ; b defeituoso; c defeituoso com rótulos retangulares de região de interesse (ROI)

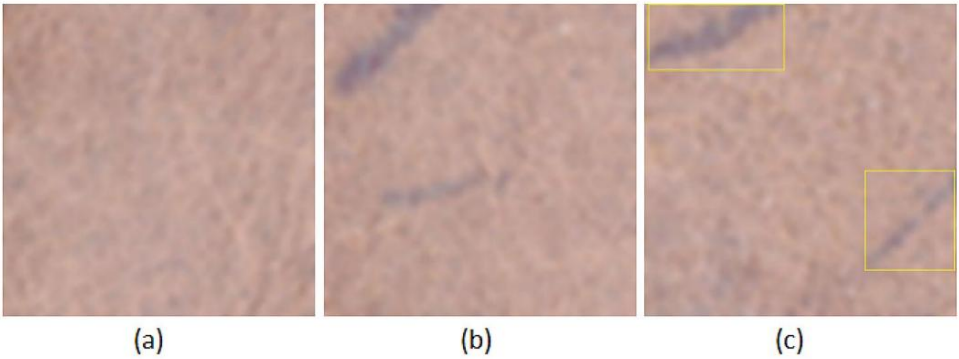


Tabela 4 Opções de treinamento dos modelos YOLOv2 e Faster R-CNN

	YOLOv2	R-CNN mais rápido
Solucionador	Adão	Adão
Taxa de aprendizado inicial	0,0005	0,0001
Tamanho do MiniLote	[2, 4, 8, 16, 32]	4
MaxEpochs	[30, 50, 100, 150]	15
Shufe	Cada época	Cada época

Quanto à localização (ou seja, para identificar as regiões defeituosas), 200 imagens defeituosas servem como dados de trem, enquanto as restantes 80 imagens defeituosas e 80 imagens não defeituosas são tratadas como dados de teste. Os detalhes do banco de dados estão descritos na Tabela 3. Antes do estágio de localização, é necessário que a verdade das regiões de defeito seja rotulada manualmente para treinar o detector de objetos. Optamos pela ferramenta de detecção de objetos integrada do MATLAB, Image Labeler. A captura de tela do aplicativo Image Labeler com rótulos retangulares de região de interesse (ROI) marcados é ilustrada na Fig. 4. Este aplicativo permite ao usuário anotar os rótulos ROI, rótulos de polilinha ROI, rótulos de ROI de pixel e rótulos de cena em uma imagem. Em nosso experimento, um rótulo retangular é escolhido como ROI. As etapas para operar este aplicativo são descritas a seguir:

1. Uma imagem é carregada do servidor local.
2. Um novo rótulo de ROI é criado e definido como “defeito”.
3. Uma caixa delimitadora retangular é desenhada ao redor de cada área defeituosa e é considerada como ROI.
4. Os assuntos marcados são salvos como objetos de verdade e usados para treinar as arquiteturas YOLOv2 e Faster R-CNN.

A Figura 5 mostra as amostras de imagens com defeito, imagens sem defeito e imagens rotuladas.

O conjunto de dados é obtido usando um braço robótico DRV70L da Delta equipado com uma câmera Canon 77D equipada com uma lente de distância focal de 135 mm. O braço do robô se move nas direções vertical e horizontal para capturar toda a superfície do couro. Além disso, a luz LED DOF D1296 Ultra High Power é usada para reduzir a oscilação da iluminação e qualquer interferência ambiental.

4.2 Configuração do experimento

Os experimentos são conduzidos usando o MATLAB 2019a em um processador Intel Core i7-6560U 2,20 GHz, RAM 8GB. Todas as imagens de entrada são primeiramente redimensionadas para 227 × 227 × 3 a partir da resolução espacial original de 140 × 140 × 3. Isso ocorre porque na classificação, o AlexNet foi treinado com valores RGB com os dados de treinamento tendo o tamanho de entrada de 227 × 227 × 3. Por outro lado, para a localização

Como não há restrição no tamanho de entrada, as imagens de entrada do YOLOv2 e do Faster R-CNN são redimensionadas para o tamanho original de 140 × 140 × 3. Isso também pode reduzir a complexidade computacional e, portanto, melhorar a execução Rapidez. As opções de treinamento para os modelos YOLOv2 e Faster R-CNN estão listadas na Tabela 4.

4.3 Métricas de desempenho

Os resultados experimentais relatam diferentes tipos de métricas de avaliação para tarefas de detecção e localização de defeitos de couro. Eles são precisão, pontuação F1 e Intersecção sobre União (IoU). As métricas para avaliar o desempenho da detecção são precisão e pontuação F1. Conceitualmente, acurácia é o número de previsões corretas em todo o conjunto de teste. O F1-score indica a média harmônica de precisão (ou seja, a taxa de amostras defeituosas corretamente previstas em todas as amostras previstas como defeitos) e recall (ou seja, a taxa de amostras defeituosas corretamente previstas em todas as amostras defeituosas reais).

Precisão = $\frac{TP + TN}{TP + FN + TN + PF};$ (13)

F1-Pontuação = $2 \times \frac{Precisão \times Recordação}{Precisão + Recall},$ (14)

por

Precisão = $\frac{TP}{TP + FP},$ (15)

e

Recuperar = $\frac{TP}{TP + FN},$ (16)

onde TP, TN, FN e FP são o verdadeiro positivo, verdadeiro negativo, falso negativo e falso positivo, respectivamente. Esses quatro termos são definidos da seguinte forma:

- TP: O número de imagens de couro com defeito que o modelo prevê corretamente.

Tabela 5 Comparação da precisão de detecção [Acc (%)] e F1-score [(F1 (%))] usando AlexNet + softmax e AlexNet + SVM

Divisão de treino/teste	AlexNet + Softmax		AlexNet + SVM	
	Conta	F1	Conta	F1
0,4	97,02	97,08	100	100
0,6	98,66	98,68	99,55	99,56
0,8	97,32	97,33	100	100

Os valores em negrito representam a pontuação mais alta

- TN: O número de imagens de couro sem defeito que o modelo prevê corretamente.
- FP: O número de imagens de couro que o modelo prevê que existe um defeito existe, mas na verdade não existe.
- FN: O número de imagens de couro que o modelo classifica erroneamente como couro sem defeito, quando na verdade há um defeito.

Por outro lado, para a tarefa de localização, a métrica utilizada é IoU. Indica a razão da sobreposição da verdade do terreno com o resultado da detecção. Os defeitos segmentados são as caixas delimitadoras de saída. Praticamente, pode existir mais de uma caixa delimitadora para cada imagem de couro, pois uma imagem pode ter várias áreas defeituosas (como mostrado na Figura 5b). Para cada imagem, todas as taxas de sobreposição serão computadas. Em seguida, a IoU média para todo o conjunto de teste é relatada como o resultado final. A IoU é expressa como:

IoU =
$$\frac{\text{Resultado da Detecção} \cap \text{Ground Truth}}{\text{Resultado da Detecção} \cup \text{Ground Truth}}$$

(17)

5 Resultados e discussão

Esta seção apresenta os resultados experimentais para detecção e localização com análise e discussão detalhadas.

5.1 Desempenho de detecção

O resultado da detecção adotando-se AlexNet + softmax e AlexNet + SVM com divisão diferente de trem/teste está tabulado na Tabela 5. Pode-se observar que tanto a precisão de detecção quanto o F1-score chegam a 100%, quando a divisão trem/teste são de 0,4 e

Fig. 6 Exemplo dos mapas de recursos de ativação das camadas de convolução AlexNet. **uma** imagem Original; **b** "conv1"; **c** "conv2"; **d** "conv3"; **e** "conv4"; **e**; **f** "conv5"

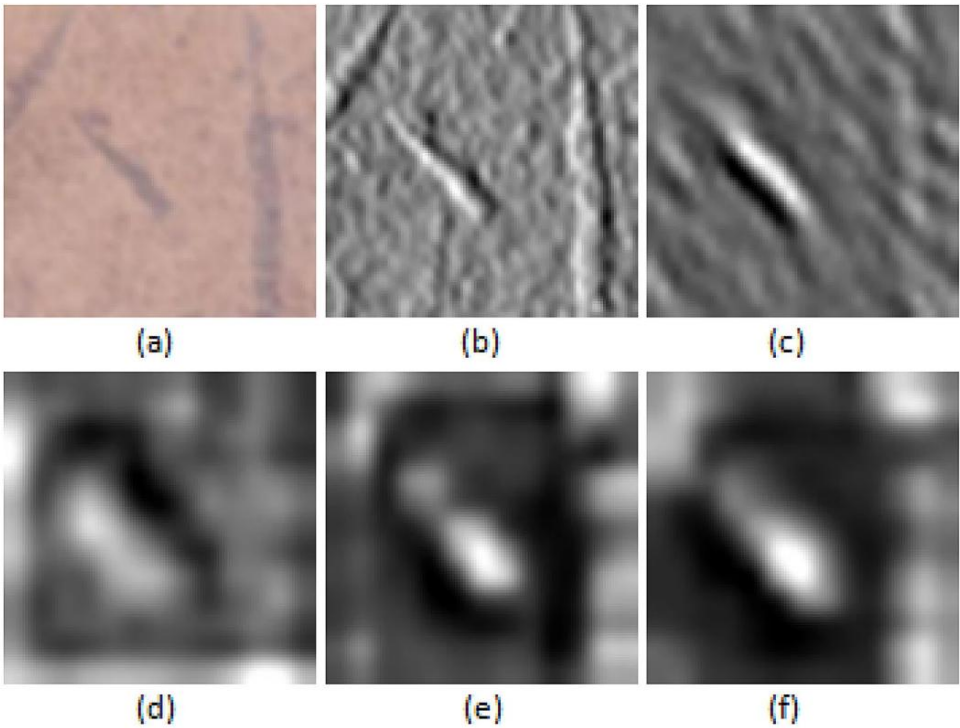
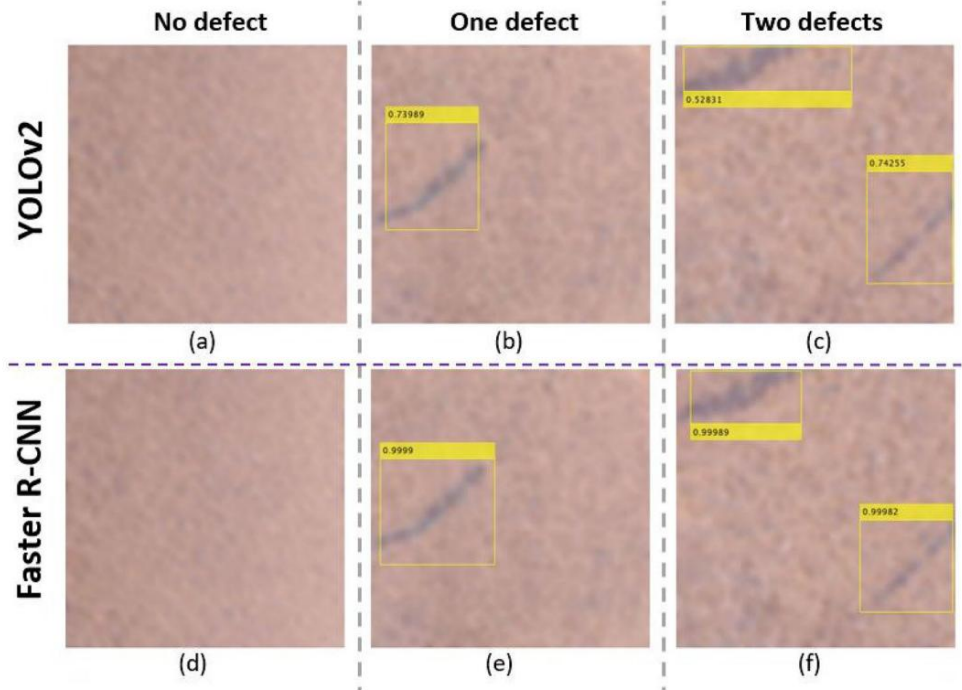


Tabela 6 Comparação da precisão de detecção com o estado da arte

Não.	Tarefa de detecção	Método	Nº da amostra	Acc (%)
1	Defeito x nenhum defeito	Método proposto	560	100
2	Defeito x nenhum defeito	ANN (Liong et al. 2019a)	2378	82,49
3	Defeito vs nenhum defeito	(WSF+WCF) + SVM (Jawahar et al. 2014)	700	99,75
4	Tipo Multi-Defeito	SFS + SVM (Villar et al. 2011)	1769	96,48
5	Tipo Multi-Defeito	GLCM + KNN (Pistori et al. 2018)	16	100
6	Tipo Multi-Defeito	FNN + DT (Jian et al. 2010)	200	94
7	Tipo Multi-Defeito	Características estatísticas + SVM (Bong et al. 2018)	2500	98,8
8	Leather (GLCM+CNN) + SVM (Winiarti et al. 2018)		4157	99,97

Os valores em negrito representam a pontuação mais alta

Fig. 7 Comparação dos resultados de a–c YOLOv2 e d–f R-CNN mais rápido



0.8 para o protocolo AlexNet + SVM. Notavelmente, AlexNet + SVM obtém resultados promissores quando o tamanho do conjunto de dados de treinamento é pequeno (ou seja, 40% dos dados são o conjunto de treinamento). Além disso, todos os resultados do AlexNet + SVM têm um desempenho melhor do que o do AlexNet + softmax. No entanto, AlexNet + softmax pode alcançar resultados significativamente promissores (ou seja, 98,66%), na divisão de 0,6 treino/teste.

Para visualizar os atributos aprendidos pelo AlexNet, investigamos os mapas de recursos de ativação após passar por cada camada de convolução. A Fig. 6 mostra o exemplo dos canais de saída da primeira para a quinta camada de convolução. Pode-se observar que existem linhas/limites óbvios detectados em “conv1” (ou seja, Fig. 6b),

Tabela 7 Resultado da avaliação de desempenho usando YOLOv2 (%): IoU médio (*AvgIoU*), precisão (*Acc*), recall (*Rec*), precisão (*Pré*) e pontuação F1 (*F1*)

Mini tamanho de lote Epoch		Média de IoU	Conta	gravando	Pré	F1
2	30	65,85	97,50	97,50	97,50	97,50
	50	64,65	97,50	96,25	98,72	97,47
	100	61,61	93,75	91,25	96,05	93,59
	150	59,64	91,88	87,50	95,89	91,50
4	30	60,49	95,00	98,75	91,86	95,18
	50	64,77	95,63	93,75	97,40	95,54
	100	58,11	90,00	93,75	87,21	90,36
	150	53,75	89,38	86,25	92,00	89,03
8	30	64,69	98,13	100	96,39	98,16
	50	69,24	96,88	98,75	95,18	96,93
	100	59,88	91,25	93,75	89,29	91,46
	150	51,14	86,25	81,25	90,28	85,53
16	30	58,13	92,50	93,75	91,46	92,59
	50	68,11	93,75	96,25	91,67	93,90
	100	64,65	91,36	92,68	90,48	91,57
	150	64,65	89,82	92,86	87,64	90,17
32	30	27,94	75,00	51,25	97,62	67,21
	50	60,86	93,75	93,75	93,75	93,75
	100	64,65	92,02	92,77	91,67	92,22
	150	51,43	89,38	83,75	94,37	88,74

Os valores em negrito representam a pontuação mais alta

Tabela 8 Matrizes de confusão usando o método YOLOv2 (tamanho do mini-lote = 8, época = 30)

		Previsto	
		Defeituoso	Não defeituoso
Real	Defeituoso	80	0
	Não defeituoso	3	77

Os valores em negrito representam a pontuação mais alta

Tabela 9 Melhor resultado de desempenho de Faster R-CNN vs melhores resultados de desempenho de YOLOv2 (%)

Método	Média de contas de IoU	gravando	Pré	F1	
R-CNN mais rápido	73,47	96,88	100	94,12	96,97
YOLOv2-1	69,24	96,88	98,75	95,18	96,93
YOLOv2-2	64,69	98,13	100	96,39	98,16

Os valores em negrito representam a pontuação mais alta

Tabela 10 Matrizes de confusão usando o método Faster R-CNN

		Previsto	
		Defeituoso	Não defeituoso
Real	Defeituoso	80	0
	Não defeituoso	5	75

Os valores em negrito representam a pontuação mais alta

indicando as áreas defeituosas. No entanto, também existe muito ruído (pixels brancos) que não correspondem às características do defeito. Após a quinta operação de convolução (ie, Fig. 6f), o ruído é eliminado, deixando as linhas de defeito perceptíveis.

A Tabela 6 demonstra ainda que nossa solução proposta é um método eficiente na classificação do tipo de defeitos do couro. Resumimos alguns dos notáveis trabalhos anteriores relacionados à tarefa de detecção de couro, tais como: (a) diferenciar entre imagens defeituosas e não defeituosas; (b) categorizar os tipos de defeitos, e; (c) reconhecer o tipo de couro. Há um trabalho que pode apresentar taxa de detecção perfeita (ou seja, # 5), mas o tamanho da amostra é de apenas 16. No entanto, é difícil tirar uma conclusão das abordagens metódicas, pois os protocolos experimentais, configurações de parâmetros e banco de dados usados são diferente.

Tabela 11 Tempo gasto para o treinamento e teste do modelo para a tarefa de classificação em diferentes configurações de divisão de trem/teste

Experimental	Estágio de treinamento		Estágio de teste	
	# amostra de dados	Tempo(s) gasto(s)	# amostra de dados	Tempo(s) gasto(s)
Detecção—[0,4 trem/divisão de teste]	224	10,44	336	0,57
Detecção—[0,6 treino/divisão de teste]	336	17,63	224	0,46
Detecção—[0,8 treino/divisão de teste]	448	24,75	112	0,34

5.2 Desempenho de localização usando YOLOv2

A Tabela 7 apresenta o desempenho de segmentação para YOLOv2, com diferentes tamanhos de mini-lotes (2, 4, 8, 16 e 32) e valores de épocas (30, 50, 100 e 150). A melhor média de IoU alcançada é de 69,25%, quando tamanho do mini-lote = 8 e época = 50. De acordo com os resultados da segmentação, é capaz de reconhecer se a amostra tem defeito ou não. Assim, pode ser considerado como um tipo de detecção. As métricas de desempenho, como precisão e pontuação F1, também são relatadas na Tabela 7. A maior acurácia alcançada é de 98,13%, quando tamanho do minilote = 8 e epoch = 30. Vale destacar que o recall chega a 100% (todas as imagens de couro com defeito são identificadas corretamente), e sua matriz de confusão é apresentada na Tabela 8. A saída de amostra usando YOLOv2 é mostrada na Fig. 7a–c. Indica que as regiões defeituosas foram identificadas com sucesso com caixas de janela retangulares.

5.3 Desempenho de localização usando Faster R-CNN

A Tabela 9 mostra o desempenho da localização usando Faster R-CNN. Além disso, os melhores resultados obtidos do YOLOv2 são comparados. Observe que os resultados mostrados na Tabela 9 são obtidos empregando diretamente a tarefa de localização, sem realizar a etapa de detecção antecipadamente. Referimo-nos às configurações experimentais de (tamanho do mini-lote, época) = (8, 50) como YOLOv2-1 e (tamanho do minilote, época) = (8, 50) como YOLOv2-2. Pode-se ver que o Faster R-CNN supera ambas as configurações YOLOv2. R-CNN mais rápido obtém um IoU médio de 73,47%, que tem + 4% em relação ao YOLOv2-1. A precisão alcançada pelo Faster R-CNN é de 96,88%, 1,25% inferior à do YOLOv2-2. Deve-se notar também que o Faster R-CNN sempre opera em uma velocidade mais rápida. A Tabela 10 mostra a matriz de confusão para o Faster R-CNN, que denota uma taxa de erro de 3% (5 imagens sem defeito são mal avaliadas).

5.4 Progresso do treinamento e tempo de computação

Como existem três tipos de configurações de divisão de treinamento/teste nos experimentos, a duração necessária para as etapas de treinamento e teste em cada configuração é ligeiramente diferente. Concretamente, o tempo exato gasto para o treinamento e teste do modelo é fornecido na Tabela 11. Observa-se que quanto maiores os dados de treinamento, menor

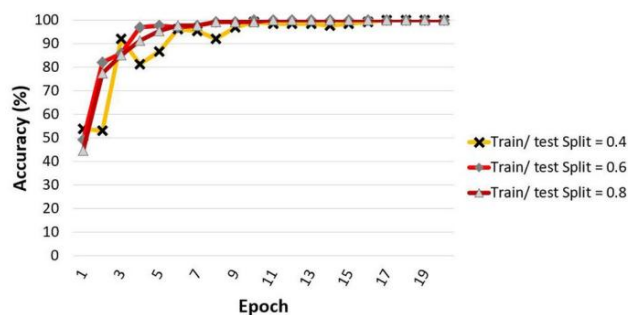


Fig. 8 O progresso do treinamento do modelo que retrata a precisão da classificação dos dados de treinamento em cada época

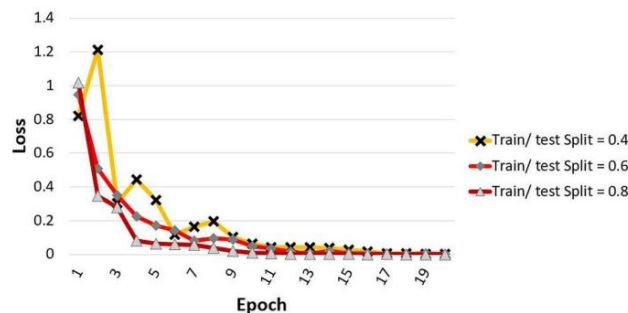


Fig. 9 O progresso do treinamento do modelo que retrata a perda de classificação dos dados de treinamento em cada época

Tabela 12 Desempenho (%) do sistema completo: detecção (AlexNet + SVM) + localização

Método	Média de contas de IoU	gravando	Pré	F1
R-CNN mais rápido	73,47	100	100	100
YOLOv2-1	69,24	100	100	100
YOLOv2-2	64,69	98,75	98,75	100
				99,37

Os valores em negrito representam a pontuação mais alta

os dados de teste. Assim, para a divisão trem/teste de 0,8, que possui o maior número de dados amostrais (ou seja, 448 imagens), o tempo gasto para o treinamento do modelo é o mais longo (ou seja, 25s). A duração do treinamento necessário reduz quando o número de dados de treinamento é menor, o que requer 10s para concluir o treinamento do modelo para a divisão de treinamento/teste de 0,4 que contém apenas 224 imagens. Este fenômeno parece ser o mesmo na fase de teste. No entanto, o processo de treinamento é sempre demorado porque o modelo ajusta repetidamente os pesos da rede até que o erro de treinamento seja minimizado. Em contraste, o processo de teste executa uma única execução pela rede treinada. Assim, o tempo gasto para a etapa de teste é significativamente menor (< 1s em todos os casos).

A Tabela 5 fornece a precisão da detecção de defeitos em três tipos de divisões de treinamento/teste (ou seja, 0,4, 0,6 e 0,8). Pode-se ver que as precisões de detecção obtidas são semelhantes (ou seja, 97,02% ~ 98,66%). Isso implica que a rede treinada acomoda uma amostra de dados relativamente pequena, mas produz alta precisão de detecção. As Figuras 8 e 9 retratam o progresso do treinamento da amostra dos três cenários em termos de precisão e perda, respectivamente. Observe que, todos os casos ilustram que as métricas de precisão e perda permanecem estagnadas após a 10ª época.

5.5 Desempenho completo do sistema

O desempenho geral para o sistema de localização e detecção de defeitos de couro de ponta a ponta proposto é tabulado na Tabela 12.

O fluxograma é mostrado na Fig. 2. AlexNet + SVM gera 100% de pontuação para as métricas de desempenho de exatidão, recall, precisão e F1-score. Então, para identificar as regiões defeituosas, a média de IoU obtida é de 73,47%, resultante do detector de objetos Faster R-CNN.

Para treinar um modelo de aprendizado profundo que seja otimamente adequado à nossa aplicação, o método de visualização Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al. 2017) é implementado na rede projetada.

É usado para examinar os pesos das ativações aprendidas

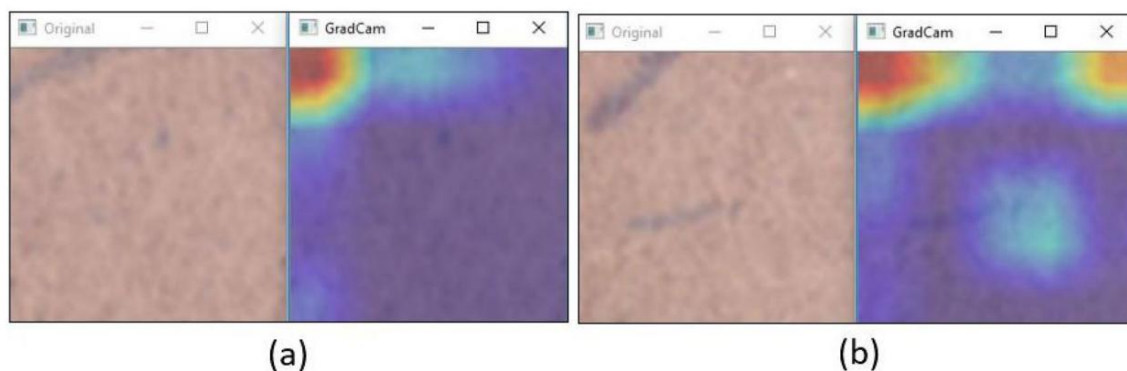


Fig. 10 Exemplo do couro original (esquerda) e saída do Grad-CAM (direita) para imagem de couro que possui **um** único defeito e **b** dois defeitos

em toda a rede. O mapa de localização discriminativo de classes Grad-CAM é o resultado da combinação linear de pesos importantes dos mapas de características A_c , e é denotado

Como:

$$L_{Grad-CAM} = \sum_c u_c v_c f(\sum_k w_{ck} A_k), \quad (18)$$

onde u_c e v_c são o peso e a altura para cada classe c , respectivamente. A função f é a função de ativação para rede. O $\sum_k w_{ck} A_k$ da classe c em k é o mapa de características é calculado usando a equação:

$$w_{ck} = \frac{\sum_{i,j} \bar{y}_{ij} A_{ij}^c}{\sum_{i,j} \bar{y}_{ij} A_{ij}^k}, \quad (19)$$

pool médio global

gradientes via retropropagação

onde Z é uma constante que representa o número de pixels no mapa de características de saída A_c , a saída da classe c (y_c) é definida como $\sum_k w_{ck} A_k$ é os pesos para a classe c em k é o mapa de características e A_{ij} é o mapa de características na posição de indexação (i, j) .

A saída do Grad-CAM nos permite fornecer pistas importantes para ajustar os parâmetros da rede que podem capturar de forma eficaz características significativas. O exemplo das imagens Grad-CAM são mostrados na Fig. 10. A figura ilustra que os defeitos foram detectados com sucesso usando o algoritmo de segmentação Grad-CAM.

6. Conclusão

Em conclusão, este artigo sugere uma abordagem para classificar e localizar as imagens de couro defeituosas. Para a tarefa de detecção, é aplicado um AlexNet pré-treinado como extrator de recursos, com SVM como classificador. Este método leva a uma precisão de detecção de 100%. Além disso, a visualização dos mapas de ativação para cada camada de convolução no AlexNet mostra que as características estão bem descritas. Como

para o estágio de localização, os métodos YOLOv2 e Faster R-CNN são testados. R-CNN mais rápido pode alcançar

IoU média mais alta (73,47%), mas tem uma velocidade computacional mais lenta em comparação com YOLOv2. No entanto, o treinamento de arquitetura de rede pode ser realizado de forma independente e, portanto, a identificação de defeitos é viável para implementação em uma situação do mundo real.

Na literatura, esta é uma tentativa bastante rara de investigar a viabilidade adotando os modelos de aprendizado profundo nas amostras de couro reais fornecidas pela indústria. O desempenho da detecção de defeitos de couro neste estudo demonstra os resultados positivos, embora a avaliação de tipos de defeitos limitados. Além disso, este estudo integra o sistema de detecção de forma que o mecanismo de ponta a ponta consiste na

procedimentos de eliciação de imagens, seleção de imagens, pré-processamento de imagens, rotulagem de verdade, classificação de defeitos e segmentação.

Para pesquisas futuras, podemos considerar a melhoria do algoritmo em três direções. Primeiro, redes neurais convolucionais pré-treinadas podem ser adotadas para extrair os recursos da área defeituosa e, em seguida, utilizá-los para detecção de defeitos. Em segundo lugar, a ferramenta de segmentação Single Shot Multibox Detector (SSD) pode ser usada para detecção de defeitos, pois pode fornecer resultados precisos com curto tempo de trem. Por último, mas não menos importante, os problemas com a detecção de vários tipos de defeitos (ou seja, picadas de insetos, rugas, cortes, arranhões, etc.), bem como em diferentes tipos de couro (vacas, porcos, ovelhas, cabras e crocodilos) podem ser abordado.

Agradecimentos Este trabalho foi financiado pelo Ministério da Ciência e Tecnologia (MOST), Taiwan (concessão nº MOST 109-2221-E-035-065-MY2, 109-2218-E-035-002- e 108-2218-E-035-018-).

Referências

- Aslam M, Khan TM, Naqvi SS, Holmes G, Nafa R (2020) Conjunto de redes neurais convolucionais com transferência de conhecimento para classificação de defeitos de couro em ambientes industriais. Acesso IEEE 8:198600–198614
- Bong HQ, Truong QB, Nguyen HC, Nguyen MT (2018) Sistema de inspeção baseado em visão para detecção e classificação de defeitos na superfície do couro. Em: 2018 5ª Conferência NAFOSTED sobre Informação e Ciência da Computação (NICS), IEEE, pp 300–304
- Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: um banco de dados de imagens hierárquicas em grande escala. In: Conferência IEEE 2009 sobre visão computacional e reconhecimento de padrões. IEEE, págs. 248–255
- Deng J, Liu J, Wu C, Zhong T, Gu G, Ling BW-K (2020) Uma nova estrutura para classificar defeitos de superfície de couro com base em uma rede residual otimizada por parâmetros. Acesso IEEE 8:192109–192118
- Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) O desafio das classes de objetos visuais pascal (voc). Int J Comput Vis 88(2):303–338
- Girshick R (2015) Fast r-cnn. In: Anais do IEEE Interna conferência nacional sobre visão computacional, pp 1440–1448
- He K, Zhang X, Ren S, Sun J (2016) Aprendizado residual profundo para reconhecimento de imagem. In: Anais da conferência IEEE sobre visão computacional e reconhecimento de padrões, pp 770–778
- Huang G, Liu Z, Van Der Maaten L, Weinberger K Q (2017) Redes convolucionais densamente conectadas. In: Anais da conferência IEEE sobre visão computacional e reconhecimento de padrões, pp 4700–4708
- Iandola F N, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: Precisão de nível Alexnet com 50x menos parâmetros e tamanho de modelo < 0,5 mb. arXiv pré-impressão [arXiv:1602.07360](https://arxiv.org/abs/1602.07360)
- Jain AK, Duin RPW, Mao J (2000) Reconhecimento de padrões estatísticos: uma revisão. IEEE Trans Pattern Anal Mach Intell 22(1):4–37
- Jawahar M, Babu NC, Vani K (2014) Classificação de textura de couro usando a técnica de extração de características wavelet. In: 2014 IEEE International Conference on Computational Intelligence and Computing Research, IEEE, pp 1–4

- Jawahar M, Babu NC, Vani K, Anbarasi LJ, Geetha S (2021) Sistema de inspeção baseado em visão para detecção de defeitos de superfície de couro usando abordagem de classificação de conjunto de otimização de enxame de partículas de convergência rápida. *Multimed Tools Appl* 80(3):4203–4235
- Jian L, Wei H, Bin H (2010) Pesquisa sobre inspeção e classificação de defeitos superficiais de couro com base em rede neural e árvore de decisão. In: 2010 International Conference On Computer Design and Applications, Vol 2, IEEE, pp V2–381
- Kasi MK, Rao JB, Sahu VK (2014) Identificação de defeitos de couro usando um algoritmo de processamento de imagem de detecção de bordas autoadaptativo. In: 2014 Conferência Internacional sobre Computação e Aplicações de Alto Desempenho (ICHPCA), IEEE, pp 1–4
- Krizhevsky A, Sutskever I, Hinton GE (2012) Classificação de imagenet com redes neurais convolucionais profundas. In: *Avanços nos sistemas de processamento de informações neurais*, pp 1097–1105
- Kwak C, Ventura JA, Tofang-Sazi K (2000) Uma abordagem de rede neural para identificação e classificação de defeitos em tecidos de couro. *J Intell Manuf* 11(5):485–499
- LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Aprendizagem baseada em gradiente aplicada ao reconhecimento de documentos. *Procedimento IEEE* 86(11):2278–2324
- Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dolár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: *Conferência Europeia de Visão Computacional*. Springer, pp 740–755
- Liong ST, Gan Y, Huang YC, Liu KH, Yau WC (2019a) Rede neural integrada e abordagem de visão de máquina para classificação de defeitos de couro. *arXiv pré-impressão* [arXiv:1905.11731](https://arxiv.org/abs/1905.11731)
- Liong ST, Gan Y, Huang YC, Yuan CA, Chang HC (2019b) Segmentação automática de defeitos em couro com aprendizado profundo. *arXiv pré-impressão* [arXiv:1903.12139](https://arxiv.org/abs/1903.12139)
- Mundt M, Majumder S, Murali S, Panetsos P, Ramesh V (2019) Arquiteturas neurais convolucionais de meta-aprendizagem para classificação de defeitos de concreto de vários alvos com o conjunto de dados de imagem de ponte de defeito de concreto. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 11196–11205
- Pistori H, Paraguassu WA, Martins PS, Conti MP, Pereira MA, Jacinto MA (2018) Detecção de defeitos em couro cru e couro wet blue. In: *Modelagem Computacional de Objetos Representados em Imagens. Fundamentos, Métodos e Aplicações: Actas do Simpósio Internacional CompIMAGE 2006* (Coimbra, Portugal, 20–21 Outubro 2006). Imprensa CRC, página 355
- Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: rumo à detecção de objetos em tempo real com redes de proposta de região. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
- Ren S, He K, Girshick R, Sun J (2015) , Faster r-cnn: rumo à detecção de objetos em tempo real com redes de proposta de região. In: *Avanços nos sistemas de processamento de informações neurais*, pp 91–99
- Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: explicações visuais de redes profundas via localização baseada em gradiente. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp 618–626
- Serafim AL (1992) Detecção de objetos em imagens de cenas naturais representadas por modelos ar usando pirâmides laplacianas: aplicação na localização de defeitos de couro. In: *Anais da Conferência Internacional de Eletrônica Industrial, Controle, Instrumentação e Automação de 1992*, IEEE, pp 716–720
- Simonyan K, Zisserman A (2014) Redes convolucionais muito profundas para reconhecimento de imagens em larga escala. *arXiv pré-impressão* [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
- Statista (2019) Artigos de couro de luxo. <https://www.statista.com>
- Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Indo mais fundo com convoluções. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1–9
- Tschandl P, Rosendahl C, Kittler H (2018) O conjunto de dados ham10000, uma grande coleção de imagens dermatoscópicas de várias fontes de lesões cutâneas pigmentadas comuns. *Dados Sci* 5(1):1–9
- Ventura JA, Chen JM (1996) Um modelo estrutural para reconhecimento de formas usando redes neurais. *J Intell Manuf* 7(1):1–11
- Villar P, Mora M, Gonzalez P (2011) Uma nova abordagem para segmentação de defeitos de couro wet blue. In: *Congresso Iberoamericano de Reconhecimento de Padrões*. Springer, pp 591–598
- Winiarti S, Prahara A, Murinto DPI (2018) Rede neural convolucional pré-treinada para classificação de imagem de couro curtido. *Trabalho líquido (CNN)* 9(1)
- Zhang X, Zhou X, Lin M, Sun J (2018) Shufenet: uma rede neural convolucional extremamente eficiente para dispositivos móveis. In: *Anais da Conferência IEEE sobre Visão Computacional e Reconhecimento de Padrões*, pp 6848–6856

Nota do editor Springer Nature permanece neutro em relação a reivindicações jurisdicionais em mapas publicados e afiliações institucionais.