JOÃO PEDRO CÂNDIDO JOSÉ

COMPARAÇÃO ENTRE DIFERENTES MODELOS DE IA NA SEGMENTAÇÃO DE SUPERFÍCIES DE FRATURA PÓS ENSAIO DE IMPACTO CHARPY



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2026

JOÃO PEDRO CÂNDIDO JOSÉ

COMPARAÇÃO ENTRE DIFERENTES MODELOS DE IA NA SEGMENTAÇÃO DE SUPERFÍCIES DE FRATURA PÓS ENSAIO DE IMPACTO CHARPY

Projeto de Conclusão de Curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Bacharel em Engenharia Aeronáutica. *VERSÃO REVISADA*

Orientador: Prof. Dr. Pedro Augusto Queiroz de Assis

Uberlândia - MG
2026

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

J83 José, João Pedro Cândido, 2001-
2026 Comparação entre diferentes Modelos de IA na segmentação de superfícies de fratura pós ensaio de impacto Charpy [recurso eletrônico] / João Pedro Cândido José. - 2026.

Orientador: Pedro Augusto Queiroz de Assis.
Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Uberlândia, Graduação em Engenharia Aeronáutica.
Modo de acesso: Internet.
Inclui bibliografia.
Inclui ilustrações.

1. Aeronáutica. I. Assis, Pedro Augusto Queiroz de,1989-, (Orient.). II. Universidade Federal de Uberlândia. Graduação em Engenharia Aeronáutica. III. Título.

CDU: 629.7

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Curso de Graduação em Engenharia Aeronáutica
Rodovia BR 050, KM 78, Bloco 1D, 2º andar - Bairro Glória, Uberlândia-MG, CEP 38400-902
Telefone: (34) 2512-6768 - www.mecanica.ufu.br - coceaero@mecanica.ufu.br

ATA DE DEFESA - GRADUAÇÃO

ATA DE DEFESA - GRADUAÇÃO

Curso de Graduação em:	ENGENHARIA AERONÁUTICA				
Defesa de:	PROJETO DE CONCLUSÃO DE CURSO - FEMEC43100				
Data:	13/03/2026	Hora de início:	15:30	Hora de encerramento:	16:40
Matrícula do Discente:	12021EAR021				
Nome do Discente:	JOÃO PEDRO CANDIDO JOSÉ				
Título do Trabalho:	Comparação entre diferentes modelos de IA na segmentação de superfícies de fratura pós ensaio de impacto Charpy.				

A carga horária curricular foi cumprida integralmente?	(X) Sim ( ) Não

Reuniu-se remotamente via Microsoft Teams (https://teams.microsoft.com/meet/2266398509944?p=iXoDorf0D9iulCCpqJ) a Banca Examinadora designada pelo Colegiado do Curso de Graduação em Engenharia Aeronáutica, assim composta: Prof.ª Rosenda Valdes Arencibia, FEMEC/UFU; Saint Clair Naves, membro externo/LTAD/UFU , e Prof. Pedro Augusto Queiroz de Assis, FEMEC/UFU, orientador do candidato.

Iniciando os trabalhos, o presidente da banca, Prof. Pedro Augusto Queiroz de Assis, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao discente a palavra para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do curso.

A seguir o(a) senhor(a) presidente concedeu a palavra, pela ordem sucessivamente, aos(às) examinadores(as), que passaram a arguir o(a) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o(a) candidato(a):

Aprovado. Nota: 100

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Pedro Augusto Queiroz de Assis**, **Professor(a) do Magistério Superior**, em 13/03/2026, às 16:40, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Saint Clair Alves Naves**, **Usuário Externo**, em 13/03/2026, às 16:41, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rosenda Valdes Arencibia**, **Professor(a) do Magistério Superior**, em 13/03/2026, às 16:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **7134558** e o código CRC **FBE957F9**.

Referência: Processo nº 23117.035866/2025-96 SEI nº 7134558

JOÃO PEDRO CÂNDIDO JOSÉ

COMPARISON BETWEEN DIFFERENT AI MODELS IN THE SEGMENTATION OF FRACTURE SURFACES AFTER THE CHARPY IMPACT TEST.

Final Paper submitted to the Faculty of Mechanical Engineering from Federal University of Uberlândia in partial fulfillment of the requirements for the degree of Aeronautical Engineer. *FINAL VERSION*

Advisor: Prof. Dr. Pedro Augusto Queiroz de Assis

Uberlândia - MG
2026

Este trabalho é dedicado à criança que fui,
que acreditava na curiosidade e colecionava porquês,
e agora, ao adulto que sou, cada vez mais próximo de respondê-los.

AGRADECIMENTOS

Agradeço, antes de tudo, à minha família, que foi meu alicerce em cada etapa desta caminhada. Obrigado pelo apoio incondicional, pela paciência nos momentos difíceis, pela compreensão diante das ausências e pela força que nunca faltou, mesmo quando eu mesmo duvidei. Este trabalho só foi possível porque vocês estiveram ao meu lado, acreditando e sustentando meus sonhos com amor, incentivo e presença.

Aos meus amigos de turma – Fillipi Rizzi, Gustavo Ramos, Lucas Evaristo, Lucas Souza, Maria Paula Almeida, Pedro Mariano, Rafael Calaça, Riann Lucas Carvalho e Vinicius Nunes – meu muito obrigado por tudo. Obrigado pelas conversas, pelas risadas, pela força nos dias puxados e por tornarem essa caminhada mais leve. Levo comigo não só o que aprendemos, mas também os momentos e a amizade de vocês.

Ao meu professor orientador, Pedro Augusto Queiroz de Assis, agradeço pela orientação atenciosa e sempre presente, pela disponibilidade e pelas contribuições fundamentais que ajudaram a dar clareza, direção e consistência a este trabalho. Obrigado por conduzir esse processo com seriedade, atenção e incentivo. À professora Rosenda Valdés Arencibia, minha sincera gratidão por ter me acompanhado durante praticamente toda a graduação, pelo acolhimento, pela confiança e por todo o apoio que marcou de forma significativa minha formação acadêmica e pessoal.

De forma geral, agradeço também à equipe do LTAD, laboratório onde realizei minha iniciação científica e tive o suporte necessário para desenvolver esta pesquisa e crescer academicamente. Foi nesse espaço que, durante três anos, vivi experiências valiosas para a minha formação, sempre encontrando apoio e aprendizado. Agradeço, ainda, à Petrobrás, por viabilizar e proporcionar condições para a realização desse trabalho.

"As pessoas que você ama fizeram quem você é.
Elas ainda são uma parte de você.
Se você parar de ser quem é,
essa última parte delas que ainda está dentro,
quem você é, desaparecerá."
(Rick Grimes, The walking dead)

RESUMO

JOSÉ, J. P. C. **Comparação entre diferentes modelos de IA na segmentação de superfícies de fratura pós ensaio de impacto Charpy**. 2026. 167 p. Projeto de Conclusão de Curso (Graduação em Engenharia Aeronáutica) – Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia - MG, 2026.

A *Shear Fracture Appearance* (SFA) é uma grandeza amplamente utilizada na interpretação de ensaios de impacto Charpy por expressar, em porcentagem, a fração de fratura dúctil observada na superfície de fratura (SF) dos corpos de prova (CPs) após o ensaio. Contudo, sua estimativa tradicional por inspeção visual e/ou procedimentos semiempíricos é trabalhosa e sujeita a variabilidade interobservador, motivando o emprego de visão computacional para automatizar a quantificação de áreas. Este trabalho propõe e avalia uma abordagem baseada em inteligência artificial (IA) para estimar a SFA a partir de imagens estereomicroscópicas de SFs. A metodologia é organizada em cascata com dois estágios: (i) um modelo fixo segmenta a SF total do fundo da imagem, resultando na região de interesse (ROI); (ii) um segundo modelo é empregado para separar as classes "frágil" e "delaminação". Particularmente, compararam-se os desempenhos de quatro conhecidas arquiteturas de segmentação na realização do segundo estágio: *U-Net*, *DeepLabV3–ResNet50*, *SegFormer* e *YOLO11l-seg*. Os modelos foram treinados sob um protocolo padronizado e avaliados de duas formas: (a) pela qualidade das máscaras previstas para as classes "frágil" e "delaminação" em comparação às anotações manuais (*ground truth*); e (b) pelo impacto dessas máscaras nas grandezas de interesse, isto é, na SFA estimada e na curva SFA×temperatura, a partir da qual se obtém a *Ductile–Brittle Transition Temperature* (DBTT) pelo critério $T_{t50\%\mathrm{SFA}}$. Em termos de segmentação, todos os modelos reproduzem com alta fidelidade a região frágil no conjunto de validação (com $\mathrm{IoU}_{\mathrm{frágil}}$ entre 0,921 e 0,935 e $\mathrm{Dice}_{\mathrm{frágil}}$ entre 0,959 e 0,966, além de precisão e *recall* próximos de 0,97), indicando que a identificação da fratura frágil é uma tarefa bem resolvida pelas arquiteturas avaliadas. Por outro lado, a delaminação permanece como o principal desafio para todos os métodos, com desempenho substancialmente inferior

(na melhor época, $\mathbf{IoU}_{\mathrm{delam}}$ entre 0,166 e 0,173 e $\mathbf{Dice}_{\mathrm{delam}}$ em torno de 0,284–0,295), refletindo o forte desbalanceamento do problema e a morfologia estreita dessa classe. Ao comparar as arquiteturas no segundo estágio, o *SegFormer* apresentou o melhor desempenho médio de segmentação ($mIoU$=0,554; $mDice$=0,631), com *U-Net* ($mIoU$=0,547; $mDice$=0,623) e *DeepLabV3–ResNet50* ($mIoU$=0,543; $mDice$=0,622) muito próximos. A *YOLO11l-seg*, após conversão das máscaras por instância para uma máscara semântica por classe, apresentou os menores valores globais ($mIoU$=0,513; $mDice$=0,585). Na avaliação das métricas derivadas, ao estimar a DBTT pelo critério $T_{t50\%\mathrm{SFA}}$, *SegFormer* (-14,44 °C) e *DeepLabV3–ResNet50* (-14,99 °C) permanecem próximos do manual (-13,74 °C), com desvios de -0,70 e -1,25 °C, respectivamente, evidenciando maior consistência na propagação dos erros de segmentação para as grandezas de interesse. Assim, o trabalho contribui com uma metodologia para estimativa automática de SFA, uma comparação controlada entre arquiteturas no estágio semântico e uma análise integrada entre desempenho de máscara e impacto em grandezas físicas como a SFA e DBTT.

Palavras-chave: Ensaio de impacto Charpy, Shear Fracture Appearance (SFA), segmentação, visão computacional, Inteligência Artificial (IA).

ABSTRACT

JOSÉ, J. P. C. **Comparison between different AI models in the segmentation of fracture surfaces after the Charpy impact test.**. 2026. 167 p. Projeto de Conclusão de Curso (Graduação em Engenharia Aeronáutica) – Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia - MG, 2026.

The *Shear Fracture Appearance* (SFA) is a quantity widely used in the interpretation of Charpy impact tests, as it expresses, in percentage, the fraction of ductile fracture observed on the fracture surface (FS) of specimens after testing. However, traditional SFA estimation by visual inspection and/or semi-empirical procedures is time-consuming and subject to inter-observer variability, which motivates the use of computer vision to automate area quantification. This work proposes and evaluates an artificial intelligence (AI)-based approach to estimate SFA from stereomicroscopic images of fracture surfaces. The methodology is organized as a two-stage cascaded pipeline: (i) a fixed model segments the entire fracture surface from the image background, yielding the region of interest (ROI); and (ii) a second model separates the "brittle" and "delamination" classes. In particular, four well-known segmentation architectures were compared in the second stage: *U-Net*, *DeepLabV3–ResNet50*, *SegFormer*, and *YOLO11l-seg*. The models were trained under a standardized protocol and evaluated in two ways: (a) by the quality of the predicted masks for "brittle" and "delamination" relative to manual annotations (*ground truth*); and (b) by the impact of these masks on the quantities of interest, namely the estimated SFA and the SFA×temperature curve, from which the *Ductile–Brittle Transition Temperature* (DBTT) is obtained using the $T_{t50\%\mathrm{SFA}}$ criterion. Regarding segmentation performance, all models reproduced the brittle region with high fidelity on the validation set (with $\mathbf{IoU}_{\mathrm{brittle}}$ between 0.921 and 0.935 and $\mathbf{Dice}_{\mathrm{brittle}}$ between 0.959 and 0.966, with precision and recall close to 0.97), indicating that brittle-fracture identification is effectively solved by the evaluated architectures. In contrast, delamination remains the main challenge for all methods, with substantially lower performance (at the best epoch, $\mathbf{IoU}_{\mathrm{delam}}$ between 0.166 and 0.173 and $\mathbf{Dice}_{\mathrm{delam}}$ around 0.284–0.295), reflecting

the strong class imbalance and the narrow morphology of this class. When comparing architectures in the second stage, *SegFormer* achieved the best average segmentation performance ($mIoU$=0.554; $mDice$=0.631), with *U-Net* ($mIoU$=0.547; $mDice$=0.623) and *DeepLabV3–ResNet50* ($mIoU$=0.543; $mDice$=0.622) being very close. *YOLO11l-seg*, after converting instance masks into a class-wise semantic mask, yielded the lowest global values ($mIoU$=0.513; $mDice$=0.585). In the evaluation of derived metrics, when estimating DBTT using the $T_{t50\%\mathrm{SFA}}$ criterion, SegFormer (-14.44 °C) and DeepLab (-14.99 °C) remain close to the manual reference (-13.74 °C), with deviations of -0.70 and -1.25 °C, respectively, indicating greater consistency in propagating segmentation errors to the quantities of interest. Therefore, this work contributes a complete methodology for automatic SFA estimation, a controlled comparison of architectures at the semantic segmentation stage, and an integrated analysis linking mask-level performance to the impact on physical quantities such as SFA and DBTT.

Keywords: Charpy impact test, Shear Fracture Appearance (SFA), segmentation, computer vision, Artificial Intelligence (AI).

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação esquemática das curvas típicas obtidas em séries de ensaios Charpy em função da temperatura. São indicados o patamar inferior (em verde), a região de transição (em amarelo) e o patamar superior (em azul), bem como a DBTT (em linha tracejada vermelha). 30

Figura 2 – Geometria típica de um CP entalhado em V apresentando suas principais dimensões (altura h, largura b, comprimento l e ângulo do entalhe α). 31

Figura 3 – Esquemático do aparato para realização ensaio de impacto Charpy: à esquerda, configuração geral do equipamento (pêndulo e sistema de leitura); à direita, detalhe do conjunto dos batentes com o CP posicionado. 32

Figura 4 – Exemplo de uma curva força × deslocamento obtida via ensaio instrumentado. Em vermelho, as grandezas características obtidas; hachurada sob a curva, a área que representa a energia. . 33

Figura 5 – Em a), região de fratura dúctil, com grande deformação; em b), região de fratura frágil, com pouca deformação e aspecto cristalino; em c) região de transição, com aspectos frágeis e dúcteis. 34

Figura 6 – Exemplo de SF de um CP Charpy com morfologia mista. A mistura de feições dúcteis e frágeis dificulta a separação óptica. 36

Figura 7 – Esquema da SF de um CP Charpy, apresentando aparência clássica de comportamento frágil (em 2) e dúctil (em 3). Em 1, o entalhe, A e B representam, respectivamente, comprimento e largura da região de fratura frágil. 37

Figura 8 – Em a), uma imagem de exemplo; em b), a mesma imagem segmentada com os *pixels* rotulados. 41

Figura 9 – Em (a), segmentação semântica, em destaque a classe “carro” (*car*); em (b), segmentação por instância, que reconhece cada objeto como distinto. 42
Figura 10 – Comparação entre arquiteturas de redes neurais. Em (a), rede neural totalmente conectada, na qual todos os neurônios (círculos) de uma camada estão ligados a todos os neurônios da camada seguinte. Em (b), rede neural com conexões parciais, apresentando uma estrutura mais simples entre as camadas de entrada, camadas escondidas e saída. 45
Figura 11 – Efeito de hiperparâmetros da convolução 2D. Em (a), o *stride* (S) controla o passo de deslocamento do filtro, alterando a densidade de amostragem e a resolução do mapa de saída; em (b), o *padding* (P) adiciona uma borda artificial à entrada, permitindo controlar dimensões e reduzir perda de informação nas bordas. Em (c), a dilatação (D) espaça os elementos do filtro, aumentando o campo receptivo efetivo sem aumentar o número de parâmetros do *kernel*. 49
Figura 12 – Exemplo do comportamento da função de ativação ReLU. . . . 51
Figura 13 – Ilustração esquemática de um bloco Conv–Norm–Act com filtro vertical aplicado a uma imagem. Em (a), a imagem de entrada; em (b) o mapa resultante após a convolução; em (c), o mapa normalizado; em (d), o mapa final após a aplicação da função de ativação ReLU. A escala de cores indica a magnitude dos valores do mapa em cada pixel (tons mais escuros representam valores menores e tons mais claros valores maiores). A mesma escala é utilizada em todos os painéis. 52
Figura 14 – Fluxo típico de uma imagem ao longo de uma rede convolucional. Em (a), imagem de entrada; Em (b), (d) e (f), o mapa de ativação após o bloco Conv–Norm–Act; em (c), (e) e (g), a saída após *downsampling* (*pooling*, $S = 2$); em (h), vetor resultante da operação de *flatten*; em (i) exemplo de saída probabilística (*softmax*) associada a classes genéricas. 58

Figura 15 – Exemplos de curvas de treinamento do tipo perda × épocas sobre o conjunto de validação e conjunto de treinamento. Em (a), caso de *overfitting*; em (b), caso de *underfitting*. 61
Figura 16 – Exemplo do comportamento da função *Sigmoide*. 66
Figura 17 – Diagrama esquemático da estrutura da rede *U-Net*. Setas em vermelho representam o caminho contrator; setas em azul representam o caminho expansor; setas cinza indicam as conexões de atalho por concatenação entre níveis de mesma escala; setas roxas e marrom representam, respectivamente, a operação de convolução 3×3 e 1×1. Blocos amarelos correspondem aos mapas de características no *encoder* e blocos verdes aos mapas no *decoder*; a camada final realiza a projeção para o mapa de saída. 75
Figura 18 – Diagrama esquemático da estrutura da rede *DeepLabV3* com estrutura de base *ResNet50*. Setas roxas representam operação de convolução; setas em azul representam concatenação; setas vermelhas e laranja representam, respectivamente, *pooling* e *max pooling*; setas rosas representam operação de *upsampling*; setas pretas representam entrada e saída da rede; o bloco amarelo representa a estrutura de base *ResNet50*; entre as chaves, a representação do ASPP. O conjunto tracejado e pontilhados correspondem, respectivamente, ao *encoder* e *decoder*. 77
Figura 19 – Diagrama esquemático da estrutura da rede *YOLO11l-seg*. Setas roxas representam operação de convolução; setas azuis representam concatenação; setas rosas representam operação de upsampling; seta verde representa aplicação de SPPF e CSP ; setas pretas representam entrada e saída da rede. 80

Figura 20 – Diagrama esquemático da estrutura da rede *SegFormer*. Setas roxas representam operação de convolução; seta rosa representa operação de *upsampling*; seta verde representa o *decoder*; seta amarela representa a operação de convolução transposta; as setas laranjas e os pentágonos representam, respectivamente, o módulo *Transformer* e sua saída; setas pretas representam entrada e saída da rede. 82
Figura 21 – Diagrama de blocos da metodologia proposta. 87
Figura 22 – Resultado do processamento via M1. Em (a), a imagem original, proveniente do estereomicroscópio; em (b), a SF segmentada do fundo da imagem. 87
Figura 23 – Resultado do processamento via M2. Em (a), a imagem de entrada de M2, proveniente da saída de M1; em (b), classificação dos *pixels* nas classes frágil e delaminação. 88
Figura 24 – Exemplo visual da metodologia proposta. 90
Figura 25 – Em (a), o suporte posicionado na base do estereomicroscópio para realizar a captura da imagem; em (b), uma visão superior do CP posicionado no suporte. 91
Figura 26 – Respectivamente, em (a), (b), (c) e (d), SFs de CPs com seção transversal 2,5 mm × 10 mm, 5 mm × 10 mm, 10 mm × 10 mm e 3 mm × 4 mm. 93
Figura 27 – Exemplo de anotação da área total da fratura (em vermelho) por polígonos no ambiente *Label Studio®*. 94
Figura 28 – Exemplo de anotação da área frágil (em azul) e delaminação (em verde) por polígonos no ambiente *Label Studio®*. 94
Figura 29 – Exemplos qualitativos de saída do M1. Na coluna (a), a imagem original; na coluna (b), a imagem recortada. 100
Figura 30 – Curvas de **mIoU** na validação ao longo das épocas para as quatro arquiteturas avaliadas, calculadas excluindo a classe *background*. Em (a), as curvas até a última época; em (b), as mesmas curvas, evidenciando, em linhas tracejadas, a melhor época de cada modelo. 104

Figura 31 – Curvas de mDice na validação ao longo das épocas para as quatro arquiteturas avaliadas, calculadas excluindo a classe *background*. 105
Figura 32 – Resumo dos resultados no conjunto de validação associadas à melhor época. 106
Figura 33 – Curvas de perda de treinamento e validação ao longo das épocas para (a) *U-Net*, (b) *DeepLabV3–ResNet50*, (c) *SegFormer* e (d) *YOLO11l-seg*. 107
Figura 34 – Matrizes de confusão avaliadas no conjunto de validação normalizadas por linha para cada arquitetura na melhor época. 110
Figura 35 – Exemplo de fratura frágil. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 113
Figura 36 – Exemplo de fratura dúctil, espera-se que as predições sejam vazias. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 115
Figura 37 – Exemplo de fratura dúctil com delaminações aparentes. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 117
Figura 38 – Exemplo de fratura em transição. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 119

Figura 39 – Exemplo de fratura em transição com presença de delaminações. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 121
Figura 40 – Curvas SFA×temperatura para as seis abordagens avaliadas (NIST, manual e quatro modelos). Em (a) pontos individuais por CP; em (b) curvas ajustadas por tangente hiperbólica simétrica e respectivas DBTTs (linhas tracejadas verticais); em (c) sobreposição dos pontos individuais com as curvas ajustadas. 127
Figura 41 – segmentação da SF do CP 1, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 141
Figura 42 – segmentação da SF do CP 2, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 142
Figura 43 – segmentação da SF do CP 3, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 143

Figura 44 – segmentação da SF do CP 4, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 144
Figura 45 – segmentação da SF do CP 5, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 145
Figura 46 – segmentação da SF do CP 6, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 146
Figura 47 – segmentação da SF do CP 7, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 147
Figura 48 – segmentação da SF do CP 8, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 148

Figura 49 – segmentação da SF do CP 9, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 149
Figura 50 – segmentação da SF do CP 10, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 150
Figura 51 – segmentação da SF do CP 11, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 151
Figura 52 – segmentação da SF do CP 12, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 152
Figura 53 – segmentação da SF do CP 13, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 153

Figura 54 – segmentação da SF do CP 14, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 154
Figura 55 – segmentação da SF do CP 15, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 155
Figura 56 – segmentação da SF do CP 16, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 156
Figura 57 – segmentação da SF do CP 17, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 157
Figura 58 – segmentação da SF do CP 18, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 158

Figura 59 – segmentação da SF do CP 19, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*. 159

LISTA DE TABELAS

Tabela 1 – Síntese comparativa dos quatro modelos selecionados, destacando paradigmas arquiteturais, mecanismos de captura de contexto/multi-escala e implicações conceituais para segmentação semântica. . 84
Tabela 3 – Parâmetros de aquisição e ajuste de imagem utilizados no software *Opticam*®. 92
Tabela 4 – Parâmetros de aumento de banco de dados utilizados no conjunto de treinamento. 96
Tabela 6 – Configuração de treinamento adotada para M1. 97
Tabela 8 – Configuração de treinamento adotada para M2. 97
Tabela 10 – Métricas avaliadas no conjunto de validação para a melhor época de M1 (em %). 99
Tabela 11 – Métricas por classe avaliadas no conjunto de validação para a melhor época de melhor época de cada arquitetura de M2 (em %). 109
Tabela 12 – Comparação da estimativa da SFA (%) nos exemplos da Seção 4.5. O erro é o erro absoluto em relação ao manual, em pontos percentuais (p.p.). 124
Tabela 13 – DBTT estimada pelo critério $T_{t50\%\mathrm{SFA}}$ para cada abordagem. . . 128

SUMÁRIO

1 INTRODUÇÃO 24
1.1 Contextualização e motivação 24
1.2 Convenções de notação 26
1.3 Organização do trabalho 27

2 FUNDAMENTAÇÃO TEÓRICA 28
2.1 Ensaio de impacto Charpy 28
2.1.1 Princípios do ensaio Charpy 29
2.1.1.1 Corpo de prova e entalhe 30
2.1.1.2 Máquina pendular e determinação da energia absorvida 31
2.1.1.3 Ensaio Charpy instrumentado 32
2.2 Fraturas frágeis, dúcteis e delaminação 33
2.2.1 A definição de SFA 35
2.3 Processamento de imagens digitais e segmentação 39
2.3.1 Conceitos básicos de processamento de imagens digitais . . . 40
2.3.1.1 Canais de cor e modelos de representação 40
2.3.2 Segmentação de imagens 40
2.3.2.1 Segmentação semântica × por instância 41
2.4 Aprendizado profundo na segmentação de imagens 42
2.4.1 Redes neurais convolucionais 44
2.4.1.1 Por que utilizar convolução para imagens? 45
2.4.1.2 Representação de dados e notação 47
2.4.1.3 Operação de convolução 2D e seus hiperparâmetros 48
2.4.1.4 Funções de ativação e normalização 50
2.4.1.5 Pooling e downsampling 54
2.4.1.6 Hierarquia de características e campo receptivo 55

2.4.1.7 Blocos e padrões arquiteturais em CNNs . . . 56
2.4.1.8 Treinamento supervisionado e otimização . . . 58
2.4.1.9 Regularização e generalização . . . 60
2.4.1.10 Transferência de aprendizado e pré-treinamento . . . 63
2.4.2 Formulação, funções de perda e métricas . . . 64
2.4.2.1 Predição densa e representação das saídas . . . 65
2.4.2.2 Funções de perda para treinamento de modelos de segmentação . . . 67
2.4.2.3 Métricas de desempenho . . . 69
2.4.3 Arquiteturas e paradigmas de segmentação profunda . . . 71
2.4.3.1 Famílias de arquiteturas para segmentação . . . 72
2.4.3.2 U-Net: encoder–decoder com conexões de atalho . . . 73
2.4.3.3 DeepLabV3 com estrutura de base ResNet50: convoluções atrous e ASPP . . . 76
2.4.3.4 YOLO11l-seg: paradigma one-stage com segmentação e adaptação semântica . . . 78
2.4.3.5 SegFormer: Transformer hierárquico para segmentação semântica . . . 80
2.4.3.6 Síntese comparativa dos paradigmas e modelos selecionados . . . 82

3 METODOLOGIA . . . 86
3.1 Banco de dados . . . 89
3.2 Treinamento dos modelos . . . 95

4 RESULTADOS E DISCUSSÕES . . . 98
4.1 Avaliação do desempenho global de M1 . . . 99
4.2 Organização dos resultados e critérios de comparação para M2 . . . 101
4.3 Avaliação quantitativa de desempenho global para M2 . . . 103
4.4 Avaliação quantitativa de desempenho por classe para M2 . . . 108
4.5 Avaliação qualitativa para M2 . . . 111
4.5.1 Fratura frágil . . . 112
4.5.2 Fratura dúctil . . . 114
4.5.3 Fratura dúctil com delaminação . . . 116
4.5.4 Fratura em transição . . . 118

4.5.5 Fratura em transição com delaminação 120
4.6 Avaliação de métricas derivadas 122
4.6.1 Avaliação da SFA 122
4.6.2 Curva SFA×temperatura e estimativa da DBTT 126

5 CONCLUSÕES E SUGESTÕES DE TRABALHOS FUTUROS 130

REFERÊNCIAS 133

APÊNDICE A DESENHO TÉCNICO DO SUPORTE DESENVOLVIDO PARA A CAPTURA DAS FRACTOGRAFIAS. 138

APÊNDICE B SEGMENTAÇÃO DOS CPS UTILIZADOS NA AVALIAÇÃO DA SEÇÃO 4.6 140

ANEXO A CERTIFICADO DE CALIBRAÇÃO DO EQUIPAMENTO CHARPY 160

CAPÍTULO

1

INTRODUÇÃO

1.1 Contextualização e motivação

O ensaio de impacto Charpy constitui um dos métodos mais difundidos para avaliação da resistência ao impacto e da tenacidade à fratura de materiais metálicos em condições de carregamento dinâmico, sendo amplamente empregado tanto em pesquisa quanto em rotinas de qualificação e controle de qualidade industrial. Em aplicações de engenharia, nas quais a integridade estrutural é crítica, como componentes submetidos a baixas temperaturas, estruturas soldadas e condições de serviço com risco de fratura instável, a caracterização do comportamento da fratura é importante por auxiliar na seleção de materiais e na tomada de decisão ao longo do ciclo de projeto, fabricação e inspeção. Nesse contexto, normas consolidadas internacionalmente, como a ASTM E23 e a ISO 14556 (ASTM, 2025c; ISO, 2023), não apenas padronizam o procedimento experimental, como também orientam critérios complementares de interpretação do resultado, associando o desempenho ao mecanismo de fratura observado.

O resultado mais tradicionalmente obtido do ensaio Charpy é a energia de impacto, conceituada como a energia requerida para quebrar um corpo de prova (CP) em uma máquina de ensaio de impacto por pêndulo, após correção para o atrito (ISO, 2016a). Entretanto, ela não descreve, de forma completa, a natureza do

processo de fratura. Em particular, em regimes de transição dúctil–frágil, também é necessário observar variações no aspecto da superfície de fratura (SF), pois essas variações refletem a concorrência entre micromecanismos distintos, como fratura por cisalhamento (associada a maior deformação plástica) e fratura por clivagem (associada a menor dissipação de energia), sendo cruciais para uma caracterização adequada do material. Nesse cenário, a *Shear Fracture Appearance* (SFA) emerge como um indicador direto da fração de fratura de caráter dúctil presente na SF, utilizada como medida para a interpretação do comportamento dúctil–frágil e, em contextos industriais, como critério de aceitação em especificações técnicas (ETSON, 2018; PETROBRAS, 2022; PHMSA, 2019). Assim, a SFA contribui para aumentar a robustez da análise do ensaio, oferecendo evidência fractográfica do mecanismo dominante, especialmente em casos em que apenas a energia absorvida é insuficiente para discriminar mudanças sutis no modo de fratura.

Tradicionalmente, dois métodos são amplamente utilizados para quantificar a SFA. A avaliação visual, realizada por operador, é atrativa por sua simplicidade, rapidez e baixo custo, sendo adequada para aplicações rotineiras. Em contrapartida, pode ser afetada por subjetividade e variabilidade intra e interobservador, além de se tornar mais trabalhosa em grandes volumes de CPs ou quando se busca repetibilidade; variações de iluminação, qualidade de imagem, delaminações e heterogeneidades superficiais podem ampliar essas discrepâncias. Outra alternativa consiste em estimar a SFA por meio de equações parametrizadas a partir de curvas instrumentadas, o que tende a reduzir a dependência do julgamento visual e pode favorecer a objetividade e repetibilidade quando o ensaio instrumentado está disponível. Contudo, a literatura reporta erros da ordem de ± 20 % em determinadas condições e a instrumentação necessária não é universalmente disponível, pois demanda infraestrutura específica (LUCON *et al.*, 2023; ASTM, 2025c; ISO, 2023).

Diante das limitações associadas aos métodos tradicionais de quantificação da SFA, este trabalho desenvolve uma abordagem automatizada a partir de imagens da SF, com o objetivo de reduzir a dependência do operador e aumentar a consistência dos resultados. Nesse contexto, técnicas de segmentação baseadas em visão computacional e inteligência artificial (IA) apresentam potencial para

lidar com variações naturais de aquisição e com a complexidade morfológica das SF, aproximando a avaliação de SFA de um processo mais reprodutível e escalável para uso em contexto experimental e em rotinas laboratoriais e de inspeção.

A abordagem proposta é organizada como um fluxo de trabalho em cascata com dois estágios. No primeiro estágio, um modelo é empregado para identificar e isolar automaticamente a SF na imagem, delimitando a região de interesse (ROI) correspondente à superfície de fratura. No segundo estágio, um modelo atua sobre a região isolada para realizar a segmentação dos *pixels* associados às novas regiões de interesse (fratura frágil e delaminação), permitindo a posterior estimativa da SFA a partir da quantificação de áreas, em conformidade com as definições adotadas pelas normas aplicáveis.

Como principais contribuições, destacam-se: (i) o desenvolvimento de um fluxo de trabalho em cascata para (a) separação do CP e extração da região de interesse e (b) segmentação pixel a pixel das regiões relevantes na SF; (ii) a comparação sistemática e padronizada de diferentes arquiteturas de segmentação em cada estágio da metodologia, sob o mesmo protocolo experimental, permitindo uma análise justa de desempenho; (iii) a avaliação do impacto do desempenho de segmentação na qualidade final da SFA estimada, comparando os resultados dos modelos com referências derivadas de anotações manuais revisadas e, quando disponível, com resultados obtidos via análise de curvas instrumentadas; e (iv) a discussão do potencial do método para reduzir subjetividade, padronizar a avaliação e acelerar rotinas de análise em ensaios Charpy.

1.2 Convenções de notação

Nesse trabalho, é convencionado que:

- Escalares são representados por letras em itálico (e.g., x);
- vetores por letras em negrito itálico (e.g., $\boldsymbol{x}$);
- matrizes e tensores por letras maiúsculas em negrito itálico (e.g., $\boldsymbol{W}$, $\boldsymbol{X}$);
- conjuntos por letras caligráficas (e.g., $\mathscr{D}$);

- operadores/métricas em fonte romana (e.g., IoU, Dice, mIoU, mDice);
- índices textuais são escritos como A_{total}, $A_{\text{frágil}}$, etc.

1.3 Organização do trabalho

Além desta introdução, o trabalho está organizado em cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica necessária para contextualizar o ensaio de impacto Charpy, os mecanismos de fratura e a definição da SFA, bem como os conceitos de processamento de imagens e segmentação profunda que suportam a abordagem proposta.

No Capítulo 3 descreve-se a metodologia adotada, incluindo o fluxo em cascata com dois estágios, a construção e preparação do banco de dados, as configurações de treinamento/validação e o protocolo de avaliação utilizado.

O Capítulo 4 apresenta e discute os resultados experimentais, organizando a comparação entre as arquiteturas avaliadas em termos de desempenho de segmentação e de métricas derivadas, incluindo a estimativa da SFA e a construção da curva SFA $\times$ temperatura para obtenção da DBTT.

Por fim, o Capítulo 5 consolida as conclusões do trabalho, destacando as principais contribuições, limitações e perspectivas de continuidade.

CAPÍTULO

2

FUNDAMENTAÇÃO TEÓRICA

2.1 Ensaio de impacto Charpy

O ensaio de impacto Charpy integra a família de ensaios de impacto desenvolvidos para investigar a sensibilidade de materiais metálicos a carregamentos rápidos e à presença de descontinuidades geométricas. Embora testes de impacto em barras existam desde o século XIX, a consolidação do método com CP entalhado e pêndulo ocorreu no início do século XX, com contribuições de Georges Charpy e a posterior padronização do procedimento, o que viabilizou sua adoção ampla na indústria como ensaio comparativo de tenacidade ao entalhe (TÓTH; ROSSMANITH; SIEWERT, 2002; SIEWERT *et al.*, 1999).

Na forma atualmente normatizada, o ensaio consiste em romper, por um único golpe, um CP entalhado apoiado em bigornas, medindo-se a energia absorvida na fratura a partir da variação da energia potencial do pêndulo (ISO, 2016a; ASTM, 2025c). A combinação de alta taxa de carregamento com a concentração de tensões introduzida pelo entalhe produz, na vizinhança da raiz, um estado de tensões multiaxial e elevada restrição à deformação plástica (alta triaxialidade), condição que pode favorecer a fratura instável em materiais suscetíveis, especialmente em temperaturas mais baixas (LUCON; MCCOWAN, 2011). Por essa razão, o ensaio Charpy em CP entalhado é amplamente empregado como ensaio

de triagem da suscetibilidade à fratura frágil em serviço, especialmente em aços ferríticos na região de transição dúctil-frágil, havendo diversas correlações práticas entre resultados de energia Charpy e medidas de tenacidade à fratura usadas em avaliações de integridade estrutural (HAHIN, 2022; American Bureau of Shipping (ABS), 2022; COLLINS *et al.*, 2023).

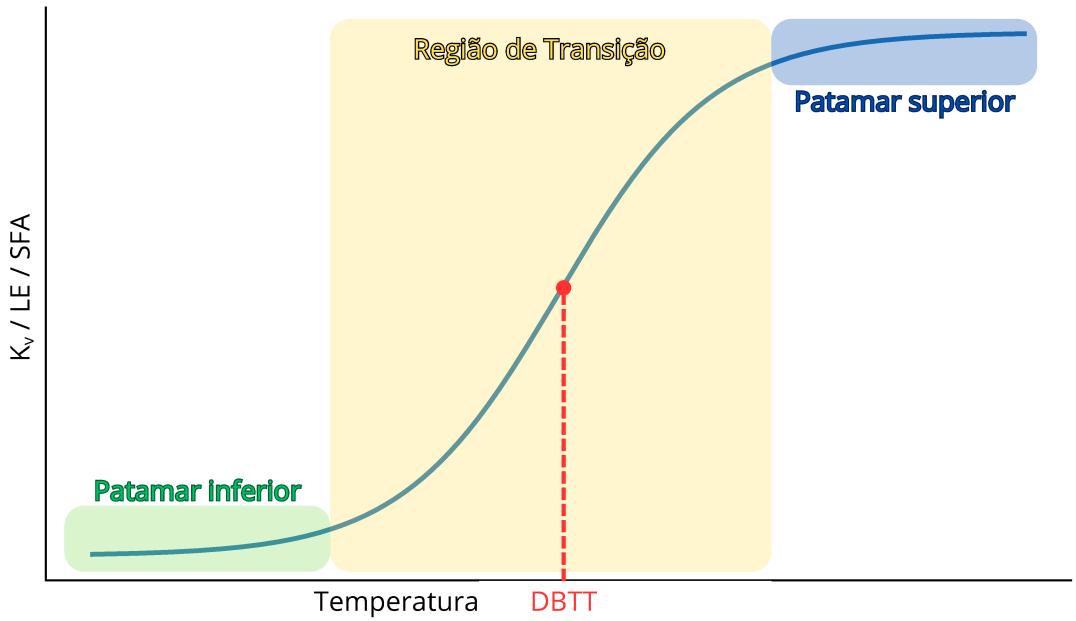
Embora a energia absorvida K_V seja o resultado clássico do ensaio Charpy, normas e prática experimental frequentemente incluem outras grandezas complementares, como a expansão lateral (*lateral expansion*, LE) e a SFA. A expansão lateral é determinada por medição dimensional nas metades fraturadas de cada CP, refletindo a deformação plástica ocorrida durante o impacto, enquanto a SFA corresponde à fração percentual da SF associada a mecanismos de cisalhamento, podendo ser estimada por análise visual ou, alternativamente, por métodos baseados em parâmetros extraídos de curvas instrumentadas, quando disponíveis (LUCON *et al.*, 2023). No ensaio Charpy instrumentado, além da leitura convencional de energia pela máquina, registra-se a resposta dinâmica (curvas força × tempo e/ou força × deslocamento), o que permite extrair parâmetros adicionais do evento de fratura (por exemplo, energias parciais e valores característicos associados a escoamento, iniciação e propagação), ao passo que a LE e a avaliação da SFA permanecem como grandezas de obtenção pós-ensaio.

Como os resultados de impacto de muitos metais variam fortemente com a temperatura T, a realização de séries de ensaios em diferentes temperaturas possibilita construir curvas $K_V \times T$, LE $\times T$ e/ou SFA $\times T$. Em materiais com transição dúctil-frágil, essas curvas tipicamente exibem um patamar inferior (*lower shelf*) em baixas temperaturas, uma região de transição e um patamar superior (*upper shelf*) em temperaturas mais elevadas. A partir dessas curvas, estimam-se temperaturas características da transição dúctil-frágil (*Ductile–Brittle Transition Temperature*, DBTT) por critérios definidos em normas (ISO, 2023; ASTM, 2025c), conforme ilustrado na Figura 1.

2.1.1 Princípios do ensaio Charpy

Esta subseção detalha os elementos fundamentais do ensaio Charpy do ponto de vista normativo: a geometria do CP entalhado, os componentes essenciais

Figura 1 – Representação esquemática das curvas típicas obtidas em séries de ensaios Charpy em função da temperatura. São indicados o patamar inferior (em verde), a região de transição (em amarelo) e o patamar superior (em azul), bem como a DBTT (em linha tracejada vermelha).



Fonte – Do autor.

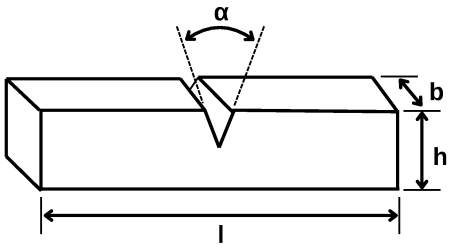
da máquina pendular e o procedimento de determinação da energia absorvida, além das particularidades do ensaio Charpy instrumentado, que complementa o ensaio convencional ao registrar a resposta dinâmica durante o impacto.

2.1.1.1 Corpo de prova e entalhe

O ensaio Charpy é realizado em um CP entalhado, cuja geometria é definida por norma (ASTM, 2025c; ISO, 2023). O CP chamado de convencional possui comprimento de 55 mm e seção quadrada de 10 mm × 10 mm, contendo um entalhe central em V. Quando não é possível obter o CP padrão, admite-se o uso de corpos de prova subdimensionados, os chamados *sub-size* (5 mm × 10 mm × 55 mm ou 2,5 mm × 10 mm × 55 mm) ou miniaturizados (3 mm × 4 mm × 27 mm). Um esquemático de um CP com entalhe em V é apresentado na Figura

2.

Figura 2 – Geometria típica de um CP entalhado em V apresentando suas principais dimensões (altura h, largura b, comprimento l e ângulo do entalhe α).



Fonte – Do autor.

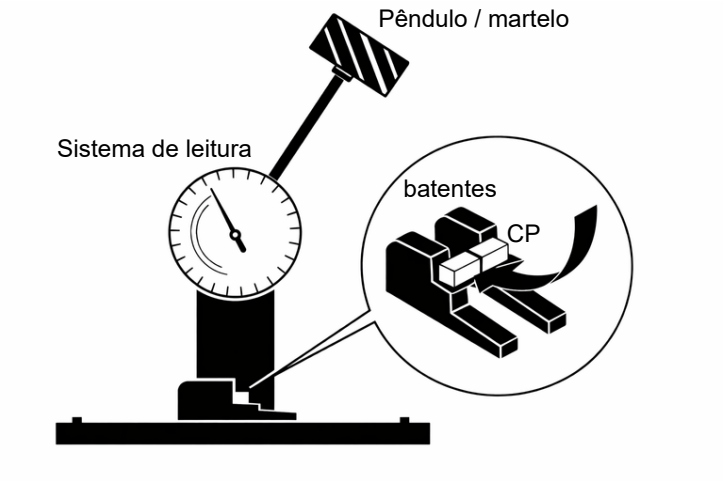
A geometria do entalhe tem papel decisivo na concentração de tensões e, portanto, na sensibilidade do ensaio à fratura frágil. No caso do entalhe em V, amplamente empregado em aplicações estruturais, a norma especifica ângulo incluso de $\alpha = 45°$, profundidade de 2 mm e raio na raiz de 0,25 mm, exceto em CPs miniaturizados, nesse caso, $\alpha = 60°$. A simetria do entalhe deve ser mantida perpendicular ao eixo longitudinal do CP, assegurando condições de carregamento consistentes entre ensaios (ASTM, 2025c; ISO, 2023; ASTM, 2025a).

2.1.1.2 Máquina pendular e determinação da energia absorvida

O equipamento responsável por realizar o ensaio é do tipo pendular, composto essencialmente por um conjunto de martelo oscilante (pêndulo) com cutelo, um conjunto de batentes de suportes que posiciona o CP, e um sistema de leitura da energia absorvida. O CP é apoiado como uma viga simplesmente apoiada, com o entalhe voltado para o lado oposto ao impacto, e o golpe é aplicado na linha de simetria do entalhe, conforme visto na Figura 3.

A energia absorvida na fratura K_V é obtida a partir da diferença entre a energia potencial inicial do pêndulo e a energia remanescente após a ruptura. Em termos operacionais, essa diferença é inferida pela altura (ou ângulo) atingida pelo pêndulo após o impacto, a partir da qual a máquina fornece a leitura indicada de energia. No procedimento normativo, recomenda-se verificar diariamente as perdas por atrito por meio de oscilações do pêndulo sem CP, quantificando-se parcelas

Figura 3 – Esquemático do aparato para realização ensaio de impacto Charpy: à esquerda, configuração geral do equipamento (pêndulo e sistema de leitura); à direita, detalhe do conjunto dos batentes com o CP posicionado.



Fonte – Do autor, adaptado de Nucor (2024).

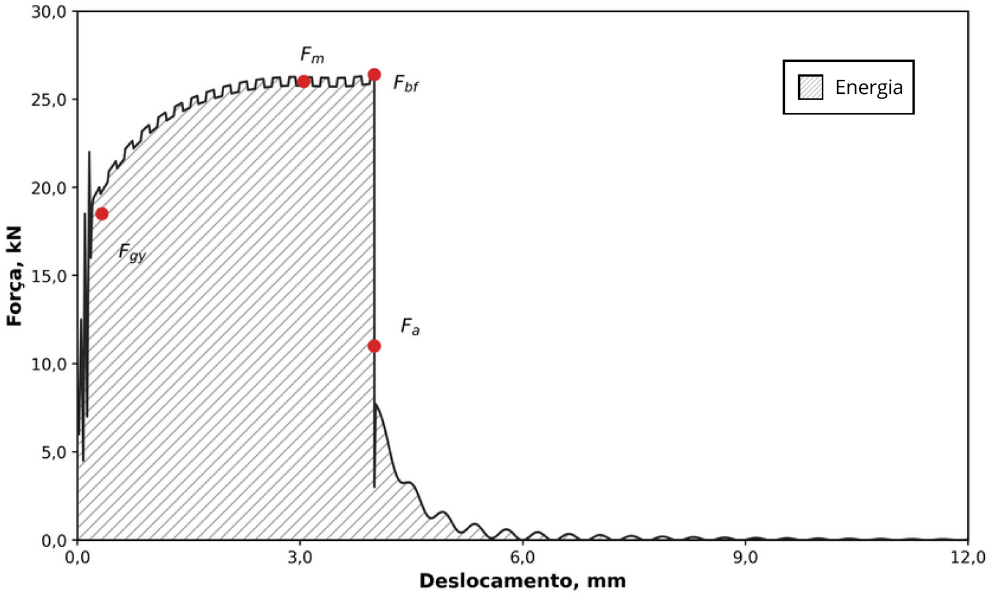
associadas ao mecanismo de indicação e ao conjunto de rolamentos/ar-resistência (ASTM, 2025c; ISO, 2023). A partir dessa verificação, é possível corrigir a energia indicada e também avaliar a conformidade do equipamento, já que perdas excessivas podem comprometer a exatidão da leitura. Além disso, a rastreabilidade metrológica e a verificação periódica da máquina (verificação direta e indireta) são requisitos essenciais para garantir comparabilidade entre resultados e confiabilidade das séries experimentais (ISO, 2016b).

2.1.1.3 Ensaio Charpy instrumentado

No ensaio Charpy instrumentado, a máquina pendular é equipada com instrumentação capaz de registrar, durante o impacto, a curva força × tempo e/ou força × deslocamento. Essa instrumentação é usualmente incorporada ao cutelo (*instrumented striker*) e associada a um sistema de aquisição apropriado para o fenômeno dinâmico. O registro da resposta permite obter informações adicionais sobre o comportamento de fratura sob altas taxas de carregamento, mesmo em situações em que a energia total absorvida seja semelhante entre materiais ou condições distintas.

A partir da curva força × deslocamento, exemplificada na Figura 4, a energia é definida como a área sob a curva (isto é, o trabalho $W = \int F \, \mathrm{d}s$, representado pela região hachurada). As normas ISO (2023) e ASTM (2025c) estabelecem que, ao longo desse registro, forças características associadas a diferentes estágios do impacto, como a força de escoamento geral (F_{gy}), a força máxima (F_{m}), a força associada ao início da fratura instável (F_{bf}) e a força de parada da trinca (F_{a}).

Figura 4 – Exemplo de uma curva força × deslocamento obtida via ensaio instrumentado. Em vermelho, as grandezas características obtidas; hachurada sob a curva, a área que representa a energia.



Fonte – Do autor.

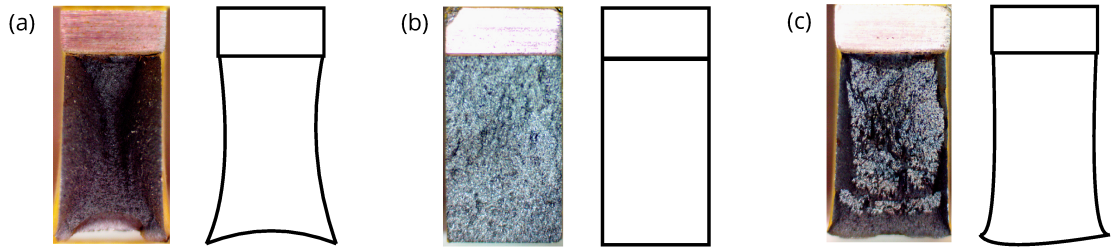
As forças características permitem o uso de métodos normatizados para estimar quantitativamente a SFA, conforme será abordado posteriormente.

2.2 Fraturas frágeis, dúcteis e delaminação

A SF resultante do ensaio Charpy frequentemente apresenta morfologia mista, cuja aparência macroscópica reflete a competição entre deformação plástica e propagação instável de trinca sob alta taxa de carregamento e elevada restrição

imposta pelo entalhe. Em termos gerais, regiões associadas à fratura dúctil tendem a exibir aspecto fibroso e deformação plástica evidente, conforme mostra a Figura 5a, enquanto regiões de fratura frágil tipicamente apresentam aspecto mais cristalino/brilhante e menor evidência de deformação, assim como é visto na Figura 5b. Na região de transição dúctil-frágil, é comum a coexistência dessas duas aparências na mesma superfície, com variação do fracionamento conforme a temperatura do ensaio e a microestrutura do material (ASM International Handbook Committee, 1987), como mostrado na Figura 5c.

Figura 5 – Em a), região de fratura dúctil, com grande deformação; em b), região de fratura frágil, com pouca deformação e aspecto cristalino; em c) região de transição, com aspectos frágeis e dúcteis.



Fonte – Do autor.

No contexto de ensaios Charpy em aços e ligas com transição dúctil-frágil, a fratura dúctil é comumente associada ao crescimento e coalescência de microcavidades, ocorrendo com maior contribuição de deformação por cisalhamento (por exemplo, formação de *shear lips* nas bordas). Em contrapartida, a fratura frágil está relacionada, de forma predominante, à clivagem transgranular e/ou a mecanismos de menor dissipação plástica, favorecendo propagação rápida e instável da trinca em baixas temperaturas. Além dessas duas classes, alguns materiais e condições de processamento podem apresentar delaminação (*splitting*), caracterizada por trincas aproximadamente paralelas ao plano de laminação e/ou à direção de processamento do material. Essas trincas podem dividir a superfície fraturada em "lâminas" e alterar a aparência global do CP, sendo frequentemente associadas à anisotropia microestrutural (por exemplo, grãos alongados, bandas de segregação e/ou alinhamento de inclusões) (ASM International Handbook Committee, 1987).

Uma vez esclarecidos os principais conceitos e princípios do ensaio de impacto Charpy, a seção seguinte dedica-se à análise da variável que constitui o eixo deste estudo, a SFA.

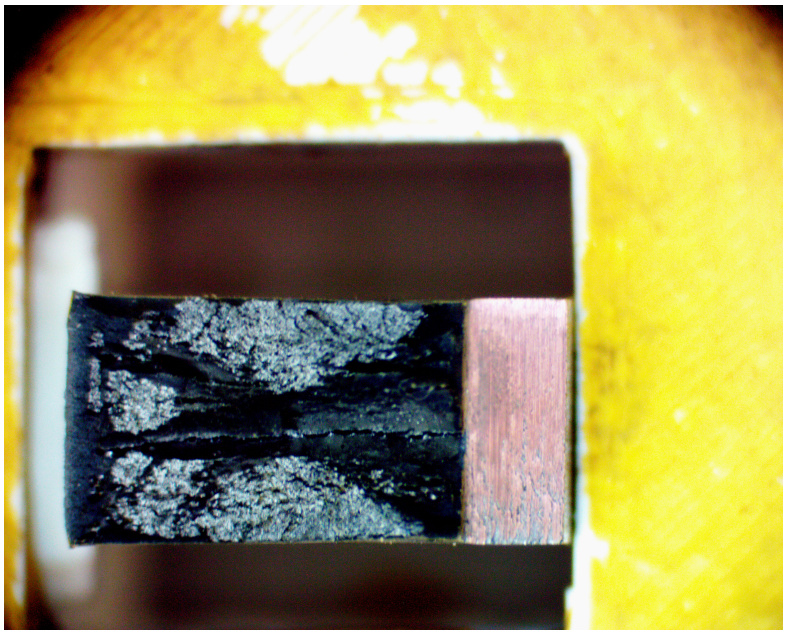
2.2.1 A definição de SFA

Tradicionalmente, a SFA é definida como a porcentagem da SF que rompeu de maneira dúctil/estável, associada macroscopicamente à fratura por cisalhamento (*shear*), em contraste com a parcela associada à fratura instável (comumente tratada como fratura frágil) (ASTM, 2025b).

Do ponto de vista operacional, normas de ensaio listam diferentes métodos para determinação de SFA, em ordem crescente de precisão: (i) medições lineares na própria superfície fraturada (comprimento e largura da região instável) para estimar a fração complementar até 100 %; (ii) comparação visual com cartas de referência; (iii) observação ampliada com *overlays* pré-calibrados ou planímetro; (iv) fotografia da superfície com medição por planímetro; e (v) captura de imagem digital e medição por *software* de análise de imagens. Em essência, esses procedimentos assumem que as regiões dúcteis e frágeis são distinguíveis macroscopicamente e que a fratura instável pode ser identificada como uma região bem delimitada (frequentemente descrita como uma "ilha" frágil circundada por áreas dúcteis), permitindo inferir a SFA como o complemento da fração frágil (ISO, 2023; ASTM, 2025c).

Entretanto, essa hipótese "idealizada" nem sempre se verifica. Em ligas modernas e aços de alta resistência, por exemplo, é comum que feições dúcteis e frágeis apareçam intimamente misturadas na SF, como exemplificado na Figura 6, tornando a tarefa de separação óptica intrinsecamente mais complexa e aumentando a subjetividade da avaliação. Nesses casos, uma alternativa prevista em normas específicas do ensaio instrumentado consiste em estimar a SFA por meio da interpretação da curva força × deslocamento, utilizando as forças características do evento, já apresentadas na Figura 4. A lógica desse método é relacionar a magnitude da queda de força associada à propagação instável (por exemplo, $F_{\mathrm{bf}} - F_{\mathrm{a}}$) com o nível de carregamento atingido (F_{m}) e, quando aplicável, com a parcela de deformação plástica desenvolvida antes de F_{m} (por exemplo, $F_{\mathrm{m}} - F_{\mathrm{gy}}$) (LUCON *et al.*, 2023).

Figura 6 – Exemplo de SF de um CP Charpy com morfologia mista. A mistura de feições dúcteis e frágeis dificulta a separação óptica.



Fonte – Do autor.

Independentemente do procedimento empregado, a determinação de SFA pode ser expressa, em termos gerais, como um problema de particionamento da SF em parcelas dúctil e frágil. Se A_{total} denota a área total considerada da SF e $A_{\text{dúctil}}$ a área atribuída à fratura estável, então a SFA pode ser estimada resolvendo-se (LUCON *et al.*, 2023)

$$\text{SFA} = 100 \cdot \frac{A_{\text{dúctil}}}{A_{\text{total}}}. \tag{2.1}$$

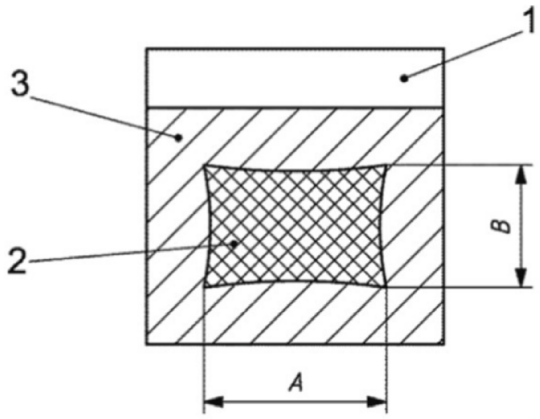
Assumindo que a área total considerada pode ser decomposta como $A_{\text{total}} = A_{\text{dúctil}} + A_{\text{frágil}}$, em que $A_{\text{frágil}}$ representa a parcela associada à fratura instável, pode-se reescrever (2.1) como

$$\text{SFA} = 100 \cdot \left(1 - \frac{A_{\text{frágil}}}{A_{\text{total}}}\right). \tag{2.2}$$

Nos métodos por medição linear, a área frágil é aproximada a partir de dimensões características medidas diretamente na superfície, tipicamente o com-

primento A e a largura B da região instável, conforme ilustra a Figura 7.

Figura 7 – Esquema da SF de um CP Charpy, apresentando aparência clássica de comportamento frágil (em 2) e dúctil (em 3). Em 1, o entalhe, A e B representam, respectivamente, comprimento e largura da região de fratura frágil.



Fonte – Lucon *et al.* (2023).

Sob a aproximação mais simples, em que a região instável é tomada como retangular no plano da fratura, obtém-se

$$\mathrm{SFA}_{\mathrm{lin}} \approx 100 \cdot \left(1 - \frac{A \cdot B}{A_{\mathrm{total}}}\right), \tag{2.3}$$

em que A_{total} é a área total da seção fraturada (nesse caso, frequentemente tratada como a área nominal da seção transversal no plano de fratura, que inclui a área do entalhe). Essa abordagem, apesar de conveniente, parte de uma simplificação geométrica que se torna insuficiente quando o contorno da região instável é irregular, quando há múltiplas "ilhas" frágeis, ou quando há mistura fina de feições dúcteis e frágeis, por exemplo, como no CP mostrado na Figura 6.

Nos métodos planimétricos e de análise digital, procura-se estimar $A_{\mathrm{frágil}}$ de forma mais direta, delineando-se o contorno da região instável (manualmente ou com auxílio de *software*) e computando-se a área correspondente. Em imagens digitais, isso se traduz, na prática, em contagem de *pixels* classificados em cada região e posterior conversão para percentual, isto é, uma implementação indireta de (2.2). Ainda assim, quando a transição entre regiões é difusa ou quando a iluminação/contraste varia, a decisão do contorno (e, portanto, $A_{\mathrm{frágil}}$) torna-se

sensível ao operador, procedimento de aquisição e pré-processamento, além do tempo elevado requerido para execução.

Para o ensaio Charpy instrumentado, as normas ASTM (2025b) e ISO (2023) propõem correlações empíricas que permitem estimar a SFA exclusivamente a partir das forças características F_{gy}, F_{m}, F_{bf} e F_{a}. Essas quatro expressões, amplamente discutidas e empregadas em publicações do NIST para a análise comparativa de métodos de determinação da SFA (LUCON *et al.*, 2023), serão denotada por $\mathrm{SFA}_{\mathrm{NIST}}$. As relações são apresentadas a seguir:

$$\mathrm{SFA}_1 = \left(1 - \frac{F_{\mathrm{bf}} - F_{\mathrm{a}}}{F_{\mathrm{m}}}\right) 100\ \%, \tag{2.4}$$

$$\mathrm{SFA}_2 = \left(1 - \frac{F_{\mathrm{bf}} - F_{\mathrm{a}}}{F_{\mathrm{m}} + (F_{\mathrm{m}} - F_{\mathrm{gy}})}\right) 100\ \%, \tag{2.5}$$

$$\mathrm{SFA}_3 = \left(1 - \frac{F_{\mathrm{bf}} - F_{\mathrm{a}}}{F_{\mathrm{m}} + 0{,}5(F_{\mathrm{m}} - F_{\mathrm{gy}})}\right) 100\ \%, \tag{2.6}$$

$$\mathrm{SFA}_4 = \left(1 - \sqrt{\frac{\frac{F_{\mathrm{gy}}}{F_{\mathrm{m}}} + 2}{3}} \left(\sqrt{\frac{F_{\mathrm{bf}}}{F_{\mathrm{m}}}} - \sqrt{\frac{F_{\mathrm{a}}}{F_{\mathrm{m}}}}\right)\right) 100\ \%, \tag{2.7}$$

nas quais a queda de força associada à propagação instável ($F_{\mathrm{bf}} - F_{\mathrm{a}}$) é interpretada como indicativa da fração de fratura instável, enquanto termos envolvendo F_{gy} procuram incorporar, direta ou indiretamente, a contribuição da deformação plástica antes de F_{m}. Como essas correlações são empíricas, seu uso deve ser entendido como estimativa indireta da composição da SF, e as próprias normas reportam uma precisão típica limitada (ordem de $\pm$ 20 %) para a SFA calculada a partir dessas relações. Além disso, a aplicação prática do método depende de instrumentação apropriada e da identificação consistente das forças características; em certas condições, podem ocorrer múltiplas quedas de força durante a propagação instável, dificultando a determinação de F_{bf} e a separação entre eventos físicos e oscilações dinâmicas do sistema. Adicionalmente, a identificação consistente das forças características pode depender da experiência do operador, especialmente quando o sinal apresenta ruído ou ambiguidades (LUCON *et al.*, 2023; ISO, 2023).

No geral, os métodos para estimar a SFA apresentam limitações complementares: abordagens ópticas/geométricas tendem a ser afetadas por subjetividade e

grande demanda de tempo para sua execução, enquanto abordagens instrumentadas requerem equipamento específico e fornecem apenas uma estimativa indireta baseada em correlações empíricas. Essas fragilidades motivam o desenvolvimento de procedimentos alternativos que preservem o caráter geométrico, mas reduzam a dependência do operador, fornecendo uma estimativa reprodutível e escalável da área frágil a partir de imagens, o que fundamenta a proposta deste trabalho de um fluxo de trabalho automatizado. Para embasar a proposta, a seção seguinte apresenta os conceitos essenciais de processamento de imagens e segmentação, que servem de base para a posterior introdução de visão computacional e redes neurais aplicadas ao problema.

2.3 Processamento de imagens digitais e segmentação

O processamento de imagens digitais reúne um conjunto de técnicas destinadas a representar, realçar, restaurar e extrair informação de imagens por meio de operações matemáticas e computacionais. Em aplicações de engenharia, essas técnicas permitem transformar dados visuais em descritores quantitativos, constituindo a base de tarefas clássicas de visão computacional, como detecção de bordas, reconhecimento de padrões e, em especial, a segmentação: a separação de regiões de interesse em uma imagem (GONZALEZ; WOODS, 2018; SZELISKI, 2022).

Nesta seção são apresentados conceitos fundamentais de imagens digitais e de segmentação, base para a introdução de conceitos mais complexos como visão computacional. Em particular, a formalização de imagens como matrizes de amostras, a noção de máscara/rotulagem e a quantificação de regiões por contagem de *pixels* são ideias que permanecem válidas tanto em abordagens tradicionais quanto em métodos modernos baseados em aprendizado profundo, conforme será discutido na próxima seção.

2.3.1 Conceitos básicos de processamento de imagens digitais

Uma imagem pode ser modelada como uma função bidimensional $f(x,y)$, em que x e y representam coordenadas espaciais e f representa a intensidade luminosa. Uma imagem digital é obtida ao amostrar e quantizar essa função, resultando em uma grade discreta de valores. Assim, uma imagem em tons de cinza pode ser representada por uma matriz

$$\boldsymbol{I} \in \mathbb{R}^{H \times W}, \tag{2.8}$$

em que H e W denotam altura e largura (em *pixels*), e cada elemento $\boldsymbol{I}(i,j)$ armazena a intensidade no *pixel* (i,j). H e W se relacionam com a resolução espacial, enquanto a resolução radiométrica (profundidade de *bits*) define a quantidade de níveis possíveis de intensidade (por exemplo, $2^8 = 256$ níveis em imagens de 8 *bits*) (GONZALEZ; WOODS, 2018).

2.3.1.1 Canais de cor e modelos de representação

Em imagens coloridas, cada *pixel* é descrito por um vetor em um determinado espaço de cor, como RGB:

$$\boldsymbol{I} \in \mathbb{R}^{H \times W \times C}, \quad C = 3, \tag{2.9}$$

em que os canais R, G e B, com valores inteiros que variam de 0 a 255, representam intensidades relativas em faixas espectrais distintas. Embora RGB seja a representação mais comum, outros espaços (por exemplo, HSV ou CIE $L^*a^*b^*$) podem facilitar determinadas operações ao separar componentes de luminância e cromaticidade. A escolha do espaço de cor influencia a robustez de procedimentos que dependem de contraste, brilho ou tonalidade (SZELISKI, 2022).

2.3.2 Segmentação de imagens

A segmentação é o processo de particionar uma imagem em regiões que sejam homogêneas segundo algum critério (intensidade, cor, textura, bordas ou características pré-definidas), de modo que cada região corresponda a uma entidade ou classe de interesse. Formalmente, pode-se definir uma segmentação como um

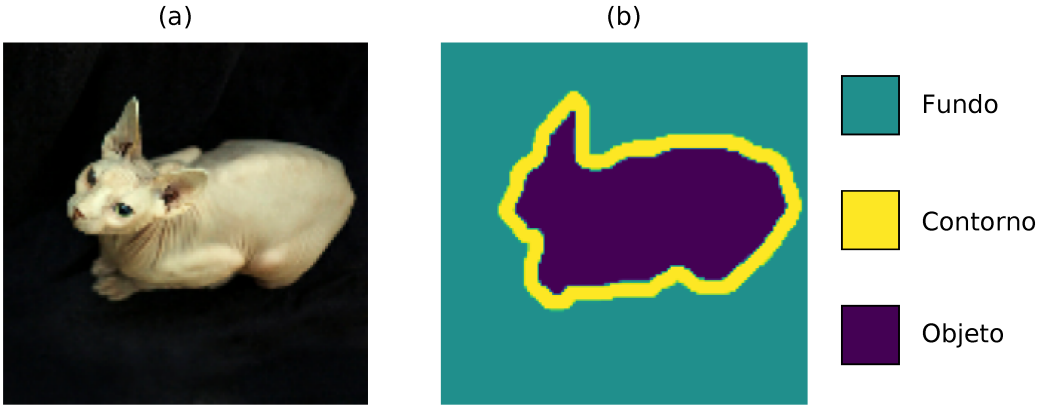
particionamento do conjunto de *pixels* Ω em subconjuntos disjuntos $\mathscr{R}_k$:

$$\Omega = \bigcup_{k=1}^{K} \mathscr{R}_k, \quad \mathscr{R}_i \cap \mathscr{R}_j = \varnothing \ (i \neq j), \tag{2.10}$$

em que cada $\mathscr{R}_k$ agrupa *pixels* com propriedades semelhantes segundo o modelo adotado (SZELISKI, 2022).

Em aplicações práticas, o resultado da segmentação é frequentemente representado por um mapa de rótulos (usualmente chamado de máscara), no qual cada *pixel* recebe um identificador de classe, assim como ilustra a Figura 8.

Figura 8 – Em a), uma imagem de exemplo; em b), a mesma imagem segmentada com os *pixels* rotulados.

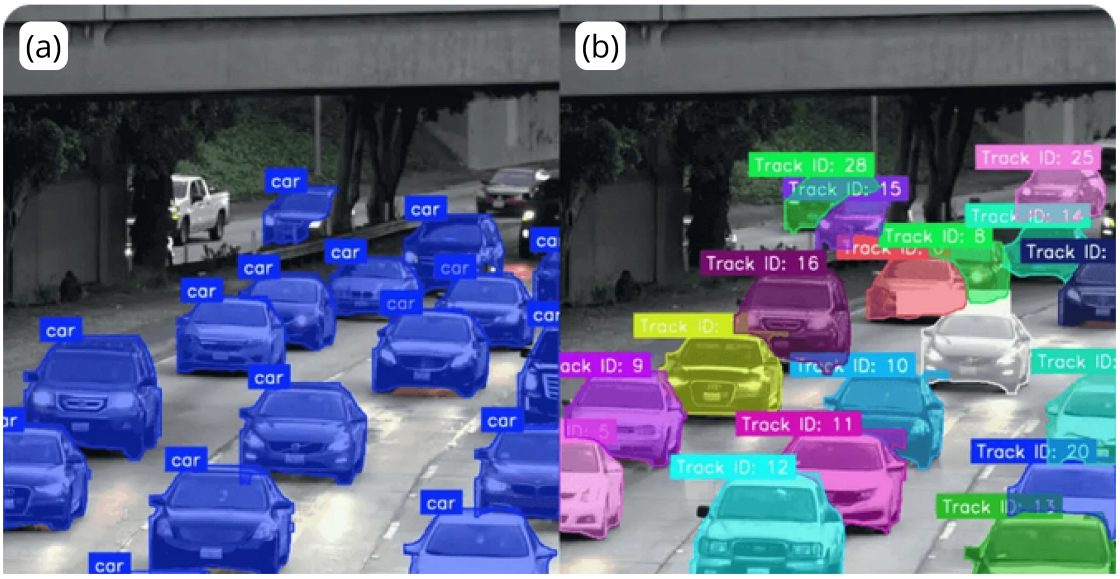


Fonte – Adaptado de TensorFlow (2022).

2.3.2.1 *Segmentação semântica × por instância*

Na segmentação semântica, todos os *pixels* que compõem uma imagem são classificados em uma categoria (por exemplo, “carro”), sem distinguir ocorrências diferentes de uma mesma classe. Já na segmentação por instância, além da classe, identifica-se cada objeto individualmente, separando instâncias distintas (e.g. ”carro 1”, ”carro 2”, ”carro 3”), como exemplificado na Figura 9 (SZELISKI, 2022). Essa distinção é importante porque determina o tipo de saída esperado: mapas de rótulos por classe (semântica) ou conjuntos de máscaras (instância).

Figura 9 – Em (a), segmentação semântica, em destaque a classe "carro" (*car*); em (b), segmentação por instância, que reconhece cada objeto como distinto.



Fonte – Ultralytics (2025a).

Hodiernamente, tarefas de segmentação (semântica ou por instância) são realizadas utilizando modelos de IA, especificamente o aprendizado profundo (DL – *Deep Learning*). Na sequência, conceitos fundamentais de DL, como redes convolucionais, são apresentados brevemente.

2.4 Aprendizado profundo na segmentação de imagens

O DL é uma vertente do aprendizado de máquina (*machine learning*, ML) que utiliza modelos com múltiplas camadas de processamento para aprender, a partir de exemplos, uma função que mapeia entradas em saídas desejadas (GOODFELLOW; BENGIO; COURVILLE, 2016). Em termos gerais, enquanto abordagens clássicas de ML frequentemente dependem da definição explícita de características (*features*) extraídas das imagens, o DL busca aprender essas representações de forma automática e hierárquica: camadas iniciais tendem a capturar padrões

locais simples (por exemplo, bordas e texturas), ao passo que camadas mais profundas combinam essas informações em estruturas mais abstratas e discriminativas (GOODFELLOW; BENGIO; COURVILLE, 2016). Em visão computacional, essa capacidade consolidou o DL como paradigma dominante em tarefas como classificação (atribuir um rótulo à imagem), detecção (localizar e classificar objetos) e, particularmente, segmentação (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; LONG; SHELHAMER; DARRELL, 2015). Assim, a segmentação profunda pode ser entendida como a aprendizagem de um mapeamento denso que prediz, para cada posição da imagem, uma classe ou máscara, aprendendo simultaneamente quais evidências visuais são relevantes para sustentar essa decisão.

O bloco básico desses modelos é o neurônio artificial, também chamado de unidade. Em sua forma mais simples, um neurônio recebe um vetor de entradas $\boldsymbol{x}$, combina esses valores por uma soma ponderada e adiciona um termo de viés, produzindo um valor intermediário $z = \boldsymbol{w}^\top \boldsymbol{x} + b$. Em seguida, aplica-se uma função de ativação $\phi(\cdot)$, gerando a saída $y = \phi(z)$. Intuitivamente, os pesos $\boldsymbol{w}$ controlam quais padrões na entrada são enfatizados ou atenuados, enquanto a ativação introduz não linearidade, permitindo que o modelo represente relações mais complexas do que combinações lineares. Ainda, uma camada corresponde ao conjunto de neurônios que opera em paralelo sobre a mesma entrada, produzindo uma representação intermediária que alimenta a próxima etapa. Ao empilhar camadas, forma-se uma rede profunda: cada camada transforma a representação anterior, construindo gradualmente descritores mais úteis para a tarefa final. De modo geral, camadas iniciais preservam maior proximidade com os dados brutos e capturam regularidades simples, enquanto camadas mais profundas integram essas regularidades em padrões de maior nível, mais discriminativos. Essa ideia de composição em níveis sustenta a noção de representações hierárquicas e explica por que arquiteturas profundas são particularmente eficazes em visão computacional, onde a decisão final depende da combinação de evidências visuais em múltiplas escalas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Entre as diversas arquiteturas empregadas em visão computacional, as Redes neurais convolucionais (CNNs) se destacam, projetadas especificamente para explorar a estrutura espacial de imagens por meio de operações de convolução.

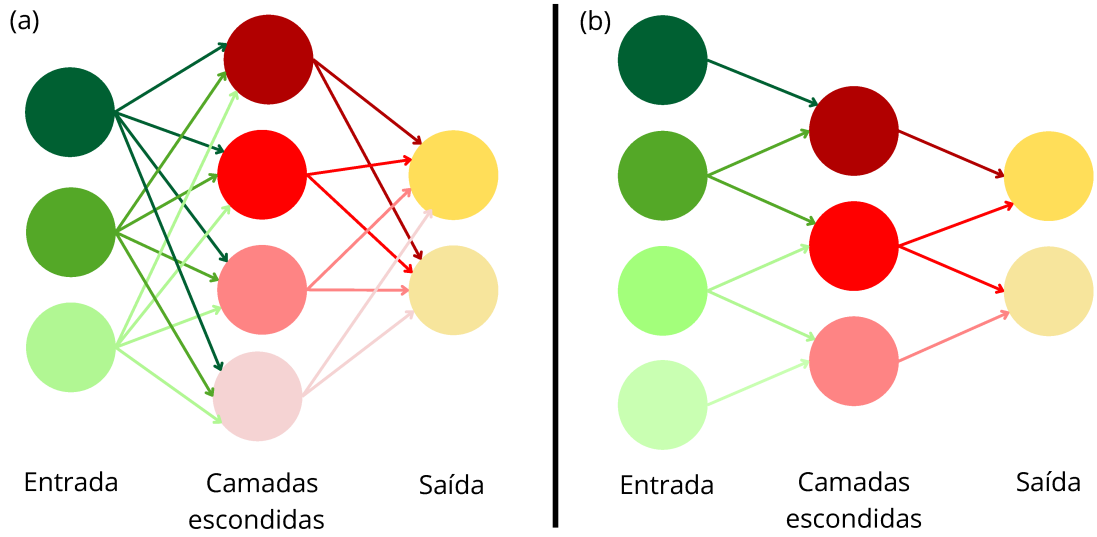
Essas redes constituem a base de grande parte dos modelos modernos de segmentação, pois permitem extrair características locais e globais de forma eficiente, mantendo a relação entre padrões visuais e sua posição na imagem. Dessa forma, a subseção seguinte apresenta os princípios fundamentais das CNNs e os principais componentes que compõem esse tipo de arquitetura.

2.4.1 Redes neurais convolucionais

As CNNs constituem a principal classe de modelos de DL empregada em visão computacional. Seu desenvolvimento foi motivado pela necessidade de lidar com a alta dimensionalidade das imagens e, ao mesmo tempo, explorar a estrutura espacial presente nesses dados, isto é, o fato de que *pixels* vizinhos tendem a estar correlacionados e padrões visuais relevantes (bordas, texturas e formas) se repetem em diferentes posições (GOODFELLOW; BENGIO; COURVILLE, 2016). Em contraste com redes totalmente conectadas, nas quais cada neurônio se conecta a todas as entradas, as CNNs introduzem mecanismos que tornam o aprendizado mais eficiente e robusto: conectividade local, compartilhamento de pesos e composição hierárquica de características ao longo das camadas (LECUN *et al.*, 1998). A Figura 10 ilustra, de forma simplificada, a diferença entre uma rede totalmente conectada a uma rede com conexões parciais.

Do ponto de vista conceitual, uma CNN pode ser entendida como uma sequência de transformações que mapeia uma imagem de entrada para um conjunto de mapas de características (conhecidas como *feature maps*), progressivamente mais informativos e mais robustos a variações irrelevantes, preservando relações espaciais importantes para tarefas de reconhecimento (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Esse princípio sustenta tanto modelos de classificação quanto arquiteturas de segmentação, nas quais é necessário equilibrar a extração de contexto global e a manutenção de detalhes locais. Na prática, isso significa que a rede transforma os valores brutos dos *pixels* em representações internas que realçam padrões relevantes ao problema (por exemplo, bordas, texturas e contornos) e suprimem variações não informativas (como pequenas mudanças de iluminação ou ruído), permitindo que a decisão final seja tomada com base em evidências visuais mais robustas. No contexto da segmentação semântica, essas

Figura 10 – Comparação entre arquiteturas de redes neurais. Em (a), rede neural totalmente conectada, na qual todos os neurônios (círculos) de uma camada estão ligados a todos os neurônios da camada seguinte. Em (b), rede neural com conexões parciais, apresentando uma estrutura mais simples entre as camadas de entrada, camadas escondidas e saída.



Fonte – Do autor.

representações são utilizadas para produzir uma predição densa, isto é, uma saída por *pixel*, de modo que a arquitetura deve combinar características de diferentes escalas: camadas profundas contribuem com informação contextual e semântica, enquanto camadas mais rasas preservam detalhes espaciais necessários para delimitar fronteiras. Nas subseções a seguir serão apresentados os fundamentos que compõem uma CNN, incluindo a motivação para o uso de convoluções, a formalização da operação, os principais componentes e aspectos de treinamento e generalização relevantes para as aplicações desse estudo.

2.4.1.1 Por que utilizar convolução para imagens?

Imagens digitais possuem uma estrutura em grade, na qual a informação visual relevante é, em grande parte, expressa por relações locais entre *pixels* vizinhos. Em redes totalmente conectadas, cada neurônio recebe como entrada todos

os valores de *pixels*, o que implica um número muito elevado de parâmetros já nas primeiras camadas, tornando o treinamento mais custoso e aumentando a propensão a sobreajuste (*overfitting*) (GOODFELLOW; BENGIO; COURVILLE, 2016). Além disso, esse tipo de conexão não incorpora, de forma explícita, a noção de vizinhança espacial que caracteriza imagens.

As CNNs contornam essas limitações ao adotar dois princípios centrais: conectividade local e compartilhamento de pesos. Na conectividade local, cada unidade em uma camada convolucional se conecta apenas a uma vizinhança restrita da entrada, denominada campo receptivo local. Isso reflete a hipótese de que padrões visuais podem ser detectados por operações locais e, ao mesmo tempo, reduz drasticamente o número de parâmetros em comparação a uma camada totalmente conectada. Já o compartilhamento de pesos significa que um mesmo filtro (um conjunto de pesos) é aplicado de forma "deslizante" ao longo de toda a imagem, permitindo detectar o mesmo padrão em diferentes posições e reduzindo ainda mais o número de parâmetros (LECUN *et al.*, 1998).

Um efeito importante das convoluções é que elas preservam a posição relativa dos padrões na imagem (propriedade conhecida como equivariância a translações). Em termos práticos, se uma borda, mancha ou textura aparece mais à esquerda ou mais à direita, a resposta do filtro também "se desloca" na mesma direção, em vez de mudar completamente. Por exemplo, se uma borda aparece 20 *pixels* mais à direita, o mapa de ativações produzido pelo filtro de convolução tende a apresentar a mesma resposta 20 *pixels* mais à direita. Isso é desejável porque o mesmo tipo de padrão pode surgir em diferentes regiões da imagem e ainda ser o mesmo. Em segmentação, essa característica ajuda a manter a correspondência espacial entre entrada e saída, facilitando a atribuição de rótulos por *pixel* e a delimitação de contornos (GOODFELLOW; BENGIO; COURVILLE, 2016).

Por fim, ao empilhar camadas convolucionais, a rede constrói uma descrição em "camadas": primeiro identifica padrões simples (bordas e texturas) e, em seguida, combina essas informações para reconhecer formas e estruturas maiores. Em outras palavras, a rede aprende do detalhe para o contexto. Essa construção hierárquica, somada à eficiência das convoluções (menor número de parâmetros do que conexões completas), ajuda a explicar por que CNNs são amplamente usadas

em visão computacional (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

2.4.1.2 Representação de dados e notação

Conforme visto na Seção 2.3, uma imagem digital pode ser representada como um arranjo multidimensional de valores. Segundo Goodfellow, Bengio e Courville (2016), para imagens em tons de cinza, a representação mais simples é uma matriz $\boldsymbol{I} \in \mathbb{R}^{H \times W}$. Para imagens coloridas, é comum utilizar uma representação tensorial $\boldsymbol{I} \in \mathbb{R}^{H \times W \times C}$, na qual C denota o número de canais (por exemplo, $C = 3$ para RGB). Em CNNs, a imagem de entrada e as ativações intermediárias também são tratadas como tensores, isto é, estruturas de dados com múltiplas dimensões, cujos eixos representam dimensões espaciais e canais de características.

Goodfellow, Bengio e Courville (2016), acrescentam, ainda, que durante o treinamento, em vez de processar uma única imagem por vez, é usual trabalhar com um conjunto de amostras de tamanho N simultaneamente, conjunto este chamado de lote (*batch*). Assim, a entrada do modelo é tipicamente representada por um tensor de ordem 4, que inclui a dimensão do lote:

$$\boldsymbol{X} \in \mathbb{R}^{N \times H \times W \times C} \quad \text{ou} \quad \boldsymbol{X} \in \mathbb{R}^{N \times C \times H \times W}. \tag{2.11}$$

A primeira forma (NHWC) organiza o canal como última dimensão, enquanto a segunda (NCHW) organiza o canal como segunda dimensão; ambas são equivalentes do ponto de vista matemático, diferindo apenas na ordem de armazenamento e em aspectos computacionais.

Após uma camada convolucional, a rede produz os mapas de características, que podem ser interpretados como respostas de filtros aprendidos aplicados à entrada. Cada filtro gera um canal de saída, de modo que as ativações em uma camada podem ser escritas como:

$$\boldsymbol{U} \in \mathbb{R}^{H' \times W' \times C'} \quad \text{ou} \quad \boldsymbol{U} \in \mathbb{R}^{C' \times H' \times W'}, \tag{2.12}$$

em que C' é o número de canais de saída (número de filtros) e H', W' dependem das escolhas de *stride*, *padding* e demais hiperparâmetros da convolução (que serão abordados em subseções adiante). Em termos intuitivos, valores elevados em um mapa de características indicam que o filtro correspondente reconheceu, naquela

posição, um padrão compatível com aquilo que foi aprendido durante o treinamento (por exemplo, uma borda, textura ou outra estrutura). Ao longo da rede, esses mapas fornecem a base para a construção de representações cada vez mais informativas, que serão utilizadas nas decisões de classificação, detecção ou segmentação.

2.4.1.3 Operação de convolução 2D e seus hiperparâmetros

A operação central em CNNs é a convolução 2D, que aplica um filtro (ou núcleo, *kernel*) sobre a entrada para produzir um mapa de resposta. Em muitas bibliotecas e implementações modernas, a operação utilizada é, estritamente, a correlação cruzada (isto é, sem inversão do filtro), embora o termo "convolução" seja mantido por conveniência e tradição (GOODFELLOW; BENGIO; COURVILLE, 2016). Conceitualmente, para uma entrada bidimensional $x(i,j)$ e um filtro w de tamanho $K_h \times K_w$, a operação (para um canal) pode ser expressa por

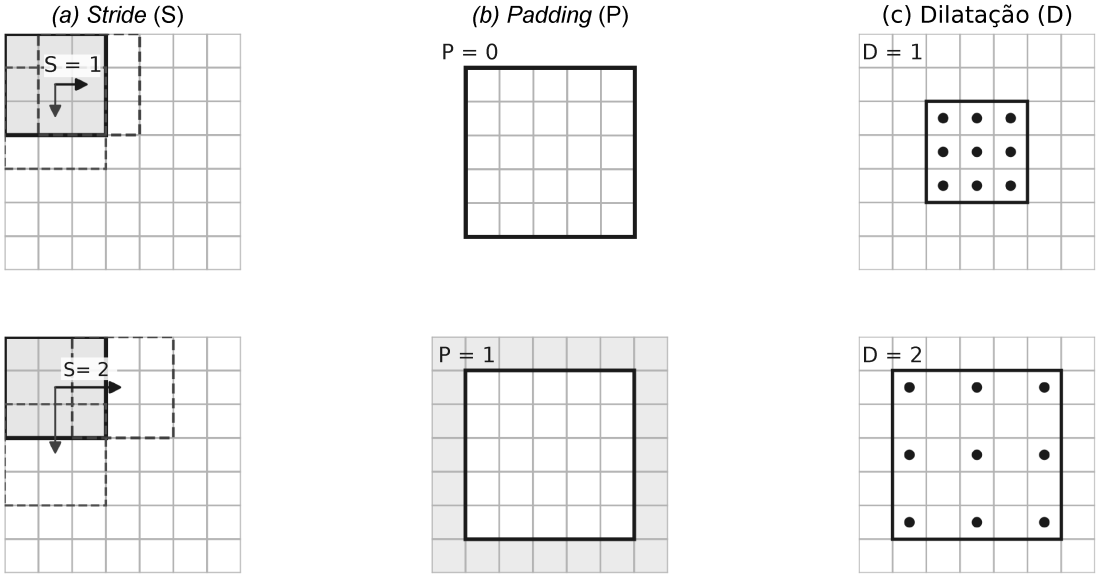
$$y(i,j) = \sum_{u=0}^{K_h-1} \sum_{v=0}^{K_w-1} x(i+u, j+v)\, w(u,v) + b, \tag{2.13}$$

em que b é um termo de viés. Em imagens multicanais, a soma se estende aos canais de entrada, e múltiplos filtros são aprendidos para produzir múltiplos canais de saída.

Os principais hiperparâmetros que controlam a convolução são: tamanho do filtro (K_h, K_w), *stride* (S_h, S_w), *padding* (P_h, P_w) e dilatação (D_h, D_w). O *stride* define o passo com que o filtro é deslocado sobre a entrada; valores maiores que 1 realizam subamostragem e reduzem a resolução espacial do resultado. O *padding* adiciona bordas artificiais (por exemplo, com zeros) ao redor da entrada, permitindo controlar o tamanho da saída e preservar informação próxima às bordas. A dilatação espaça os elementos do filtro, aumentando o campo receptivo sem aumentar o número de parâmetros (GOODFELLOW; BENGIO; COURVILLE, 2016). Na prática, a escolha desses hiperparâmetros determina uma relação entre nível de detalhe e contexto: convoluções com *stride* maior e/ou sem *padding* tendem a reduzir rapidamente a resolução e privilegiar informação mais global, enquanto *padding* adequado preserva melhor as dimensões e evita perda de informação nas

bordas. Já a dilatação permite incorporar contexto de regiões mais amplas mantendo a resolução, sendo particularmente útil em tarefas de segmentação nas quais se deseja combinar informação de escala maior sem degradar os contornos finos. A influência desses hiperparâmetros pode ser compreendida de forma esquemática: a Figura 11 ilustra como o *stride* controla o espaçamento das posições amostradas pelo filtro, o *padding* adiciona uma borda artificial para preservar dimensões e informação nas extremidades, e como a dilatação expande o campo receptivo efetivo sem aumentar o número de parâmetros.

Figura 11 – Efeito de hiperparâmetros da convolução 2D. Em (a), o *stride* (S) controla o passo de deslocamento do filtro, alterando a densidade de amostragem e a resolução do mapa de saída; em (b), o *padding* (P) adiciona uma borda artificial à entrada, permitindo controlar dimensões e reduzir perda de informação nas bordas. Em (c), a dilatação (D) espaça os elementos do filtro, aumentando o campo receptivo efetivo sem aumentar o número de parâmetros do *kernel*.



Fonte – Do autor.

Considerando uma entrada de altura H e largura W, uma convolução com

filtro $K_h \times K_w$, *stride* $S_h \times S_w$ e *padding* $P_h \times P_w$ produz uma saída com dimensões:

$$H' = \left\lfloor \frac{H + 2P_h - K_h}{S_h} \right\rfloor + 1, \tag{2.14}$$

$$W' = \left\lfloor \frac{W + 2P_w - K_w}{S_w} \right\rfloor + 1, \tag{2.15}$$

em que $\lfloor \cdot \rfloor$ denota a expressão piso. Para convoluções dilatadas, o tamanho efetivo do filtro é ampliado, substituindo-se K_h e K_w por:

$$K_{h,\text{ef}} = D_h\,(K_h - 1) + 1, \qquad K_{w,\text{ef}} = D_w\,(K_w - 1) + 1, \tag{2.16}$$

e as expressões de H' e W' passam a depender desses tamanhos efetivos (GOODFELLOW; BENGIO; COURVILLE, 2016).

Na prática, Goodfellow, Bengio e Courville (2016) afirmam que filtros 3×3 são amplamente utilizados por fornecerem um desempenho satisfatório entre capacidade de modelagem e custo computacional, sendo comum empilhar múltiplas convoluções 3×3 para aumentar o campo receptivo de maneira gradual. Já convoluções 1×1 (também chamadas de convoluções pontuais) atuam principalmente na dimensão de canais, combinando linearmente os mapas de características em cada posição espacial, úteis para ajustar o número de canais, reduzir custo computacional e introduzir combinações entre características aprendidas sem alterar a resolução espacial.

Uma vez detalhada a operação convolucional e os hiperparâmetros que controlam sua aplicação, é necessário introduzir os elementos que tornam as CNNs efetivamente capazes de aprender representações complexas. Em particular, a combinação de funções de ativação não lineares e mecanismos de normalização são fundamentais para aumentar o poder de modelagem da rede e tornar o treinamento mais estável. Esses componentes são discutidos a seguir.

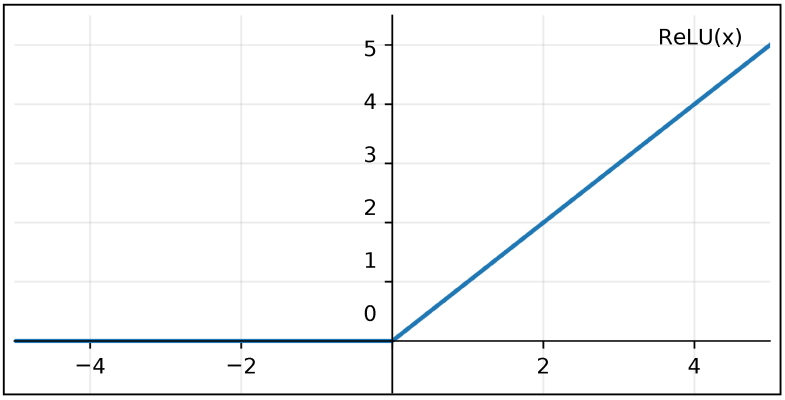
2.4.1.4 Funções de ativação e normalização

Camadas convolucionais, por si só, realizam transformações lineares sobre a entrada. Para que uma CNN seja capaz de modelar relações complexas e não lineares, intercala-se entre camadas lineares uma função de ativação não linear

capaz de introduzir não linearidade na transformação (GOODFELLOW; BENGIO; COURVILLE, 2016), ou seja, fazer com que a saída de uma camada não seja apenas uma "média ponderada" (combinação linear) da entrada, mas sim uma transformação que consegue separar e destacar estruturas diferentes na imagem. Entre as ativações mais utilizadas em visão computacional está a Unidade Linear Retificada (ou *Rectified Linear Unit*, ReLU), ilustrada na Figura 12 e definida por

$$\mathrm{ReLU}(x) = \max(0, x). \tag{2.17}$$

Figura 12 – Exemplo do comportamento da função de ativação ReLU.

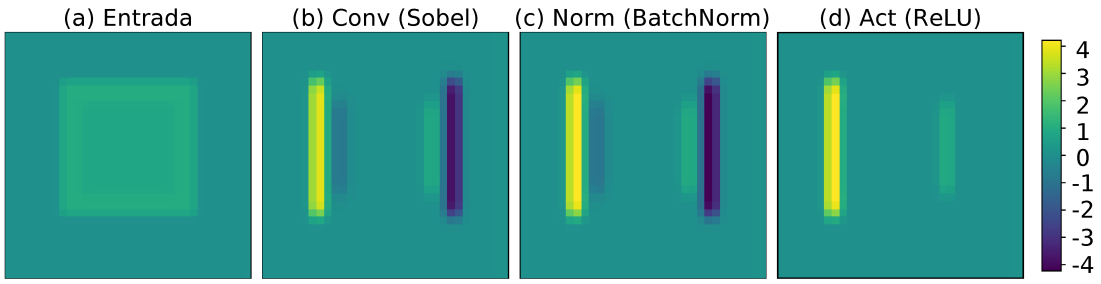


Fonte – Do autor.

Além das ativações, é comum empregar camadas de normalização para estabilizar e acelerar o treinamento. A chamada normalização de lote (ou *Batch Normalization*, BN) normaliza as ativações de cada canal com base nas estatísticas do lote (média e variância), seguidas de parâmetros aprendíveis de escala e deslocamento (IOFFE; SZEGEDY, 2015). Intuitivamente, a BN reduz variações de escala entre camadas e torna a otimização mais estável, permitindo o uso de taxas de aprendizado (*learning rate*, que será explicado na Subseção 2.4.1.8) maiores e reduzindo a sensibilidade à inicialização. Quando o tamanho do lote é pequeno, pode-se preferir normalizações alternativas, como a normalização por camada (*Layer Normalization*) ou a normalização por grupo (*Group Normalization*), que calculam estatísticas por amostra (ou por grupos de canais), diminuindo a dependência do lote (BA; KIROS; HINTON, 2016; WU; HE, 2018).

Na prática, um bloco típico em CNNs segue a composição convolução → normalização → ativação (frequentemente abreviada como Conv–Norm–Act). Essa organização é adotada em diversas arquiteturas por favorecer estabilidade numérica e boa propagação de gradientes, constituindo um padrão recorrente em redes profundas aplicadas a visão computacional (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa sequência pode ser visualizada por meio de um exemplo bidimensional simplificado, no qual uma imagem sintética é submetida sucessivamente às operações de convolução, normalização e ativação. A Figura 13 ilustra a evolução do mapa de características ao longo dessas etapas, evidenciando como a convolução realça estruturas locais, a normalização redistribui os valores de ativação e a função de ativação introduz não linearidade ao suprimir valores negativos.

Figura 13 – Ilustração esquemática de um bloco Conv–Norm–Act com filtro vertical aplicado a uma imagem. Em (a), a imagem de entrada; em (b) o mapa resultante após a convolução; em (c), o mapa normalizado; em (d), o mapa final após a aplicação da função de ativação ReLU. A escala de cores indica a magnitude dos valores do mapa em cada pixel (tons mais escuros representam valores menores e tons mais claros valores maiores). A mesma escala é utilizada em todos os painéis.



Fonte – Do autor.

Considere a imagem bidimensional sintética ilustrada na Figura 13, representada por uma matriz

$$\boldsymbol{S} \in \mathbb{R}^{32\times 32}, \tag{2.18}$$

cujos elementos assumem valores binários, definidos como

$$\boldsymbol{S}_{i,j} = \begin{cases} 1, & \text{se } (i,j) \in \Omega, \\ 0, & \text{caso contrário,} \end{cases} \tag{2.19}$$

sendo Ω corresponde às regiões do quadrado central e da linha horizontal presentes na imagem de entrada.

Na etapa de convolução, a imagem $\boldsymbol{S}$ é processada por um filtro $\boldsymbol{K} \in \mathbb{R}^{3\times3}$, definido por

$$\boldsymbol{K} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \tag{2.20}$$

o qual atua como um operador realçador de bordas (GONZALEZ; WOODS, 2018). O mapa resultante da convolução $\boldsymbol{F} \in \mathbb{R}^{32\times32}$ é obtido por

$$\boldsymbol{F}_{i,j} = \sum_{u=-1}^{1} \sum_{v=-1}^{1} \boldsymbol{K}_{u+2,v+2}\, \boldsymbol{S}_{i+u,j+v}, \tag{2.21}$$

assumindo *padding* apropriado que preserva as dimensões espaciais da imagem.

Em seguida, aplica-se uma normalização global por *Batch Normal*, dada por

$$\widehat{\boldsymbol{F}}_{i,j} = \frac{\boldsymbol{F}_{i,j} - \mu_{\boldsymbol{F}}}{\sigma_{\boldsymbol{F}} + \varepsilon}, \tag{2.22}$$

em que $\mu_{\boldsymbol{F}}$ e $\sigma_{\boldsymbol{F}}$ representam, respectivamente, a média e o desvio-padrão calculados sobre todos os elementos de $\boldsymbol{F}$, e $\varepsilon > 0$ é um termo de estabilidade numérica. Por fim, a função de ativação ReLU é aplicada elemento a elemento, produzindo a saída final

$$\boldsymbol{Z}_{i,j} = \max\left(0, \widehat{\boldsymbol{F}}_{i,j}\right). \tag{2.23}$$

As matrizes $\boldsymbol{S}$, $\boldsymbol{F}$, $\widehat{\boldsymbol{F}}$ e $\boldsymbol{Z}$ correspondem, respectivamente, aos mapas exibidos nos painéis da Figura 13.

De forma mais explícita, considerando inicialmente um *pixel* (i, j) localizado na região de maior intensidade da imagem de entrada, de modo que todos os valores em sua vizinhança 3×3 sejam iguais a 1. Nesse caso, o valor após a convolução é

$$\boldsymbol{F}_{i,j} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = 0, \tag{2.24}$$

indicando que regiões homogêneas não produzem resposta significativa para esse tipo de filtro.

Em contraste, para um *pixel* (p, q) situado sobre a borda dessa região, cuja vizinhança 3×3 contém valores distintos entre fundo e objeto, a aplicação do mesmo *kernel* resulta em uma resposta não nula, evidenciando a presença de uma transição abrupta de intensidade. Após as etapas de normalização e ativação, respostas negativas são suprimidas, enquanto ativações positivas associadas às bordas são preservadas, conforme ilustrado na Figura 13 (d).

2.4.1.5 Pooling e downsampling

Em CNNs, operações de redução de resolução (ou *downsampling*) são utilizadas para diminuir o custo computacional, aumentar o campo receptivo efetivo das camadas subsequentes e tornar as representações menos sensíveis a pequenas variações de posição. Uma forma clássica de realizar *downsampling* é o *pooling*, que resume valores em janelas locais, produzindo um mapa de características com menor dimensão espacial (GOODFELLOW; BENGIO; COURVILLE, 2016).

No *max pooling*, cada janela é substituída pelo maior valor no seu interior, preservando respostas fortes a padrões detectados (por exemplo, bordas e texturas), enquanto no *average pooling* utiliza-se a média dos valores, produzindo uma representação mais suavizada. Um exemplo simplificado do *max pooling* pode ser expresso como

$$y(i, j) = \max_{(u,v) \in \mathscr{W}_{i,j}} x(u, v), \tag{2.25}$$

em que $\mathscr{W}_{i,j}$ denota a janela local associada à posição (i, j) (GOODFELLOW; BENGIO; COURVILLE, 2016). Na prática, isso se traduz em uma redução direta

do tamanho dos mapas. Por exemplo, um *pooling* 2×2 com *stride* 2 transforma um mapa $H \times W$ em aproximadamente $(H/2) \times (W/2)$, diminuindo o número de ativações a serem processadas e, consequentemente, o custo computacional. Ao mesmo tempo, cada ativação nas camadas seguintes passa a resumir informação de uma região maior da imagem original, o que favorece a captura de contexto em níveis mais profundos.

Apesar de seu uso tradicional, muitas arquiteturas modernas substituem o *pooling* explícito por convoluções com *stride* maior que 1, que também realizam redução espacial, mas de forma aprendida (pois os filtros convolucionais são ajustados durante o treinamento). Ainda assim, o princípio permanece: a rede troca resolução espacial por uma representação mais compacta e, em geral, mais rica semanticamente.

Em tarefas de segmentação, reduções sucessivas de resolução tendem a facilitar a captura de estruturas globais, mas podem degradar a precisão de contornos e a identificação de objetos pequenos. Por isso, arquiteturas de segmentação frequentemente incluem mecanismos para recuperar resolução (por exemplo, *upsampling*) (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.4.1.6 Hierarquia de características e campo receptivo

Um dos conceitos centrais para entender CNNs é que as representações aprendidas tendem a se organizar de forma hierárquica ao longo das camadas. Em geral, camadas iniciais respondem a padrões locais simples, como bordas, cantos e texturas; camadas intermediárias combinam essas respostas para representar estruturas maiores; e camadas mais profundas agregam informação em descritores mais abstratos e semânticos, úteis para distinguir classes ou regiões de interesse (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; GOODFELLOW; BENGIO; COURVILLE, 2016). Essa progressão do local para o global está diretamente relacionada a como cada unidade em uma camada "enxerga" a entrada.

Esse "alcance" é formalizado pelo conceito de campo receptivo (ou *receptive field*). O campo receptivo de uma ativação é a região da imagem de entrada que pode influenciar o seu valor. À medida que se empilham convoluções e operações

de *downsampling*, o campo receptivo efetivo aumenta: uma ativação em camadas profundas depende de uma área maior da imagem original, incorporando mais contexto. Intuitivamente, isso permite que a rede interprete padrões considerando informações do entorno, o que é essencial quando a distinção entre classes depende não apenas de detalhes locais, mas também de sua organização em escala maior.

Para aumentar o campo receptivo sem reduzir a resolução espacial de forma agressiva, pode-se empregar convoluções dilatadas (também chamadas de *atrous convolutions*). Nessa operação, os elementos do filtro são espaçados por um fator de dilatação, ampliando o tamanho efetivo do *kernel* e permitindo integrar contexto de uma região maior sem aumentar o número de parâmetros e sem recorrer a *stride* elevado (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa estratégia é particularmente útil em segmentação, pois auxilia na captura de contexto global mantendo informação espacial mais detalhada.

Em síntese, a hierarquia de características e o crescimento do campo receptivo ajudam a explicar o compromisso inerente em segmentação: camadas profundas fornecem contexto e robustez semântica, mas a perda de resolução decorrente de *downsampling* pode prejudicar contornos finos e objetos pequenos. Por essa razão, arquiteturas de segmentação frequentemente combinam características de diferentes profundidades e escalas, preservando detalhes de camadas iniciais e, ao mesmo tempo em que explorando o contexto das camadas profundas, como será discutido adiante.

2.4.1.7 Blocos e padrões arquiteturais em CNNs

Embora CNNs possam ser descritas como sequências de camadas, arquiteturas modernas são tipicamente organizadas em blocos, que combinam operações elementares (convolução, normalização, ativação e, por vezes, *downsampling*) em unidades estruturais. Um bloco convolucional básico pode ser representado, por exemplo, pela já descrita composição Conv–Norm–Act, eventualmente repetida. Essa organização modular facilita o projeto de redes profundas e permite controlar a complexidade do modelo por meio do número de canais e da profundidade de cada estágio (GOODFELLOW; BENGIO; COURVILLE, 2016).

À medida que redes se tornam mais profundas, problemas de otimização podem surgir, como dificuldade de propagação de gradientes e piora de desempenho ao simplesmente empilhar camadas. Uma solução amplamente adotada é o uso das chamadas conexões residuais, nas quais a entrada de um bloco é somada à sua saída, formando um atalho (*skip connection*). Em sua forma conceitual, um bloco residual aprende uma função $F(x)$ e produz:

$$\{y\} = \{x\} + F(\{x\}), \tag{2.26}$$

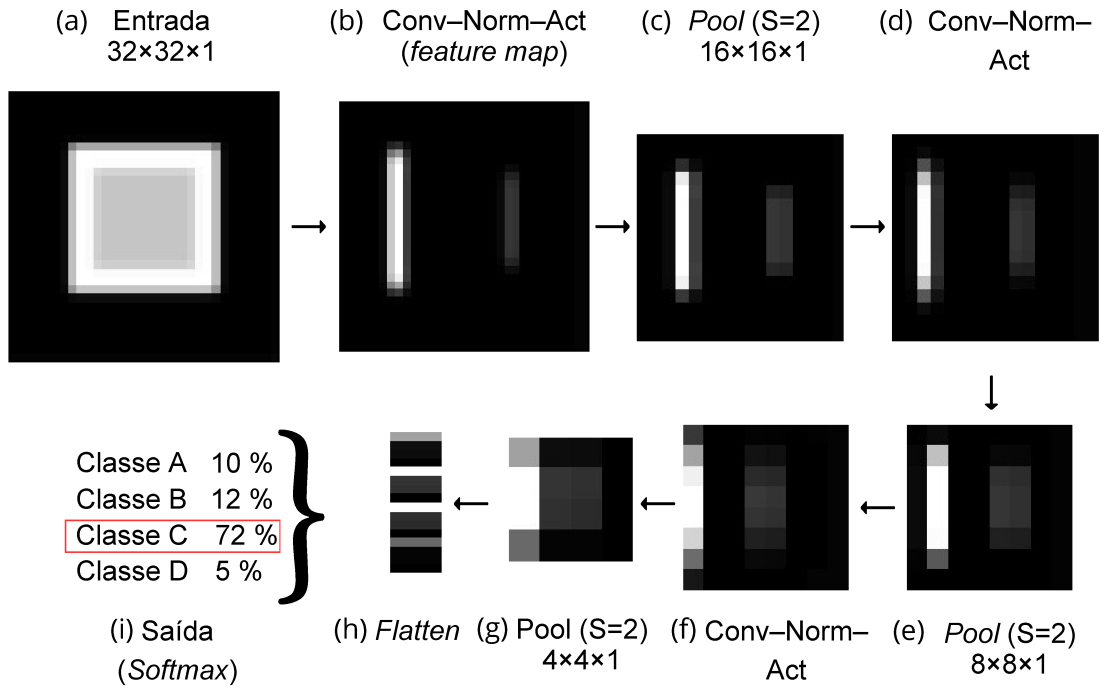
o que permite que o bloco modele um "resíduo" em relação à identidade. Essa formulação facilita o treinamento de redes muito profundas e constitui a base de arquiteturas como a ResNet, frequentemente empregada como estrutura de base (também conhecida como *backbone*) em tarefas de visão computacional (HE *et al.*, 2016).

Outro padrão arquitetural central, especialmente em segmentação, é o arranjo *encoder–decoder*. No *encoder*, a rede aplica sucessivas camadas convolucionais e operações de *downsampling* para extrair características e aumentar o campo receptivo, comprimindo gradualmente a representação espacial. No *decoder*, busca-se recuperar a resolução por meio de operações de *upsampling*, convertendo as características aprendidas em uma predição densa (LONG; SHELHAMER; DARRELL, 2015). Em segmentação, é comum complementar esse padrão com conexões de atalho entre níveis correspondentes do *encoder* e do *decoder*, combinando informação de alto nível (mais semântica) com detalhes espaciais preservados em camadas rasas, como é o caso da *U-Net* (RONNEBERGER; FISCHER; BROX, 2015).

A Figura 14 apresenta, de forma ilustrativa, o fluxo de processamento de uma CNN aplicado à imagem de entrada, destacando a sequência de extração de características por convolução e a redução progressiva da resolução espacial ao longo dos estágios. Nessa Figura, o bloco "Conv-Norm-Act" corresponde ao apresentado de forma expandida na Figura 13.

Em conjunto, esses blocos e padrões formam a base conceitual das principais famílias de modelos modernos de segmentação. Esses elementos serão retomados posteriormente, ao descrever as arquiteturas adotadas no presente trabalho para segmentação das classes frágil e delaminação.

Figura 14 – Fluxo típico de uma imagem ao longo de uma rede convolucional. Em (a), imagem de entrada; Em (b), (d) e (f), o mapa de ativação após o bloco Conv–Norm–Act; em (c), (e) e (g), a saída após *downsampling* (*pooling*, $S = 2$); em (h), vetor resultante da operação de *flatten*; em (i) exemplo de saída probabilística (*softmax*) associada a classes genéricas.



Fonte – Do autor.

2.4.1.8 *Treinamento supervisionado e otimização*

O treinamento de um modelo em regime supervisionado consiste em ajustar seus parâmetros (pesos e vieses) de modo que a saída do modelo se aproxime das anotações de referência fornecidas no chamado "conjunto de treinamento". Para isso, define-se uma função de perda $\mathscr{L}(\boldsymbol{\theta})$, que mede o erro entre a predição e a saída verdadeira, em que $\boldsymbol{\theta}$ denota o conjunto de parâmetros da rede. O objetivo do treinamento é encontrar parâmetros que minimizem essa perda, tipicamente por métodos iterativos baseados em gradiente (GOODFELLOW; BENGIO; COURVILLE, 2016).

A minimização é realizada via retropropagação do erro (*backpropagation*), na qual o gradiente $\nabla_{\theta}\mathscr{L}$ é calculado pela regra da cadeia, propagando derivadas da saída em direção às camadas iniciais. Em seguida, os parâmetros são atualizados por um algoritmo de otimização. O método clássico é o gradiente descendente estocástico (*Stochastic Gradient Descent*, SGD). Na prática, utiliza-se SGD com variantes como *momentum*, que suavizam oscilações e aceleram a convergência em vales alongados da superfície de erro (GOODFELLOW; BENGIO; COURVILLE, 2016).

Além do SGD, são amplamente utilizados otimizadores adaptativos como o Adam, que ajusta automaticamente a escala das atualizações por parâmetro com base em estimativas de primeiro e segundo momentos do gradiente (KINGMA; BA, 2015). Esses métodos facilitam o treinamento em certos cenários, embora a escolha do otimizador e de seus hiperparâmetros dependa do problema e do comportamento observado durante a validação.

Outro hiperparâmetro essencial no treinamento de um modelo é a taxa de aprendizado (η), um dos mais sensíveis. Valores muito altos podem levar a instabilidade e divergência, enquanto valores muito baixos tornam a convergência lenta ou podem aprisionar o treinamento em mínimos locais. Por essa razão, é comum empregar agendadores de taxa de aprendizado (*learning rate schedulers*), que reduzem η ao longo do treinamento segundo regras pré-definidas (por exemplo, decaimento em degraus, exponencial ou cosseno) (GOODFELLOW; BENGIO; COURVILLE, 2016).

Outro aspecto prático importante é o tamanho do lote (*batch size*). Lotes maiores fornecem estimativas de gradiente menos ruidosas, mas exigem mais memória e podem reduzir a frequência de atualizações; lotes menores, por outro lado, introduzem maior ruído, o que pode atuar como regularização, mas torna o treinamento mais sensível à taxa de aprendizado. Um ciclo completo de processamento de todo o conjunto de treinamento é chamado de época (*epoch*); ao longo de múltiplas épocas, o modelo é repetidamente exposto aos exemplos, refinando seus parâmetros.

De forma geral, várias amostras (imagens) formam um lote. A rede pro-

cessa esse lote e atualiza seus pesos uma vez (uma iteração). Um conjunto de iterações que percorre todo o conjunto de treinamento corresponde a uma época. O treinamento é composto por várias épocas e, ao final, obtém-se o modelo treinado, isto é, a arquitetura com os parâmetros ajustados pelos dados. Em síntese, o desempenho desse modelo treinado depende não apenas de sua arquitetura, mas também de escolhas de otimização (função de perda, otimizador, taxa de aprendizado, *scheduler* e tamanho de lote), que controlam a estabilidade e a eficiência do treinamento. Esses conceitos são essenciais para compreender a adaptação de CNNs a tarefas de segmentação, em que a função de perda e o balanceamento entre classes assumem papel central, como será formalizado na subseção seguinte. Entretanto, um bom desempenho durante o treinamento não garante, por si só, que o modelo apresente resultados satisfatórios em dados inéditos. Em redes com grande capacidade, é comum ocorrer o sobreajuste, isto é, o ajuste excessivo às particularidades do conjunto de treinamento, com perda de desempenho na validação, ao passo que configurações insuficientemente expressivas podem levar a *underfitting*. Assim, além das escolhas de otimização, torna-se necessário empregar estratégias que favoreçam a generalização do modelo, controlando o aprendizado efetivo.

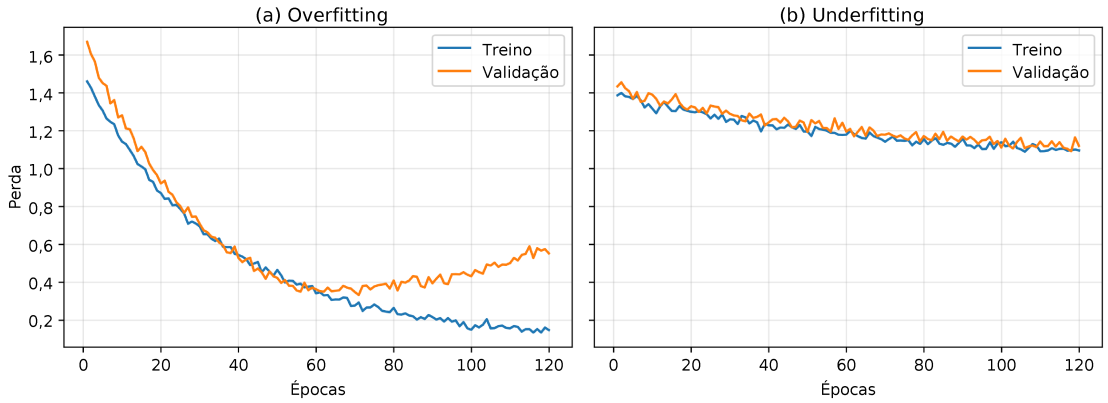
A subseção seguinte discute os principais mecanismos de regularização utilizados em CNNs, como penalizações nos pesos, *dropout*, *early stopping* e aumento de dados, bem como seu papel na redução do risco de sobreajuste.

2.4.1.9 Regularização e generalização

Em aprendizado supervisionado, busca-se que o modelo apresente bom desempenho não apenas nos dados usados para ajuste dos parâmetros, mas também em amostras novas, isto é, que possua boa capacidade de generalização. Em redes profundas, cuja capacidade de representação é elevada, pode ocorrer *overfitting*: o modelo reduz a perda no treinamento, porém passa a capturar particularidades e ruídos do conjunto de treino, deteriorando o desempenho em validação/teste. O caso oposto, *underfitting*, ocorre quando o modelo é incapaz de representar adequadamente o problema, resultando em desempenho baixo tanto em treinamento quanto em validação (GOODFELLOW; BENGIO; COURVILLE, 2016). Na prá-

tica, isso significa que, na verdade, um desempenho muito alto no conjunto de treinamento não é, por si só, evidência de um modelo "bom": se houver *overfitting*, a rede pode estar apenas "memorizando" padrões específicos daqueles exemplos e falhar quando confrontada com dados novos. Por outro lado, se houver *underfitting*, mesmo o treinamento não evolui de forma satisfatória, indicando que o modelo (ou sua configuração) não consegue capturar as relações necessárias para a tarefa. A Figura 15 apresenta exemplos de curvas de treinamento Perda × épocas que apresentam *overfitting*, em (a), caracterizado por redução contínua da perda de treinamento acompanhada por estagnação ou aumento da perda de validação após certo número de épocas) e *underfitting*, em (b), caracterizado por perdas de treinamento e validação elevadas e próximas entre si, indicando incapacidade do modelo de capturar adequadamente o padrão dos dados).

Figura 15 – Exemplos de curvas de treinamento do tipo perda × épocas sobre o conjunto de validação e conjunto de treinamento. Em (a), caso de *overfitting*; em (b), caso de *underfitting*.



Fonte – Do autor.

Para mitigar esses fenômenos, empregam-se técnicas de regularização, que limitam a complexidade efetiva do modelo ou tornam o treinamento mais robusto. Uma forma clássica de regularização é a penalização dos parâmetros na função de perda, conhecida como *weight decay* (regularização L_2). Nesse caso, minimiza-se

uma perda modificada:

$$\mathscr{L}_{\text{reg}}(\boldsymbol{\theta}) = \mathscr{L}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{2.27}$$

em que λ controla a intensidade da penalização. Intuitivamente, esse termo desencoraja pesos com magnitudes muito elevadas, favorecendo soluções mais "suaves" e menos sensíveis a variações pequenas na entrada (GOODFELLOW; BENGIO; COURVILLE, 2016).

Outra técnica amplamente utilizada é o *dropout*, no qual ativações são aleatoriamente anuladas durante o treinamento com uma probabilidade p, forçando a rede a não depender excessivamente de um subconjunto específico de neurônios (SRIVASTAVA *et al.*, 2014). Esse mecanismo pode ser interpretado como o treinamento de uma coleção de sub-redes e, na fase de inferência, a rede completa é utilizada com reescalonamento apropriado das ativações, reduzindo correlações internas e melhorando generalização.

Além disso, estratégias de parada baseadas em validação, como o *early stopping*, interrompem o treinamento quando o desempenho em validação deixa de melhorar, evitando que o modelo continue a se ajustar ao ruído do treinamento. Na prática, define-se uma tolerância para uma métrica (chamada de *patience*) para a interrupção, buscando um compromisso entre ajuste suficiente e prevenção de sobreajuste.

Uma abordagem complementar, especialmente importante em visão computacional, é o aumento de dados (*data augmentation*). Nessa estratégia, exemplos sintéticos são gerados a partir das amostras originais por transformações que preservam o rótulo (por exemplo, rotações, espelhamentos, alterações controladas de brilho/contraste ou ruído), ampliando a variabilidade do treinamento sem a necessidade de novas anotações. Conceitualmente, o aumento de dados atua como regularizador ao induzir invariâncias desejadas e reduzir a chance de o modelo "memorizar" padrões específicos do conjunto de treino (GOODFELLOW; BENGIO; COURVILLE, 2016).

Em conjunto, *weight decay*, *dropout*, *early stopping* e aumento de dados constituem mecanismos complementares para melhorar a generalização de CNNs. Em tarefas de segmentação, essas técnicas são particularmente relevantes devido à

alta dimensionalidade da saída e à presença frequente de desbalanceamento entre classes, aspectos que serão retomados ao discutir a formulação do problema e funções de perda específicas para segmentação.

2.4.1.10 Transferência de aprendizado e pré-treinamento

Em muitas aplicações de visão computacional, o treinamento de uma CNN a partir de valores iniciais aleatórios para o conjunto de parâmetros $\boldsymbol{\theta}$ pode demandar grandes quantidades de dados anotados para alcançar bom desempenho. Uma alternativa amplamente adotada é a transferência de aprendizado (*transfer learning*), no qual o modelo é inicializado com pesos obtidos em um pré-treinamento realizado em uma base grande e diversa (por exemplo, *ImageNet*), e posteriormente adaptado à tarefa de interesse (GOODFELLOW; BENGIO; COURVILLE, 2016; KRIZHEVSKY; SUTSKEVER; HINTON, 2012). De maneira simplificada, é que camadas iniciais aprendem detectores genéricos (bordas, texturas e padrões simples) que tendem a ser úteis em diferentes domínios visuais, enquanto camadas mais profundas podem ser refinadas para capturar características específicas do problema-alvo.

A adaptação do modelo pré-treinado pode ser feita de diferentes maneiras. Uma estratégia comum é utilizar a rede pré-treinada como extratora de características, mantendo seus pesos iniciais fixos (congelamento, ou *freezing*) e treinando apenas as camadas finais responsáveis pela predição na nova tarefa. Alternativamente, pode-se realizar o ajuste fino (ou *fine-tuning*), no qual parte (ou a totalidade) das camadas tem seus pesos ajustados com os dados do novo domínio, frequentemente com taxas de aprendizado menores para evitar degradação das representações pré-aprendidas (GOODFELLOW; BENGIO; COURVILLE, 2016). Em problemas de segmentação, é especialmente comum empregar um estrutura de base pré-treinado como codificador (*encoder*), acoplado a uma "cabeça" (*head*) de segmentação treinada especificamente para produzir máscaras.

Em bases moderadas ou pequenas, a transferência de aprendizado tende a trazer benefícios práticos importantes: acelera a convergência, melhora a estabilidade do treinamento e pode aumentar o desempenho final, sobretudo quando a variabilidade visual do problema é alta e a quantidade de dados rotulados é

limitada. Assim, o pré-treinamento e o *fine-tuning* constituem ferramentas fundamentais no desenvolvimento de modelos de segmentação profunda, preparando o terreno para a discussão das arquiteturas específicas empregadas em diferentes paradigmas.

Os conceitos discutidos nesta subseção fundamentam tanto tarefas de classificação quanto de segmentação. Entretanto, na segmentação o objetivo difere de maneira essencial: em vez de produzir um único rótulo para a imagem, o modelo deve gerar uma predição densa, atribuindo uma classe a cada *pixel* ou produzindo máscaras para cada instância. Isso impõe um requisito adicional: além de reconhecer padrões, a rede precisa preservar informação espacial suficiente para localizar com precisão as fronteiras entre regiões (GOODFELLOW; BENGIO; COURVILLE, 2016). Nesse contexto, surge um compromisso central: o *downsampling* progressivo, típico de CNNs, é eficiente para aumentar contexto e capturar informação semântica em camadas profundas, mas reduz a resolução e pode degradar detalhes finos, afetando contornos e estruturas pequenas. Por essa razão, arquiteturas de segmentação frequentemente introduzem mecanismos de recuperação de resolução (*upsampling*), como interpolação seguida de convoluções, convoluções transpostas ou módulos específicos de decodificação. Além disso, é comum empregar conexões de atalho que combinam características de camadas rasas (mais detalhadas) com características profundas (mais contextuais), buscando conciliar precisão espacial e robustez semântica.

Em síntese, a segmentação profunda pode ser vista como uma extensão natural das CNNs: aprende representações hierárquicas, mas requer mecanismos adicionais para transformar essas representações em máscaras coerentes à geometria da cena. Na sequência, essa conexão será formalizada ao apresentar a formulação do problema de segmentação como predição densa e as funções de perda e métricas mais utilizadas para treinar e avaliar modelos de segmentação.

2.4.2 Formulação, funções de perda e métricas

Uma vez estabelecidos os fundamentos de CNNs, é necessário formalizar como o problema de segmentação é representado e como se definem os critérios que orientam o treinamento e a avaliação dos modelos. Conforme visto, no contexto da

segmentação, a saída desejada não é um único rótulo por imagem, mas um conjunto de rótulos alinhado espacialmente à entrada, de modo que cada *pixel* seja associado a uma classe de interesse. Essa particularidade exige tratar explicitamente a forma das saídas, as funções de ativação utilizadas na etapa final (por exemplo, $ReLU$) e a relação entre valores intermediários (*logits*) e probabilidades da predição.

Além disso, o treinamento supervisionado depende da escolha de uma função de perda que penalize discrepâncias entre a segmentação prevista e a anotação de referência, enquanto a comparação de desempenho entre modelos requer **métricas** capazes de quantificar a qualidade de sobreposição e os erros de classificação em nível de *pixel*. Embora perdas e métricas frequentemente se baseiem em conceitos semelhantes (como interseção e união), elas desempenham papéis distintos: a perda é o critério otimizado durante o treinamento, ao passo que as métricas são usadas principalmente para monitoramento e comparação de desempenho (LONG; SHELHAMER; DARRELL, 2015).

Na sequência, apresenta-se a formulação matemática da predição densa e das representações usuais de máscaras, descrevem-se funções de perda amplamente empregadas no treinamento de modelos de segmentação. Por fim, são introduzidas métricas relacionadas, que serão retomadas posteriormente na etapa de avaliação dos modelos desenvolvidos nesse estudo.

2.4.2.1 Predição densa e representação das saídas

Pode-se simplificar o objetivo do modelo de segmentação como a produção de uma predição densa, isto é, uma saída alinhada espacialmente à entrada. Considerando uma imagem de entrada representada por um tensor $\boldsymbol{X}$ (conforme visto definido na Seção 2.4.1.2), modelo de segmentação implementa uma função parametrizada $f_{\boldsymbol{\theta}}$ que mapeia essa entrada para um conjunto de mapas de saída, os chamados *logits*, valores reais não normalizados que representam, para cada *pixel* (i, j) e classe k, a evidência predita pela rede antes da conversão em probabilidades.

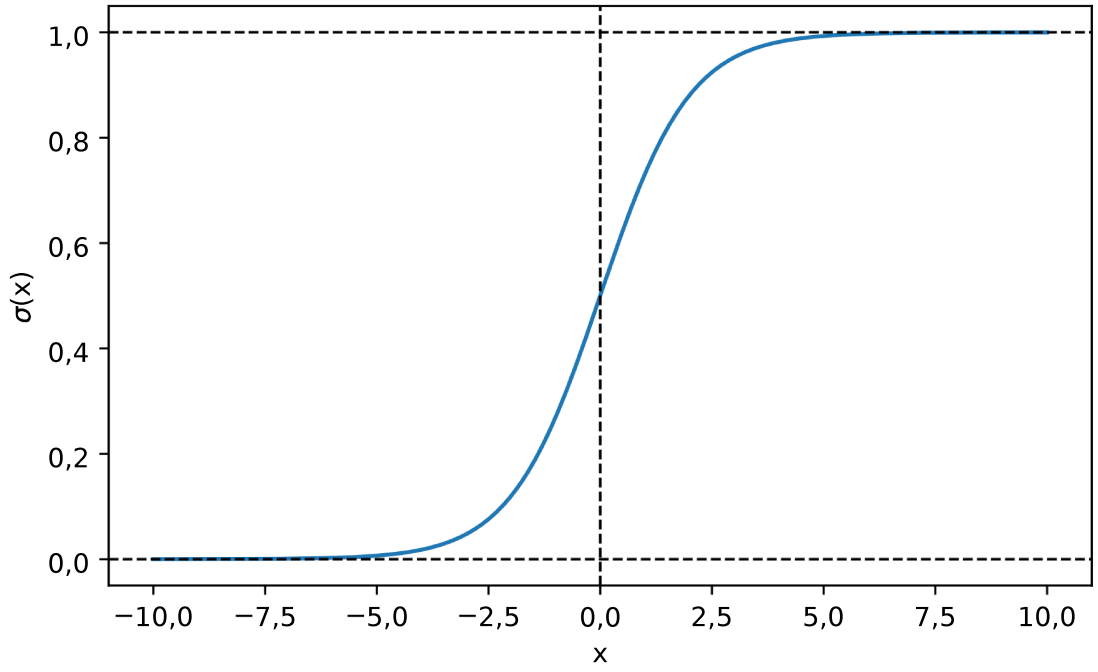
A conversão de *logits* em probabilidades depende da natureza do problema. Na segmentação binária, em que cada *pixel* é classificado como pertencente ou não a uma classe de interesse, utiliza-se tipicamente a função *Sigmoide*, definida

por (2.28) e representada da Figura 16, resultando em um mapa $\hat{\boldsymbol{P}} \in [0,1]^{H \times W}$, no qual $\hat{p}(i,j)$ pode ser interpretado como a probabilidade do *pixel* pertencer à classe positiva (LONG; SHELHAMER; DARRELL, 2015).

$$\hat{p}(i,j) = \sigma(z(i,j)) = \frac{1}{1+e^{-z(i,j)}}, \tag{2.28}$$

em que $z(i,j)$ é o valor de entrada da função *Sigmoide* na posição (i,j).

Figura 16 – Exemplo do comportamento da função *Sigmoide*.



Fonte – Do autor.

Na segmentação multiclasse (mutuamente exclusiva), isto é, quando cada pixel pertence a apenas uma dentre K classes, utiliza-se a função *softmax* ao longo do eixo de classes:

$$\hat{p}_k(i,j) = \frac{\exp(z_k(i,j))}{\sum_{c=1}^{K} \exp(z_c(i,j))}, \tag{2.29}$$

produzindo $\hat{\boldsymbol{P}} \in [0,1]^{H \times W \times K}$ com a restrição $\sum_{k=1}^{K} \hat{p}_k(i,j) = 1$ para cada *pixel*. A predição final de rótulos pode ser obtida por limiarização (no caso binário) ou

por $\arg\max$ (no caso multiclasse), resultando em uma máscara $\hat{\boldsymbol{Y}} \in \{1,\ldots,K\}^{H\times W}$ (LONG; SHELHAMER; DARRELL, 2015).

As anotações de referência (ou *ground truth*) são usualmente representadas como máscaras por *pixel*. No caso binário, a máscara pode ser escrita como

$$\boldsymbol{Y} \in \{0,1\}^{H\times W}, \tag{2.30}$$

enquanto no caso multiclasse é comum utilizar uma codificação chamada de *one-hot*, definida como:

$$\boldsymbol{Y} \in \{0,1\}^{H\times W\times K}, \qquad \sum_{k=1}^{K} y_k(i,j) = 1, \tag{2.31}$$

em que $y_k(i,j) = 1$ indica que o *pixel* (i,j) pertence à classe k (LONG; SHELHAMER; DARRELL, 2015).

Uma vez estabelecida a representação das saídas e a forma como probabilidades são obtidas a partir dos *logits*, pode-se definir funções de perda que quantificam a discrepância entre predição e referência e orientam o ajuste dos parâmetros durante o treinamento, tema da subseção seguinte.

2.4.2.2 Funções de perda para treinamento de modelos de segmentação

Conforme visto, no treinamento supervisionado, a função de perda define o critério numérico que o otimizador busca minimizar, quantificando a discrepância entre a predição do modelo e a máscara de referência. Em segmentação, perdas usuais podem ser agrupadas em duas famílias principais: (i) perdas baseadas em classificação por *pixel*, como a entropia cruzada; e (ii) perdas baseadas em sobreposição de regiões, como Dice e *Intersection over Union* (IoU). Em muitos casos, combinações dessas perdas são empregadas para conciliar e avaliar estabilidade de otimização e qualidade de contorno.

Considerando probabilidades preditas $p_k(i,j)$ obtidas por *softmax* e rótulos *one-hot* $y_k(i,j)$, a função de perda denominada entropia cruzada por *pixel* é definida como

$$\mathscr{L}_{\mathrm{CE}} = -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{K} y_k(i,j)\log(p_k(i,j))\,. \tag{2.32}$$

Para segmentação binária ($K = 1$), com probabilidade $p(i, j)$ e rótulo $y(i, j) \in \{0, 1\}$, tem-se a forma conhecida como entropia cruzada binária:

$$\mathscr{L}_{\text{BCE}} = -\frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \Big[y(i,j) \log(p(i,j)) + (1 - y(i,j)) \log(1 - p(i,j)) \Big]. \quad (2.33)$$

A entropia cruzada é amplamente utilizada por ter interpretação probabilística direta e por fornecer gradientes bem comportados, tratando a segmentação como um problema de classificação em nível de *pixel*. Entretanto, quando as classes são muito desbalanceadas (i.e. quando há muito mais pixels no banco de dados de treinamento de uma classe do que de outra), a otimização pode ser dominada pela classe majoritária, motivando versões ponderadas e/ou perdas alternativas (SZELISKI, 2022; LONG; SHELHAMER; DARRELL, 2015).

Perdas baseadas em sobreposição procuram alinhar diretamente a predição com a máscara de referência, sendo menos sensíveis ao desbalanceamento. Para o caso binário, definindo o coeficiente **Dice** na forma "suavizada" (com $\varepsilon > 0$), tem-se

$$\text{Dice}(\{y\}, \{\hat{p}\}) = \frac{2 \sum_{i,j} y(i,j) p(i,j) + \varepsilon}{\sum_{i,j} y(i,j) + \sum_{i,j} p(i,j) + \varepsilon}, \quad (2.34)$$

a perda correspondente pode ser escrita como

$$\mathscr{L}_{\text{Dice}} = 1 - \text{Dice}(\{y\}, \{p\}). \quad (2.35)$$

Em segmentação multiclasse, é comum calcular **Dice** por classe e então ponderar entre classes. A principal vantagem da função de perda **Dice** é a penalização direta de erros de sobreposição (falsos positivos e falsos negativos) e tende a funcionar bem quando a região de interesse ocupa pequena fração da imagem (EVERINGHAM *et al.*, 2010).

Outra medida de sobreposição amplamente usada é a **IoU**, conhecida como "índice de Jaccard" (JACCARD, 1901). Para o caso binário, na forma contínua, tem-se que

$$\text{IoU}(\{y\}, \{\hat{p}\}) = \frac{\sum_{i,j} y(i,j) p(i,j) + \varepsilon}{\sum_{i,j} y(i,j) + \sum_{i,j} p(i,j) - \sum_{i,j} y(i,j) p(i,j) + \varepsilon}, \quad (2.36)$$

e a perda correspondente

$$\mathscr{L}_{\text{IoU}} = 1 - \text{IoU}(\{y\}, \{p\}). \tag{2.37}$$

De modo geral, **IoU** tende a ser mais "rígido" do que **Dice**, pois penaliza fortemente discrepâncias relativas entre regiões previstas e reais, sendo frequentemente utilizado como métrica principal de avaliação em tarefas de segmentação (EVERINGHAM *et al.*, 2010).

Na prática, é comum combinar perdas complementares para obter treinamento mais estável e boa qualidade de segmentação. Uma combinação frequente é

$$\mathscr{L} = \alpha \mathscr{L}_{\text{CE}} + (1 - \alpha) \mathscr{L}_{\text{Dice}}, \tag{2.38}$$

em que $\alpha \in [0, 1]$ controla o compromisso entre um termo de classificação por *pixel* e um termo de sobreposição. Combinações análogas podem ser feitas com **IoU**, ou ainda com "versões" da entropia cruzada (EVERINGHAM *et al.*, 2010).

A escolha da função de perda a ser minimizada influencia fortemente o comportamento do treinamento, especialmente quando há desbalanceamento entre classes ou quando regiões de interesse são pequenas. Esses aspectos motivam perdas ponderadas e perdas robustas, discutidas na subseção seguinte.

2.4.2.3 Métricas de desempenho

A função de perda é o critério otimizado durante o treinamento, isto é, define o gradiente usado na atualização dos pesos (ou parâmetros) do modelo de IA. Já as métricas de desempenho são utilizadas principalmente para monitorar desempenho e comparar modelos. Em tarefas de segmentação, tais métricas são geralmente definidas a partir da comparação entre a máscara prevista e a máscara de referência em nível de *pixel*.

Considerando uma classe de interesse e uma máscara prevista binária $\hat{\boldsymbol{Y}}$, e a máscara de referência $\boldsymbol{Y}$. Define-se, então, a contagem de *pixels* como: verdadeiros positivos (TP – *True Positive*), falsos positivos (FP – *False Positive*), falsos negativos (FN – *False Negative*) e verdadeiros negativos (TN – *True Negative*). A partir desses termos, métricas clássicas de classificação podem ser computadas,

conforme

$$\textbf{Precisão} = \frac{TP}{TP+FP}, \tag{2.39}$$

$$\textit{Recall} = \frac{TP}{TP+FN}, \tag{2.40}$$

$$\text{F1} = \frac{2 \cdot \textbf{Precisão} \cdot \textit{Recall}}{\textbf{Precisão} + \textit{Recall}}. \tag{2.41}$$

Essas métricas ajudam a interpretar o tipo de erro predominante: baixa precisão indica excesso de falsos positivos, isto é, o modelo está classificando como pertencentes à classe de interesse regiões que não deveriam ser incluídas (supersegmentação). Em contrapartida, baixo *recall* indica muitos falsos negativos, ou seja, partes da região verdadeira não estão sendo detectadas pelo modelo (subsegmentação). Assim, a análise conjunta de precisão e *recall* permite distinguir se o erro está associado principalmente a "vazar" para áreas indevidas (chamados de erros de comissão) ou a "perder" partes relevantes do objeto (chamados de erros de omissão) (EVERINGHAM *et al.*, 2010). Além dessas métricas, medidas de sobreposição são particularmente importantes, como **Dice** (que é $F1$ escrito em termos de conjuntos) e **IoU**, pois quantificam diretamente o alinhamento geométrico entre regiões. Tanto a **IoU** quanto o **Dice** podem ser definidos em função de TP, FP e FN da seguinte forma:

$$\text{IoU} = \frac{|\hat{Y} \cap Y|}{|\hat{Y} \cup Y|} = \frac{TP}{TP+FP+FN}, \tag{2.42}$$

$$\text{Dice} = \frac{2|\hat{Y} \cap Y|}{|\hat{Y}| + |Y|} = \frac{2TP}{2TP+FP+FN}. \tag{2.43}$$

Ambas variam entre 0 e 1, sendo 1 o caso de sobreposição perfeita. Em problemas multiclasse, é comum que essas métricas sejam calculadas por classe e então sejam calculadas as médias (simples ou ponderada), de modo a evitar que classes dominantes ocultem o desempenho em classes minoritárias. Têm-se, portanto, o **mIoU** (*mean IoU*) e o **mDice** (*mean Dice*).

Os conceitos apresentados até aqui estabelecem uma base comum do aprendizado profundo para segmentação: a predição densa, as representações de saída, as principais funções de perda e métricas de desempenho. Entretanto, modelos de

segmentação diferem substancialmente na forma como organizam esses elementos arquiteturais, em particular, na maneira de equilibrar extração de contexto (camadas profundas e *downsampling*) e preservação de detalhes espaciais (recuperação de resolução e conexões de atalho), bem como na forma como produzem a saída (segmentação semântica, por instância ou modelos fundacionais com *prompts*). Assim, a próxima subseção apresenta um panorama das arquiteturas e paradigmas de segmentação considerados neste trabalho para a tarefa de segmentação das áreas frágeis e de delaminação da SF, destacando diferenças conceituais e implicações práticas.

2.4.3 Arquiteturas e paradigmas de segmentação profunda

Embora a formulação do problema, as funções de perda e as métricas estabeleçam uma base comum para o treinamento de modelos de segmentação, o desempenho prático depende fortemente de como a arquitetura constrói e combina representações ao longo das camadas. Em particular, diferentes famílias de modelos adotam estratégias distintas para lidar com o compromisso entre capturar contexto e informações de escala mais global e preservar detalhes espaciais necessários para delimitar contornos e estruturas finas (RONNEBERGER; FISCHER; BROX, 2015; CHEN *et al.*, 2017). Nesse cenário, a comparação entre arquiteturas não implica simplesmente na escolha da "rede mais profunda", mas na compreensão dos diferentes paradigmas de projeto e suas implicações.

Modelos do tipo *encoder–decoder* com *skip connections* (como a *U-Net*) enfatizam localização precisa ao reutilizar características rasas (RONNEBERGER; FISCHER; BROX, 2015); arquiteturas que exploram convoluções dilatadas e agregação multi-escala (como *DeepLabV3*) buscam aumentar o campo receptivo e modelar variações de escala sem reduzir agressivamente a resolução (CHEN *et al.*, 2017); abordagens *one-stage* que integram detecção e segmentação (como variantes *YOLO* com segmentação) produzem predições associadas a classes e escores, tradicionalmente no nível de instâncias, podendo ser adaptadas para saídas semânticas por classe quando o interesse é a máscara final (KHANAM; HUSSAIN, 2024); e, mais recentemente, modelos baseados em *Transformers* (como o *SegFormer*) utilizam codificadores hierárquicos e decodificadores leves para combinar informação

local e global de forma eficiente (XIE *et al.*, 2021).

Assim, esta subseção apresenta as principais características e diferenças entre esses paradigmas, tomando como referência arquiteturas representativas utilizadas no desenvolvimento deste trabalho. O objetivo é explicitar como cada arquitetura organiza a extração de características, o tratamento de multi-escala e a preservação de detalhe espacial, preparando o terreno para o ponto central do trabalho: uma discussão comparativa de desempenho desse modelos em uma mesma tarefa.

2.4.3.1 Famílias de arquiteturas para segmentação

De maneira geral, arquiteturas de segmentação profunda podem ser organizadas em famílias conforme a estratégia de extração de características e agregação de contexto e o mecanismo empregado para recuperação de resolução e preservação de detalhes. Embora essas famílias compartilhem a mesma formulação básica, já discutida na Subseção 2.4.1, isto é, aprender um mapeamento que atribui rótulos por *pixel*, elas diferem no "como" combinam informação local e global ao longo do processamento (GOODFELLOW; BENGIO; COURVILLE, 2016; LONG; SHELHAMER; DARRELL, 2015).

Uma primeira família, bastante influente, é a dos modelos do tipo *encoder–decoder* com conexões de atalho (*skip connections*). Nesses modelos, o *encoder* reduz progressivamente a resolução espacial para extrair representações mais ricas e contextuais, enquanto o *decoder* recupera a resolução por *upsampling*. As conexões de atalho entre níveis correspondentes permitem reutilizar características rasas (mais detalhadas), favorecendo a delimitação de contornos e estruturas finas; a *U-Net* é o exemplo mais clássico dessa filosofia (RONNEBERGER; FISCHER; BROX, 2015).

Uma segunda família, também amplamente utilizada, enfatiza explicitamente o contexto multi-escala por meio de convoluções dilatadas (*atrous*) e mecanismos de agregação em múltiplas taxas de amostragem, buscando aumentar o campo receptivo sem reduzir agressivamente a resolução, ou seja, intuitivamente, permitir que cada ativação "enxergue" uma região maior da imagem (incorporando

mais contexto) sem que seja necessário aplicar sucessivas etapas de *downsampling*. O *DeepLabV3* é representativo desse paradigma ao empregar o módulo *Atrous Spatial Pyramid Pooling* (ASPP) para combinar contexto em diferentes escalas (CHEN *et al.*, 2017).

Uma terceira família reúne arquiteturas *one-stage* que integram detecção e segmentação, produzindo máscaras associadas a classes e escores de confiança, tipicamente no nível de instâncias. Essa abordagem é atrativa por unificar localização e segmentação em uma única passagem (por isso, a sigla "*YOLO*" se refere à expressão "*You only look once*", ou "Você só olha uma vez") e por ser computacionalmente eficiente. Entretanto, quando o objetivo final é uma máscara semântica por classe (isto é, sem distinguir instâncias), as saídas por instância podem ser agregadas por classe (por exemplo, por união das máscaras previstas), obtendo-se uma representação diretamente compatível com a avaliação semântica (KHANAM; HUSSAIN, 2024). Por fim, uma quarta família, mais recente, compreende modelos baseados em *Transformers* para segmentação, nos quais mecanismos de atenção possibilitam combinar informação local e global de maneira flexível. O *SegFormer* é um exemplo relevante: utiliza um codificador *Transformer* hierárquico (gerando características multi-escala) e um *decoder* leve (*multilayer perceptron* – MLP) para agregar essas escalas e produzir a predição (XIE *et al.*, 2021).

A seguir, essas famílias são melhor abordadas a partir de arquiteturas representativas, destacando como suas escolhas estruturais se relacionam com os compromissos fundamentais de segmentação já discutidos.

2.4.3.2 U-Net: encoder–decoder com conexões de atalho

A *U-Net* é uma das arquiteturas mais influentes para segmentação semântica, proposta originalmente no contexto de imagens biomédicas e posteriormente amplamente adotada em diversas aplicações de visão computacional. Seu princípio central é combinar um caminho contrator (*encoder*), responsável por extrair representações progressivamente mais abstratas e contextuais por meio de convoluções e operações de *downsampling*, com um caminho expansor (*decoder*), que recupera a resolução espacial para produzir uma predição densa. Diferentemente de arquiteturas puramente *feedforward* para classificação, a *U-Net* foi desenhada

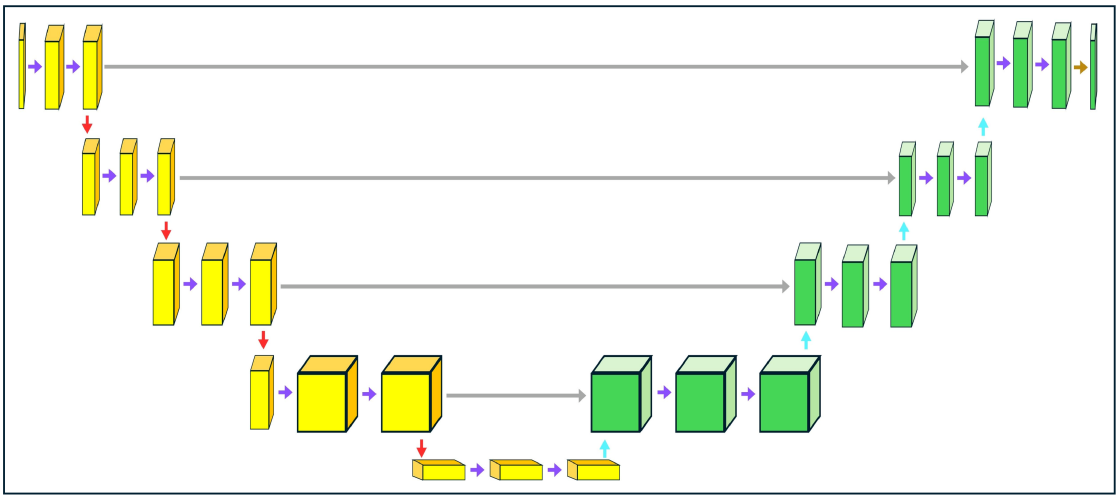
para manter informação espacial suficiente para delinear contornos com precisão, equilibrando contexto e detalhe local (RONNEBERGER; FISCHER; BROX, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016).

No *encoder*, a imagem de entrada é processada por blocos convolucionais (tipicamente duas convoluções seguidas de ativação e, frequentemente, normalização) e então submetida a uma etapa de redução espacial (por exemplo, *max pooling* ou convolução com *stride* maior que 1). A cada nível, a resolução espacial diminui, enquanto o número de canais tende a aumentar, permitindo que a rede represente padrões de maior complexidade e incorpore um campo receptivo efetivo mais amplo. Já no *decoder*, realiza-se o processo inverso: aplica-se *upsampling* (por interpolação ou convolução transposta) para aumentar a resolução e, em seguida, convoluções refinam a representação reconstruída, aproximando-a da geometria da entrada (RONNEBERGER; FISCHER; BROX, 2015).

O elemento distintivo da *U-Net* são as conexões de atalho entre níveis correspondentes do *encoder* e do *decoder*. Em vez de depender apenas das ativações profundas, a *U-Net* combina mapas de características de camadas rasas do *encoder* com as ativações em resolução compatível no *decoder*. Intuitivamente, essas conexões reintroduzem detalhes espaciais (bordas, texturas e informações de contorno) que tendem a se degradar com o *downsampling*, permitindo que a saída preserve melhor limites entre regiões e estruturas finas (RONNEBERGER; FISCHER; BROX, 2015). Do ponto de vista do treinamento, as conexões também favorecem a propagação de gradientes e reduzem a dificuldade de otimização em redes mais profundas (GOODFELLOW; BENGIO; COURVILLE, 2016).

A camada final da *U-Net* é tipicamente uma convolução 1×1 que projeta os canais para o número de classes $\boldsymbol{K}$, produzindo *logits* por *pixel* e, após *softmax* (multiclasse) ou *sigmoid* (binário/multirrótulo), obtém-se a probabilidade associada a cada classe por posição (RONNEBERGER; FISCHER; BROX, 2015). Por esse motivo, a *U-Net* se integra naturalmente às funções de perda discutidas anteriormente (por exemplo, entropia cruzada, Dice/IoU e combinações), e sua modularidade facilita adaptações de profundidade, largura (número de canais) e mecanismos de *upsampling*. Um diagrama esquemático da estrutura da *U-Net* é apresentado na Figura 17.

Figura 17 – Diagrama esquemático da estrutura da rede *U-Net*. Setas em vermelho representam o caminho contrator; setas em azul representam o caminho expansor; setas cinza indicam as conexões de atalho por concatenação entre níveis de mesma escala; setas roxas e marrom representam, respectivamente, a operação de convolução 3×3 e 1×1. Blocos amarelos correspondem aos mapas de características no *encoder* e blocos verdes aos mapas no *decoder*; a camada final realiza a projeção para o mapa de saída.



Fonte – Do autor. Adaptado de Wei *et al.* (2019).

Entre os pontos fortes da *U-Net* destacam-se: (i) boa capacidade de localizar contornos devido às *skip connections*; (ii) desempenho competitivo em bases moderadas, especialmente quando combinado com aumento de dados; e (iii) simplicidade estrutural, que a torna um *baseline* robusto e amplamente reprodutível (RONNEBERGER; FISCHER; BROX, 2015). Como limitações, pode-se mencionar que a *U-Net* clássica depende de múltiplas reduções de resolução para incorporar contexto global, o que pode não ser ideal quando a distinção entre classes exige informação de longo alcance; além disso, em altas resoluções, a concatenação de características pode elevar o custo de memória no *decoder*. Essas limitações motivaram extensões e variações (por exemplo, modificações no *encoder*, atenção e módulos multi-escala), mas o princípio *encoder–decoder* com conexões de atalho permanece como um dos paradigmas fundamentais da segmentação profunda.

2.4.3.3 DeepLabV3 com estrutura de base ResNet50: convoluções atrous e ASPP

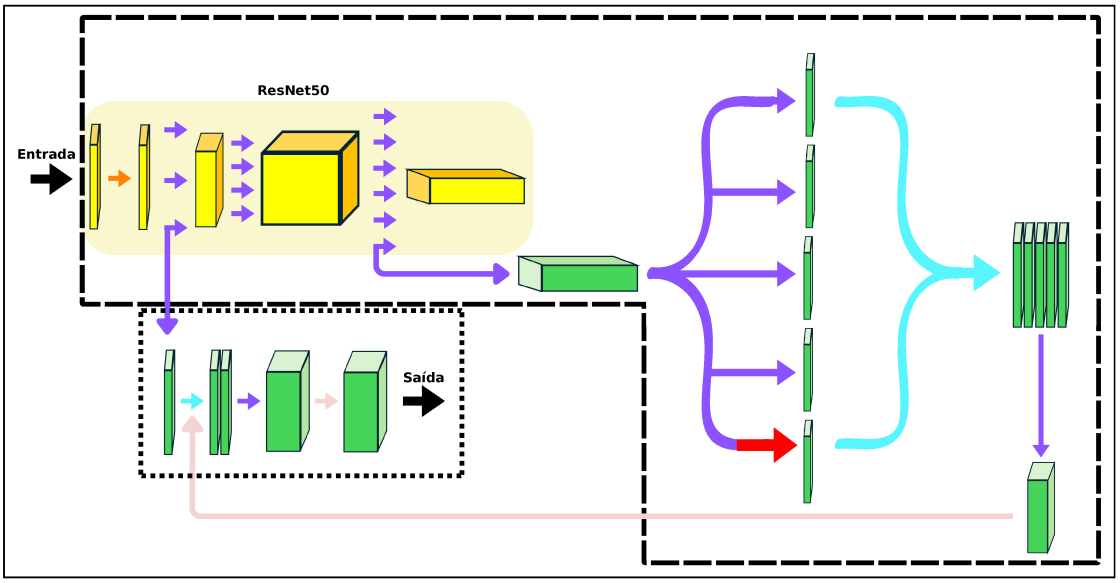
O *DeepLabV3* é uma arquitetura de segmentação semântica projetada para aumentar a capacidade do modelo de incorporar contexto e lidar com variações de escala sem recorrer a reduções agressivas de resolução espacial (CHEN *et al.*, 2017). Diferentemente de abordagens *encoder–decoder* clássicas, nas quais a recuperação de detalhes depende fortemente de um decodificador explícito, o *DeepLabV3* enfatiza a extração de representações ricas e multi-escala no estágio de codificação, combinando convoluções dilatadas com um módulo de agregação de contexto em múltiplas taxas (ASPP). No presente trabalho, o modelo é considerado na forma "*DeepLabV3* com estrutura de base *ResNet50*", em que a *ResNet50* atua como extratora de características e a cabeça *DeepLabV3* produz o mapa de predição denso (HE *et al.*, 2016; CHEN *et al.*, 2017).

A *ResNet50* é uma CNN baseada em blocos residuais, nos quais conexões de atalho somam a entrada do bloco à sua saída. Essa estratégia facilita a otimização e permite treinar redes com maior profundidade sem degradação de desempenho, sendo amplamente utilizada como estrutura de base em segmentação, detecção e reconhecimento (HE *et al.*, 2016). No contexto do *DeepLabV3*, a *ResNet50* fornece mapas de características em níveis sucessivos, e parte das convoluções mais profundas pode ser modificada para operar com convoluções dilatadas (isto é, com dilatação), de modo a manter maior resolução espacial no mapa final do *encoder* ao mesmo tempo em que se preserva o campo receptivo efetivo (CHEN *et al.*, 2017). Intuitivamente, ao substituir *strides* por dilatação em camadas profundas, evita-se que a malha de ativações se torne excessivamente grossa, reduzindo perdas de detalhe que seriam difíceis de recuperar posteriormente.

O componente característico do *DeepLabV3* é o ASPP. Esse módulo aplica múltiplas convoluções em paralelo com diferentes taxas de dilatação, além de um caminho de agregação global (por exemplo, *global average pooling*), e então combina essas respostas para formar uma representação que incorpora evidências em diferentes escalas (CHEN *et al.*, 2017). Assim, o modelo pode responder a estruturas com tamanhos variados, mitigando o problema de que um único campo receptivo (ou uma única escala) pode ser insuficiente para segmentar objetos grandes e pequenos de maneira consistente. Após o ASPP, uma camada de classificação

(normalmente uma convolução 1×1) projeta os canais para K classes, resultando em *logits* por *pixel*; a máscara final é obtida por *softmax* (multiclasse) ou *Sigmoide* (casos binários/multirrótulo), analogamente ao discutido em 2.4.2.1. Um diagrama esquemático da estrutura da *DeepLabV3–ResNet50* é apresentado na Figura 18.

Figura 18 – Diagrama esquemático da estrutura da rede *DeepLabV3* com estrutura de base *ResNet50*. Setas roxas representam operação de convolução; setas em azul representam concatenação; setas vermelhas e laranja representam, respectivamente, *pooling* e *max pooling*; setas rosas representam operação de *upsampling*; setas pretas representam entrada e saída da rede; o bloco amarelo representa a estrutura de base *ResNet50*; entre as chaves, a representação do ASPP. O conjunto tracejado e pontilhados correspondem, respectivamente, ao *encoder* e *decoder*.



Fonte – Do autor.

Entre as vantagens do *DeepLabV3* destacam-se: (i) forte modelagem de contexto e multi-escala via ASPP; (ii) possibilidade de manter maior resolução no *encoder* com convoluções dilatadas; e (iii) uso de estrutura de base bem estabelecida, que podem ser pré-treinados e transferidos para tarefas com bases menores (HE *et al.*, 2016; CHEN *et al.*, 2017). Como limitações, pode-se mencionar que, como não há um decodificador explícito, a recuperação de contornos finos pode ser mais desafiadora em cenários onde as fronteiras são críticas; além disso, o custo

computacional pode aumentar com a preservação de resoluções mais altas e com múltiplos ramos no ASPP. Ainda assim, o *DeepLabV3–ResNet50* constitui uma referência consolidada para segmentação semântica, representando um paradigma distinto da *U-Net* ao enfatizar principalmente contexto multi-escala no estágio de codificação.

2.4.3.4 YOLO11l-seg: paradigma one-stage com segmentação e adaptação semântica

A família *YOLO* consolidou o paradigma de *one-stage*, no qual a localização e a classificação de objetos são preditas diretamente em uma única passagem pela rede, em contraste com abordagens *two-stage* que separam geração de propostas e refinamento. De forma geral, essa arquitetura privilegia eficiência computacional e baixa latência, tornando-se atrativo em cenários em que velocidade e simplicidade do processamento são desejáveis (REDMON *et al.*, 2016).

Nas variantes modernas com segmentação (como o *YOLO11l-seg*), o modelo estende a predição clássica de detecção (caixas delimitadoras e escores de classe) para também produzir, para cada detecção, uma máscara associada ao objeto (segmentação por instância). Assim, a saída do modelo pode ser interpretada como um conjunto de instâncias $\{(\{b\}_i, c_i, s_i, \boldsymbol{M}_i)\}_{i=1}^{N}$, em que $\{b\}_i$ representa a caixa, c_i a classe prevista, s_i uma pontuação de confiança e $\boldsymbol{M}_i$ a máscara da i-ésima instância (Ultralytics, 2024b; Ultralytics, 2024a). Esse tipo de predição é especialmente útil quando é necessário separar múltiplas ocorrências de uma mesma classe e quando se deseja uma segmentação guiada pelo mecanismo de detecção.

Entretanto, em aplicações cujo objetivo final é uma máscara semântica por classe, a saída por instância pode ser convertida conceitualmente em uma rotulagem semântica por meio de agregação por classe. Em sua forma mais simples, para cada classe k, define-se uma máscara semântica $\boldsymbol{Q}_k$ como a união das máscaras das instâncias atribuídas a essa classe (após critério um limiar pré-definido de confiança), por exemplo:

$$\boldsymbol{Q}_k(p) = \bigvee_{i:\, c_i = k} \mathbb{I}\big[s_i \geq \tau\big]\, \boldsymbol{M}_i(p), \tag{2.44}$$

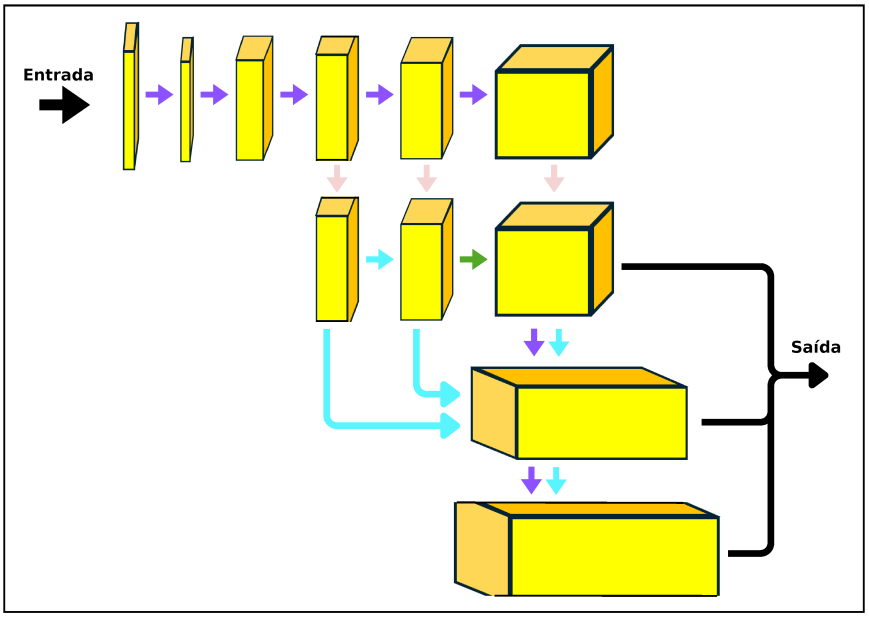
em que p denota a posição do *pixel*, τ é um limiar de confiança, $\mathbb{I}[\cdot]$ é a função indicadora e $\vee$ representa a operação lógica "OU". Alternativamente, quando se

trabalha com máscaras probabilísticas, pode-se empregar uma combinação por máximo (ou soma limitada), resultando em uma probabilidade por *pixel* para cada classe, que é então convertida em rótulo por *argmax* (no caso multiclasse). Essa adaptação permite avaliar o modelo no mesmo regime semântico discutido na Subseção 2.4.2, utilizando perdas e métricas por *pixel* (por exemplo, IoU/Dice), ao mesmo tempo em que se explora a eficiência do paradigma *one-stage* (Ultralytics, 2024a; SAPKOTA *et al.*, 2025).

Nessa arquitetura, dois blocos frequentemente citados como parte das otimizações arquiteturais são o *Spatial Pyramid Pooling Fast* (SPPF) e o *Cross Stage Partial* (CSP). O SPPF é um módulo de *pooling* em pirâmide que agrega contexto multi-escala a partir de um mesmo mapa de características, tipicamente combinando operações de *max pooling* com diferentes "alcances" efetivos para que a rede incorpore informação local e mais global sem aumentar drasticamente o custo computacional, isso ajuda o modelo a manter sensibilidade a estruturas de tamanhos distintos e a enriquecer a representação com contexto antes da predição de saída (Ultralytics, 2023; Ultralytics, 2025b). Já o CSP consiste em uma estratégia de conexão na qual o fluxo de características é particionado em dois caminhos, de modo que apenas parte das características atravessa blocos convolucionais mais profundos, sendo posteriormente recombinada (por concatenação) com o caminho de atalho. Essa configuração busca reduzir redundância computacional e favorecer o fluxo de gradientes, aumentando a eficiência e a estabilidade do treinamento, sem alterar o paradigma convolucional básico da estrutura de base e conector (também conhecido como *neck*). Um diagrama esquemático da estrutura da *YOLO11l-seg* é apresentado na Figura 19.

Como vantagens, modelos do tipo *YOLO11l-seg* tendem a oferecer boa relação entre desempenho e velocidade, com uma formulação unificada que associa naturalmente predição de classe, localização e máscara. Como limitações, destaca-se que a saída original é estruturada por instâncias e depende de critérios como limiares de confiança e supressão de duplicatas (NMS), de modo que a conversão para uma máscara estritamente semântica pode introduzir dependência adicional desses limiares e de como se resolvem sobreposições entre máscaras.

Figura 19 – Diagrama esquemático da estrutura da rede *YOLO11l-seg*. Setas roxas representam operação de convolução; setas azuis representam concatenação; setas rosas representam operação de upsampling; seta verde representa aplicação de SPPF e CSP ; setas pretas representam entrada e saída da rede.



Fonte – Do autor.

2.4.3.5 SegFormer: Transformer hierárquico para segmentação semântica

O *SegFormer* é uma arquitetura de segmentação semântica baseada em *Transformers*, proposta com o objetivo de combinar a capacidade de modelar dependências de longo alcance (característica de mecanismos de atenção) com um desenho simples e eficiente para uso prático em segmentação. Diferentemente de CNNs puramente convolucionais, nas quais o campo receptivo cresce gradualmente com a profundidade e com operações de *downsampling*, *Transformers* permitem que a representação em uma posição incorpore informação de outras regiões da imagem de maneira mais direta, por meio de atenção. Em visão computacional, essa propriedade é particularmente atrativa quando o rótulo de um *pixel* depende do contexto ao redor ou de relações em escala mais ampla (XIE *et al.*, 2021).

O núcleo do *SegFormer* é um codificador *Transformer* hierárquico denominado MiT (*Mix Transformer*). Esse codificador produz mapas de características

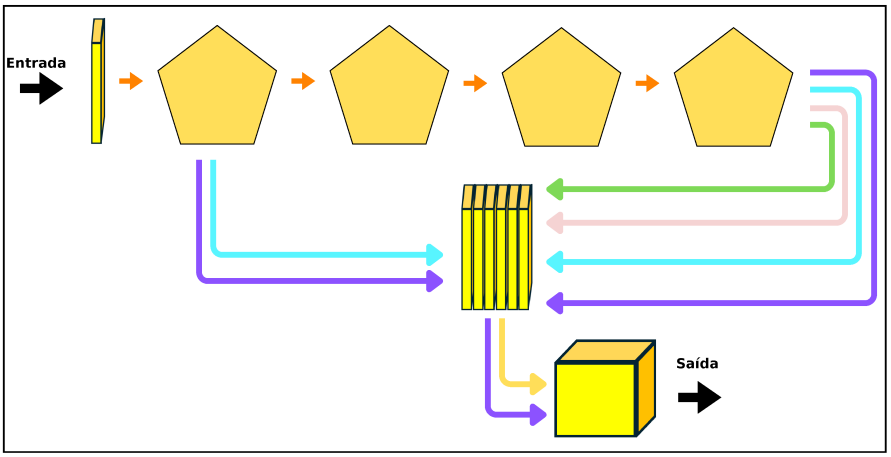
em múltiplas resoluções (multi-escala), em um arranjo análogo ao de *stages* em CNNs: camadas iniciais operam com resolução mais alta e capturam padrões mais locais, enquanto camadas mais profundas produzem representações mais compactas e contextuais. Para tornar o custo computacional viável em imagens, o MiT emprega mecanismos de atenção mais eficientes do que a atenção global, reduzindo a complexidade ao trabalhar com representações espacialmente reduzidas e/ou com estratégias de mistura (*mixing*) que preservam informação local e global de forma balanceada. Assim, o modelo obtém uma representação rica em contexto sem exigir um decodificador pesado para agregar escalas (XIE *et al.*, 2021).

A etapa de decodificação no *SegFormer* é deliberadamente simples: em vez de um *decoder* profundo com múltiplos blocos convolucionais, utiliza-se um decodificador leve baseado em MLP que projeta os mapas multi-escala para uma dimensão comum, realiza sua fusão (após reamostragem para uma resolução compatível) e produz os *logits* por *pixel*. A predição final segue o mesmo princípio discutido na Subseção 2.4.2.1: obtêm-se *logits* por *pixel* e aplica-se *softmax* (multiclasse) ou *Sigmoide* (binário/multirrótulo), treinando-se com perdas adequadas e avaliando-se com métricas como **IoU** e **Dice**.

Um diagrama esquemático da estrutura da *SegFormer* é apresentado na Figura 20.

Entre os pontos fortes do *SegFormer* destacam-se: (i) boa capacidade de capturar contexto global e relações em diferentes escalas, graças ao codificador *Transformer* hierárquico; (ii) decodificador leve, favorecendo eficiência; e (iii) flexibilidade para transferência de aprendizado e ajuste fino, uma vez que variantes do MiT podem ser pré-treinadas em bases genéricas e adaptadas a domínios específicos (XIE *et al.*, 2021). Como limitações, modelos baseados em atenção podem ser mais sensíveis a configurações de treinamento e, em alguns cenários, demandar maior cuidado com pré-treinamento e regularização para alcançar desempenho estável em bases menores; além disso, apesar de mais eficiente do que *Transformers* globais, o custo computacional do mecanismo de atenção pode crescer com a resolução, exigindo escolhas cuidadosas de tamanho de entrada e variante do modelo.

Figura 20 – Diagrama esquemático da estrutura da rede *SegFormer*. Setas roxas representam operação de convolução; seta rosa representa operação de *upsampling*; seta verde representa o *decoder*; seta amarela representa a operação de convolução transposta; as setas laranjas e os pentágonos representam, respectivamente, o módulo *Transformer* e sua saída; setas pretas representam entrada e saída da rede.



Fonte – Do autor.

Em conjunto, as arquiteturas discutidas ilustram quatro paradigmas complementares de segmentação. Como esses modelos diferem na forma de equilibrar contexto global e preservação de detalhes espaciais, é esperado que apresentem comportamentos distintos na delimitação de contornos, na segmentação de regiões pequenas e na robustez a variações de aparência. A seguir, essas diferenças são consolidadas em uma síntese comparativa, destacando as principais implicações conceituais de cada arquitetura e preparando o terreno para a definição do protocolo experimental e das métricas de avaliação.

2.4.3.6 Síntese comparativa dos paradigmas e modelos selecionados

As arquiteturas apresentadas representam escolhas deliberadamente heterogêneas dentro do problema de segmentação semântica: elas diferem no mecanismo principal de captura de contexto (convolução com atalhos, convolução dilatada multi-escala, paradigma *one-stage* orientado à detecção e atenção baseada em *Transformer*), e também na forma como preservam e recuperam detalhe espa-

cial, sintetizadas na Tabela 1. Essa diversidade é desejável porque, ao controlar o protocolo de treinamento e avaliação, torna-se possível isolar como decisões arquiteturais afetam a qualidade de contornos, a robustez a variações de aparência e o comportamento em regiões pequenas.

Tabela 1 – Síntese comparativa dos quatro modelos selecionados, destacando paradigmas arquiteturais, mecanismos de captura de contexto/multi-escala e implicações conceituais para segmentação semântica.

Modelo	**Paradigma**	**Contexto / multi-escala**	**Preservação de detalhe**	**Saída nativa**	**Implicações conceituais (pontos fortes / limitações)**
U-Net	*Encoder–decoder* com *skip connections*	Contexto cresce via *downsampling* e profundidade; combina escalas ao fundir características rasas e profundas	Forte preservação de contornos por concatenação de características do *encoder* no *decoder*	Semântica por *pixel* (logits por classe)	*Forte*: boa localização e contornos finos; bom *baseline* e alta reprodutibilidade. *Limite*: contexto global depende do grau de compressão; pode exigir maior profundidade para estruturas de longo alcance; custo/memória no *decoder* ao concatenar mapas.
DeepLabV3–ResNet50	*Estrutura de base* residual + cabeçote de segmentação	*Atrous* para manter resolução + ASPP para agregar contexto em múltiplas taxas (multi-escala explícita)	Preserva resolução do *encoder* (via dilatação), mas sem *decoder* dedicado pode suavizar contornos	Semântica por *pixel* (*logits* por classe)	*Forte*: contexto multi-escala robusto; bom compromisso entre resolução e campo receptivo. *Limite*: sem módulo de refinamento dedicado, bordas muito finas podem degradar; custo pode aumentar ao manter mapas em alta resolução e múltiplos ramos no ASPP.
YOLO11l-seg	*One-stage* (detecção + máscara por instância)	Contexto via estrutura de base/conector + fusão multi-escala típica de detectores; predições orientadas a instâncias	Detalhe de máscara depende do cabeçote de segmentação e do mecanismo de detecção (limiares, NMS, etc.)	Instâncias (máscaras + escores); pode ser agregado por classe para uso semântico	*Forte*: eficiência/velocidade e processamento unificado; tende a ser robusto quando a detecção é estável. *Limite*: paradigma nativo é por instância; a passagem para semântica depende de regras de agregação e limiares (confiança/overlap), que podem impactar contornos e regiões pequenas; erros de detecção podem "cascatear" para a máscara.
SegFormer	*Transformer* hierárquico (MiT) + *decoder* MLP leve	Atenção favorece dependências de longo alcance; codificador multi-escala hierárquico (análoga a *stages*)	Detalhe depende da qualidade das características multi-escala e da fusão no *decoder*; pode exigir cuidado no ajuste fino	Semântica por *pixel* (logits por classe)	*Forte*: bom modelamento de contexto global e consistência semântica; *decoder* simples reduz complexidade. *Limite*: pode ser mais sensível a pré-treinamento/regularização em bases menores; custo de atenção cresce com a resolução (embora o MiT seja projetado para eficiência).

Neste Capítulo, foram apresentados os fundamentos necessários para sustentar a proposta de automação da estimativa da SFA em SF provenientes de CP pós ensaio de impacto Charpy: (i) os conceitos do ensaio e as grandezas associadas, enfatizando a definição e os métodos normatizados de obtenção da SFA; (ii) bases de processamento de imagens e segmentação, formalizando a ideia de máscaras e quantificação por *pixels*; e (iii) fundamentos de segmentação profunda, incluindo CNNs, treinamento supervisionado, funções de perda, métricas e paradigmas arquiteturais que serão utilizados. Com essa base estabelecida, o próximo capítulo descreve a metodologia adotada no trabalho, detalhando o fluxo de processamento em cascata, o preparo do conjunto de dados, as configurações de treinamento/validação e o protocolo de avaliação empregado para comparar os modelos e quantificar seu impacto na estimativa final da SFA.

CAPÍTULO

3

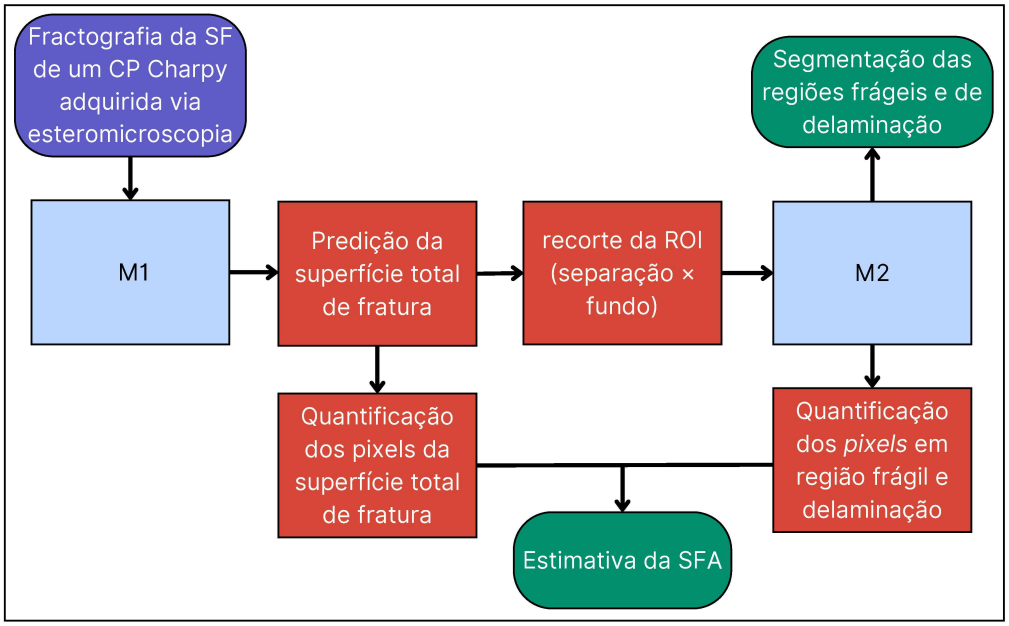
METODOLOGIA

A metodologia automática para determinar a SFA a partir da SF de CPs fraturados é apresentada no diagrama de blocos da Figura 21. A estimativa é realizada por meio de dois estágios de segmentação em cascata. No primeiro estágio, o modelo M1, implementado com a arquitetura *YYOLO11l-seg*, segmenta os *pixels* pertencentes à SF na imagem original e delimita a ROI utilizada nas etapas seguintes. Esse recorte isola a SF em relação ao fundo (Figura 22) e padroniza a entrada do segundo estágio, reduzindo a influência de variações de enquadramento e de fundo entre as imagens.

No segundo estágio, o modelo M2 processa exclusivamente a ROI e classifica os *pixels* nas classes "frágil" e "delaminação", como mostrado na Figura 23. Para essa tarefa, são comparadas quatro arquiteturas de visão computacional descritas no Capítulo 2: *U-Net*, *DeepLabV3–ResNet50*, *YOLO11l-seg* e *SegFormer*. Ao final do processamento, obtêm-se as quantidades de *pixels* correspondentes à SF, à região frágil e às delaminações e, a partir de (2.2), estima-se a SFA.

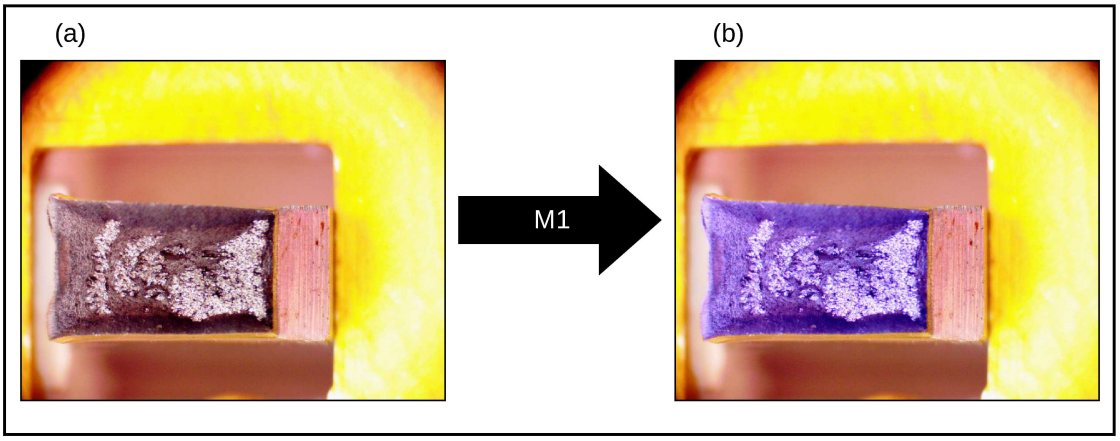
No primeiro estágio, segmenta-se apenas uma única classe (denominada "CP"). Adota-se um limiar de inferência de 0,6 e limita-se a, no máximo, uma detecção por imagem, ou seja, quando o modelo retorna múltiplas propostas, retém-se a predição associada ao maior valor de confiança. Neste contexto, o limiar de inferência corresponde ao valor mínimo de confiança exigido para que uma predição

Figura 21 – Diagrama de blocos da metodologia proposta.



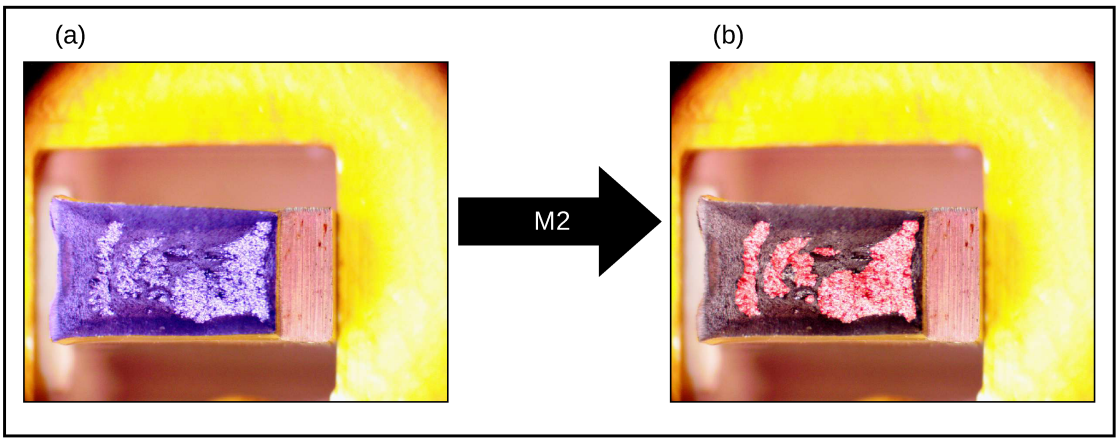
Fonte – Do autor.

Figura 22 – Resultado do processamento via M1. Em (a), a imagem original, proveniente do estereomicroscópio; em (b), a SF segmentada do fundo da imagem.



Fonte – Do autor.

Figura 23 – Resultado do processamento via M2. Em (a), a imagem de entrada de M2, proveniente da saída de M1; em (b), classificação dos *pixels* nas classes frágil e delaminação.



Fonte – Do autor.

seja considerada válida, enquanto a confiança representa o escore atribuído pelo modelo à probabilidade de a detecção estar correta para a classe prevista (i.e, um indicativo de quão consistente é a predição com os padrões aprendidos durante o treinamento).

Após a segmentação realizada com base na saída de M1, aplica-se um pós-processamento de suavização de contorno para reduzir serrilhados na borda da máscara. Em seguida, obtém-se o menor retângulo delimitador (*bounding box*) que contém a máscara prevista e recorta-se a ROI correspondente. Por fim, aplica-se a lógica de máscara negativa: após o recorte pelo retângulo delimitador, os *pixels* que ficam dentro da ROI (isto é, dentro do retângulo delimitador), mas que não pertencem à SF segundo a máscara de M1, têm seus valores substituídos por um valor constante (cor sólida, e.g., amarelo). Já os *pixels* classificados como fratura mantêm seus valores originais. Esse procedimento neutraliza o fundo dentro da ROI e padroniza a entrada do M2.

As máscaras baseadas na predição de M2 também são submetidas a um pós-processamento de suavização de contorno, visando melhorar a continuidade dos limites dos contornos reduzindo serrilhados nas bordas e contornos mais regulares, resultando em uma delimitação mais fiel à morfologia observada na superfície

de fratura. É importante notar que a quantificação dos *pixels* classificados como "frágil" por M2 é realizada antes de qualquer etapa de redimensionamento, isto é, no mesmo domínio espacial em que a área total da superfície de fratura foi definida pelo M1. Para isso, a máscara de fratura frágil inferida por M2 é remapeada para o recorte proveniente do M1, preservando a correspondência 1:1 entre os conjuntos de *pixels*.

Seja Q_{total} o número de *pixels* pertencentes à SF e $Q_{\text{frágil}}$ o número de *pixels* rotulados como frágil. Assim, (2.2) pode ser reescrita como

$$SFA = 100\left(1 - \frac{Q_{\text{frágil}}}{Q_{\text{total}}}\right), \tag{3.1}$$

permitindo a estimativa da SFA.

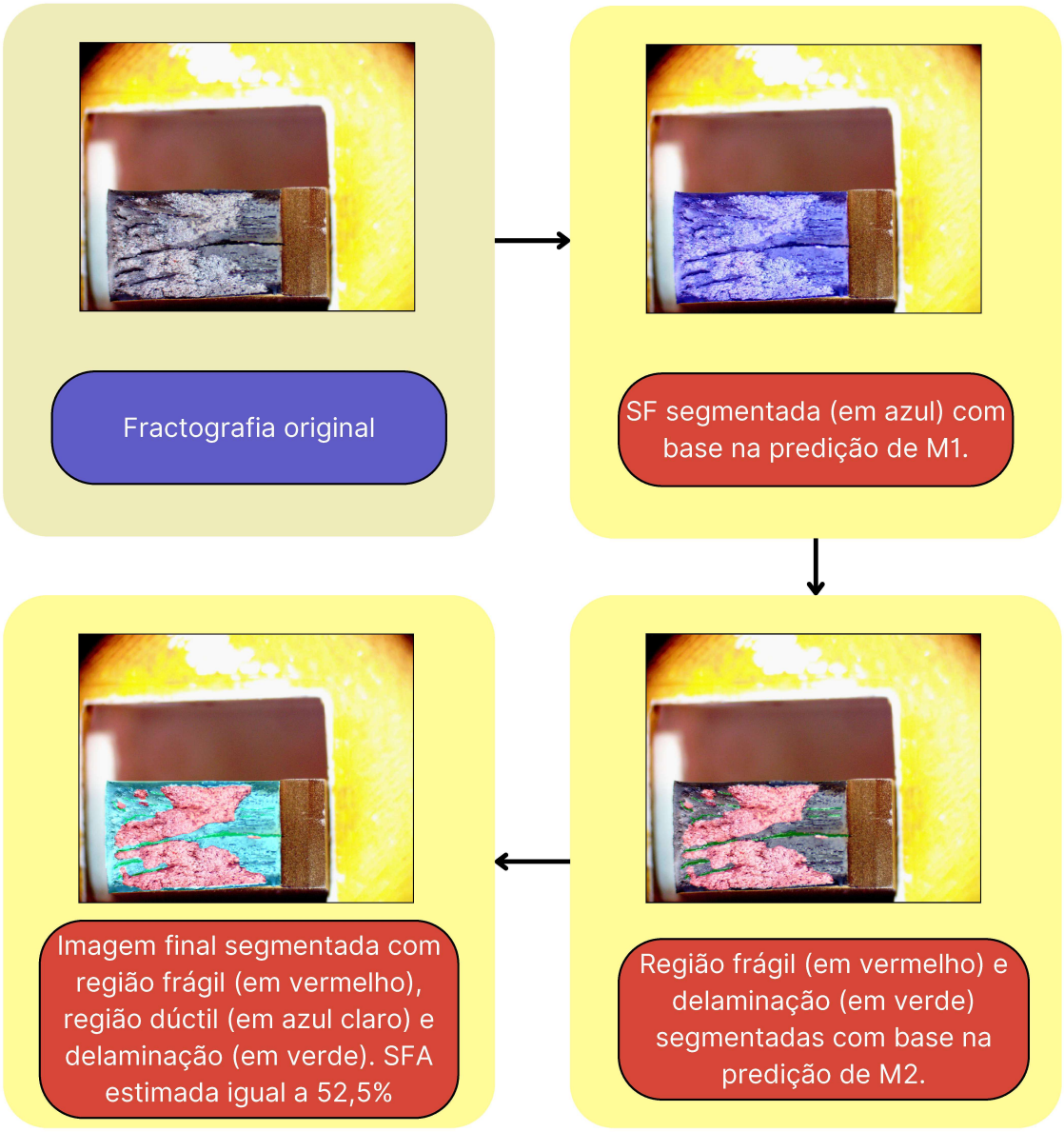
Dessa forma, a Figura 24 resume visualmente, por meio de um exemplo, a metodologia proposta.

3.1 Banco de dados

Os modelos M1 e M2 foram treinados de maneira supervisionada, isto é, utilizando bancos de dados previamente anotados manualmente por especialistas da área. Essas anotações fornecem as informações de referência necessárias para o aprendizado dos modelos. Na sequência, serão descritos o processo de construção desse banco de dados, incluindo, nessa ordem, preparação do CP, condições e parâmetros de captura das fractografias, procedimento de rotulagem, técnicas de aplicação de aumento de dados e preparação para o treinamento.

Para garantir maior consistência na construção do banco de dados, utilizou-se, para a captura das imagens dos CPs fraturados, o estereomicroscópio *Opticam*® *OPZTS*. Acoplada ao estereomicroscópio, foi utilizada uma câmera de 14,0 Mega Pixels do modelo *LOPT14003 0,35x Stereo*. Além disso, foi aplicada a ampliação de 1,0x em todas as geometrias exceto para a miniaturizada (3 mm × 4 mm × 27 mm), cuja ampliação utilizada foi de 1,5x, dessa forma, em uma única imagem é possível visualizar toda a área transversal do CP. Também foi necessária a concepção de um suporte para assegurar o correto posicionamento dos CPs dentro da área de foco

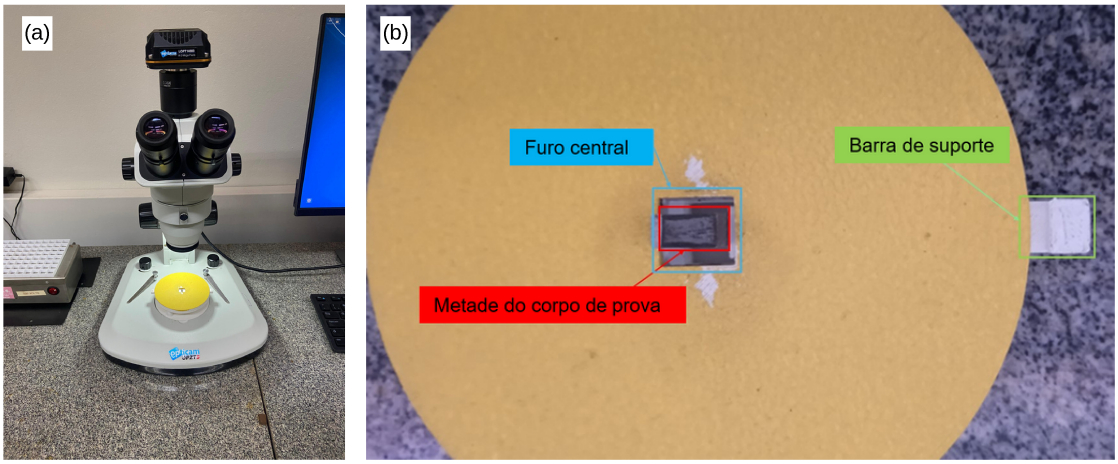
Figura 24 – Exemplo visual da metodologia proposta.



Fonte – Do autor.

da lente, minimizando variações durante a aquisição das imagens. Esse suporte (mostrado na Figura 25) foi modelado em software de *CAD Autodesk® Inventor 2024* e fabricado por meio de impressão 3D em acrilonitrilo-butadieno-estireno (ABS). O desenho técnico do suporte desenvolvido é apresentado no Apêndice A.

Figura 25 – Em (a), o suporte posicionado na base do estereomicroscópio para realizar a captura da imagem; em (b), uma visão superior do CP posicionado no suporte.



Fonte – Do autor.

Antes de cada sessão de aquisição de imagens, com o objetivo de garantir a padronização e a repetibilidade do processo, os parâmetros de digitalização e obtenção das imagens disponíveis no software foram previamente configurados. Esses parâmetros, apresentados na Tabela 3, permaneceram constantes ao longo de todas as sessões.

A Figura 26 apresenta imagens de SFs de CPs de diferentes geometrias, capturadas a partir do procedimento descrito. Observa-se que, em todos os casos, a fratura pode ser visualizada em sua totalidade, com boa distinção entre o CP e o fundo.

O processo de anotação das imagens para o M1 foi conduzido com o objetivo de treinar um modelo capaz de identificar e demarcar automaticamente a SF do CP. Para isso, utilizou-se o software *Label Studio®* (TKACHENKO *et al.*, 2020), uma plataforma de código aberto que permite adaptações para atender às necessidades

Tabela 3 – Parâmetros de aquisição e ajuste de imagem utilizados no software *Opticam*®.

Classificação do ajuste	**Ajuste**	**Valor**
Exposição e ganho	Exposição de alvo	120
	Tempo de exposição	888,389 ms
	Ganho	2,95
Balanço de branco	Temperatura da cor	5496 K
	Tom	954
Balanço de preto	*Red*	0
	Green	0
	Blue	0
Ajuste de cor	Tonalidade	8
	Saturação	142
	Brilho	-44
	Contraste	42
	Gamma	0,96

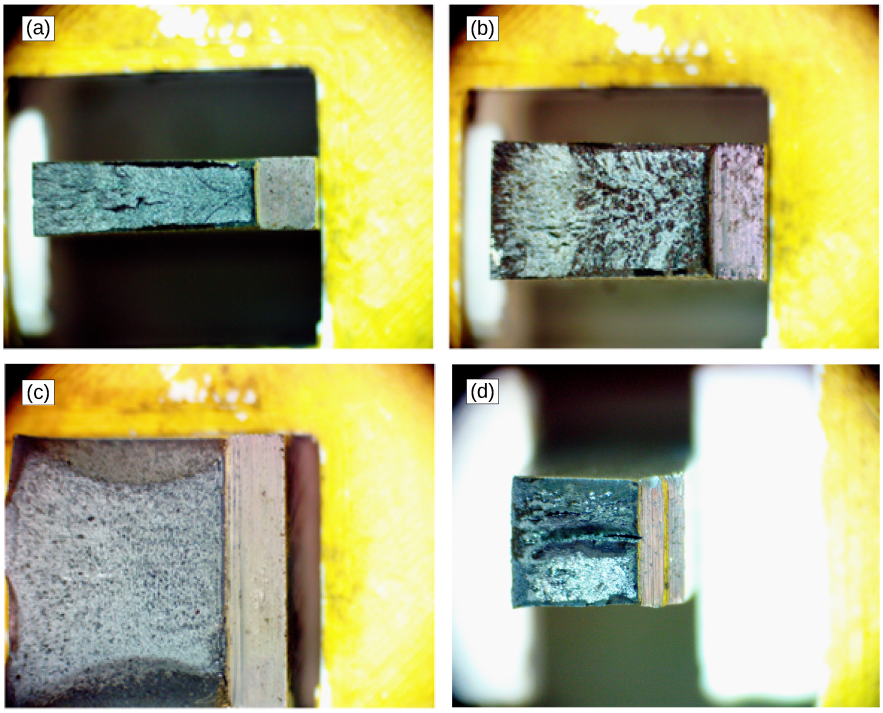
Fonte – Do autor.

específicas do projeto, eliminando custos de licenciamento. As imagens foram então anotadas manualmente por meio da definição de polígonos ao redor da SF dos CPs, de modo que cada polígono representa um conjunto de *pixels* associado a uma determinada classe. Um exemplo de imagem anotada é apresentado na Figura 27. Ressalta-se que o entalhe não foi considerado como parte da fratura.

A anotação das imagens para M2 seguiu uma metodologia semelhante, entretanto, foram anotadas apenas a área frágil e as regiões de delaminação e cada fratura, conforme exemplificado na Figura 28. Após o processo de anotação, os dados foram exportados em formato TXT para a *YOLO* e PNG para as demais arquiteturas.

O banco de dados foi dividido em duas partes: 70 % (427 imagens) para treinamento e 30 % (183 imagens) para validação, totalizando 610 imagens. Como os próprios termos indicam, o primeiro conjunto é utilizado para ajustar os parâmetros do modelo, enquanto o segundo é reservado para avaliar o aprendizado obtido. É importante destacar que as imagens de validação não são apresentadas ao modelo durante o processo de treinamento, de modo que funcionam como dados inéditos. Essa prática é amplamente adotada na área para verificar a capacidade

Figura 26 – Respectivamente, em (a), (b), (c) e (d), SFs de CPs com seção transversal 2,5 mm × 10 mm, 5 mm × 10 mm, 10 mm × 10 mm e 3 mm × 4 mm.

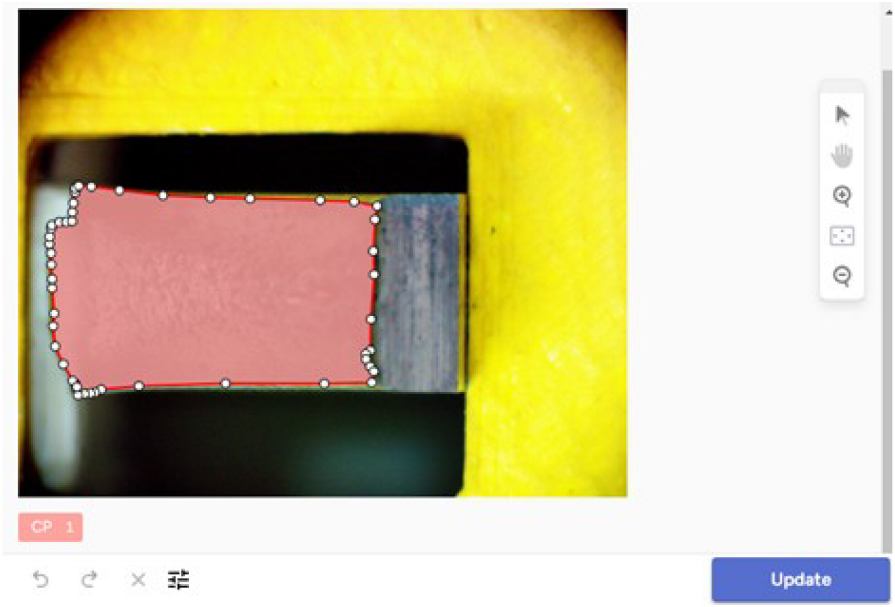


Fonte – Do autor.

de generalização do modelo, ou seja, sua habilidade de manter bom desempenho em situações não vistas anteriormente (GOODFELLOW; BENGIO; COURVILLE, 2016).

Para melhorar a generalização do aprendizado, o conjunto de imagens de treinamento foi ampliado utilizando a biblioteca para Python *Albumentations* (versão 2.0.8) e, quando necessário, pequenos ajustes manuais realizados com a biblioteca *OpenCV* (versão 4.11.0.86). Como resultado, o conjunto de treinamento foi expandido, totalizando 1281 imagens. Ressalta-se que as imagens de validação não foram alteradas. Na sequência, descrevem-se as operações de aumento de dados aplicadas. Foram empregados espelhamentos horizontal e vertical para tornar o modelo menos sensível à orientação do CP no campo de visão do estereomicroscópio. Também foram aplicadas rotações para emular pequenos desalinhamentos de posicionamento. Transformações afins complementam a rotação com transla-

Figura 27 – Exemplo de anotação da área total da fratura (em vermelho) por polígonos no ambiente *Label Studio®*.



Fonte – Do autor.

Figura 28 – Exemplo de anotação da área frágil (em azul) e delaminação (em verde) por polígonos no ambiente *Label Studio®*.



Fonte – Do autor.

ção de até ±10 % (simulando deslocamentos do CP), além de variações de escala. Adicionalmente, aplicou-se *RandomScale* como perturbação extra de tamanho, técnica que simula variações associadas à distância focal e à altura da câmera. Para as transformações fotométricas, utilizaram-se *RandomBrightnessContrast* para modelar variações de iluminação, ganho da câmera e diferenças de exposição, e *ColorJitter*, que introduz perturbações controladas nos canais de cor, simulando pequenas variações de balanço de branco, temperatura de cor e reflexos especulares do conjunto óptico.

Transformações geométricas (como rotação, translação e mudança de escala) podem resultar em *pixels* mapeados para coordenadas não inteiras. Para reconstruir os valores nos novos *pixels*, é necessária uma interpolação, isto é, a estimativa do valor em uma posição que fica "entre" *pixels* da imagem original. A escolha do método de interpolação pode impactar significativamente o resultado, especialmente em bordas e texturas finas. Neste trabalho, adotou-se a interpolação bilinear nas imagens, que estima o valor de cada novo *pixel* a partir dos quatro vizinhos imediatos (janela 2×2), realizando uma média ponderada pelas distâncias nas direções x e y. Por fim, quando transformações geométricas projetam partes da imagem para fora do quadro, adotou-se preenchimento com bordas pretas. Os parâmetros utilizados no aumento do banco de dados são apresentados na Tabela 4.

Com o banco de dados devidamente particionado, anotado e ampliado por meio das estratégias de aumento descritas, estabeleceu-se um conjunto de entrada mais robusto e representativo para o desenvolvimento dos modelos. A definição dessas etapas prévias assegura condições adequadas para avaliar o aprendizado e a capacidade de generalização durante a modelagem. Nesse sentido, a próxima seção trata sobre os procedimentos de treinamento de cada modelo e suas configurações.

3.2 Treinamento dos modelos

Como o fluxo é em cascata, os modelos foram treinados em dois contextos distintos. O M1 foi treinado separadamente com uma configuração específica baseada na *YOLO11l-seg*, apresentada na Tabela 6.

Visando comparabilidade entre arquiteturas no M2, adotou-se um proto-

Tabela 4 – Parâmetros de aumento de banco de dados utilizados no conjunto de treinamento.

Transformação	**Parâmetros principais**	**Prob. (p)**	**Interpol.**
HorizontalFlip (Espelhamento horizontal)	-	0,20	Bilinear
VerticalFlip (Espelhamento vertical)	-	0,20	Bilinear
Rotate (Rotação)	*limit* $= 6°$	0,50	Bilinear
Affine (Transformação afim)	*translate_percent* $= 0{,}1$; *scale* $= (0{,}15, 1{,}7)$; *rotate* $= 7°$	0,50	Bilinear
RandomScale (Escalonamento aleatório)	*scale_limit* $= 0{,}5$	0,60	Bilinear
RandomBrightnessContrast (Brilho e contraste aleatórios)	*brightness_limit* $= 0{,}3$; *contrast_limit* $= 0{,}4$	0,65	-
ColorJitter (Perturbação de cor)	*brightness* $= 0{,}35$; *contrast* $= 0{,}4$; *saturation* $= 0{,}4$; *hue* $= 0{,}2$	0,60	-

Fonte – Do autor.

colo padronizado de treinamento, mantendo constantes (na medida do possível) a resolução de entrada, a política de otimização, o número máximo de épocas, a estratégia de interrupção por estagnação e a forma de avaliação no conjunto de validação. As imagens foram processadas com resolução 640×640 px. Para reduzir a influência de variações de aquisição e tornar a comparação mais controlada, o aumento de dados foi aplicado *off-line* na etapa de preparação do banco de dados, conforme descrito na seção anterior.

A Tabela 8 resume o protocolo comum adotado para o treinamento das arquiteturas avaliadas em M2. O valor de *seed* foi fixado para garantir reprodu-

Tabela 6 – Configuração de treinamento adotada para M1.

Parâmetro	**Valor**
Resolução de entrada	640 × 640 px
Épocas máximas	541
Early stopping (*patience*)	45
Otimizador	AdamW
Taxa de aprendizado inicial	$\eta_0 = 1 \times 10^{-4}$
Weight decay	5×10^{-4}
Warmup	3 épocas
Scheduler	decaimento cossenoidal até $\eta = 0{,}01$
Precisão mista (*AMP*)	ativado
Augmentations on-line	desativado
Batch size	4
seed	30

Fonte – Do autor.

tibilidade do particionamento e das inicializações quando aplicável. Em todos os treinamentos, foi utilizada precisão mista para reduzir custo computacional e acelerar o processo (PyTorch Contributors, 2025). A taxa de aprendizado inicial foi definida como 3×10^{-4}, com *warmup* nas primeiras épocas e decaimento cossenoidal até um fator final $\eta_f = 0{,}01$. O critério de interrupção por estagnação (*early stopping*) foi controlado por um fator de 25 épocas sem melhora no desempenho do conjunto de validação. A avaliação durante o treinamento foi realizada no conjunto de validação, reportando métricas por classe (delaminação e frágil), incluindo **IoU** e **Dice**, precisão, *recall*, e medidas agregadas como **mIoU** e **mDice**.

Tabela 8 – Configuração de treinamento adotada para M2.

Parâmetro	**Configuração**
Resolução de entrada	640 × 640 px
Épocas máximas	150
Early stopping	*patience*=25
Otimizador	AdamW
Taxa de aprendizado inicial	$\eta_0 = 3 \times 10^{-4}$
Weight decay	5×10^{-4}
Warmup	3 épocas
Scheduler	decaimento cossenoidal até $\eta_f = 0{,}01$
Precisão mista	*AMP* ativado
Augmentations on-line	desativado
Batch size	8
Seed	42

Fonte – Do autor.

CAPÍTULO

4

RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados e discutidos os resultados obtidos no estágio de separação do *background* (M1), responsável por definir a ROI, e no estágio de segmentação semântica (M2), responsável por identificar regiões de fratura frágil e delaminação na ROI previamente definida. A comparação é conduzida entre as quatro arquiteturas de segmentação profunda descritas no Capítulo 2: *U-Net*, *DeepLabV3–ResNet50*, *SegFormer* e *YOLO11l-seg*.

Os resultados reportados de M2 referem-se ao conjunto de validação composto por 255 imagens, empregando uma formulação de três rótulos (0: *background* classe implícita, 1: delaminação, 2: frágil). Para quantificar o desempenho no nível de máscara, são analisadas métricas por classe (IoU, Dice, Precisão e *Recall* para delaminação e frágil, a classe *background* não é avaliada por ser auxiliar), além de medidas agregadas (mIoU e mDice) calculadas a partir das classes de interesse. Complementarmente, a discussão inclui uma análise qualitativa por meio de exemplos visuais, destacando padrões de acerto e erro associados a diferentes morfologias de fratura e a variações naturais de aquisição.

4.1 Avaliação do desempenho global de M1

Embora o objetivo central deste trabalho seja comparar arquiteturas no segundo estágio (M2), é importante apresentar evidências do desempenho do primeiro estágio (M1), pois ele define a ROI utilizada por todos os modelos e fornece a área total de fratura empregada no cálculo da SFA. Assim, erros em M1 poderiam propagar-se para as métricas derivadas e comprometer a rastreabilidade da comparação entre os modelos do M2.

Nessa avaliação de desempenho, M1 é adotado como um componente de referência (*baseline*) a partir do trabalho de José *et al.* (2025), no qual o modelo foi treinado e validado especificamente para segmentar a SF e separá-la do fundo em imagens estereomicroscópicas. Como reportado, o modelo atinge desempenho elevado no conjunto de validação, com AP_{50}^{mask} na faixa 0,994–0,995 e $AP_{50\text{–}95}^{\text{mask}}$ em torno de 0,992–0,994 (pico de 0,995 em época inicial), indicando alta aderência dos contornos previstos mesmo sob critérios mais restritivos. Também são reportados valores de precisão superiores a 0,998 e *recall* próximo de 1,0 após a fase inicial de treinamento, sugerindo baixa incidência de falsos positivos e falsos negativos no ponto de operação adotado. A Tabela 10 sintetiza esses resultados.

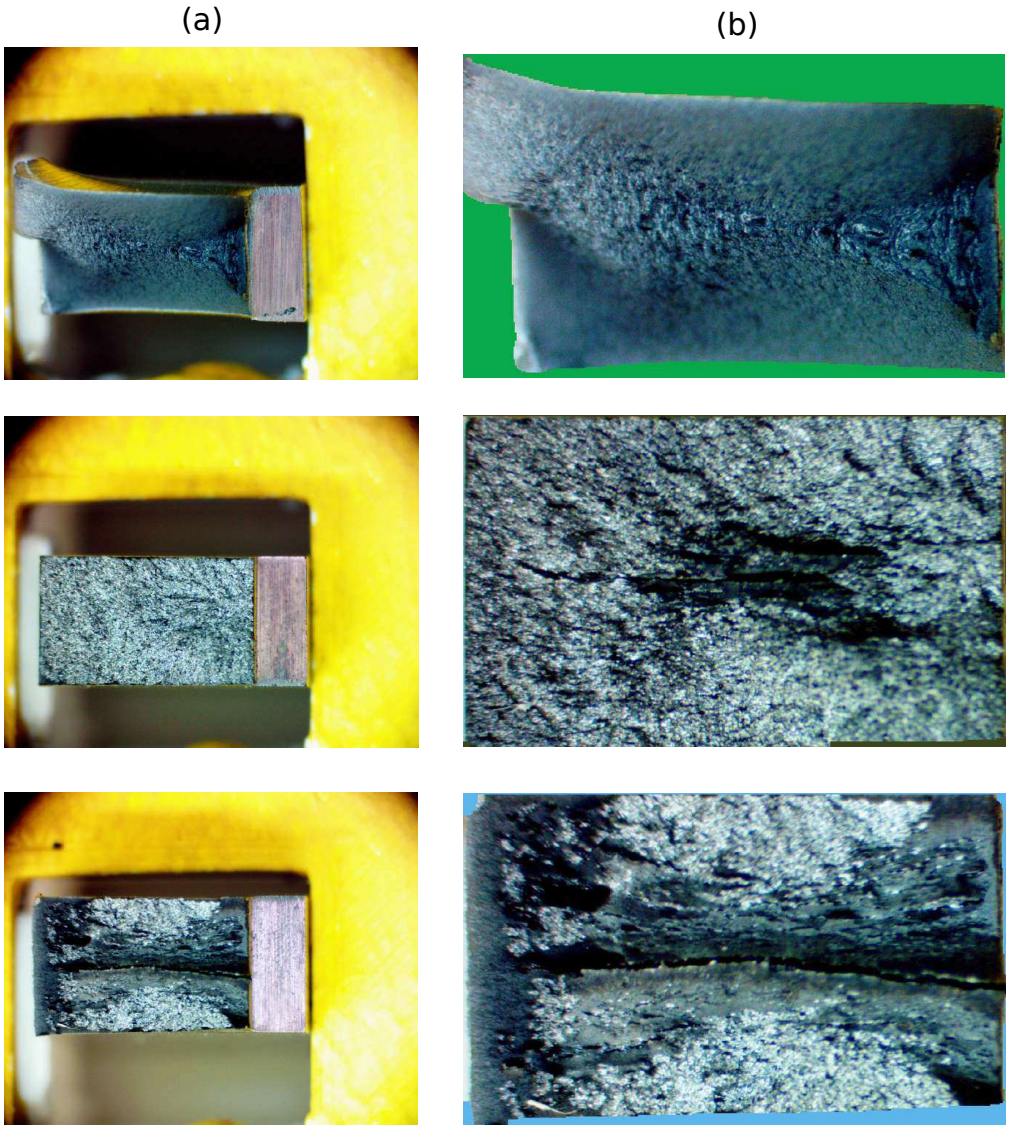
Tabela 10 – Métricas avaliadas no conjunto de validação para a melhor época de M1 (em %).

Métrica	**Valor**
AP_{50}	99,5
$AP_{50\text{–}95}$	99,4
Precisão	**99,8**
Recall	**99,9**

Fonte – Adaptado de José *et al.* (2025).

Além das métricas reportadas, a Figura 29 apresenta exemplos qualitativos de saída do M1. Observa-se que a segmentação da SF produz contornos contínuos e recortes consistentes, mesmo sob variações de iluminação e textura, reforçando a adequação do M1 como etapa fixa de padronização da ROI para o segundo estágio.

Figura 29 – Exemplos qualitativos de saída do M1. Na coluna (a), a imagem original; na coluna (b), a imagem recortada.



Fonte – Do autor.

A partir desta seção, a análise passa a se concentrar no segundo estágio (M2), responsável pela segmentação das classes *frágil* e *delaminação* e ponto chave do trabalho. Como o M1 é mantido fixo ao longo de todos os experimentos, as diferenças de desempenho reportadas nas seções seguintes refletem diretamente o comportamento das arquiteturas avaliadas para M2 e seus efeitos sobre as grandezas derivadas de interesse.

4.2 Organização dos resultados e critérios de comparação para M2

As médias **mIoU** e **mDice** são definidas como a média aritmética simples das duas classes de interesse, excluindo o *background*, conforme

$$\mathrm{mIoU} = \frac{\mathrm{IoU}_{\mathrm{delam}} + \mathrm{IoU}_{\mathrm{frágil}}}{2}, \tag{4.1}$$

$$\mathrm{mDice} = \frac{\mathrm{Dice}_{\mathrm{delam}} + \mathrm{Dice}_{\mathrm{frágil}}}{2}. \tag{4.2}$$

Sendo os valores de **IoU** e **Dice** de cada classe calculados respectivamente por (2.42) e (2.43). A escolha de avaliação por essas métricas se justifica por evitar que o desempenho da classe *background* (que mesmo com a imagem recortada, pode chegar a ter um número expressivo *pixels*) mascare o comportamento do modelo nas demais classes diretamente relacionadas ao cálculo subsequente de grandezas de interesse, como a SFA (BERNHARD *et al.*, 2024). Na prática, essa decisão faz com que variações no acerto de *pixels* fora das regiões de fratura (por exemplo, pequenas porções residuais de fundo preservadas após o recorte, e essas regiões tendem a aumentar em CPs conforme a ductilidade aumenta) não dominem as métricas agregadas, concentrando a avaliação na capacidade do M2 de predizer, com robustez, as regiões de fratura frágil e de delaminação dentro da área de fratura predita por M1.

Para os modelos *U-Net*, *DeepLabV3–ResNet50* e *SegFormer*, a cada época, durante o treinamento foi calculada a **mIoU** no conjunto de validação e o *checkpoint* do modelo foi salvo como o melhor sempre que esse valor superou o melhor

resultado até então por uma margem mínima (Δ). Especificamente, adotou-se uma tolerância absoluta de $\Delta = 1 \times 10^{-4}$, de modo que a melhor época só é atualizada quando $\mathbf{mIoU} \geq \mathbf{mIoU}_{best} + \Delta$. Esse procedimento reduz a troca de *checkpoint* por flutuações numéricas muito pequenas e mantém o critério de seleção alinhado ao objetivo principal, isto é, maximizar a sobreposição entre predição e referência nas classes de interesse. Em paralelo, empregou-se *early stopping* com *patience*=25, técnica que interrompe o treinamento quando não há melhora relevante no critério monitorado ao longo de 25 épocas consecutivas; dessa forma, o *checkpoint* final de treinamento não necessariamente coincide com a melhor época porque o término do treinamento é determinado pela ausência de ganho adicional após o melhor valor já ter sido atingido, enquanto a melhor época corresponde à época exata em que ocorreu o maior valor do critério monitorado (**mIoU**), podendo ocorrer antes do encerramento e ser seguido apenas por oscilações de validação sem melhoria consistente. O valor de perda na validação também foi registrado, mas apenas como indicador auxiliar de convergência e verificação de *overfitting*, não sendo utilizado como critério de seleção do *checkpoint*. No caso do *YOLO11l-seg*, cuja saída original é por instância, a avaliação foi realizada pós-treinamento, após a conversão das predições para uma representação semântica por classe, por meio da união das máscaras de todas as instâncias pertencentes a cada classe assim obtendo-se, para cada imagem, uma máscara em PNG com rótulos mutuamente exclusivos no mesmo domínio espacial dos demais modelos, o que permite calcular as métricas por *pixel* e as médias **mIoU**/**mDice** sob o mesmo protocolo de avaliação e, consequentemente, comparar de forma direta o desempenho do *YOLO* com as arquiteturas estritamente semânticas no estágio M2.

Com base nesses critérios e definições, a comparação entre arquiteturas é conduzida no conjunto de validação, permitindo interpretar as diferenças de desempenho tanto em termos de sobreposição por *pixel* quanto de estabilidade ao longo do treinamento. Na sequência, a Seção 4.3 apresenta a síntese quantitativa dos resultados, reunindo as métricas agregadas e a referência temporal dos *checkpoints* para cada modelo.

4.3 Avaliação quantitativa de desempenho global para M2

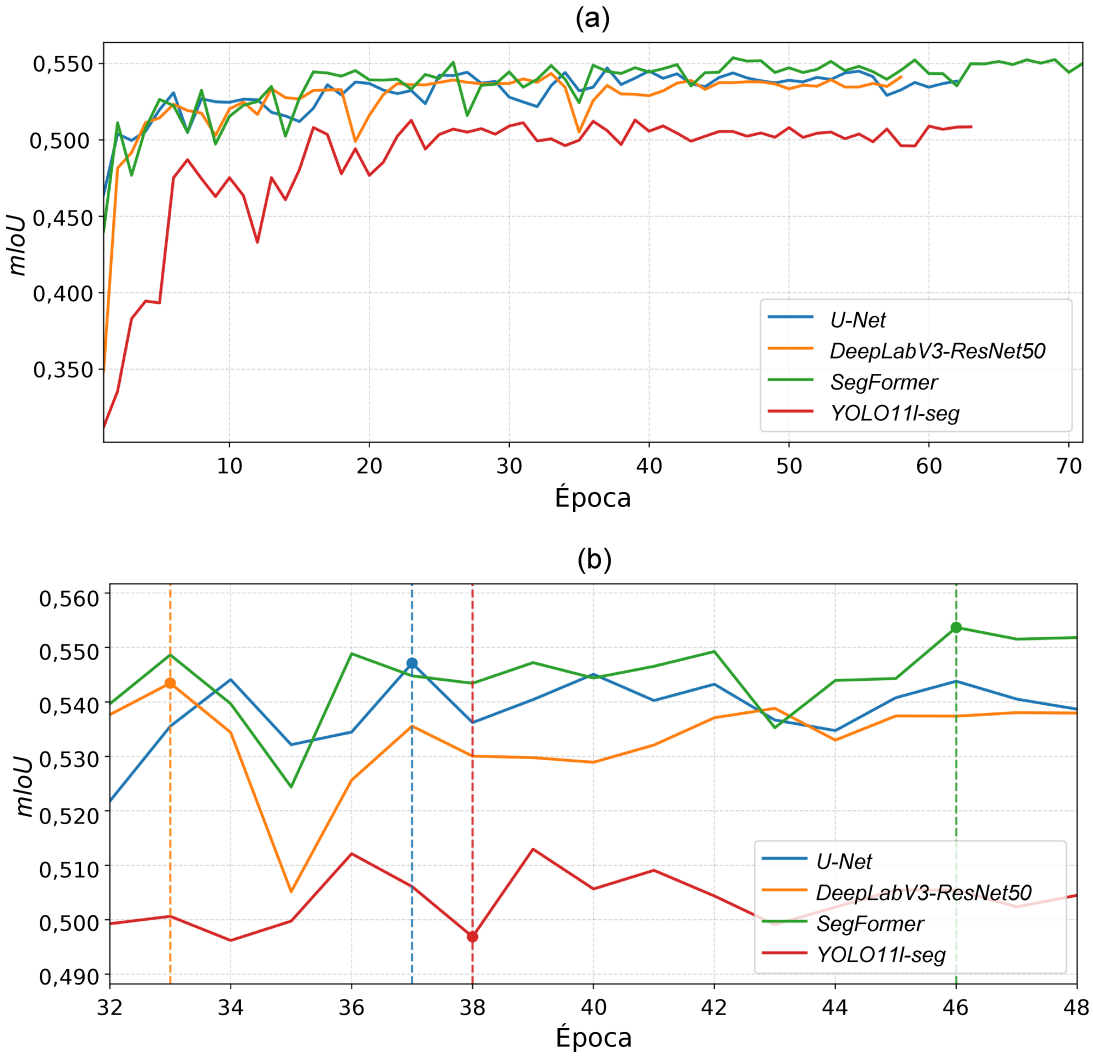
A comparação quantitativa entre as arquiteturas foi iniciada pela análise do comportamento das métricas agregadas na validação ao longo das épocas, de modo a avaliar convergência, estabilidade e o momento em que cada modelo atinge seu melhor desempenho. A Figura 30 apresenta as curvas de *mIoU* para os quatro modelos, enquanto a Figura 31 mostra as curvas correspondentes de mDice.

De maneira geral, as curvas indicam um ganho rápido de desempenho nas primeiras épocas, seguido por um regime de estabilização em que mIoU e mDice passam a oscilar em torno de um patamar próximo ao máximo. Esse padrão é particularmente evidente nos três modelos estritamente semânticos: após a subida inicial, *U-Net* e *DeepLabV3–ResNet50* evoluem de forma muito próxima e com variações pequenas em torno do platô, enquanto o *SegFormer* tende a sustentar, na maior parte do intervalo observado, valores ligeiramente superiores. Além disso, nota-se que, a partir de aproximadamente 15–20 épocas, os ganhos passam a ser marginais, e as diferenças entre *checkpoints* consecutivos ficam frequentemente na escala de poucos milésimos, tornando natural a ocorrência de pequenas oscilações na validação sem melhora consistente.

Essas oscilações são compatíveis com a natureza do problema: mIoU/mDice são calculadas sobre duas classes de interesse com forte desbalanceamento, de modo que pequenas variações na recuperação de estruturas finas (como delaminações) podem produzir flutuações perceptíveis nas métricas agregadas, mesmo quando a segmentação de "frágil" já se encontra estável. Observa-se também que a mDice tende a assumir valores sistematicamente mais altos que a mIoU, como esperado pela própria definição das métricas, o que não altera o *ranking* entre modelos.

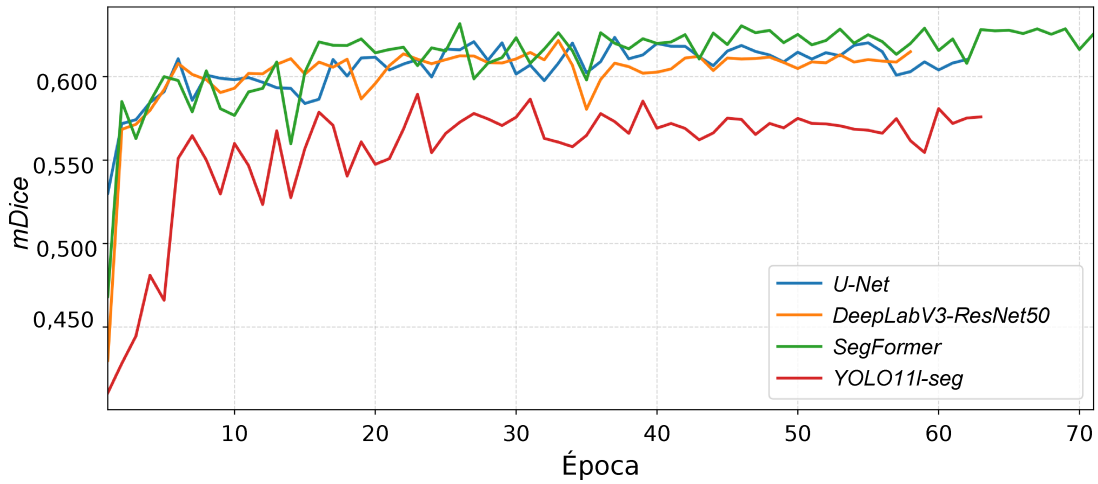
No caso do *YOLO11l-seg*, as curvas exibem uma fase inicial mais lenta e ruidosa, com valores substancialmente inferiores nas primeiras épocas e posterior aproximação a um patamar mais estável, ainda abaixo dos modelos originalmente semânticos. Esse comportamento é coerente com a diferença de paradigma (instância $\times$ semântica) e com o fato de a avaliação semântica depender do pós-processamento por união de instâncias e da resolução de sobreposições por confi-

Figura 30 – Curvas de **mIoU** na validação ao longo das épocas para as quatro arquiteturas avaliadas, calculadas excluindo a classe *background*. Em (a), as curvas até a última época; em (b), as mesmas curvas, evidenciando, em linhas tracejadas, a melhor época de cada modelo.



Fonte – Do autor.

Figura 31 – Curvas de **mDice** na validação ao longo das épocas para as quatro arquiteturas avaliadas, calculadas excluindo a classe *background*.



Fonte – Do autor.

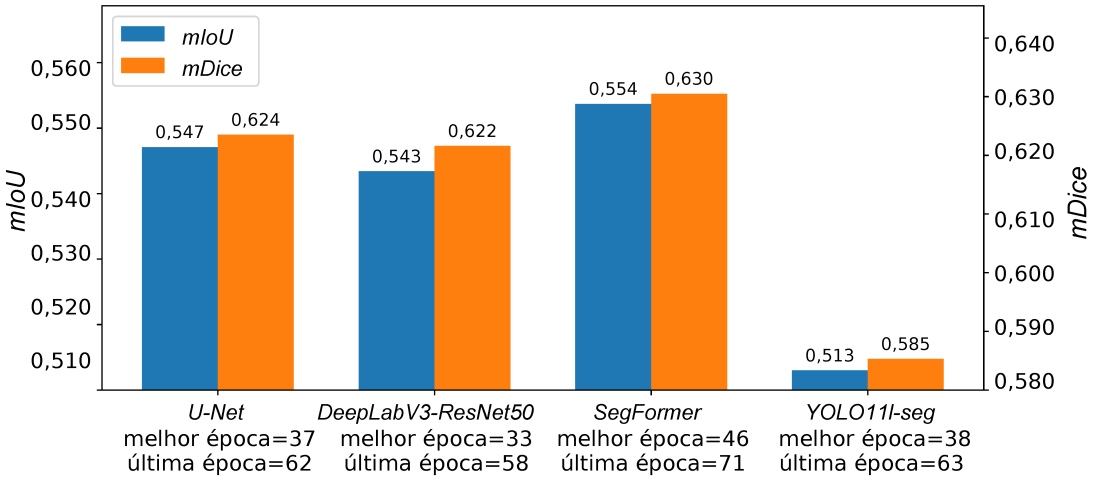
ança por *pixel*, etapas que podem amplificar a sensibilidade do resultado final a variações no conjunto de instâncias previstas.

É possível, ainda, relacionar o comportamento das curvas com as diferenças arquiteturais entre os modelos. *U-Net* e *DeepLabV3–ResNet50* são arquiteturas convolucionais do tipo *encoder–decoder*, com forte viés indutivo local e agregação multiescala, o que tende a favorecer aprendizado rápido de padrões de textura e bordas nas primeiras épocas e a produzir trajetórias de validação semelhantes quando treinadas sob protocolo idêntico. Assim, de forma coerente, é possível notar que suas curvas de **mIoU** e **mDice** evoluem de maneira muito próxima e estabilizam em patamares comparáveis. Já o *SegFormer*, por empregar um *encoder* baseado em *Transformer* com mecanismos de atenção, tem maior capacidade de integrar contexto global e relações de longo alcance, o que pode beneficiar a delimitação de regiões mais ambíguas e estruturas finas; isso é compatível com um patamar ligeiramente superior nas métricas agregadas, sobretudo quando o desempenho em "frágil" já se encontra próximo do teto e a classe delaminação passa a ser o fator limitante. Por fim, a *YOLO11l-seg* é treinada originalmente no paradigma de segmentação por instância, no qual a otimização envolve componentes de detecção

(associação de instâncias e escores de confiança) além da máscara, essa diferença de objetivo e de pós-processamento tende a tornar a curva semântica mais sensível a variações na predição de instâncias (especialmente nas primeiras épocas), o que é consistente com a fase inicial mais ruidosa e com o patamar final inferior quando comparado às arquiteturas estritamente semânticas (KHANAM; HUSSAIN, 2024; XIE *et al.*, 2021; CHEN *et al.*, 2017; RONNEBERGER; FISCHER; BROX, 2015).

Por fim, o comportamento observado é coerente com o procedimento de seleção da melhor época por maior **mIoU** e com o uso de *early stopping*: o melhor desempenho pode ocorrer antes do encerramento do treinamento e ser seguido majoritariamente por oscilações em torno do platô, sem novas melhorias relevantes no critério monitorado. A Figura 32 sumariza o desempenho das quatro arquiteturas avaliadas no conjunto de validação, reportando a melhor época e as métricas agregadas correspondentes, calculadas exclusivamente sobre as classes de interesse, bem como a última época registrada durante o treinamento.

Figura 32 – Resumo dos resultados no conjunto de validação associadas à melhor época.

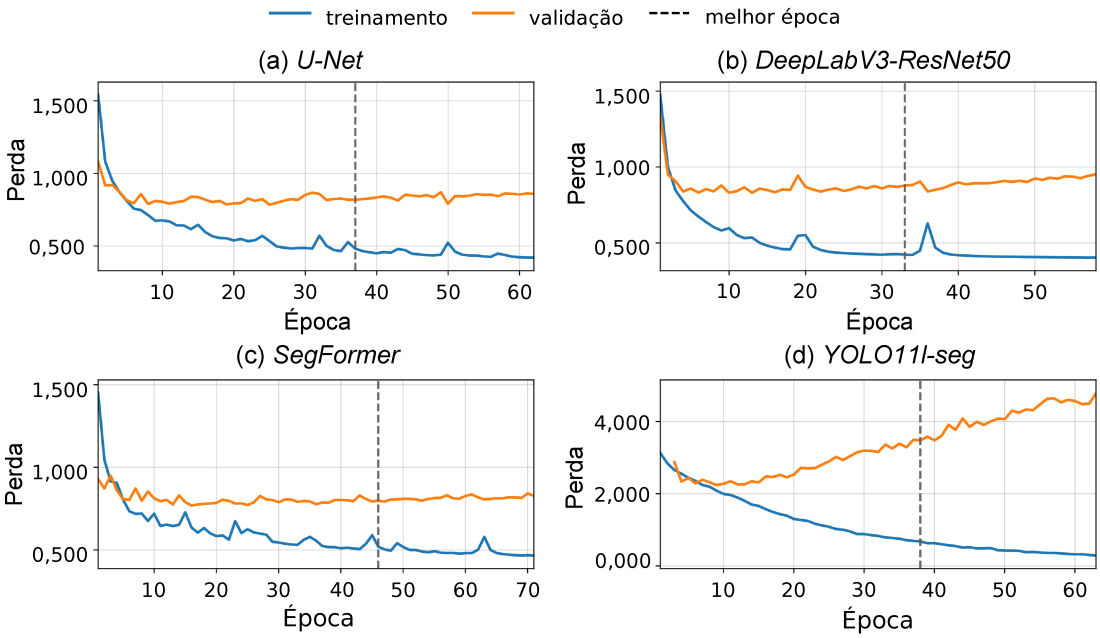


Fonte – Do autor.

No panorama global, o *SegFormer* apresentou o melhor desempenho médio, seguido por *U-Net* e *DeepLabV3-ResNet50*, com diferenças pequenas entre esses três modelos. Já a *YOLO11l-seg* obteve valores substancialmente inferiores de **mIoU**/**mDice**.

A informação da última época registrada é útil para discutir estabilidade e convergência. Em geral, diferenças pequenas entre o desempenho na melhor época e na última época confirmam que o treinamento atingiu um regime de platô, no qual oscilações naturais de validação podem ocorrer sem ganhos consistentes no critério monitorado. Ademais das métricas baseadas em sobreposição por *pixel*, as curvas de perda (*loss*) ao longo das épocas fornecem uma interpretação que complementa a análise de desempenho dos modelos. Dessa forma, a Figura 33 apresenta as perdas de treino e validação para cada arquitetura.

Figura 33 – Curvas de perda de treinamento e validação ao longo das épocas para (a) *U-Net*, (b) *DeepLabV3–ResNet50*, (c) *SegFormer* e (d) *YOLO11l-seg*.



Fonte – Do autor.

Nos modelos *U-Net*, *DeepLabV3–ResNet50* e *SegFormer*, observa-se uma queda acentuada da perda de treino nas primeiras épocas, seguida por redução mais gradual, enquanto a perda de validação diminui rapidamente no início e passa a oscilar em torno de um patamar aproximadamente constante. Na *U-Net* e *SegFormer*, esse patamar se mantém relativamente estável, com pequenas flutuações; já no *DeepLabV3–ResNet50* nota-se uma tendência levemente crescente da

perda de validação ao longo das épocas, após a fase inicial de queda. Esse comportamento (redução contínua em treino com estabilização ou leve aumento em validação) é compatível com um regime de ganhos marginais de generalização e início de sobreajuste, no qual melhorias adicionais passam a ser limitadas principalmente pela dificuldade do problema, e não pela capacidade de ajuste no treino. Como a melhor época foi selecionada por **mIoU**, não se espera necessariamente que o menor valor de perda em validação coincida com a época de maior valor de **mIoU**. No caso da *YOLO11l-seg*, a curva apresenta um comportamento qualitativamente distinto. Enquanto a perda de treino decresce de forma contínua, a perda de validação diminui apenas nas primeiras épocas e, em seguida, cresce de maneira pronunciada ao longo do treinamento, padrão que sugere sobreajuste (*overfitting*. Apesar desse aumento, observa-se nas Figuras 30 e 31 que o desempenho tende a estabilizar em um patamar, o que evidencia a importância de avaliar diretamente a métrica-alvo do problema.

4.4 Avaliação quantitativa de desempenho por classe para M2

Embora as métricas agregadas apresentadas na seção anterior sejam úteis para comparar o desempenho global das arquiteturas, a interpretação do problema e seu impacto nas grandezas derivadas requerem a análise separada das duas classes de interesse. Isso se deve ao fato de que essas classes apresentam características morfológicas e representatividade distintas: conforme já dito, a região frágil tende a ocupar uma fração maior da SF, enquanto a delaminação, quando presente, ocorre tipicamente em regiões estreitas e com baixa fração de *pixels*. Assim, diferenças entre arquiteturas podem refletir desempenho distinto na delimitação da região frágil (classe dominante) e/ou limitações sistemáticas na identificação de delaminações (classe minoritária), além de confusões entre delaminação, frágil e a classe implícita *background*.

A Tabela 11 apresenta as métricas por classe na melhor época para cada arquitetura. Para “delaminação” e “frágil” são reportados **IoU**, **Dice**, Precisão e *Recall*, permitindo identificar não apenas a sobreposição média, mas também o

trade-off entre falsos positivos e falsos negativos em cada classe.

Tabela 11 – Métricas por classe avaliadas no conjunto de validação para a melhor época de melhor época de cada arquitetura de M2 (em %).

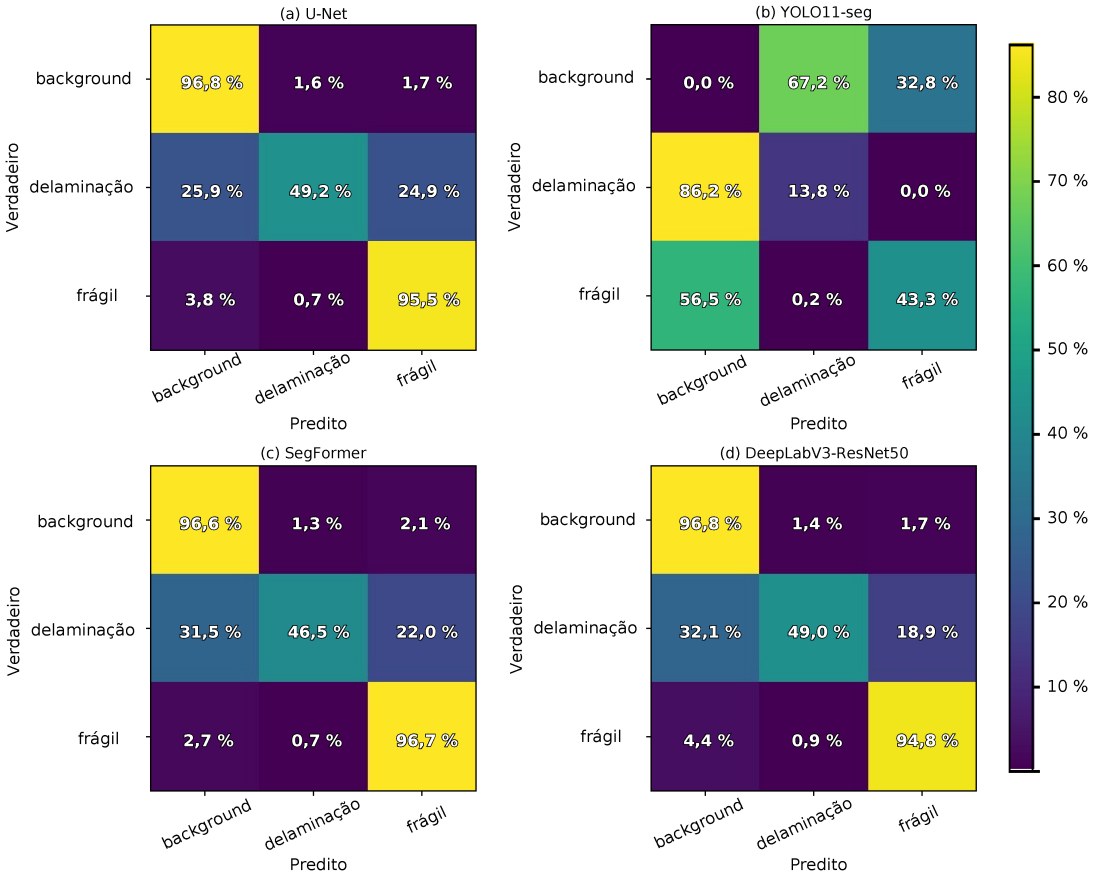
Modelo	Melhor época	Delaminação				Frágil			
		Prec.	Rec.	IoU	Dice	Prec.	Rec.	IoU	Dice
U-Net	37	20,0	**49,2**	16,5	28,4	**97,1**	95,5	**92,9**	96,3
DeepLabV3-ResNet50	33	20,0	49,0	16,6	28,4	**97,1**	94,8	92,1	95,9
SegFormer	46	**21,6**	46,5	**17,3**	**29,5**	96,5	**96,7**	93,4	**96,6**
YOLO11-seg	38	15,3	41,4	12,5	22,3	94,6	94,9	90,0	94,8

Fonte – Do autor.

Como complemento, a Figura 34 apresenta as matrizes de confusão normalizadas por linha para cada arquitetura na melhor época. A normalização por linha (classe verdadeira) permite interpretar diretamente, para cada classe, como os *pixels* são distribuídos entre as classes previstas, evidenciando os principais padrões de confusão independentemente da diferença de representatividade entre classes.

Sobre a classe frágil, em concordância com os altos valores de IoU/Dice observados na Tabela 11, as arquiteturas *U-Net*, *DeepLabV3–ResNet50* e *SegFormer* apresentam elevada taxa de acerto, como visto na matriz de confusão, a maior parte dos *pixels* verdadeiros de "frágil" é corretamente atribuída à própria classe (na faixa de ∼95–97 %), com erros residuais concentrados principalmente na confusão entre "frágil" e "*background*" (poucos pontos percentuais) e, em menor grau, entre "frágil" e "delaminação". Na prática, esse comportamento indica que, para esses modelos, a segmentação de "frágil" é robusta e próxima do teto de desempenho, de modo que as diferenças entre arquiteturas tendem a se manifestar mais como variações sutis de contorno e pequenas oscilações de falsos negativos/positivos nas bordas. Para a *YOLO11l-seg*, entretanto, a matriz de confusão revela um padrão distinto: uma fração substancial dos *pixels* verdadeiros de "frágil" é atribuída ao "*background*" (56,5 %), com apenas 43,3 % sendo recuperados como "frágil". Esse resultado sugere uma tendência de subsegmentação quando o modelo é avaliado semanticamente após a conversão das instância: regiões que não são cobertas por instâncias previstas (ou que são eliminadas por limiares de confiança e supressão)

Figura 34 – Matrizes de confusão avaliadas no conjunto de validação normalizadas por linha para cada arquitetura na melhor época.



Fonte – Do autor.

acabam rotuladas como "*background*" na máscara final, reduzindo o *recall* efetivo da classe frágil.

Em contraste, a classe delaminação permanece como a classe mais desafiadora em todas as arquiteturas. Nos modelos *U-Net*, *DeepLabV3–ResNet50* e *SegFormer* a matriz normalizada mostra que apenas cerca de 46,5–49,2 % dos *pixels* verdadeiros de "delaminação" são corretamente recuperados como "delaminação", enquanto o restante é dividido principalmente entre "*background*" (25,9–32,1 %) e "frágil" (18,9–24,9 %). Esse padrão explica por que a **IoU/Dice** de "delaminação" permanece baixa mesmo quando o desempenho em "frágil" já é elevado: a

"delaminação" é frequentemente absorvida pelas demais classes, seja por perda de contorno em regiões estreitas, ou por ambiguidade local de textura e contraste. No *YOLO11l-seg*, o efeito é ainda mais pronunciado: 86,2 % dos *pixels* verdadeiros de delaminação são classificados como "*background*", e apenas 13,8 % são recuperados como delaminação. Diferentemente dos demais, em que parte da delaminação migra também para frágil, a *YOLO11l-seg* tende a "apagar" a delaminação para *background*, o que é coerente com um regime em que instâncias de delaminação são pouco detectadas e, portanto, não contribuem para a máscara semântica final após a união das instâncias.

Em conjunto, os dados indicam que (i) as arquiteturas estritamente semânticas (*U-Net*, *DeepLabV3–ResNet50* e *SegFormer*) segmentam a classe frágil com alta confiabilidade, enquanto a classe delaminação é limitada por confusões simultâneas com a classe *background* e frágil; e (ii) no *YOLO11l-seg*, os erros dominantes para ambas as classes de interesse são do tipo classe→*background*, caracterizando subsegmentação no formato semântico final. Esses achados motivam a análise qualitativa da próxima seção, na qual são inspecionados exemplos visuais para contextualizar como tais padrões se manifestam em SFs reais e, posteriormente como podem influenciar o cálculo de métricas derivadas.

4.5 Avaliação qualitativa para M2

As análises quantitativas apresentadas nas Seções 4.3 e 4.4 permitem comparar o desempenho médio das arquiteturas e identificar padrões de confusão dominantes. Entretanto, tais métricas não evidenciam diretamente como os erros se manifestam na SF, nem quais morfologias e condições de aquisição estão mais associadas às falhas observadas. Nesta seção, a avaliação é complementada por inspeção qualitativa de exemplos visuais de segmentação, obtidos a partir das inferências dos diferentes modelos após o treinamento, com o objetivo de verificar a coerência visual das máscaras previstas e caracterizar erros típicos por classe.

Para tornar a discussão mais objetiva e reprodutível, os exemplos foram organizados em cinco casos representativos de fratura, todos inéditos para os modelos (ou seja, não fazem parte do conjunto de treinamento ou de validação), seleciona-

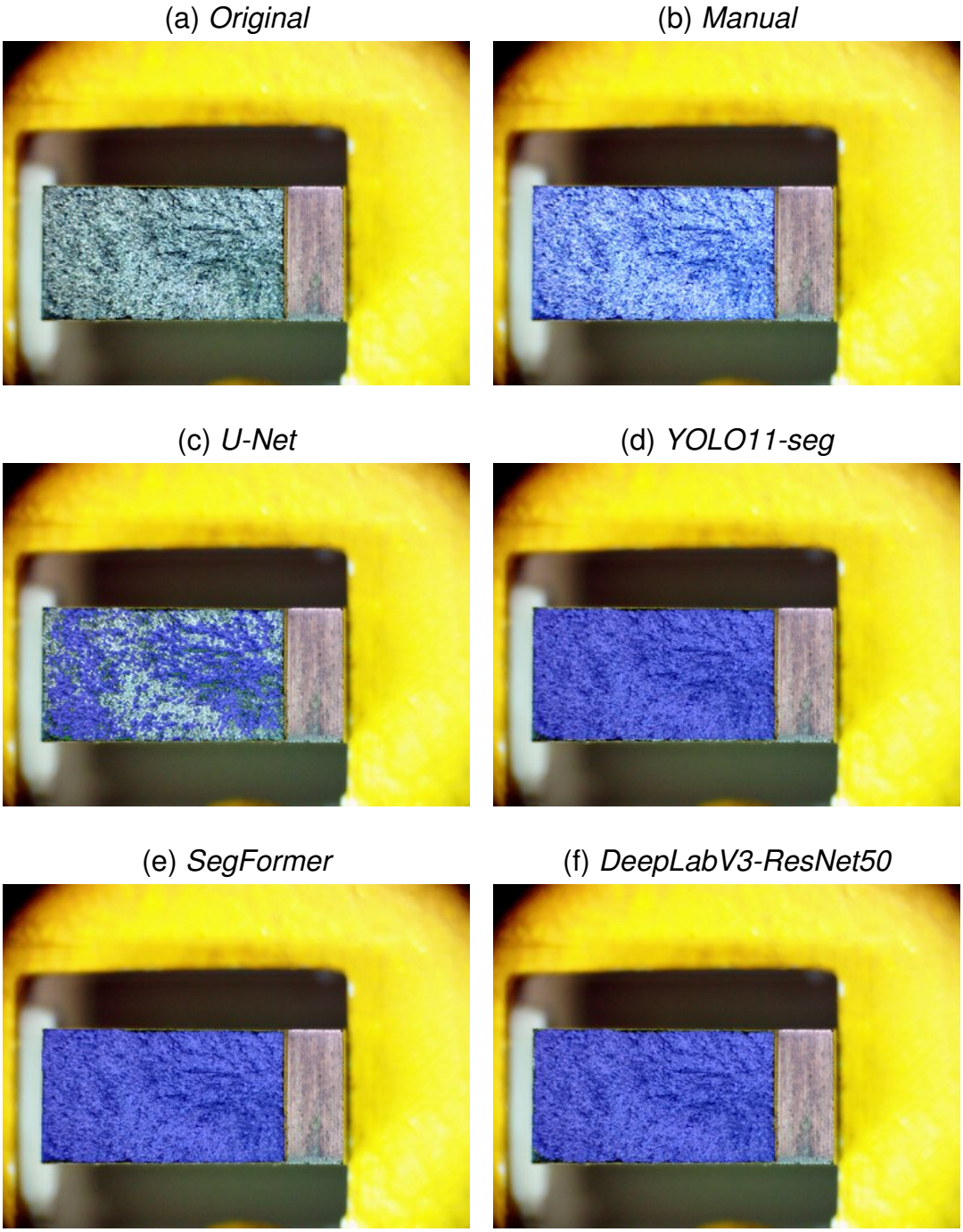
dos por apresentarem padrões recorrentes e por evidenciarem, de maneira clara, situações de sucesso e de falha dos modelos. Em cada caso, seguindo a metodologia proposta, a entrada para esses modelos é imagem recortada da SF, e a saída a imagem original segmentada com região frágil e delaminação, quando presentes.

4.5.1 Fratura frágil

A Figura 35 apresenta um caso de fratura frágil, no qual a máscara de referência indica praticamente toda a área da SF como frágil, sem evidência de delaminações relevantes. Esse cenário é particularmente útil por funcionar como uma condição limite (patamar inferior): como $A_{\text{frágil}} \approx A_{\text{total}}$, erros por subsegmentação da classe frágil (falsos negativos, frágil→*background*) têm impacto direto na área frágil estimada e, consequentemente, na SFA (isto é, reduzir $A_{\text{frágil}}$ tende a inflar a SFA).

Visualmente, *SegFormer* e *DeepLabV3–ResNet50* reproduzem a referência com alta semelhança, preenchendo praticamente toda a SF com o rótulo frágil e preservando o contorno global da área segmentada, comportamento alinhado com o desempenho por classe apresentado anteriormente, no qual a classe frágil apresentou **IoU** e **Dice** elevados. A *YOLO11l-seg* também produz, neste exemplo, uma máscara visualmente consistente e quase totalmente frágil. Isso evidencia que, em situações com fratura frágil dominante e textura relativamente homogênea, a conversão instância→semântica pode manter boa cobertura espacial. Ao mesmo tempo, esse resultado ajuda a contextualizar o contraste com as métricas globais da *YOLO*: a redução observada no desempenho médio na classe frágil (Tabela 11) tende a estar concentrada em casos mais desafiadores (conforme apresentado mais adiante, por exemplo, em CPs na transição e/ou estruturas mais finas), e não necessariamente em condições limite como a apresentada aqui. Por fim, a *U-Net* apresenta regiões pontuais com rotulação divergente, na forma de fragmentação local e ativações espúrias ("ilhas") que se afastam do padrão quase uniforme da referência. Embora o desempenho médio em frágil da *U-Net* seja alto, esse tipo de erro localizado é compatível com o fato de que as métricas agregadas (**IoU**/**Dice**) podem permanecer elevadas mesmo na presença de pequenas regiões incorretamente rotuladas. Além disso, a presença de ativações espúrias em uma imagem

Figura 35 – Exemplo de fratura frágil. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



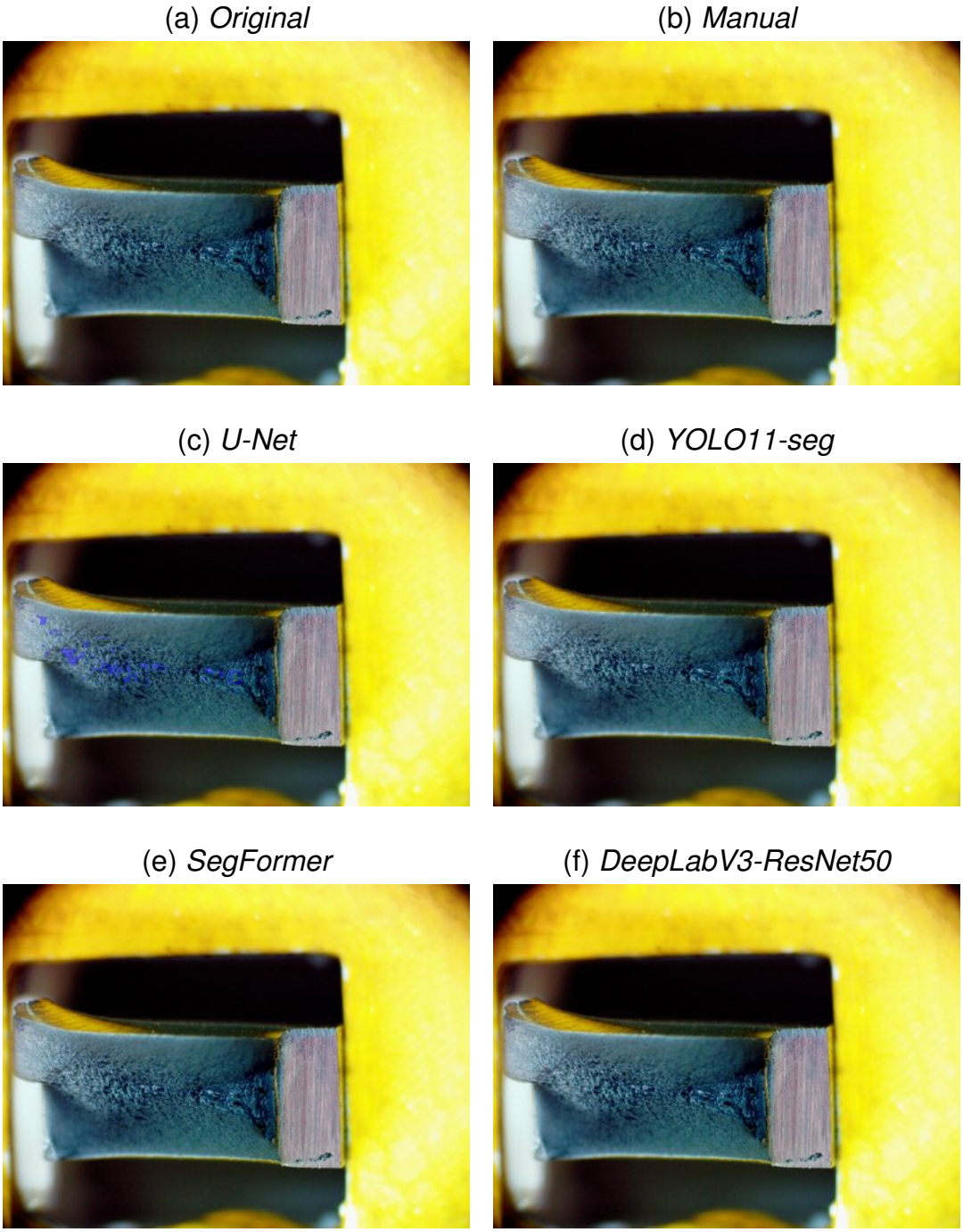
totalmente frágil ilustra, qualitativamente, como falsos positivos em classes minoritárias podem surgir em texturas específicas, o que é coerente com a dificuldade geral observada para delaminação na avaliação quantitativa (baixa precisão e **IoU** nessa classe).

4.5.2 Fratura dúctil

A Figura 36 apresenta um caso de fratura dúctil, no qual a referência manual não indica regiões classificadas como frágil ou delaminação. Esse cenário constitui o extremo oposto ao caso de fratura frágil: como $A_{\text{frágil}} \approx 0$, a SFA de referência tende ao patamar superior (próxima de 100 %), e o erro mais crítico passa a ser a geração de *falsos positivos*. Na prática, qualquer ativação espúria de "frágil" aumenta $A_{\text{frágil}}$ e tende a subestimar a SFA (isto é, reduzir artificialmente a porcentagem de fratura dúctil).

Neste caso, *SegFormer* e *DeepLabV3–ResNet50* preservam a ausência de frágil e delaminação observada na referência, mantendo a região de fratura sem ativações espúrias perceptíveis. Esse comportamento é coerente com as elevadas precisões observadas para a classe frágil, indicando baixa propensão a rotular como frágil padrões de textura que, neste caso, correspondem à fratura dúctil. A *YOLO11l-seg* também apresenta uma saída qualitativamente limpa neste exemplo, sem regiões rotuladas como frágil, sugerindo que, quando não há instâncias convincentes, o pós-processamento instância→semântica tende a resultar em máscara vazia para as classes de interesse (isto é, atribuição predominante ao *background*). Esse tipo de comportamento, embora possa contribuir para subsegmentação em outros cenários (como discutido na Seção 4.4), é desejável em fraturas totalmente dúcteis por reduzir falsos positivos de "frágil" e evitar subestimação da SFA no patamar superior. Em contraste, a *U-Net* exibe pequenas ativações pontuais na região de fratura ("ilhas"), ausentes na referência. Embora localizadas, essas ativações ilustram um aspecto importante: em condições de fratura totalmente dúctil, poucos *pixels* incorretamente rotulados como frágil já são suficientes para introduzir um viés na área frágil estimada, afetando desproporcionalmente a SFA quando o valor verdadeiro de $A_{\text{frágil}}$ é próximo de zero. Esse tipo de observação complementa as métricas médias, pois evidencia que, além do desempenho global em **IoU**/**Dice**,

Figura 36 – Exemplo de fratura dúctil, espera-se que as predições sejam vazias. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



Fonte – Do autor.

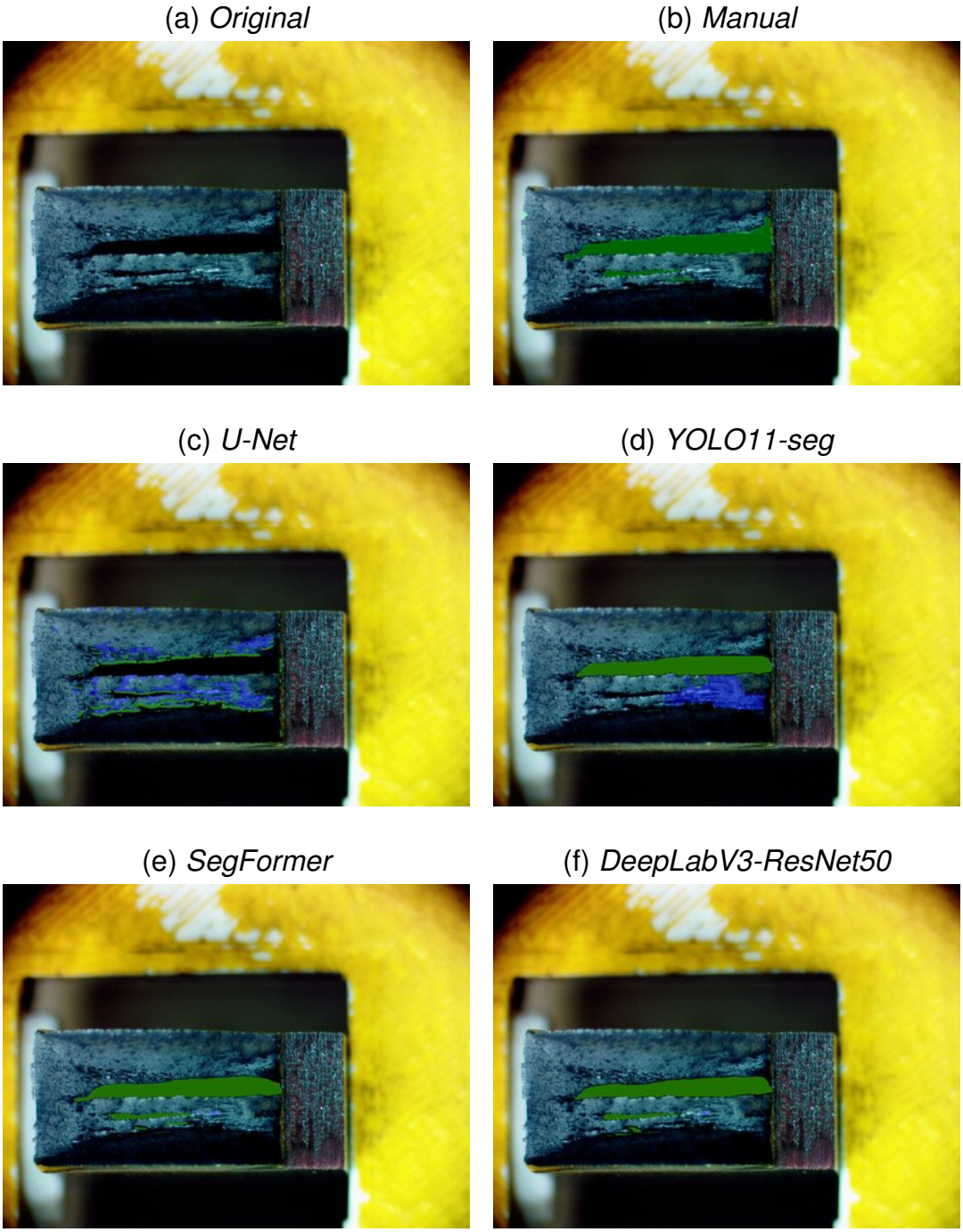
a robustez contra falsos positivos em "frágil" é particularmente relevante.

4.5.3 Fratura dúctil com delaminação

A Figura 37 apresenta um caso de fratura dúctil com presença de delaminação, em que a referência manual indica uma faixa delaminada bem definida, enquanto a classe frágil permanece ausente. Esse cenário é particularmente informativo para discutir a classe limitante identificada na avaliação quantitativa por classe: espera-se que o modelo seja capaz de recuperar a delaminação (evitando falsos negativos delaminação→*background*) sem, ao mesmo tempo, identificar regiões frágeis (falsos positivos de frágil).

Conforme pode ser avaliado, *SegFormer* e *DeepLabV3–ResNet50* recuperam a delaminação com boa continuidade espacial e com pouca ativação fora da região marcada na referência, preservando simultaneamente a ausência de frágil neste caso. Esse comportamento é coerente com a leitura quantitativa de que, embora a delaminação tenha desempenho global baixo, parte relevante do erro decorre de casos difíceis e de estruturas finas; em exemplos com delaminação bem contrastada, a predição tende a ser mais consistente. A *U-Net*, por sua vez, recupera a faixa delaminada, porém apresenta ativações adicionais ao longo da superfície, incluindo regiões rotuladas como frágil que não aparecem na referência manual. Esse padrão ilustra, qualitativamente, como falsos positivos de "frágil" podem emergir mesmo em fraturas dúcteis, o que é particularmente indesejado quando o valor de $A_{\text{frágil}}$ verdadeiro é próximo de zero: poucos *pixels* indevidamente classificados como frágil já são suficientes para introduzir viés na área frágil estimada e subestimar a SFA. No caso da *YOLO11l-seg*, observa-se boa recuperação da delaminação principal, porém com pequenas discrepâncias locais e sinais de rotulação residual fora da região delaminada. Esse exemplo é útil para contextualizar o desempenho médio inferior da *YOLO* em delaminação apresentado: a baixa performance agregada discutida anteriormente não implica falha sistemática em todas as amostras, mas sim maior sensibilidade do modelo a variações de morfologia/contraste e à própria etapa de pós-processamento instância→semântica, que pode amplificar omissões quando a cobertura por instâncias é incompleta.

Figura 37 – Exemplo de fratura dúctil com delaminações aparentes. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



Fonte – Do autor.

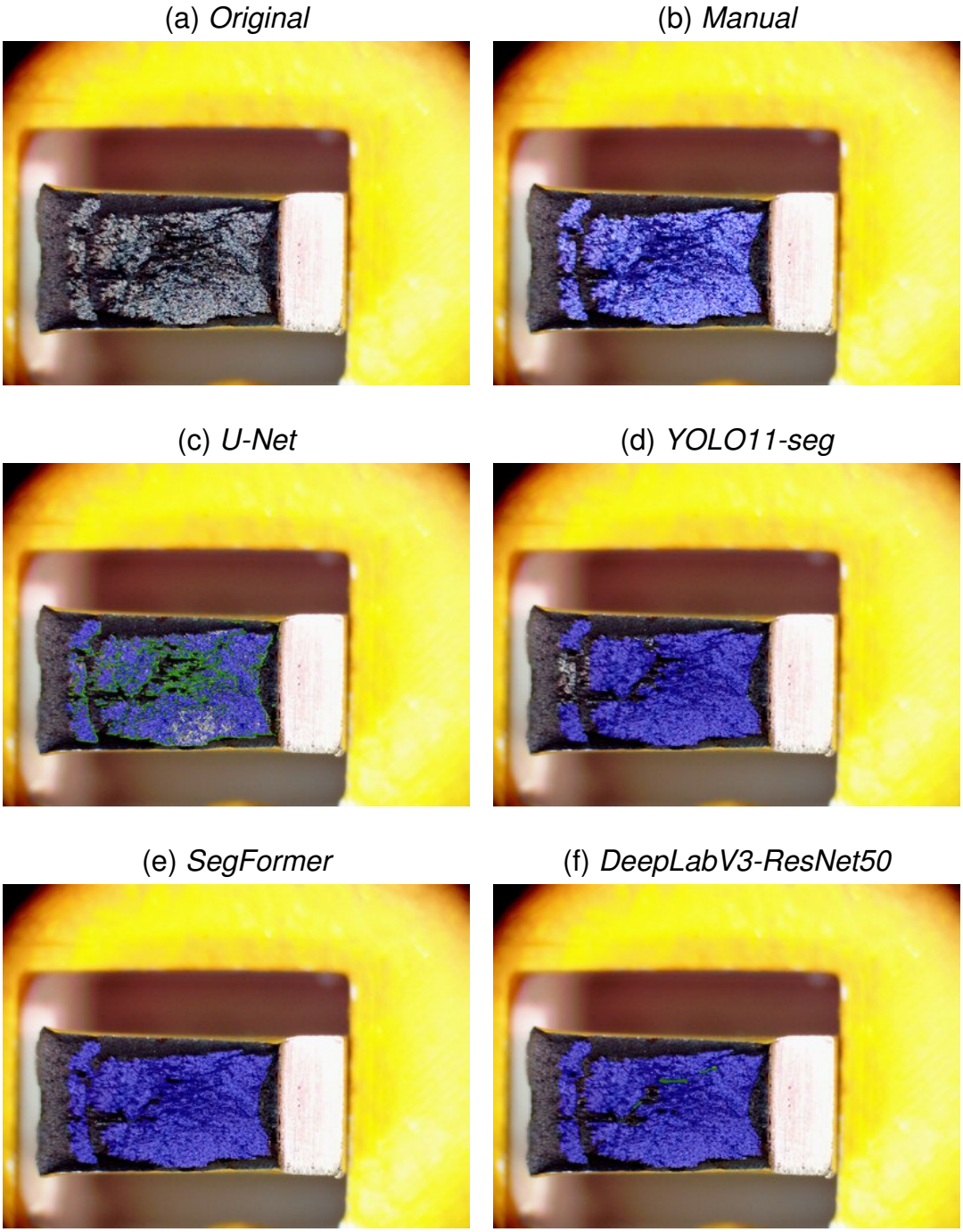
4.5.4 Fratura em transição

A Figura 38 apresenta um exemplo de fratura em transição, no qual coexistem regiões dúcteis e frágeis na SF, caracterizando um cenário intermediário de SFA (isto é, $0 < A_{\text{frágil}} < A_{\text{total}}$). Esse caso é, em geral, o mais relevante do ponto de vista de aplicação, pois pequenas variações na delimitação da fronteira frágil–dúctil tendem a produzir variações perceptíveis na área frágil estimada e, consequentemente, na SFA, influenciando diretamente o formato da curva SFA × temperatura e a estimativa de parâmetros de transição (por exemplo, a DBTT).

Acordante com a avaliação quantitativa, *SegFormer* e *DeepLabV3–ResNet50* apresentam boa concordância com a referência na separação entre as regiões frágeis e dúcteis, com contorno relativamente contínuo e menor fragmentação. Esse comportamento é consistente com o desempenho global ligeiramente superior da *SegFormer* nas métricas agregadas e com os altos valores de **IoU**/**Dice** da classe frágil discutidos na Seção 4.4. Em casos de transição, em que a classe frágil deixa de ser quase total ou quase nula, diferenças sutis de contorno e pequenas "ilhas" tendem a ter maior peso na métrica e no erro de área, o que ajuda a explicar por que o *ranking* global entre os melhores modelos decorre de ganhos marginais. A *U-Net*, por sua vez, tende a apresentar maior irregularidade de contorno e regiões fragmentadas, com pequenas áreas isoladas de frágil e/ou erosões locais na fronteira frágil–dúctil. Embora essas discrepâncias possam ser pouco expressivas em termos absolutos de *pixels*, elas são relevantes por ocorrerem justamente na interface entre regiões, onde a ambiguidade visual é maior; por isso, podem contribuir para oscilações de **mIoU**/**mDice** ao longo do treinamento, como observado nas curvas das Figuras 30 e 31. No caso da *YOLO11l-seg*, observa-se maior tendência à subsegmentação na região frágil, com porções da área que a referência indica como frágil sendo atribuídas ao *background* na máscara final. Esse padrão é coerente com a análise quantitativa por classe e com as matrizes de confusão, nas quais a *YOLO* apresentou confusão dominante frágil→*background*, característica que tende a se tornar mais evidente em fraturas de transição, onde a região frágil é menos extensa e pode depender de instâncias detectadas com maior sensibilidade a limiares e pós-processamento.

No geral, este exemplo reforça que, na região de transição, o desempenho

Figura 38 – Exemplo de fratura em transição. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



Fonte – Do autor.

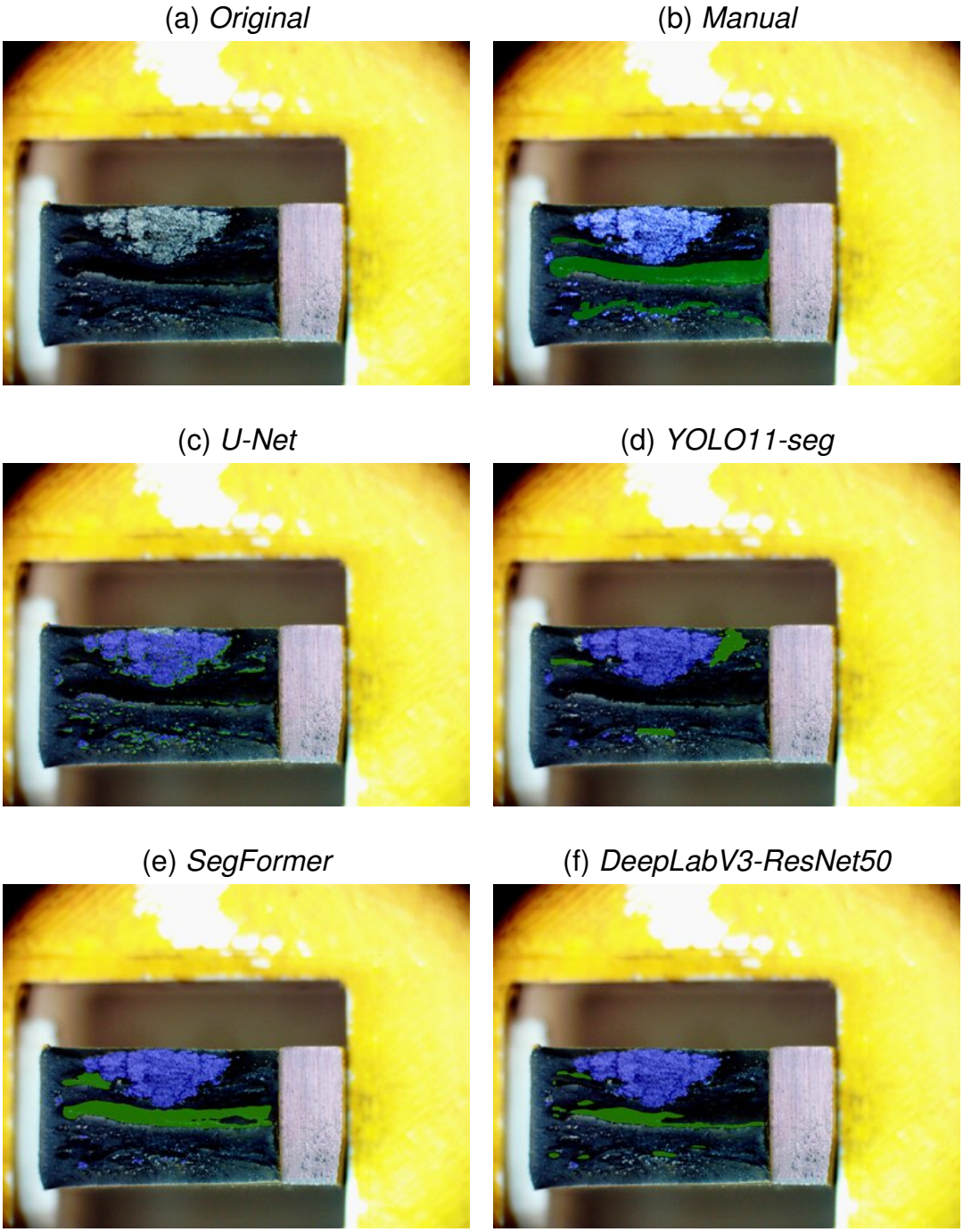
visual está fortemente associado à qualidade do contorno e à coerência espacial das máscaras, aspectos que não são totalmente capturados por um único valor agregado, mas que se conectam diretamente à estimativa de área frágil e ao comportamento da SFA. Na subseção seguinte, considera-se o caso de fratura em transição com delaminação, que adiciona uma fonte adicional de ambiguidade e confusão entre classes.

4.5.5 Fratura em transição com delaminação

A Figura 39 apresenta um exemplo de fratura em transição com presença de delaminação, combinando dois fatores que aumentam a complexidade do problema: a coexistência de regiões frágeis e dúcteis e a ocorrência de delaminações, tipicamente manifestadas como faixas estreitas e alongadas. Do ponto de vista aplicado, este é um cenário crítico, pois discrepâncias localizadas nos contorno afetam diretamente $A_{\text{frágil}}$, enquanto falhas na delaminação evidenciam o comportamento limitante dessa classe observado na avaliação quantitativa por classe, descrito anteriormente.

Comparando-se as predições com a referência, nota-se que *SegFormer* e *DeepLabV3–ResNet50* recuperam de forma mais coerente a estrutura principal de delaminação, preservando simultaneamente a região frágil no topo, com contorno mais contínuo. Esse comportamento é compatível com o *trade-off* observado quantitativamente: embora a delaminação apresente **IoU**/**Dice** baixos no geral, quando ela se manifesta como uma faixa bem definida, esses modelos tendem a manter maior continuidade espacial e menor fragmentação, reduzindo falsos negativos. Em contraste, a *U-Net*, apresenta delaminação mais fragmentada e pontual, com detecções de delaminação dispersas e menor recuperação da faixa contínua observada na referência. Esse padrão é consistente com a dificuldade geral da classe delaminação (baixa **IoU**/precisão) e com a tendência de a delaminação ser parcialmente "absorvida" por outras classes ou se perder em estruturas estreitas, como discutido na análise das matrizes de confusão. No caso da *YOLO11l-seg*, observa-se uma recuperação significativamente mais limitada da delaminação, com detecções parciais e ausência de cobertura da faixa principal em grande parte da sua extensão. Essa omissão é esperada ao se analisar os os resultados quantitativos, nos

Figura 39 – Exemplo de fratura em transição com presença de delaminações. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



Fonte – Do autor.

quais a *YOLO* apresentou menor *recall* em delaminação e um padrão dominante de delaminação→*background* nas matrizes normalizadas. Além disso, em um contexto de transição, a não detecção de delaminações pode se combinar com ajustes de borda na região frágil, contribuindo para variações em $A_{\text{frágil}}$.

Em conjunto, este exemplo reforça que a região de transição com delaminação concentra os mecanismos de erro mais relevantes observados nas métricas: variações de contorno e coerência espacial na classe frágil e dificuldade persistente em delaminação, frequentemente por omissão ou fragmentação.

Na seção seguinte, esses resultados são conectados explicitamente às métricas derivadas, discutindo como as diferenças entre máscaras se propagam para a estimativa da SFA e, posteriormente, parâmetros de transição como a DBTT.

4.6 Avaliação de métricas derivadas

As métricas de segmentação discutidas nas seções anteriores quantificam a qualidade das máscaras no nível de *pixels* e são importantes para comparar arquiteturas sob um protocolo comum. Entretanto, a metodologia apresentada propõe não apenas produzir segmentações visualmente, mas também utilizar essas máscaras para estimar grandezas derivadas associadas ao ensaio Charpy, em particular a SFA e parâmetros extraídos da curva SFA×temperatura, como a DBTT e patamares superiores e inferiores. Assim, esta seção avalia o desempenho das abordagens sob uma perspectiva voltada à essas métricas calculadas a partir das predições dos modelos, investigando como discrepâncias nas máscaras previstas se propagam para erros em áreas por classe.

4.6.1 Avaliação da SFA

Com o objetivo de conectar a qualidade das segmentações ao desempenho dos modelos a nível de grandezas físicas, esta subseção compara a SFA obtida nos cinco exemplos discutidos na Seção 4.5. Para cada imagem, a SFA foi calculada por três vias: (i) método do NIST (descrito no Capítulo 2), (ii) referência manual (a partir de máscaras anotadas) e (iii) máscaras inferidas pelas quatro arquiteturas

avaliadas. A Tabela 12 reporta os valores de SFA e o erro absoluto em relação à referência manual, em pontos percentuais (p.p.).

Tabela 12 – Comparação da estimativa da SFA (%) nos exemplos da Seção 4.5. O erro é o erro absoluto em relação ao manual, em pontos percentuais (p.p.).

Imagem	**SFA manual**	**SFA NIST**		**SFA *U-Net***		**SFA *SegFormer***		***DeepLab-ResNet50***		***YOLO11l-seg***	
	Valor (%)	**Valor (%)**	**Erro (p.p.)**	**Valor (%)**	**Erro (p.p.)**	**Valor (%)**	**Erro (p.p.)**	**Valor (%)**	**Erro (p.p.)**	**Valor (%)**	**Erro (p.p.)**
Frágil (Fig. 35)	0,0	19,3	19,3	48,0	48,0	**2,3**	**2,3**	3,6	3,6	4,0	4,0
Dúctil (Fig. 36)	100,0	100,0	0,0	96,6	3,4	**100,0**	**0,0**	**100,0**	**0,0**	10,0	90,0
Dúctil c/ delaminação (Fig. 37)	100,0	100,0	0,0	90,5	9,5	**99,9**	**0,1**	**99,9**	**0,1**	93,9	6,1
Transição (Fig. 38)	41,4	55,5	14,1	65,3	23,9	**41,8**	**0,4**	43,4	2,0	47,4	6,0
Transição c/ delaminação (Fig. 39)	83,5	100,0	16,5	85,2	1,7	**83,9**	**0,4**	84,3	0,8	**83,1**	**0,4**

Fonte – Do autor.

Os resultados da Tabela 12 são consistentes com a interpretação geométrica da SFA. No caso de fratura frágil, em que $\mathbf{SFA}_{\text{manual}} = 0$ %, qualquer subsegmentação da classe frágil (redução de $A_{\text{frágil}}$) leva a uma superestimação da SFA. Isso é observado principalmente na *U-Net* (+48,0 p.p.) e, em menor grau, no método do NIST (+19,3 p.p.), enquanto *SegFormer* (+2,3 p.p.) e *DeepLab-ResNet50* (+3,6 p.p.) permanecem próximos da referência, em linha com os maiores valores de **IoU**/**Dice** em "frágil" apresentados.

No extremo oposto, na fratura dúctil, o erro dominante é a ocorrência de falsos positivos de frágil, que aumentam $A_{\text{frágil}}$ e subestimam a SFA. Nesse caso, *SegFormer*, *DeepLab-ResNet50* e NIST preservam o patamar superior, e o *U-Net* apresenta pequena subestimação (-3,4 p.p.). Já a *YOLO11l-seg* apresenta um desvio pronunciado (-90,0 p.p.), indicando que, para este exemplo específico, a máscara semântica final resultou em uma área frágil muito superestimada. Esse comportamento reforça que, além das médias globais, a variabilidade caso a caso pode ser relevante quando a SFA se encontra em regiões de patamar (próxima de 0% ou 100%), nas quais poucos erros na classe frágil podem induzir grandes desvios relativos.

Nos casos de transição (com e sem delaminação), as diferenças entre métodos passam a refletir principalmente a qualidade da delimitação do contorno das regiões, justamente a região mais sensível para a construção da curva SFA × temperatura e para a estimativa da DBTT. Observa-se que *SegFormer* e *DeepLab-ResNet50* mantêm erros baixos (+0,4 a +2,0 p.p.), enquanto NIST e *U-Net* apresentam superestimação mais elevada (+14,1 a +23,9 p.p.), sugerindo subsegmentação sistemática de frágil nesses exemplos. Para a *YOLO11l-seg*, o erro é moderado no caso de transição (+6,0 p.p.) e baixo no caso de transição com delaminação (-0,4 p.p.), indicando que seu desempenho pode ser heterogêneo e fortemente dependente do padrão morfológico presente na imagem. Por fim, nos exemplos com delaminação, vale notar que, embora a delaminação não entre diretamente no cálculo da estimativa da SFA, ela pode influenciar indiretamente quando há confusão entre delaminação e frágil (ou *background*). Assim, discrepâncias de delaminação observadas qualitativamente na Seção 4.5 ajudam a contextualizar diferenças de SFA em imagens nas quais a delaminação está próxima da fronteira de outras

classes ou ocupa regiões estreitas.

4.6.2 Curva SFA×temperatura e estimativa da DBTT

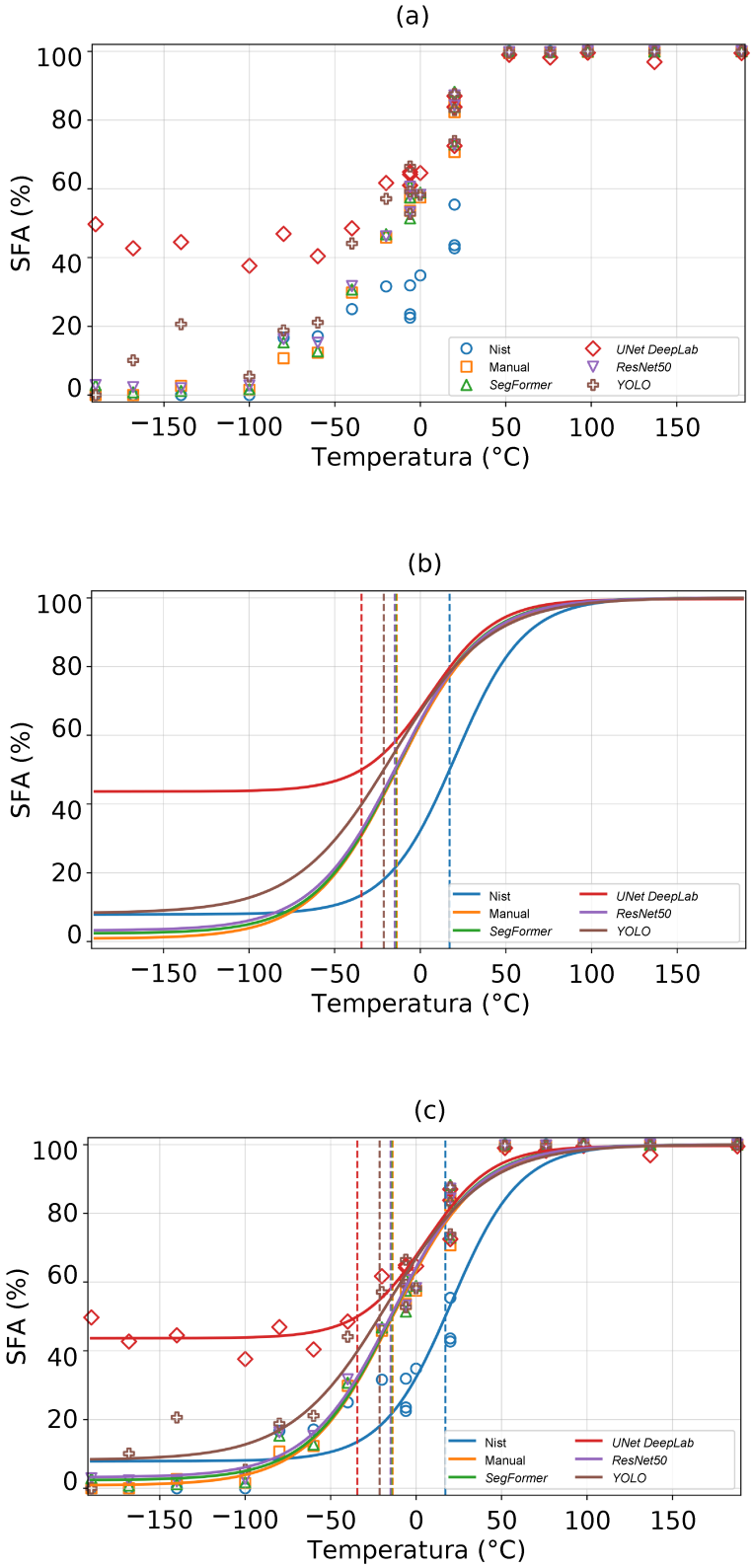
Em seguida, avaliou-se o desempenho das abordagens em um cenário cotidiano de uso, construindo-se a curva SFA × temperatura para um lote de ensaios Charpy do mesmo material (variando apenas a temperatura de ensaio). Para cada CP, a SFA foi estimada por seis abordagens (NIST, manual e os quatro modelos), resultando em 20 pontos individuais distribuídos em 16 temperaturas, de -190 a 188 °C. Os ensaios foram realizados por um equipamento do modelo HIT450P, da fabricante Zwick Roell, cujo certificado de calibração é apresentado no Anexo A. A referência manual desta etapa foi obtida a partir de máscaras anotadas em imagens independentes (não utilizadas nos conjuntos de treino/validação), de modo a refletir o comportamento das abordagens fora do conjunto de desenvolvimento. As 20 imagens da SF, e as segmentações por cada abordagem estão disponíveis no Apêndice B. A Figura 40 apresenta os pontos individuais e os ajustes obtidos para cada método.

Como indicado na norma ISO (2016a), a temperatura de transição pode ser definida por diferentes critérios, incluindo o critério associado à SFA, $T_{t50\%\mathrm{SFA}}$, correspondente a 50 % de SFA. Além disso, a própria norma registra que o modelo de ajuste mais empregado para curvas de transição é o da tangente hiperbólica. Assim, para assegurar uma estimativa consistente entre as seis abordagens e reduzir o efeito de dispersão ponto a ponto, ajustou-se cada conjunto de dados por uma função tangente hiperbólica simétrica, na forma

$$\mathrm{SFA}(T) = A + B\tanh\left(\frac{T - T_0}{C}\right), \tag{4.3}$$

em que T é a temperatura de ensaio, A e B controlam o nível médio e a amplitude entre os patamares inferior e superior, C está associado à largura da região de transição (controlando a inclinação da curva) e T_0 é a abscissa do ponto de inflexão do ajuste. Para fins de comparação, a DBTT foi definida aqui como a temperatura em que a curva ajustada atinge 50 % de SFA. Esse critério é diretamente compatível

Figura 40 – Curvas SFA×temperatura para as seis abordagens avaliadas (NIST, manual e quatro modelos). Em (a) pontos individuais por CP; em (b) curvas ajustadas por tangente hiperbólica simétrica e respectivas DBTTs (linhas tracejadas verticais); em (c) sobreposição dos pontos individuais com as curvas ajustadas.



Fonte – Do autor.

com a definição $T_{t50\%\mathrm{SFA}}$ e independe de eventuais desvios dos patamares. Os valores estimados de DBTT para cada abordagem são apresentados na Tabela 13.

Tabela 13 – DBTT estimada pelo critério $T_{t50\%\mathrm{SFA}}$ para cada abordagem.

Abordagem	**DBTT (°C)**
NIST	17,12
Manual	-13,74
SegFormer	-14,44
DeepLabV3–ResNet50	-14,99
YOLO11l-seg	-21,33
U-Net	-34,36

Observa-se que *SegFormer* e *DeepLabV3–ResNet50* apresentam DBTTs muito próximas da referência manual, indicando boa preservação do posicionamento global da região de transição. Esse resultado é coerente com o desempenho quantitativo elevado na classe frágil e com menor incidência de erros sistemáticos de contorno nos casos de transição discutidos. Em contraste, a *YOLO11l-seg* tende deslocar a transição para temperaturas mais baixas, sugerindo um viés moderado na estimativa de área frágil ao longo da faixa de transição, compatível com a maior variabilidade observada na avaliação semântica por classe.

O *U-Net* apresenta o maior deslocamento para temperaturas mais baixas. Esse comportamento indica que a curva ajustada do *U-Net* tende a exibir um patamar inferior significativamente acima de 0 % (isto é, SFA elevada mesmo em temperaturas muito baixas), indicando superestimação do valor da SFA em condições onde se esperaria fratura predominantemente frágil. Como consequência, a curva atinge 50 % de SFA em temperaturas mais baixas, deslocando a DBTT para valores menores.

Por outro lado, o método do NIST, pelas equações empíricas, resulta em DBTT significativamente mais alta do que as demais abordagens. Visualmente, isso se manifesta como uma curva mais "tardia", com SFA mais baixa na faixa de transição quando comparada ao manual e aos modelos na Figura 40. Em termos práticos, esse deslocamento implica diferenças relevantes na caracterização do

material quando a DBTT é empregada como parâmetro único de transição, reforçando a importância de avaliar não apenas métricas de segmentação, mas também o impacto do método escolhido na grandeza física.

CAPÍTULO

5

CONCLUSÕES E SUGESTÕES DE TRABALHOS FUTUROS

Este trabalho desenvolveu e avaliou uma abordagem baseada em IA para automatizar a estimativa da SFA em CPs após o ensaio de impacto Charpy, a partir de imagens estereomicroscópicas das superfícies de fratura (SFs). A metodologia foi estruturada como: em um primeiro estágio, o modelo M1 segmentou a SF total; logo após, foram comparadas quatro arquiteturas (*U-Net*, *DeepLabV3–ResNet50*, *SegFormer* e *YOLO11l-seg*), chamadas de M2, para a segmentação das regiões frágeis e de delaminação, permitindo a estimativa da SFA pela razão de áreas. Para garantir comparabilidade, os modelos foram treinados sob protocolo padronizado e a melhor época foi selecionada com base no maior valor de **mIoU** no conjunto de validação (com tolerância absoluta de $\Delta = 1 \times 10^{-4}$), calculado apenas sobre as classes de interesse, excluindo a classe implícita *background*.

Os resultados no nível de máscara mostraram que a classe frágil foi segmentada com desempenho elevado e consistente pelas arquiteturas estritamente semânticas, com valores de **IoU** / **Dice** elevados na melhor época, ao passo que a delaminação permaneceu como o principal gargalo, apresentando métricas substancialmente inferiores em todas as arquiteturas. A análise por classe e as matrizes de confusão indicaram que a delaminação foi frequentemente confundida com *background* e, em menor grau, com frágil, comportamento compatível com sua baixa

representatividade em *pixels* e com sua morfologia tipicamente estreita e alongada. A avaliação qualitativa corroborou esses achados ao evidenciar, em casos típicos de fratura (frágil, dúctil, transição e variantes com delaminação), que as diferenças entre modelos se manifestaram principalmente na estabilidade dos contornos e na capacidade de recuperar delaminações sem fragmentação, além de revelar padrões de subsegmentação e falsos positivos em situações específicas.

A nível de grandezas físicas, a avaliação de métricas derivadas demonstrou que discrepâncias nas máscaras se propagaram para a SFA de forma dependente do regime: em fraturas próximas dos extremos (totalmente frágeis ou totalmente dúcteis), erros relativamente localizados puderam induzir desvios expressivos, enquanto na região de transição pequenas variações de contorno e de área frágil influenciaram a forma da curva SFA× temperatura. Ao ajustar as curvas por tangente hiperbólica simétrica e estimar a DBTT pelo critério $T_{t50\%\mathrm{SFA}}$, observou-se que *SegFormer* e *DeepLabV3–ResNet50* produziram valores de DBTT muito próximos da referência manual (diferenças da ordem de ∼1 °C), enquanto o *YOLO11l-seg* apresentou deslocamento moderado e a *U-Net* apresentou o maior deslocamento para temperaturas mais baixas. O método do NIST, por sua vez, resultou em DBTT substancialmente mais alta do que a obtida manualmente, evidenciando que abordagens distintas de quantificação de SFA podem levar a interpretações significativamente diferentes quando a DBTT é usada como parâmetro resumido de transição.

Como limitações, identificou-se que a classe delaminação foi severamente desbalanceada no conjunto de dados, o que restringiu a capacidade dos modelos de aprender padrões robustos para essa classe e aumentou a sensibilidade a variações de textura e iluminação. Além disso, a avaliação baseada na curva SFA× temperatura foi realizada em um lote específico, com número limitado de corpos de prova em algumas temperaturas, o que restringiu a generalização dos parâmetros ajustados. Também se observou que, no caso da *YOLO11l-seg*, a necessidade de conversão de segmentação por instância para máscara semântica introduziu dependência adicional de limiares e regras de resolução de sobreposição, que podem ter afetado a estabilidade em cenários de baixa evidência visual. Por fim, como M1 foi mantido fixo, erros sistemáticos na definição da ROI não foram explorados como

fonte de incerteza no desempenho final, ainda que possam impactar diretamente a contabilização de áreas.

De forma consolidada, o *ranking* das arquiteturas indicou que a *SegFormer* apresentou o melhor desempenho global no conjunto de validação (maiores **mIoU** / **mDice**), com *U-Net* e *DeepLabV3–ResNet50* virtualmente empatados na sequência, enquanto a *YOLO11l-seg* obteve os menores valores quando avaliado no formato semântico. Sob a ótica das grandezas físicas, ao comparar a DBTT estimada com a referência manual, *SegFormer* e *DeepLabV3–ResNet50* foram os mais consistentes (menores desvios), ao passo que a *YOLO11l-seg* apresentou desvio intermediário e a *U-Net* o maior deslocamento. Assim, considerando conjuntamente desempenho de máscara e impacto nas grandezas derivadas, *SegFormer* e *DeepLabV3–ResNet50* configuraram as alternativas mais adequadas para a aplicação proposta.

Como perspectivas de trabalhos futuros, recomenda-se: (i) ampliar e diversificar o banco de imagens, incluindo maior variabilidade de aquisição e, quando possível, diferentes materiais e condições de ensaio, para avaliar generalização; (ii) investigar estratégias específicas para delaminação, como reamostragem balanceada, perdas robustas a desbalanceamento, treinamento *multi-scale*, supervisão de bordas e pós-processamento morfológico; (iii) incorporar análise de incerteza e calibração de confiança, de modo a quantificar a confiabilidade das áreas estimadas e da SFA em regimes críticos; (iv) reavaliar o desempenho das arquiteturas variando os hiperparâmetros de treinamento, assim como avaliar diferentes valores de confiança nas inferências; e (v) expandir a validação metrológica com lotes maiores e replicações por temperatura, reportando dispersão e intervalos de confiança para a curva ajustada e para a DBTT. Em conjunto, tais aprimoramentos tenderiam a aumentar a robustez do método e sua adequação ao uso prático em rotinas experimentais de caracterização por ensaio Charpy.

REFERÊNCIAS

American Bureau of Shipping (ABS). **Guidance Notes on Fracture Analysis for Marine and Offshore Structures**. Spring, EUA, 2022.

ASM International Handbook Committee. **ASM Handbook, Volume 12: Fractography**. 9. ed. Metals Park, OH: ASM International, 1987. ISBN 9780871700186.

ASTM. **Standard Test Method for Impact Testing of Miniaturized Charpy V-Notch Specimens**. West Conshohocken, PA, 2025. Disponível em: <https://store.astm.org/e2248-25.html>.

_____. **Standard Test Method for Instrumented Impact Testing of Metallic Materials**. West Conshohocken, PA, 2025. Disponível em: <https://store.astm.org/e2298-25.html>.

_____. **Standard Test Methods for Notched Bar Impact Testing of Metallic Materials**. West Conshohocken, PA, 2025. Disponível em: <https://store.astm.org/standards/e23>.

BA, J. L.; KIROS, J. R.; HINTON, G. E. **Layer Normalization**. 2016. ArXiv preprint. Disponível em: <https://arxiv.org/abs/1607.06450>.

BERNHARD, M.; AMOROSO, R.; KINDERMANN, Y.; BARALDI, L.; CUCCHIARA, R.; TRESP, V.; SCHUBERT, M. What's outside the intersection? fine-grained error analysis for semantic segmentation beyond iou. In: **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)**. IEEE Computer Society, 2024. p. 968–977. Disponível em: <https://openaccess.thecvf.com/content/WACV2024/html/Bernhard_Whats_Outside_the_Intersection_Fine-Grained_Error_Analysis_for_Semantic_Segmentation_WACV_2024_paper.html>.

CHEN, L.-C.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. **Rethinking Atrous Convolution for Semantic Image Segmentation**. 2017. ArXiv preprint. Disponível em: <https://arxiv.org/abs/1706.05587>.

COLLINS, W. N.; YOUNT, T. D.; SHERMAN, R. J. *et al.* Dynamic fracture and crack arrest toughness evaluation of high-performance steel used in highway bridges. **Materials**, v. 16, n. 9, p. 3402, 2023.

ETSON. **Comparison of Rules-Making and Practices Concerning Reactor Pressure Vessel Integrity Assessment**. Fontenay-aux-Roses, France, 2018.

EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K. I.; WINN, J.; ZISSERMAN, A. The pascal visual object classes (voc) challenge. **International Journal of Computer Vision**, v. 88, n. 2, p. 303–338, 2010.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 4. ed. New York, NY: Pearson, 2018. ISBN 9780133356724.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA: MIT Press, 2016. ISBN 9780262035613.

HAHIN, C. **Impact and Fracture Toughness Properties of Structural Steels for Design Engineers**. Springfield, EUA, 2022.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, EUA: IEEE, 2016. p. 770–778.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: **Proceedings of the 32nd International Conference on Machine Learning (ICML)**. PMLR, 2015. v. 37, p. 448–456. Disponível em: <https://proceedings.mlr.press/v37/ioffe15.html>.

ISO. **Metallic materials — Charpy pendulum impact test — Part 1: Test method**. Geneva, Switzerland, 2016. Disponível em: <https://www.iso.org/standard/63802.html>.

_____. **Metallic materials — Charpy pendulum impact test — Part 2: Verification of testing machines**. Geneva, Switzerland, 2016. Disponível em: <https://www.iso.org/standard/63812.html>.

_____. **Metallic materials — Charpy V-notch pendulum impact test — Instrumented test method**. Geneva, Switzerland, 2023. Disponível em: <https://www.iso.org/standard/82164.html>.

JACCARD, P. étude comparative de la distribution florale dans une portion des alpes et des jura. **Bulletin de la Société Vaudoise des Sciences Naturelles**, v. 37, p. 547–579, 1901.

JOSÉ, J. P. C.; DUARTE, B. B.; ASSIS, P. A. Q. d.; ARENCIBIA, R. V.; FERREIRA, D. C. F. Desenvolvimento de um modelo de inteligência artificial para segmentação da superfície de fratura de corpos de prova charpy. **Anais do POSMEC 2025 – 33° Simpósio do Programa de Pós-graduação em Engenharia Mecânica**, Uberlândia, MG, Brasil, out. 2025. 22–24 de outubro. Disponível em: <https://www.even3.com.br/posmec2025_ufu-585564/>.

KHANAM, R.; HUSSAIN, M. **YOLOv11: An Overview of the Key Architectural Enhancements**. 2024. ArXiv preprint. Disponível em: <https://arxiv.org/abs/2410.17725>.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: **International Conference on Learning Representations**. ICLR, 2015. Disponível em: <https://arxiv.org/abs/1412.6980>.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in Neural Information Processing Systems**. [s.n.], 2012. v. 25, p. 1106–1114. Disponível em: <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, Nova Jersey, EUA, v. 86, n. 11, p. 2278–2324, 1998.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Boston, EUA: IEEE, 2015. p. 3431–3440.

LUCON, E.; MARTIN, M. L.; CALVO, B. L.; CHEN, X.; SOKOLOV, M. A. Assessment of different approaches for measuring shear fracture appearance in charpy tests. **Journal of Testing and Evaluation**, v. 51, n. 5, p. 2833–2849, 2023.

LUCON, E.; MCCOWAN, C. N. **Impact Testing — Yesterday and Today**. 2011. Disponível em: <https://www.nist.gov/publications/impact-testing-yesterday-and-today>.

Nucor. **What is the Charpy Impact Test and Why is it Important?** 2024. Disponível em: <https://nucor.com/newsroom/what-is-the-charpy-impact-test-and-why-is-it-important>. Acesso em: 03 jan. 2026.

PETROBRAS. **Steel Forging Requirements for Submarine Rigid Pipeline and Riser**. Rio de Janeiro, Brazil, 2022.

PHMSA. **Metallurgical Laboratory Failure Examination Protocol**. Washington, DC, 2019. Created 23/08/2006. Revised 04/04/2019.

PyTorch Contributors. **Automatic Mixed Precision package — torch.amp**. 2025. Documentação do PyTorch. Disponível em: <https://docs.pytorch.org/docs/stable/amp.html>. Acesso em: 14 fev. 2026.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. YOLO: You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. IEEE, 2016. p. 779–788. Disponível em: <https://doi.org/10.1109/CVPR.2016.91>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: Convolutional networks for biomedical image segmentation. In: **Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)**. Munich, Alemanha: Springer, 2015. p. 234–241.

SAPKOTA, R.; FLORES-CALERO, M.; QURESHI, R.; BADGUJAR, C.; NEPAL, U.; POULOSE, A.; ZENO, P.; VADDEVOLU, U. B. P.; KHAN, S.; SHOMAN, M.; YAN, H.; KARKEE, M. Yolo advances to its genesis: a decadal and comprehensive review of the you only look once (yolo) series. **Artificial Intelligence Review**, Springer, v. 58, n. 9, p. 274, 2025. Disponível em: <https://link.springer.com/article/10.1007/s10462-025-11253-3>.

SIEWERT, T. A.; MANAHAN, M. P.; MCCOWAN, C. N.; HOLT, J. M.; MARSH, F. J.; RUTH, E. A. The history and importance of impact testing. In: SIEWERT, T. A.; MANAHAN, M. P. (Ed.). **Pendulum Impact Testing: A Century of Progress**. West Conshohocken, PA: ASTM International, 1999, (ASTM STP 1380).

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, p. 1929–1958, 2014.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. 2. ed. Cham, Switzerland: Springer, 2022. ISBN 9783030343729.

TensorFlow. **Segmentação de Imagem**. 2022. Tutorial. Última atualização: 24 jan. 2022 (UTC). Disponível em: <https://www.tensorflow.org/tutorials/images/segmentation?hl=pt-br>. Acesso em: 14 fev. 2026.

TKACHENKO, M.; MALYUK, M.; HOLMANYUK, A.; LIUBIMOV, N. **Label Studio: Data labeling software**. 2020. Software (open source). Disponível em: <https://labelstud.io>.

TÓTH, L.; ROSSMANITH, H.-P.; SIEWERT, T. A. Historical background and development of the charpy test. In: FRANÇOIS, D.; PINEAU, A. (Ed.). **From Charpy to Present Impact Testing**. Vienna, Áustria: Elsevier Science Ltd. and ESIS, 2002.

Ultralytics. **Ultralytics YOLOv5 Architecture**. 2023. Disponível em: <https://docs.ultralytics.com/yolov5/tutorials/architecture_description/>. Acesso em: 10 jan. 2026.

_____. **Segmentation Tasks in YOLO: Instance and Semantic Mask Prediction**. 2024. Disponível em: <https://docs.ultralytics.com/tasks/segment/>. Acesso em: jan. 2026.

_____. **YOLOv11: Real-Time Object Detection and Segmentation**. 2024. Disponível em: <https://docs.ultralytics.com/models/yolo11/>. Acesso em: jan. 2026.

_____. **Segmentação de instâncias: um guia rápido**. 2025. Disponível em: <https://www.ultralytics.com/pt/blog/what-is-instance-segmentation-a-quick-guide>. Acesso em: 05 jan. 2026.

_____. **YOLO11 vs YOLOv8: Architectural Evolution and Performance Comparison**. 2025. Disponível em: <https://docs.ultralytics.com/compare/yolo11-vs-yolov8/>. Acesso em: 10 jan. 2026.

WEI, S.; ZHANG, H.; WANG, C.; XU, L.; WU, F.; ZHANG, B. Large-scale rice mapping of thailand using sentinel-1 multi-temporal sar data. In: IEEE. **Proceedings of the 2019 SAR in Big Data Era (BIGSARDATA)**. Beijing, China, 2019.

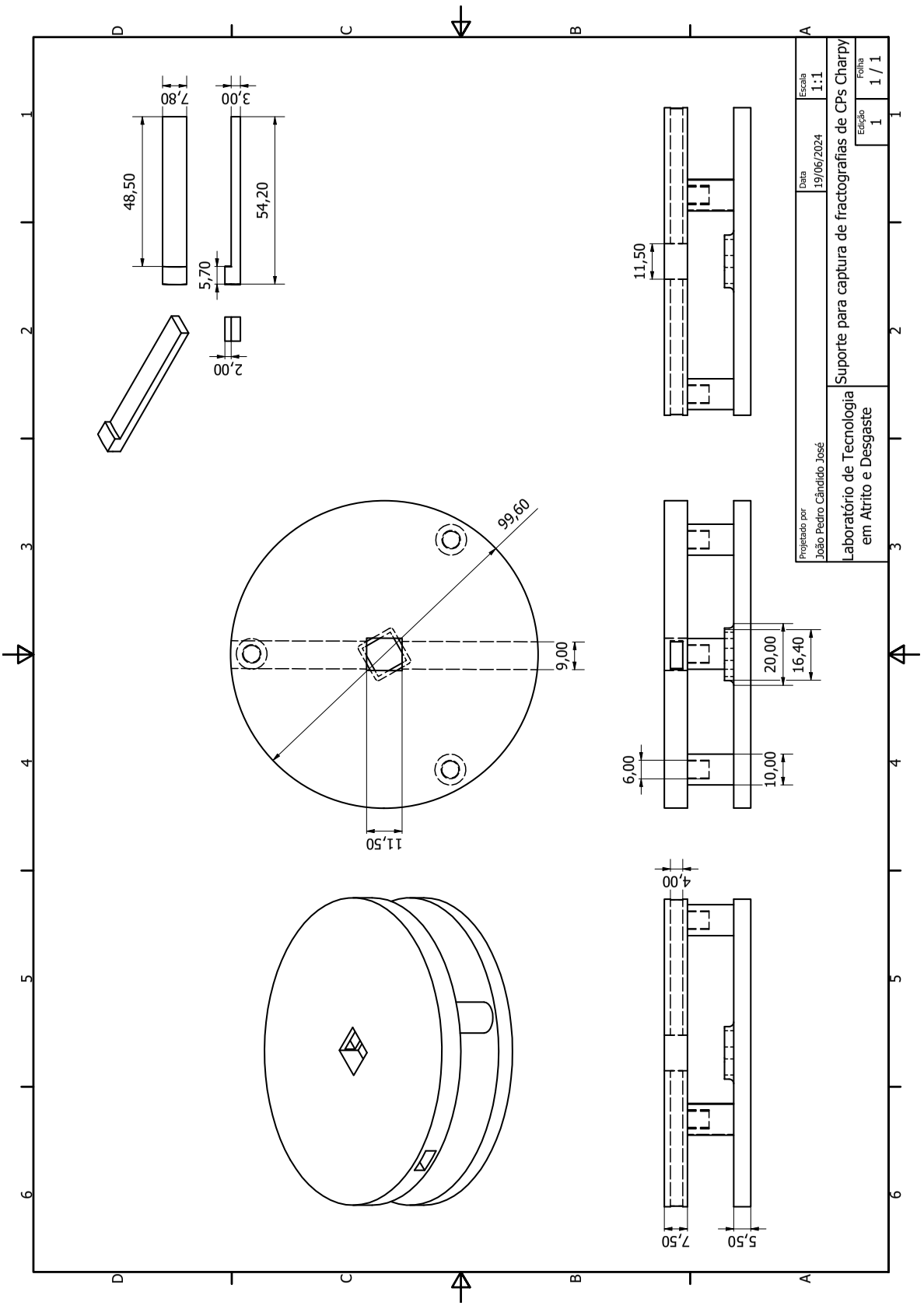
WU, Y.; HE, K. Group normalization. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. Munich, Alemanha: Springer, 2018. p. 3–19.

XIE, E.; WANG, W.; YU, Z.; ANANDKUMAR, A.; ALVAREZ, J. M.; LUO, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. In: **Advances in Neural Information Processing Systems**. Nova Iorque, EUA: NeurIPS, 2021. v. 34, p. 12077–12090.

APÊNDICE

A

DESENHO TÉCNICO DO SUPORTE DESENVOLVIDO PARA A CAPTURA DAS FRACTOGRAFIAS.

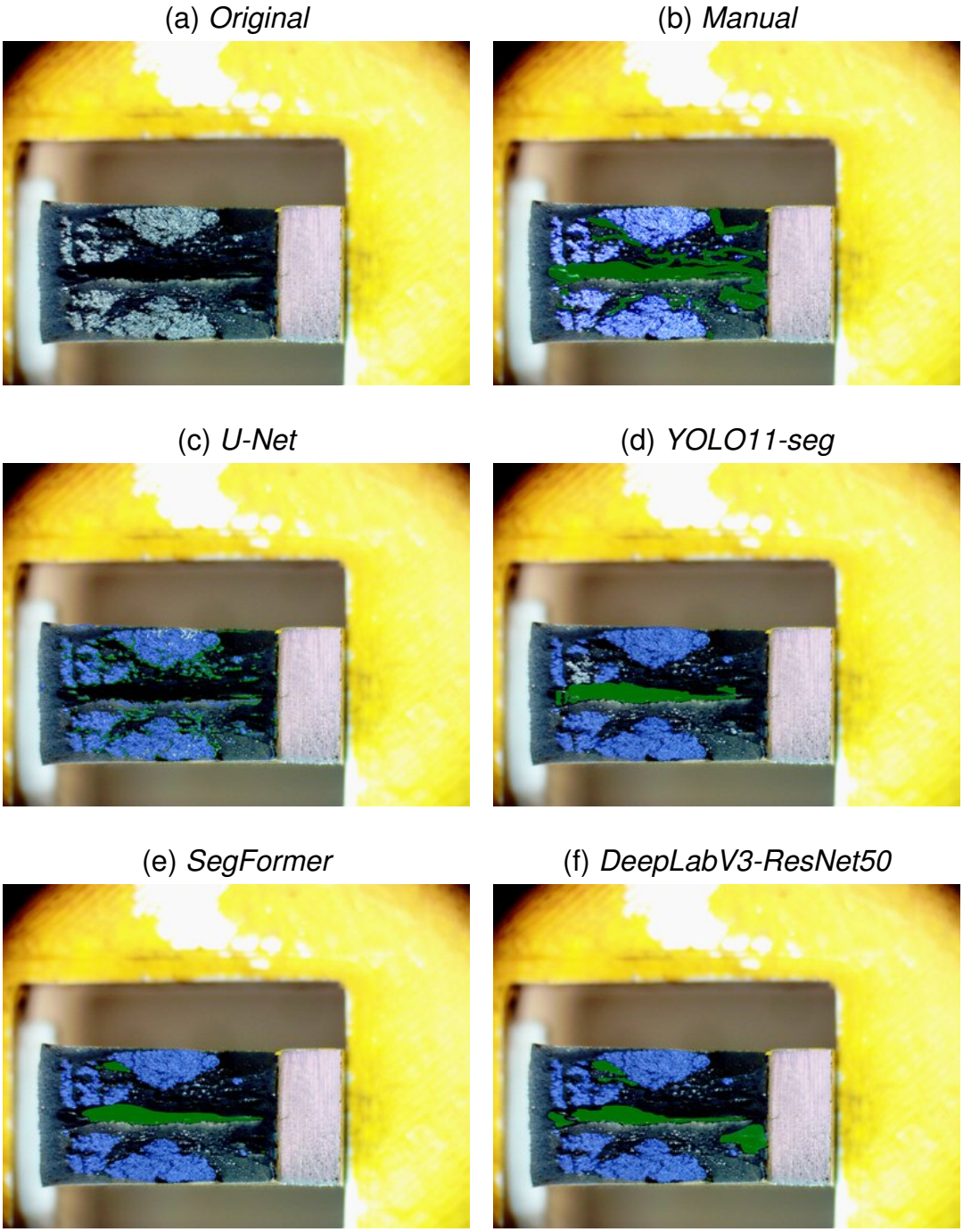

7,80
3,00
48,50
54,20
5,70
2,00
11,50
99,60
9,00
11,50
20,00
16,40
6,00
10,00
4,00
7,50
5,50
Projetado por
João Pedro Cândido José
Data
19/06/2024
Escala
1:1
Edição
1
Folha
1 / 1
Laboratório de Tecnologia em Atrito e Desgaste
Suporte para captura de fractografias de CPs Charpy

APÊNDICE

B

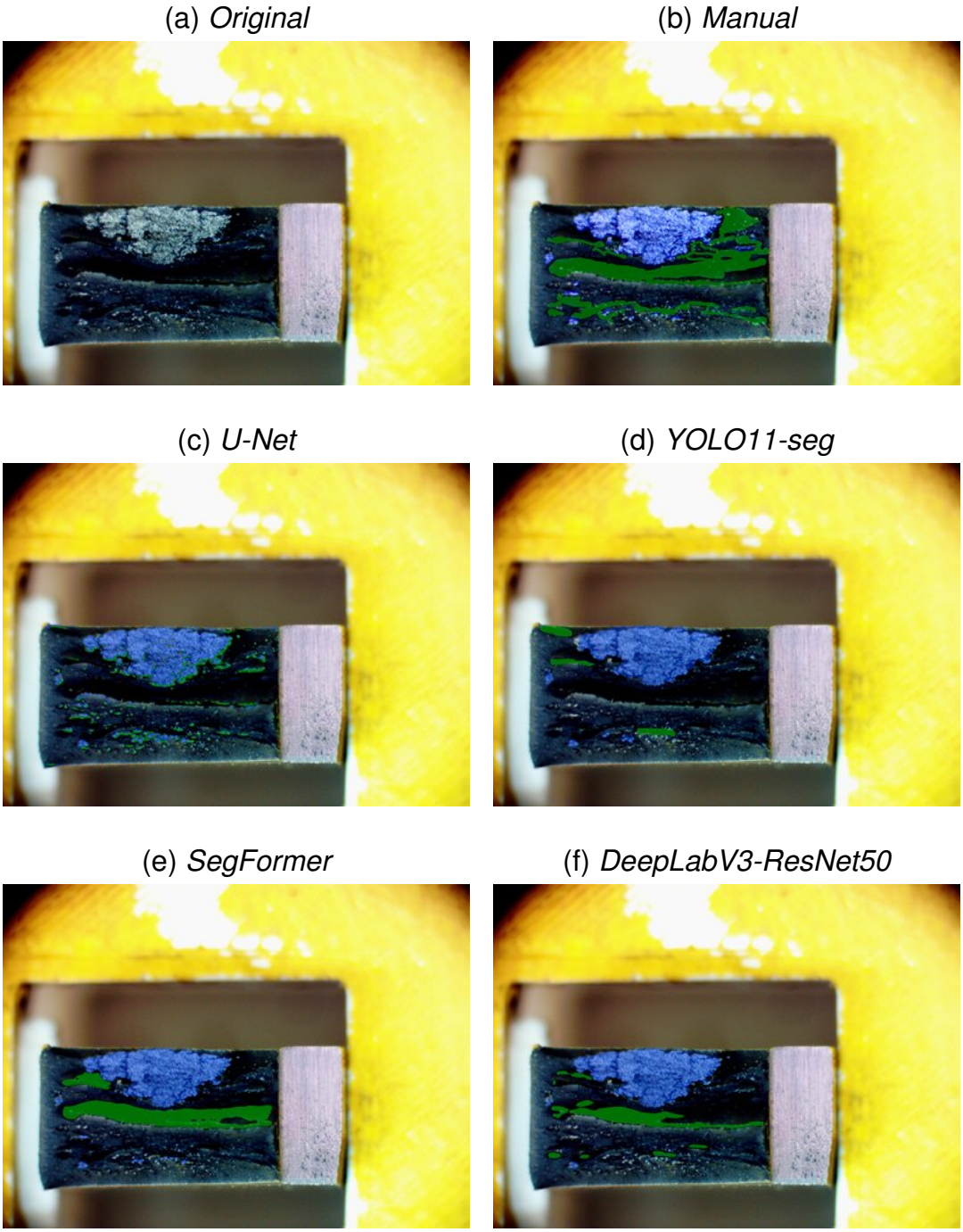
SEGMENTAÇÃO DOS CPS UTILIZADOS NA AVALIAÇÃO DA SEÇÃO 4.6

Figura 41 – segmentação da SF do CP 1, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



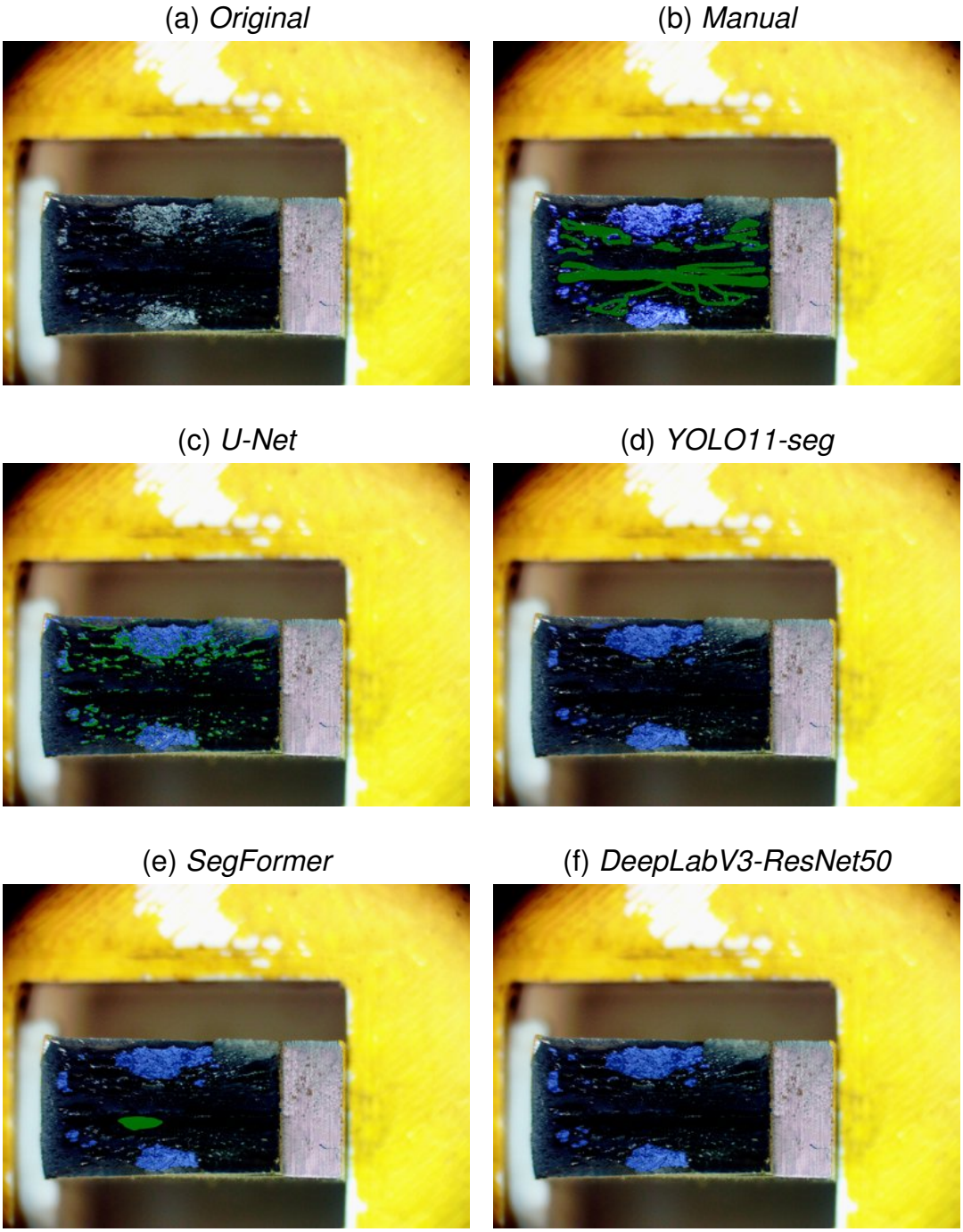
Fonte – Do autor.

Figura 42 – segmentação da SF do CP 2, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



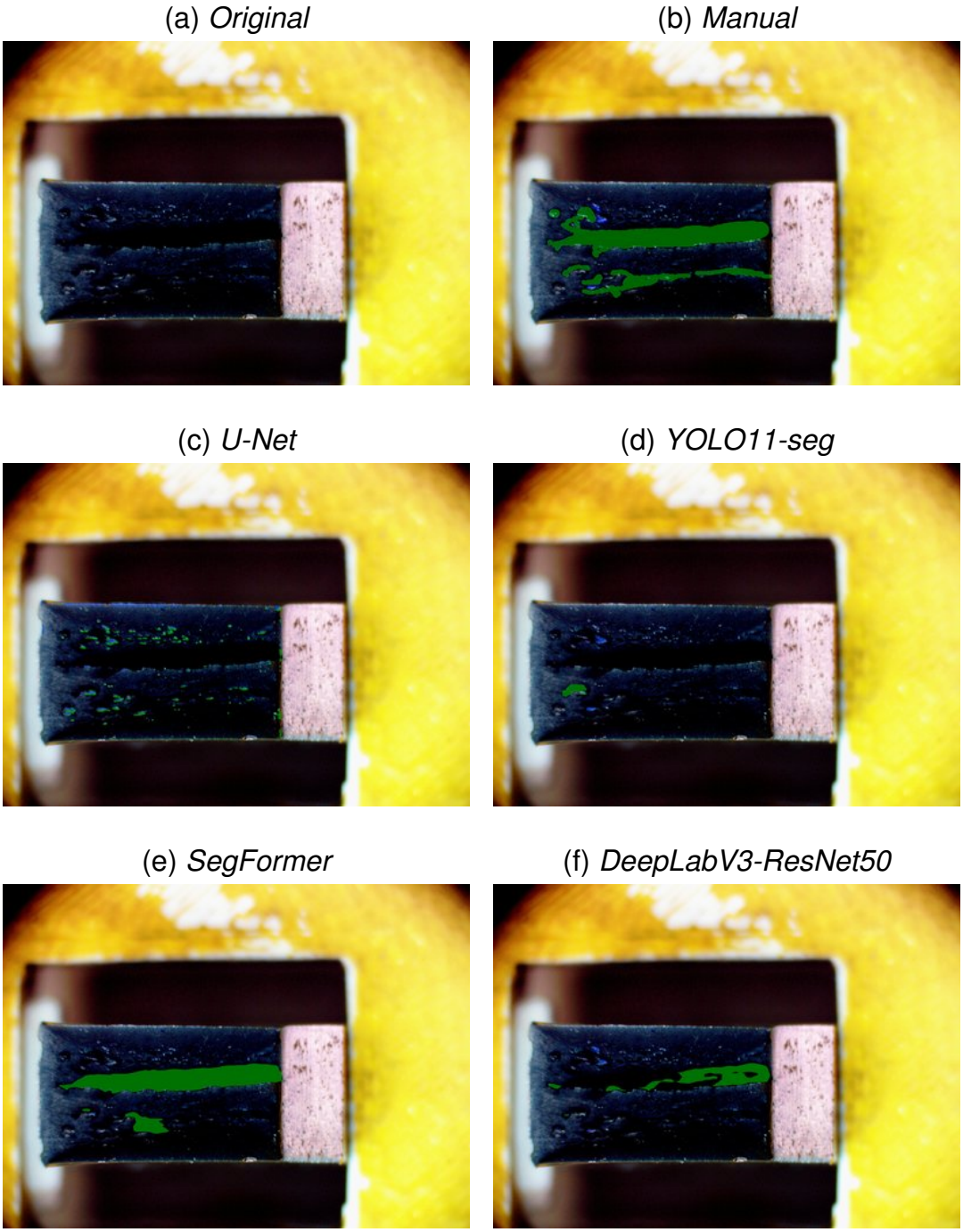
Fonte – Do autor.

Figura 43 – segmentação da SF do CP 3, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



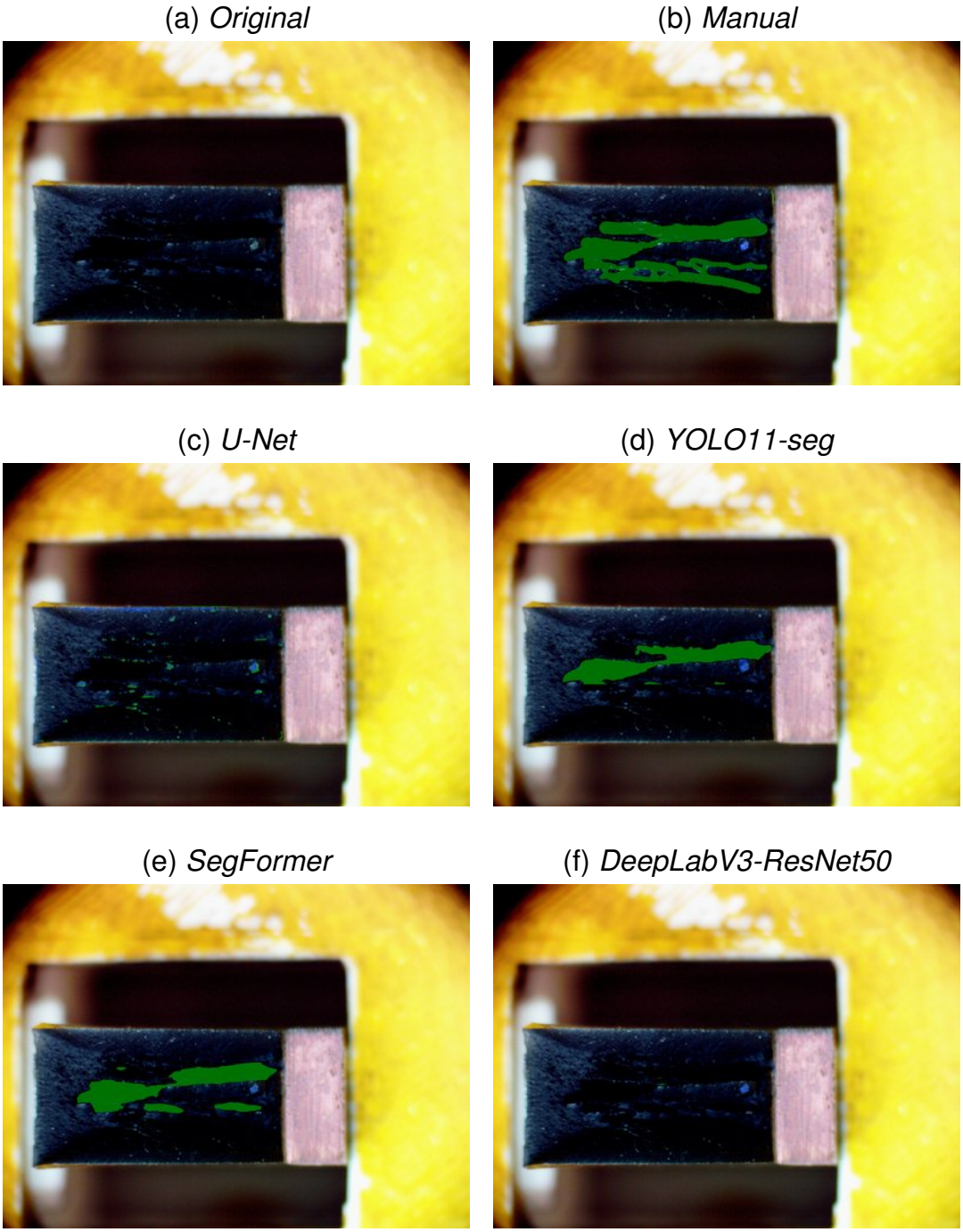
Fonte – Do autor.

Figura 44 – segmentação da SF do CP 4, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



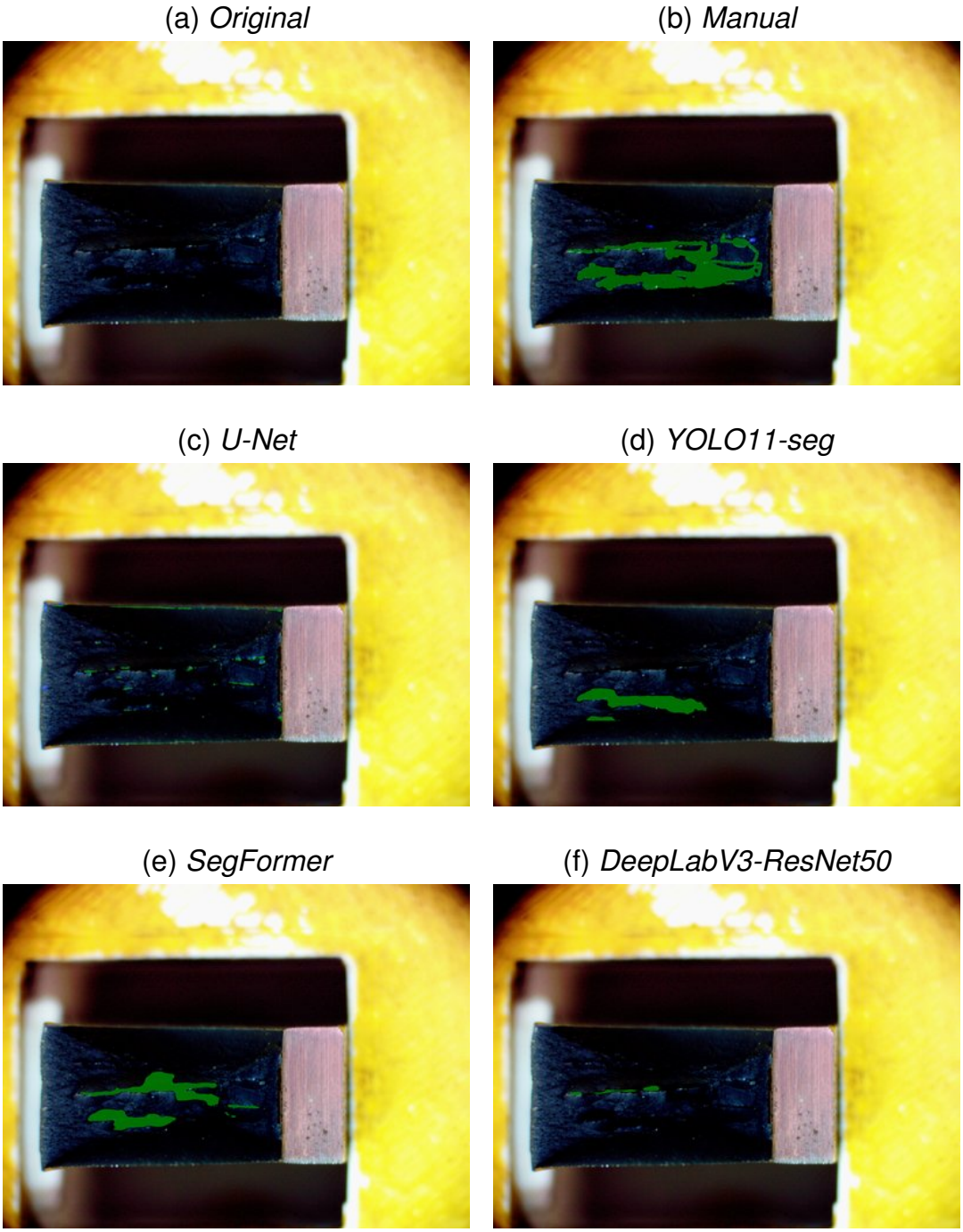
Fonte – Do autor.

Figura 45 – segmentação da SF do CP 5, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



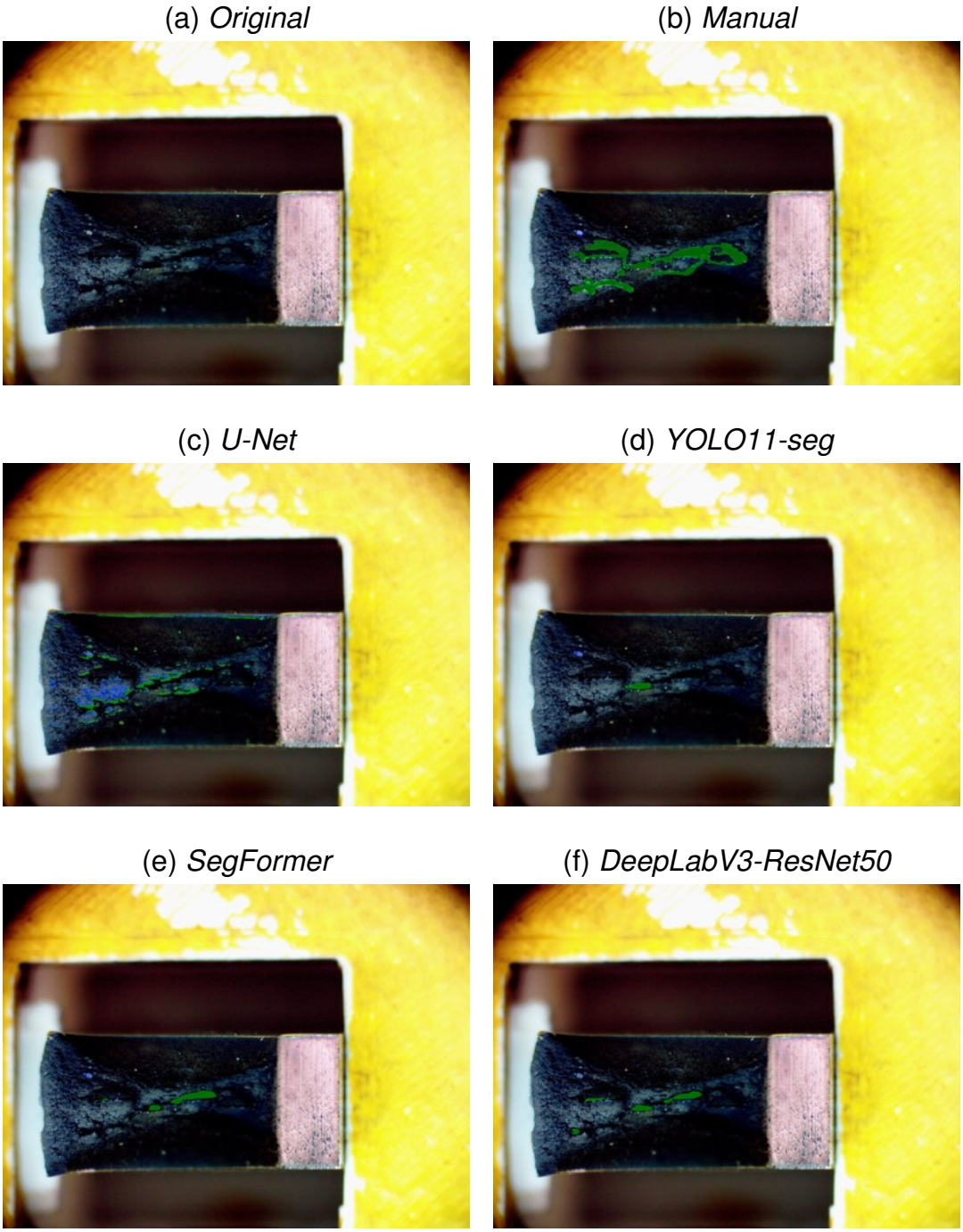
Fonte – Do autor.

Figura 46 – segmentação da SF do CP 6, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



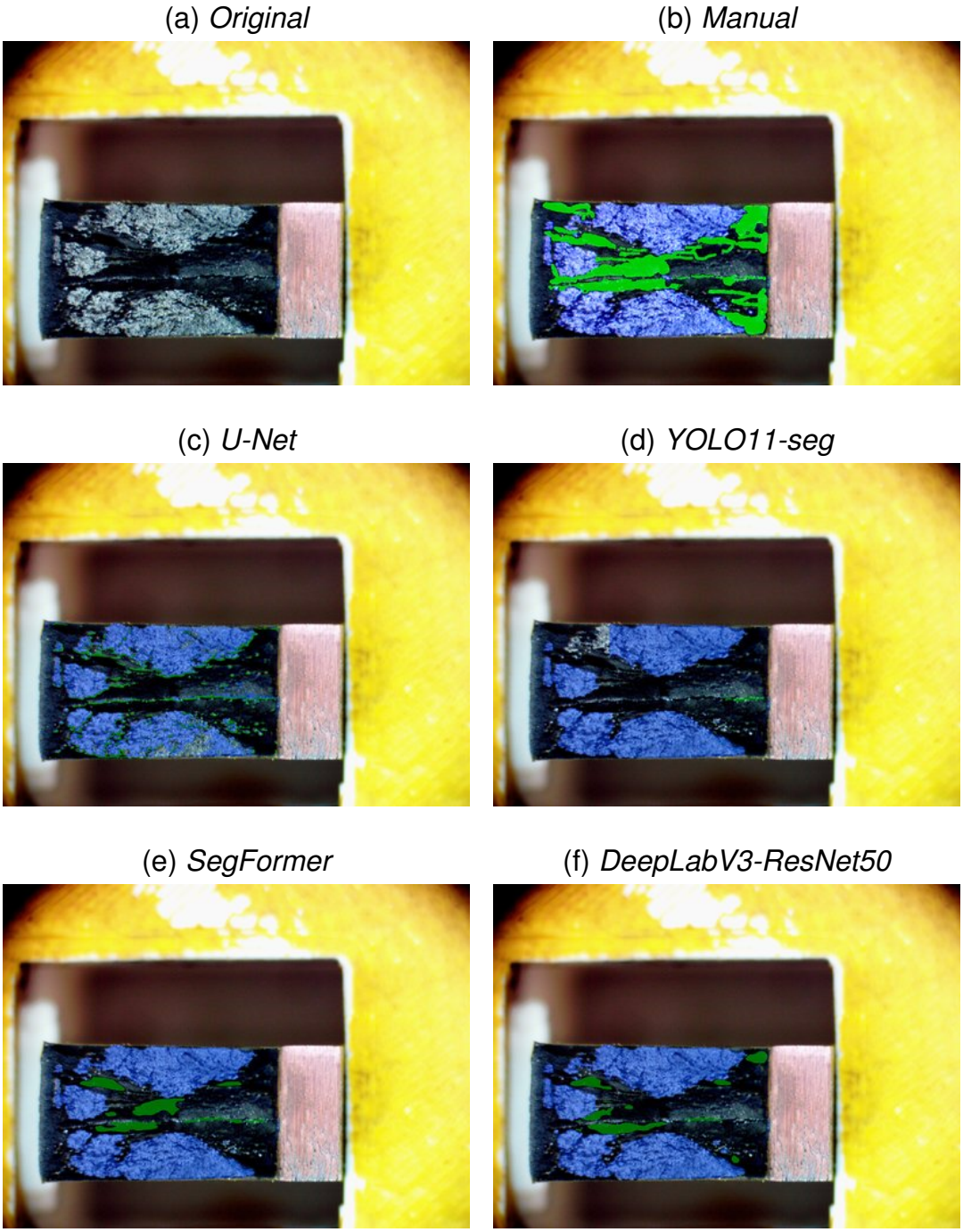
Fonte – Do autor.

Figura 47 – segmentação da SF do CP 7, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



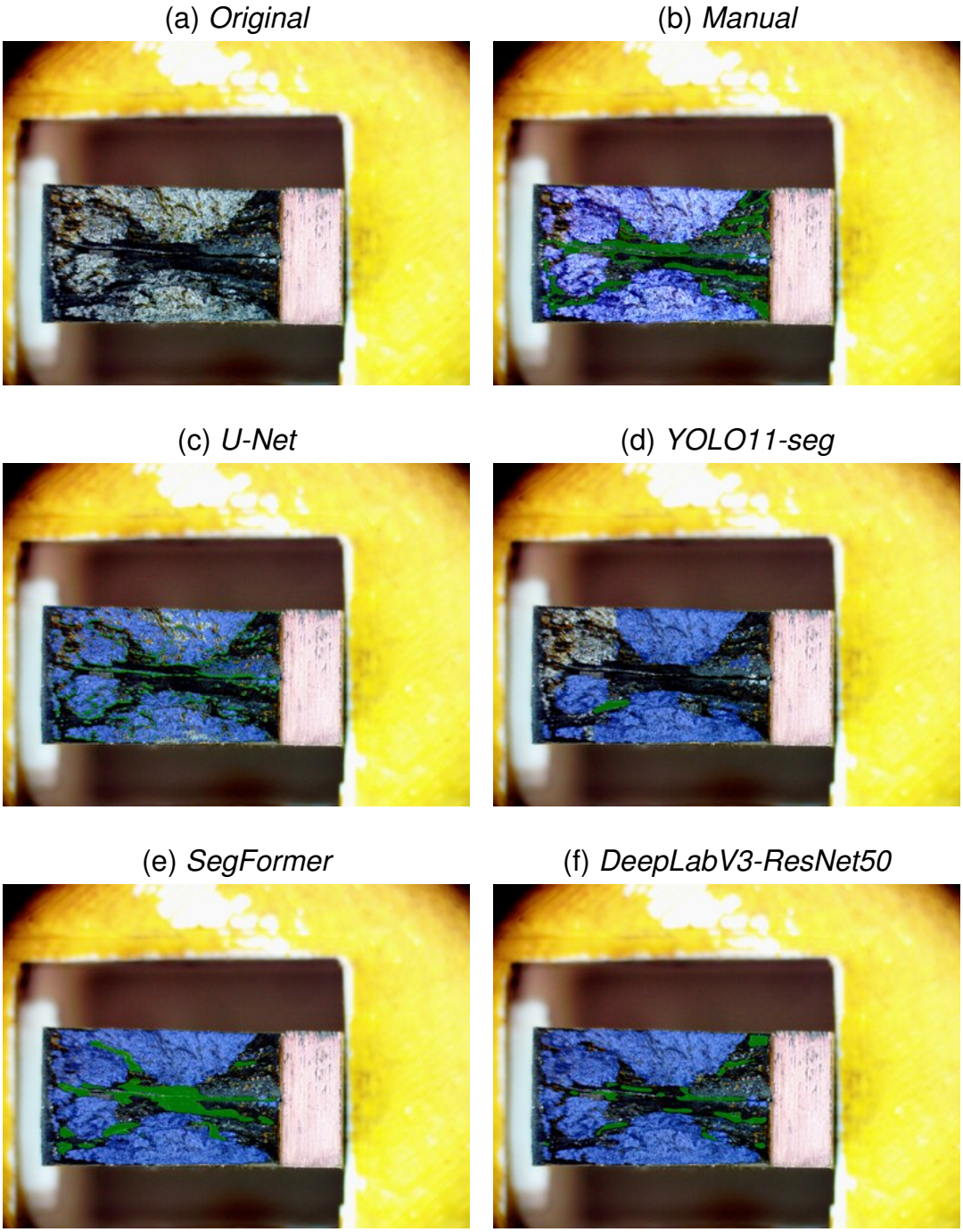
Fonte – Do autor.

Figura 48 – segmentação da SF do CP 8, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



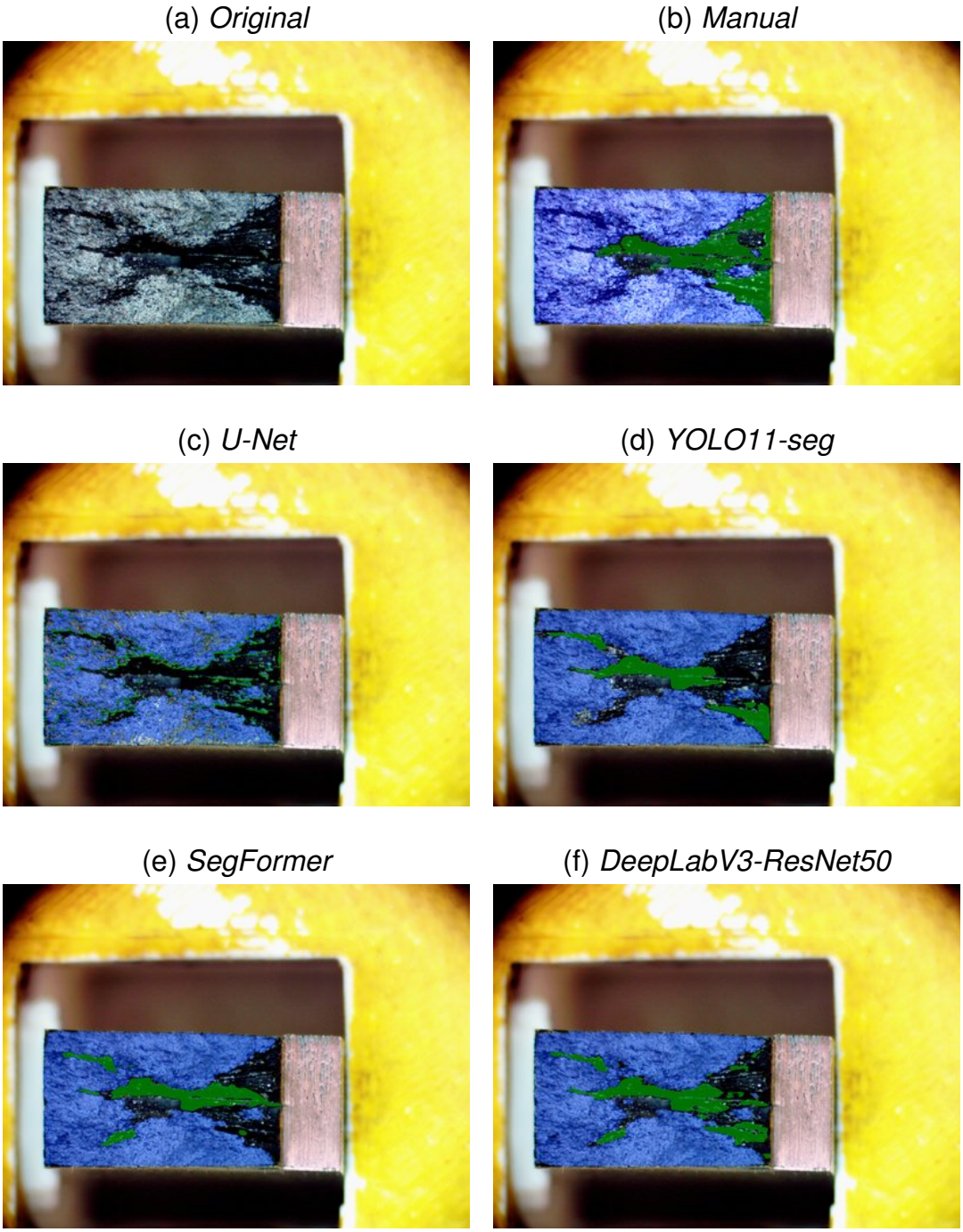
Fonte – Do autor.

Figura 49 – segmentação da SF do CP 9, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



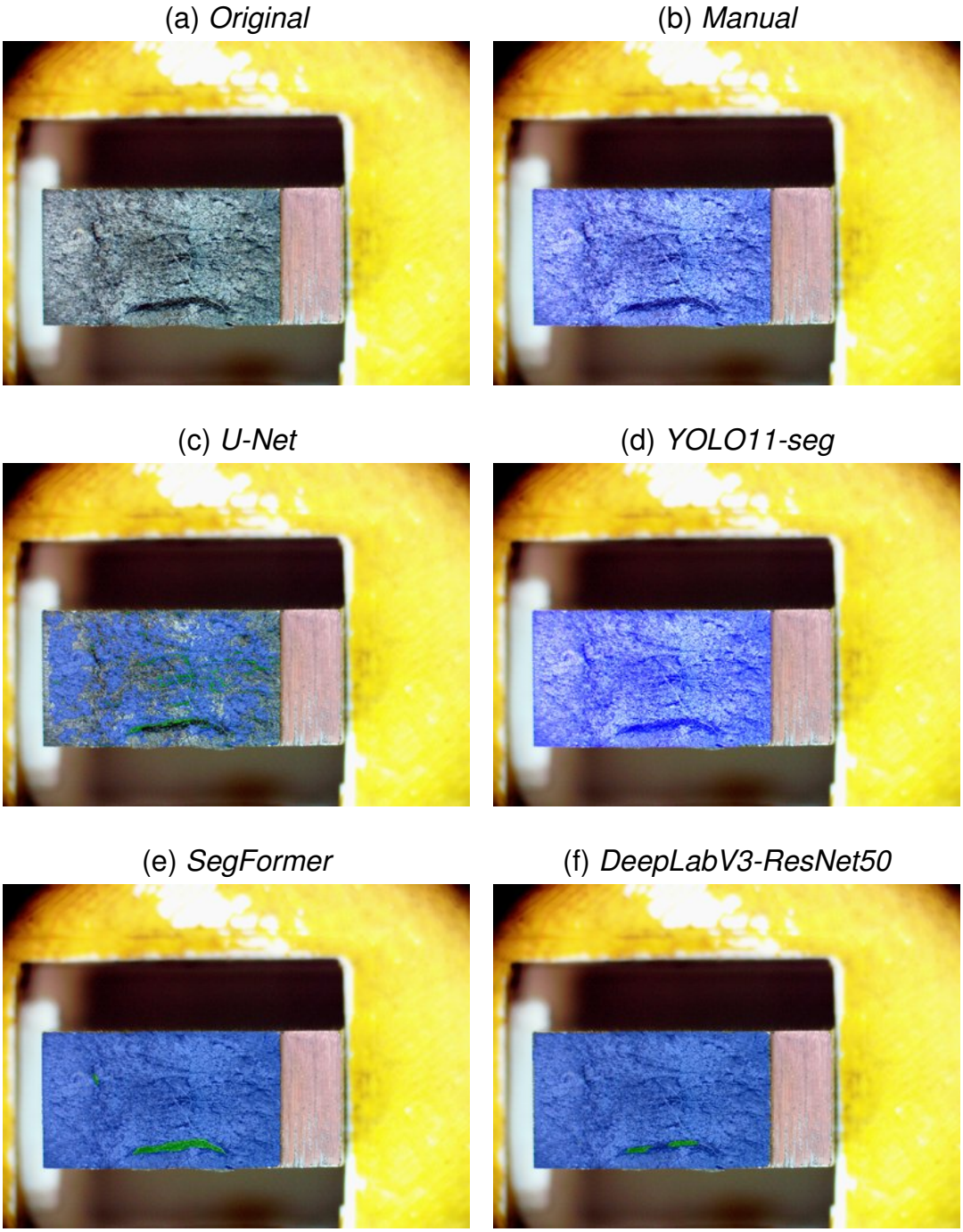
Fonte – Do autor.

Figura 50 – segmentação da SF do CP 10, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



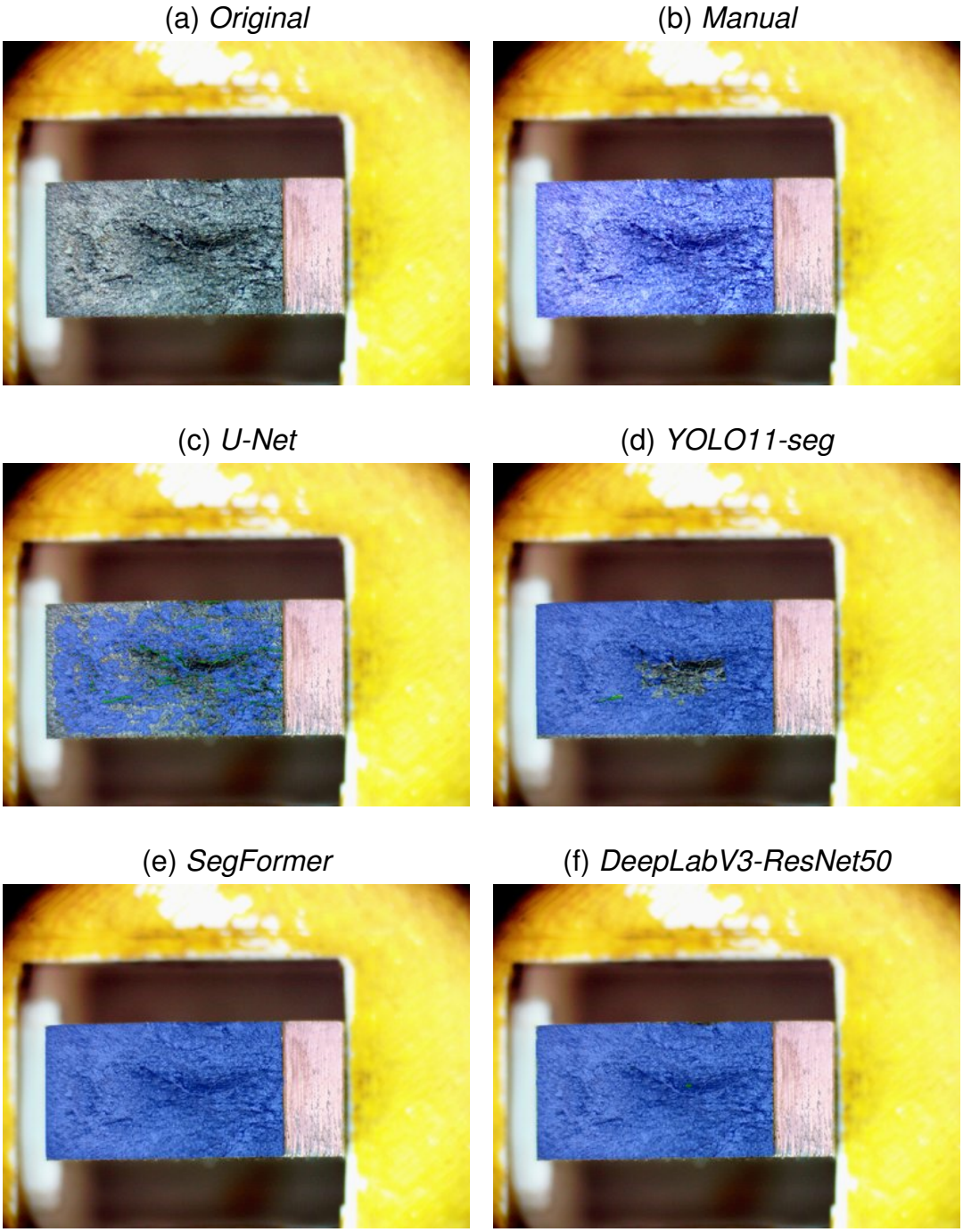
Fonte – Do autor.

Figura 51 – segmentação da SF do CP 11, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



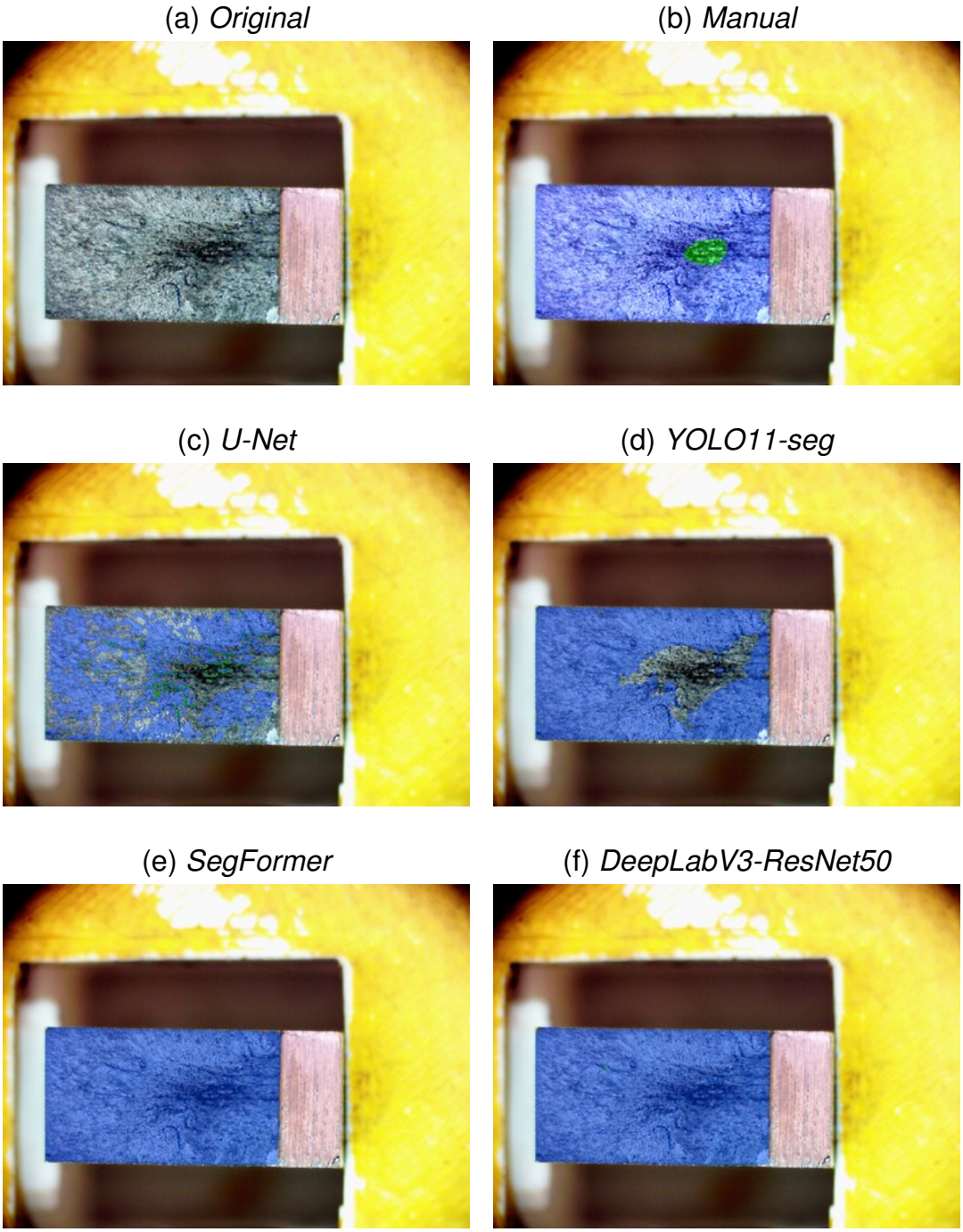
Fonte – Do autor.

Figura 52 – segmentação da SF do CP 12, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



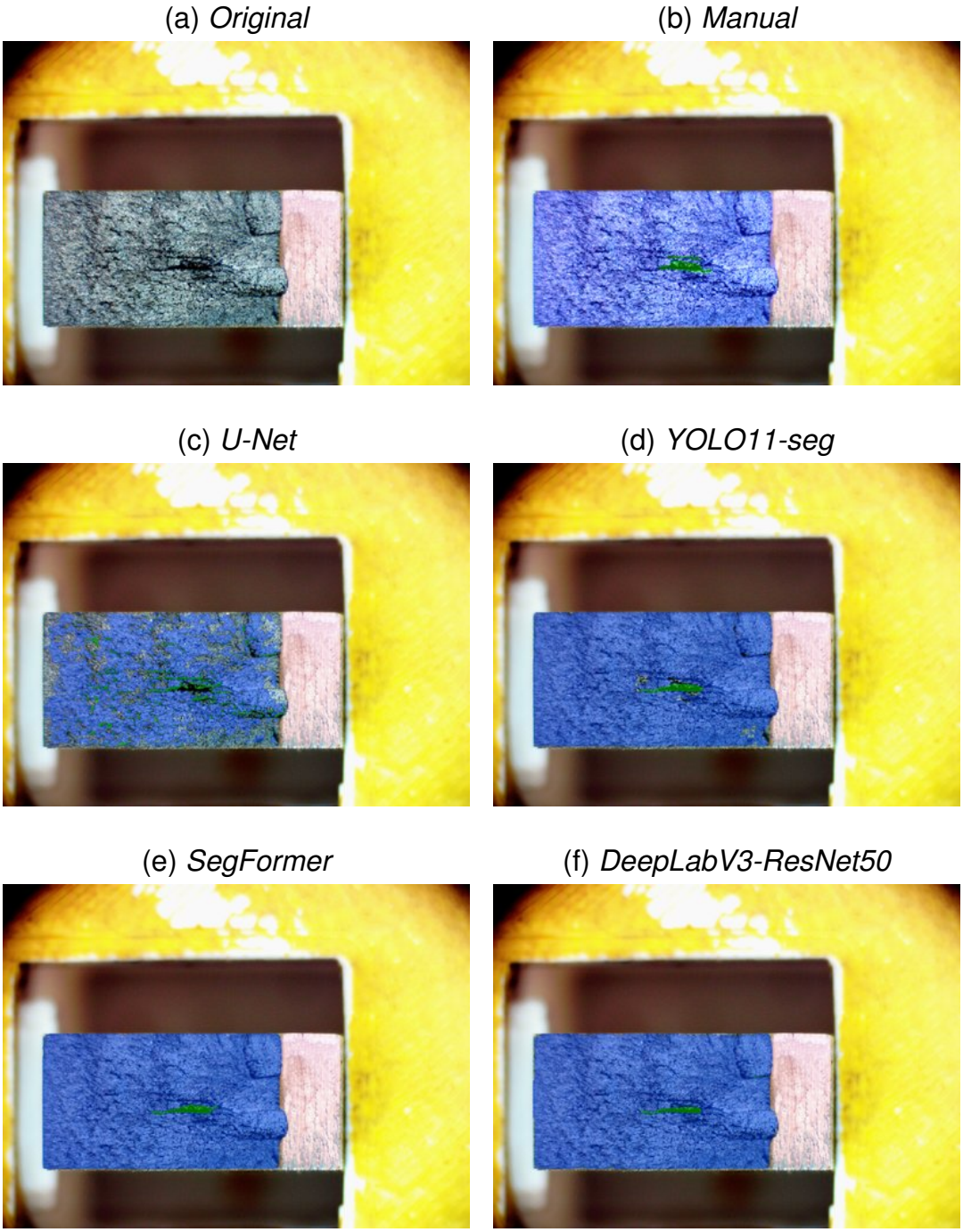
Fonte – Do autor.

Figura 53 – segmentação da SF do CP 13, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



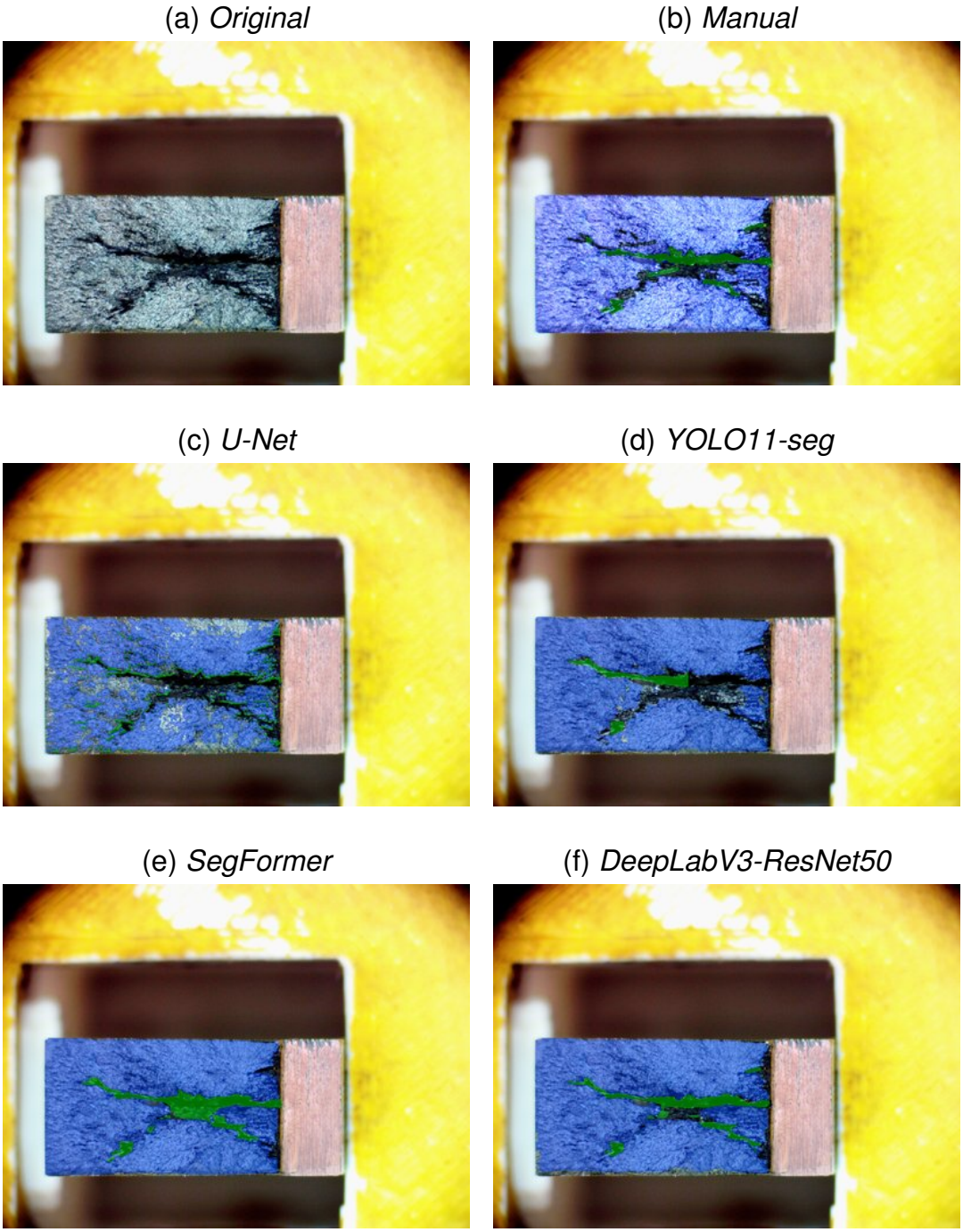
Fonte – Do autor.

Figura 54 – segmentação da SF do CP 14, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



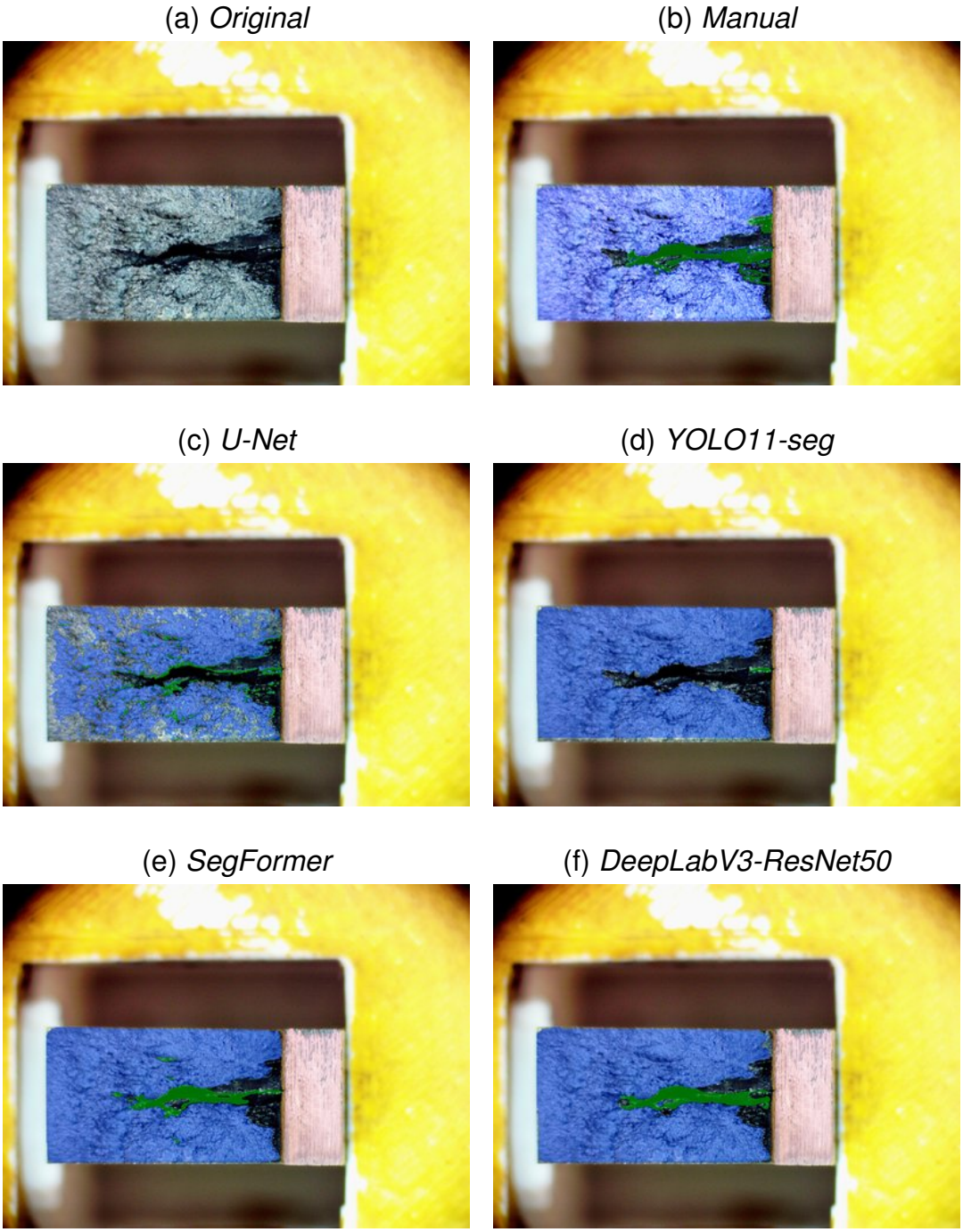
Fonte – Do autor.

Figura 55 – segmentação da SF do CP 15, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



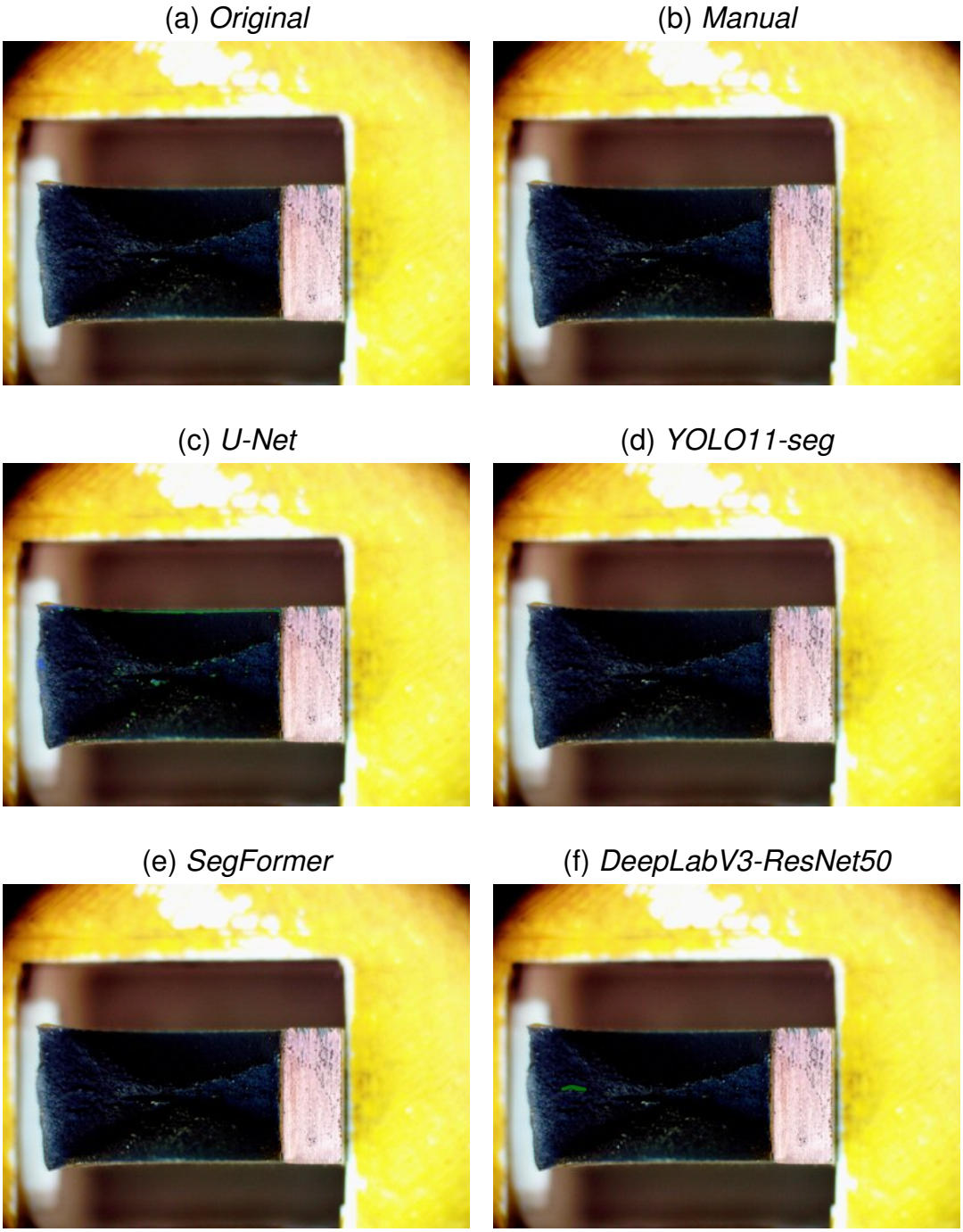
Fonte – Do autor.

Figura 56 – segmentação da SF do CP 16, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



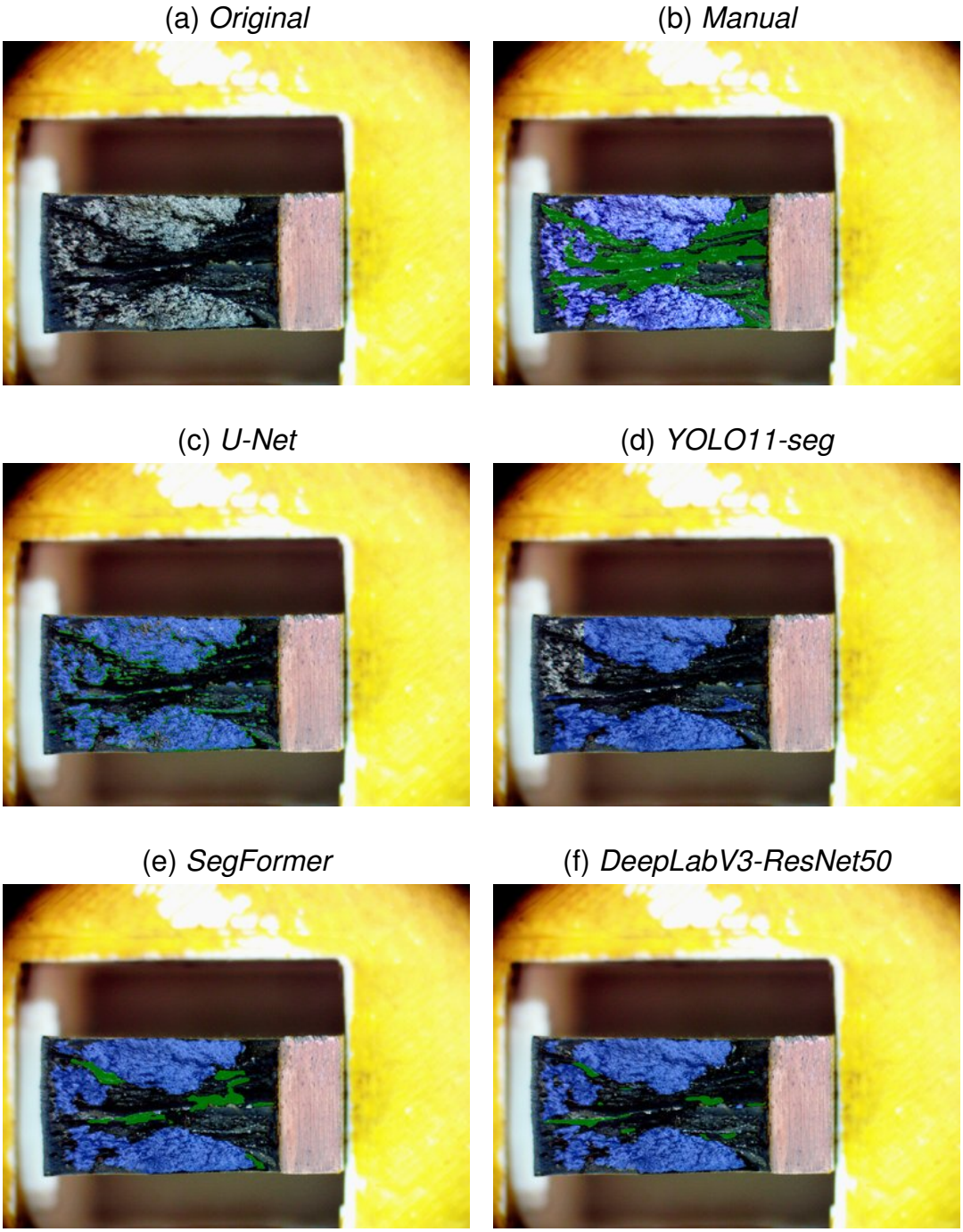
Fonte – Do autor.

Figura 57 – segmentação da SF do CP 17, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



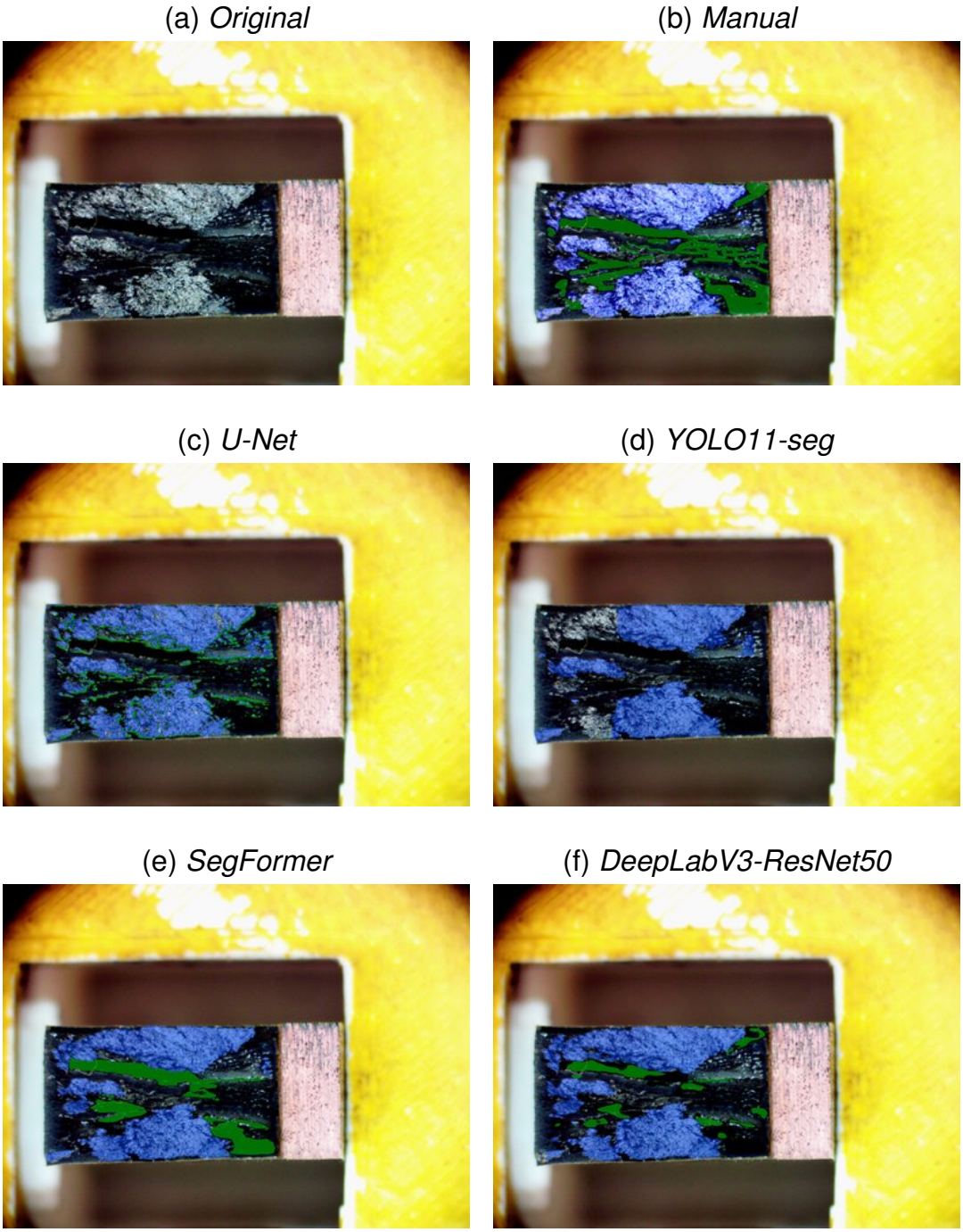
Fonte – Do autor.

Figura 58 – segmentação da SF do CP 18, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



Fonte – Do autor.

Figura 59 – segmentação da SF do CP 19, utilizado na avaliação da Seção 4.6. Em azul, a classe frágil; em verde, a classe delaminação. Em (a), a imagem original; em (b), a referência anotada manualmente; em (c), (d), (e) e (f), respectivamente, a segmentação baseada nas predições dos modelos *U-Net*, *YOLO11-seg*, *SegFormer* e *DeepLabV3–ResNet50*.



Fonte – Do autor.

ANEXO

A

CERTIFICADO DE CALIBRAÇÃO DO EQUIPAMENTO CHARPY

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
LABORATÓRIO DE TECNOLOGIA EM ATRITO E DESGASTE
Avenida João Naves de Ávila, 2121, bloco 5F, Santa Mônica
38.400-902 – Uberlândia/MG

Certificado de Calibração do Charpy (Verificação Indireta)

Documento do Sistema da Qualidade

CERTIFICADO DE CALIBRAÇÃO N. 02/2023

1.1. IDENTIFICAÇÃO DA MÁQUINA

Charpy HIT450P
Fabricante: ZwickRoell GmbH & Co. KG
Modelo: HIT450P
Raio do cutelo: 8 mm
Faixa nominal: 450 J

1.2. PADRÕES UTILIZADOS NA CALIBRAÇÃO

Unidade SRM 2113 de corpos de prova convencionais (força) com a seguinte especificação:
SRM 2113 LL103 (ID: 606)
SRM 2113 LL103 (ID: 955)
SRM 2113 LL103 (ID: 622)
SRM 2113 LL103 (ID: 769)
Certificado de calibração Lot No. LL-103

Unidade SRM 2112 de corpos de prova convencionais (força) com a seguinte especificação:
SRM 2112 LL103 (ID: 270)
SRM 2112 LL103 (ID: 675)
SRM 2112 LL103 (ID: 297)
SRM 2112 LL103 (ID: 506)
Certificado de calibração Lot No. LL-103

Unidade SRM 2098 de corpos de prova convencionais (energia) com a seguinte especificação:
SRM 2098 SH-66 (ID: 1421)
SRM 2098 SH-66 (ID: 1422)
SRM 2098 SH-66 (ID: 1423)
SRM 2098 SH-66 (ID: 1424)
Certificado de calibração Lot No. SH-66

1.3. PROCEDIMENTO DE MEDIÇÃO

A verificação indireta foi conduzida seguindo a ASTM E23 - Standard Test Methods for Notched Bar Impact Testing of Metallic Materials. 2023 e a ISO 148-2 – Materiais metálicos – Ensaio de impacto por pêndulo Charpy (Parte 2: Verificação de máquinas de ensaio). 2016.
Os valores de energia absorbida e de força máxima indicados pela máquina de ensaio foram comparados com os valores de energia absorvida declarados no certificado de calibração de cada unidade de corpos de prova de referência.

Data da calibração: 03 de outubro 2023

Data de emissão do certificado: 23 de outubro de 2023

Os resultados deste certificado referem-se exclusivamente à máquina submetida à verificação nas condições especificadas, não sendo extensivo a outras máquinas.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
LABORATÓRIO DE TECNOLOGIA EM ATRITO E DESGASTE
Avenida João Naves de Ávila, 2121, bloco 5F, Santa Mônica
38.400-902 – Uberlândia/MG

1.4. RESULTADOS DA VERIFICAÇÃO Página 2/6

Tabela 1 – Resultados dos ensaios de verificação indireta (Energia absorvida)

Resultados do ensaio e dados de certificados			Erro e Incerteza	
Identificação dos corpos de prova de referência	KVn_v		$\overline{KV_V}$	17,87 J
SRM 2113 LL103 (ID: 606)	18,15 J		s_V	0,61 J
SRM 2113 LL103 (ID: 955)	18,60 J		n_V	4
SRM 2113 LL103 (ID: 622)	17,36 J		$u(\overline{KV_V})$	0,30 J
SRM 2113 LL103 (ID: 769)	17,38 J			
Do certificado: Energia de Referência	18,20 J		B_V	-0,33 J
Do certificado: graus de liberdade, ν_{MR}	71		$u(B_V)$	0,31 J
Do certificado: Incerteza expandida U_{RM} com 95 % de abrangência e k_{RM}=1,9939	0,10 J		u_V	0,45 J
Incerteza padrão combinada u_{RM}	0,05 J		νB_V	6,12
SRM 2112 LL103 (ID: 270)	105,19		$\overline{KV_V}$	104,54 J
SRM 2112 LL103 (ID: 675)	104,85		s_V	0,61 J
SRM 2112 LL103 (ID: 297)	104,33		n_V	4
SRM 2112 LL103 (ID: 506)	103,80		$u(\overline{KV_V})$	0,30 J
Do certificado: Energia de Referência	105,30 J		B_V	-0,76 J
Do certificado: graus de liberdade, ν_{MR}	70		$u(B_V)$	0,31 J
Do certificado: Incerteza expandida U_{RM} com 95 % de abrangência e k_{RM}=1,9944	0,60 J		u_V	0,87 J
Incerteza padrão combinada u_{RM}	0,30 J		νB_V	5,08
SRM 2098 SH-66 (ID: 1421)	197,05 J		$\overline{KV_V}$	194,27 J
SRM 2098 SH-66 (ID: 1422)	193,64 J		s_V	3,12 J
SRM 2098 SH-66 (ID: 1423)	190,12 J		n_V	4
SRM 2098 SH-66 (ID: 1424)	196,25 J		$u(\overline{KV_V})$	1,57 J
Do certificado: Energia de Referência	197,8 J		B_V	-3,54 J
Do certificado: graus de liberdade, ν_{MR}	69		$u(B_V)$	1,72 j
Do certificado: Incerteza expandida U_{RM} com 95 % de abrangência e k_{RM}=1,423	1,423 J		u_V	3,93 j
Incerteza padrão combinada u_{RM}	0,713 J		νB_V	4,41

Os resultados deste certificado referem-se exclusivamente à máquina submetida à verificação nas condições especificadas, não sendo extensivo a outras máquinas.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
LABORATÓRIO DE TECNOLOGIA EM ATRITO E DESGASTE
Avenida João Naves de Ávila, 2121, bloco 5F, Santa Mônica
38.400-902 – Uberlândia/MG

Tabela 2 – Resultados dos ensaios de verificação indireta (Força máxima) Página 3/6

Resultados do ensaio e dados de certificados		Cálculo do erro e da incerteza	
Identificação dos corpos de prova de referência	Força máxima		
SRM 2113 LL103 (ID: 606)	32,30 kN	$\bar{F}_V$	32,30 kN
SRM 2113 LL103 (ID: 955)	32,14 kN	s_{FV}	0,24 kN
SRM 2113 LL103 (ID: 622)	32,64 kN	n_{FV}	4
SRM 2113 LL103 (ID: 769)	32,12 kN	$u(\bar{F}_V)$	0,12 kN
Do certificado: Energia de Referência	33,00 kN	BF_v	-0,70 kN
Do certificado: graus de liberdade, ν_{MR}	6	u(BFv)	0,77 kN
Do certificado: Incerteza expandida U_{RM} com 95 % de abrangência e k=2,447	1,86 kN	uF_V	1,04 kN
Incerteza padrão combinada u_{cRM}:	0,76 kN	vF_V	8,63
SRM 2112 LL103 (ID: 270)	25,47 kN	$\bar{F}_V$	25,56 kN
SRM 2112 LL103 (ID: 675)	25,47 kN	s_{FV}	0,12 kN
SRM 2112 LL103 (ID: 297)	25,72 kN	n_{FV}	4
SRM 2112 LL103 (ID: 506)	25,56 kN	$u(\bar{F}_V)$	0,06 kN
Do certificado: Energia de Referência	1,78 kN	BF_v	1,50 kN
Do certificado: graus de liberdade, ν_{MR}	6	u(BFv)	0,29 kN
Do certificado: Incerteza expandida U_{RM} com 95 % de abrangência e k=2,447	0,70 kN	uF_V	1,52 kn
Incerteza padrão combinada u_{cRM}:	0,29 kN	vF_V	3,23

Os resultados deste certificado referem-se exclusivamente à máquina submetida à verificação nas condições especificadas, não sendo extensivo a outras máquinas.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
LABORATÓRIO DE TECNOLOGIA EM ATRITO E DESGASTE
Avenida João Naves de Ávila, 2121, bloco 5F, Santa Mônica
38.400-902 – Uberlândia/MG

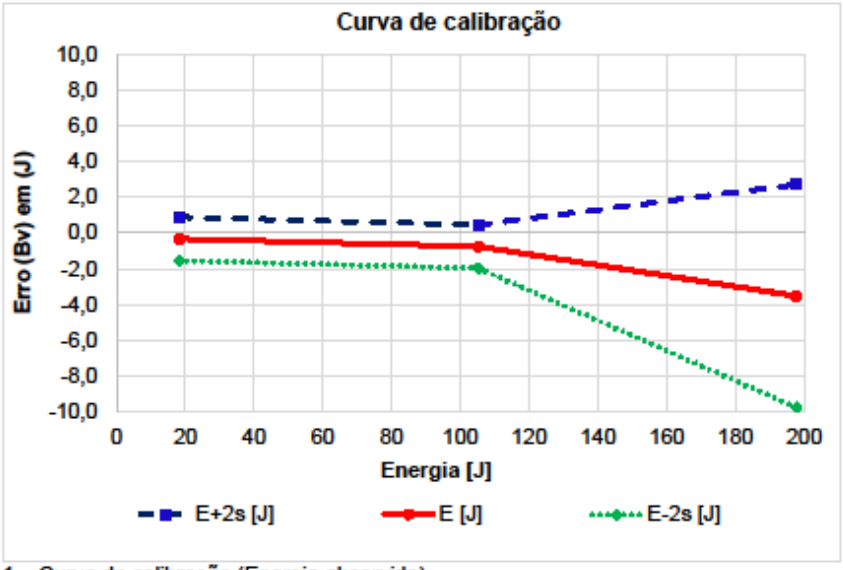


Figura 1 – Curva de calibração (Energia absorvida).

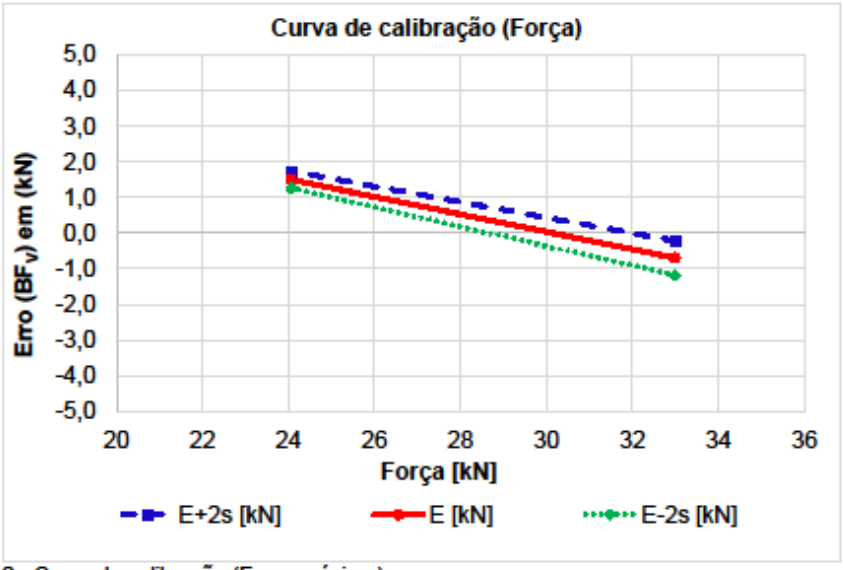


Figura 2 - Curva de calibração (Força máxima).

Os resultados deste certificado referem-se exclusivamente à máquina submetida à verificação nas condições especificadas, não sendo extensivo a outras máquinas.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
LABORATÓRIO DE TECNOLOGIA EM ATRITO E DESGASTE
Avenida João Naves de Ávila, 2121, bloco 5F, Santa Mônica
38.400-902 – Uberlândia/MG

1.5. ESTABILIDADE

O Charpy HIT450P até a presente data passou por duas verificações indiretas. A primeira verificação foi efetuada em 20 de julho 2022 e a segunda 03 de outubro de 2023. A Tabela 3 mostra os valores dos parâmetros metrológicos (erro, repetibilidade, incerteza e erro máximo) obtidos em ambas as verificações. Vale ressaltar que há diferenças entre os valores de energia absorbida dos corpos de prova de referência utilizados nas duas verificações realizadas, bem como na temperatura dos corpos de prova ensaiados, Tabela 3. Outra diferença digna de nota é que na verificação anterior foram ensaiados cinco corpos de prova de referência por nível de energia e na verificação atual quatro. Ainda, antes da atual verificação o Charpy HIT450P foi instrumentado. Desta forma cutelos diferentes foram utilizados em cada uma das verificações realizadas.

Tabela 31 - Valores de energia absorbida dos corpos de prova de referência ensaiados nas verificações indiretas realizadas. Temperatura dos corpos de prova durante os ensaios.

Nível de energia	KV_R (J)		T (°C)	
	20/07/2022	03/10/2023	20/07/2022	03/10/2023
Baixa	14,9	18,2	-40 ± 1 °C	21 ± 1°C
Alta	94,8	105,3	-40 ± 1 °C	21 ± 1°C
Super alta	222,3	197,8	21 ± 1°C	21 ± 1°C

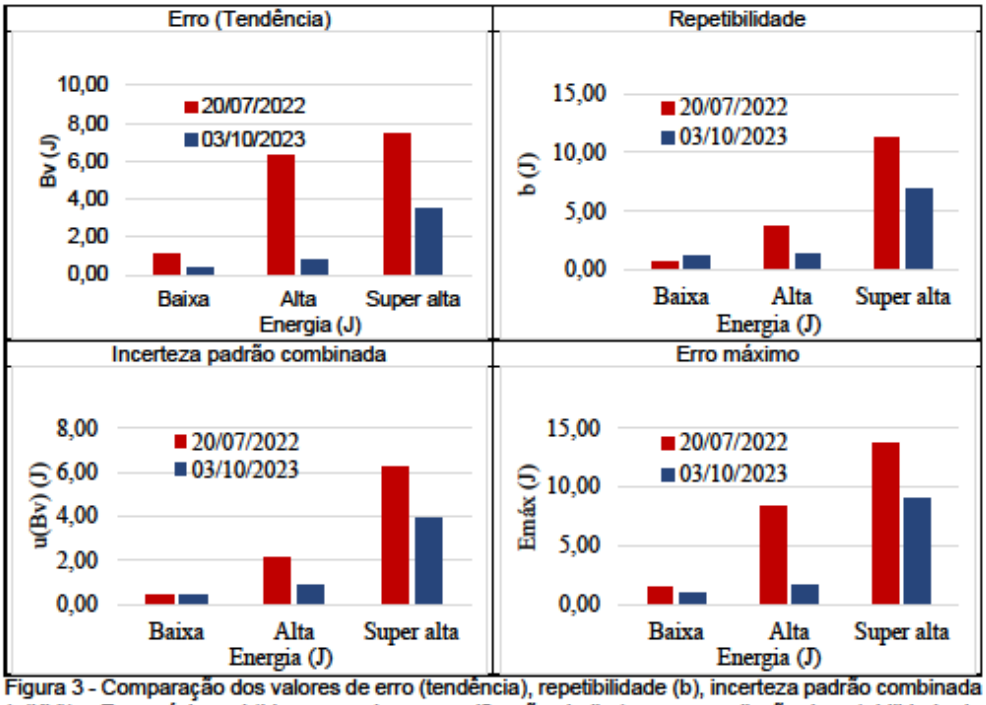


Figura 3 - Comparação dos valores de erro (tendência), repetibilidade (b), incerteza padrão combinada (u(KV)) e Erro máximo obtidos em ambas as verificações indiretas para avaliação da estabilidade do Charpy HIT450P. Energia absorbida.

Os resultados deste certificado referem-se exclusivamente à máquina submetida à verificação nas condições especificadas, não sendo extensivo a outras máquinas.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
LABORATÓRIO DE TECNOLOGIA EM ATRITO E DESGASTE
Avenida João Naves de Ávila, 2121, bloco 5F, Santa Mônica
38.400-902 – Uberlândia/MG

1.6. INCERTEZA DE MEDIÇÃO

A incerteza padrão combinada relatada foi calculada seguindo as determinações da ISO 128-2 (ISO, 2016) a qual considera a incerteza padrão associada aos valores de energia absorbida obtidos durante os ensaios, a incerteza padrão associada aos valores de energia absorbida de referência e a tendência. Esta última foi adicionada uma vez que a correção dos efeitos sistemáticos não foi realizada.

1.7. TEMPERATURA

Todos os ensaios foram realizados à temperatura ambiente de 23,0 ± 1,0 °C. Esta foi a temperatura da máquina Charpy. Os corpos de prova foram ensaiados a 21,0 ± 1,0 °C.

gov.br

Eng. Guilherme Henrique Alves Andrade
Operador

Prof. Luiz Eduardo dos Santos Paes
Líder do projeto

ROSENDA VALDES ARENCIBIA:21460806808

Assinado de forma digital por ROSENDA VALDES ARENCIBIA:21460806808
Dados: 2023.11.05 08:28:33 -03'00'

Profa Rosenda Valdés Arencibia
Coordenadora do projeto

Os resultados deste certificado referem-se exclusivamente à máquina submetida à verificação nas condições especificadas, não sendo extensivo a outras máquinas.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
LABORATÓRIO DE TECNOLOGIA EM ATRITO E DESGASTE
Avenida João Naves de Ávila, 2121, bloco 5F, Santa Mônica
38.400-902 – Uberlândia/MG

Certificado de Calibração do Charpy (Verificação Indireta)

Documento do Sistema da Qualidade

CERTIFICADO DE CALIBRAÇÃO N. 02/2023

1.1. IDENTIFICAÇÃO DA MÁQUINA

Charpy HIT450P
Fabricante: ZwickRoell GmbH & Co. KG
Modelo: HIT450P
Raio do cutelo: 8 mm
Faixa nominal: 450 J

1.2. PADRÕES UTILIZADOS NA CALIBRAÇÃO

Unidade SRM 2113 de corpos de prova convencionais (força) com a seguinte especificação:
SRM 2113 LL103 (ID: 606)
SRM 2113 LL103 (ID: 955)
SRM 2113 LL103 (ID: 622)
SRM 2113 LL103 (ID: 769)
Certificado de calibração Lot No. LL-103

Unidade SRM 2112 de corpos de prova convencionais (força) com a seguinte especificação:
SRM 2112 LL103 (ID: 270)
SRM 2112 LL103 (ID: 675)
SRM 2112 LL103 (ID: 297)
SRM 2112 LL103 (ID: 506)
Certificado de calibração Lot No. LL-103

Unidade SRM 2098 de corpos de prova convencionais (energia) com a seguinte especificação:
SRM 2098 SH-66 (ID: 1421)
SRM 2098 SH-66 (ID: 1422)
SRM 2098 SH-66 (ID: 1423)
SRM 2098 SH-66 (ID: 1424)
Certificado de calibração Lot No. SH-66

1.3. PROCEDIMENTO DE MEDIÇÃO

A verificação indireta foi conduzida seguindo a ASTM E23 - Standard Test Methods for Notched Bar Impact Testing of Metallic Materials. 2023 e a ISO 148-2 – Materiais metálicos – Ensaio de impacto por pêndulo Charpy (Parte 2: Verificação de máquinas de ensaio). 2016.
Os valores de energia absorbida e de força máxima indicados pela máquina de ensaio foram comparados com os valores de energia absorvida declarados no certificado de calibração de cada unidade de corpos de prova de referência.

Data da calibração: 03 de outubro 2023

Data de emissão do certificado: 23 de outubro de 2023

Os resultados deste certificado referem-se exclusivamente à máquina submetida à verificação nas condições especificadas, não sendo extensivo a outras máquinas.