

# Detecção de Corrosão em Instalações *Offshore* de Óleo e Gás com Uso de Computação na Borda

Pedro G. O. de M. Couto, Rodrigo S. Couto, Eduardo A. B. da Silva, Sergio L. Netto

**Resumo**—Tratar corrosão é um dos grandes desafios da produção *offshore* na indústria de Óleo e Gás, pois demanda inspeções custosas. O objetivo deste trabalho é desenvolver uma ferramenta que acelere a análise de corrosão nesse cenário. Para isso, propõe-se uma base de dados de segmentação de corrosão e um método de segmentação baseado em *transformers*, adaptado para restrições típicas do cenário *offshore*. Resultados indicam a viabilidade dos métodos propostos e sua potencial aplicação em setores além do *offshore*, permitindo contribuições futuras para análises gerais de corrosão.

**Palavras-Chave**—Processamento de Imagens, Aprendizado de Máquina, Visão Computacional, Óleo e Gás, E&P *Offshore*

**Abstract**—Dealing with corrosion is a major challenge in offshore Oil and Gas production, as costly inspections are needed. We propose, then, the development of a tool to accelerate corrosion analysis in this scenario. To this end, we construct a corrosion segmentation dataset and a segmentation method based on transformers, adapted for offshore restrictions. Results indicate the feasibility of the proposed methods and their potential use beyond offshore applications, enabling future contributions to general corrosion analysis.

**Keywords**—Image Processing, Machine Learning, Computer Vision, Oil and Gas, Offshore E&P.

## I. INTRODUÇÃO

O Brasil tem papel de destaque na indústria global de Óleo e Gás (O&G), estando entre os principais produtores, consumidores e exportadores de petróleo. A maior parte da produção nacional se dá no *offshore*, correspondendo a 95% da produção total de O&G e 97,5% das reservas comprovadas [1].

A produção *offshore*, no entanto, é desafiadora. Plataformas e embarcações operam por décadas no meio do oceano, expostas a intempéries e à salinidade marinha – o que leva à incidência elevada de corrosão. Portanto, monitorar corrosão é essencial para sua manutenção. Por isso, são realizadas inspeções periódicas, conduzidas por especialistas treinados que realizam avaliações visuais. O processo, porém, é dispendioso, demorado e sujeito a inconsistências devidas ao viés humano.

Para ajudar a sanar esses problemas, novas tecnologias vêm sendo adotadas. Contudo, têm enfrentado restrições relacionadas à operação no *offshore*. Por exemplo, conexões de internet limitadas e instáveis dificultam a transmissão de dados e, como instalações *offshore* são construídas sob medida, adicionar infraestrutura não é simples. Alternativas que têm

obtido mais sucesso envolvem a aplicação de aprendizado de máquina (*Machine Learning* – ML) em computação de borda, permitindo a automatização da detecção de corrosão a baixo custo. Modelos de ML desse tipo poderiam ser incorporados aos cotidianos das tripulações, habilitando avaliações contingentes de corrosão sem a presença de especialistas – o que permitiria reduzir a duração das inspeções, implementar alarmes automatizados e a diminuir a frequência de visitas de inspeção.

O objetivo deste trabalho, então, é propor uma solução de ML para avaliar níveis de corrosão em imagens capturadas em instalações *offshore*, respeitando restrições de uso de recursos. Para alcançar esse objetivo, também é proposta a criação uma base de dados de segmentação de corrosão com anotações no nível do pixel, que é usada para treinar e avaliar a solução desenvolvida.

O restante do trabalho está organizado da seguinte forma: a Seção II apresenta breve revisão bibliográfica; a Seção III detalha metodologias utilizadas para a criação do conjunto de dados, o desenvolvimento do modelo e a avaliação dos resultados; a Seção IV discute os resultados obtidos, e, por fim, a Seção V oferece conclusões e trabalhos futuros.

## II. TRABALHOS RELACIONADOS

Lidar com corrosão em ambientes *offshore* é um desafio amplamente explorado. Os trabalhos em [2] e [3] compilam técnicas de ML utilizadas para avaliação de corrosão na indústria de O&G. A maioria desses trabalhos utiliza conjuntos de dados privados e arquiteturas baseadas em redes neurais convolucionais (*Convolutional Neural Networks* – CNNs). O uso de computação de borda em aplicações *offshore*, contudo, ainda é pouco adotado pela indústria de O&G; ainda assim, os autores de [4] propõem possíveis usos dessas técnicas. Ampliando-se o escopo da revisão bibliográfica para além do *offshore*, o leque se torna mais amplo. Em [5], é proposta metodologia para segmentar corrosão em estruturas de aço industriais, utilizando técnicas de aumento de dados e duas arquiteturas de redes neurais: Mask R-CNN e PSPNet. Os autores de [6] abordam a classificação de níveis de corrosão em pontes de aço, propondo uma metodologia de aumento de dados baseada em redes adversariais. Para lidar com questões de generalização, devido à falta de conjuntos de dados públicos sobre corrosão, os autores de [7] apresentam um modelo com três variantes bayesianas que fornecem estimativas de incerteza. Por fim, [8] propõe modificações à estrutura da rede convolucional U-Net, especializando-a para problemas de corrosão.

Pedro Couto, Petrobras, RJ-RJ, e-mail: pedro.g.couto@petrobras.com.br; Rodrigo Couto, GTA/PEE/COPPE, Universidade Federal do Rio de Janeiro (UFRJ), RJ-RJ, e-mail: rodrigo@gta.ufrj.br, Eduardo da Silva, SMT/PEE/COPPE, Universidade Federal do Rio de Janeiro (UFRJ), RJ-RJ, e-mail: eduardo@smt.ufrj.br, Sergio Lima Netto, SMT/PEE/COPPE, Universidade Federal do Rio de Janeiro (UFRJ), RJ-RJ, e-mail: sergioln@smt.ufrj.br. Este trabalho foi realizado com apoio da Petrobras, CNPq, Capes e FAPERJ.

Embora existam técnicas promissoras de ML sendo aplicadas à avaliação de corrosão, adaptações ao ambiente *offshore* são raras. Além disso, a maioria dos trabalhos focados em *offshore* carece de testes com arquiteturas mais recentes, como, por exemplo, *transformers*. Mais importante, bases de dados anotadas de corrosão, especialmente as voltadas para o *offshore*, são raras, possuem poucos exemplos e, em sua maioria, não são públicas. Para lidar com esses desafios, este trabalho propõe duas frentes: (i) começar o processo de construção de uma base de dados anotada de segmentação de corrosão *offshore* de alta qualidade que, quando completa, seja tornada pública; e (ii) aplicar modelos Estado-da-Arte (*State of the Art* – SOTA) sobre esta base, levando em conta restrições de recurso típicas de atividades *offshore*. Para a segunda frente, é conduzida revisão bibliográfica adicional, relacionada a modelos SOTA para segmentação de imagens. Essa revisão possui objetivo de encontrar modelos leves, capazes de executar inferência em dispositivos de borda. Entretanto, apesar de leves, esses modelos devem ter alta capacidade, sendo capazes de segmentar corrosão em imagens complexas com sucesso.

Dois arquiteturas que permitem variação de custo computacional são identificadas: DeepLab [9] e SegFormer [10]. O DeepLab é composto por um módulo de segmentação anexado a CNNs tradicionais – o que permite modificações em sua estrutura. A família de modelos SegFormer, por outro lado, apresenta cinco categorias de complexidade computacional pré-definidas, cujos resultados de *benchmark* crescem proporcionalmente a seus custos computacionais. Sua arquitetura, baseada em *transformers*, propõe adaptações não só à estrutura do Vision Transformer (ViT) [11], mas também à do decodificador – que é simplificado significativamente.

Buscando outros modelos de alta capacidade, encontram-se arquiteturas quase exclusivamente baseadas em *transformers*. O Segmentation Transformer (SETR) [12] é pioneiro no uso de *transformers* para segmentação semântica, utilizando ViTs para capturar informações contextuais globais, gerando previsões densas por meio de um decodificador. Os Swin Transformers (SWIN) [13] são *transformers* hierárquicos que processam imagens em janelas não sobrepostas, aumentando a eficiência computacional. Os Mask Transformers, como o Mask2Former (M2F) [14], inferem máscaras sem depender da geração de *bounding boxes*. O Mask DINO [15] permite unificar detecção e segmentação de objetos. O ViT-Adapter [16] melhora o ViT para tarefas de previsão densa através da adição de recursos multiescalares e de detalhamento espacial. O TransNeXt [17] combina mecanismos de atenção biomiméticos – que simulam características do sistema visual humano – com estruturas convolucionais para melhorar resultados com menor sobrecarga computacional. O OneFormer [18] unifica tarefas de segmentação através da adoção de uma arquitetura unificada. Outras arquiteturas baseadas em *transformers* e que apresentam bons resultados foram identificadas, mas a maioria ultrapassa 1 bilhão de parâmetros e, portanto, seu uso é inviável no contexto deste trabalho – devido às restrições de recurso típicas do *offshore*.

Com base na revisão feita, elaborou-se tabela comparativa, abordando as arquiteturas identificadas. Essa tabela é disposta – e utilizada como base para a escolha do modelo aplicado

neste trabalho – na Seção III.

### III. METODOLOGIA

Nesta seção, descrevemos a construção da base de dados de segmentação de corrosão *offshore*. Então, trazemos os processos de escolha e treinamento dos modelos. Por fim, abordamos as transformações necessárias para adequar a base de dados construída às especificações dos modelos utilizados.

#### A. Base de Dados

Normas regulatórias exigem que empresas de O&G operando no Brasil forneçam evidências visuais de procedimentos de manutenção. Tratando-se de corrosão no *offshore*, essas evidências são obtidas na forma de fotografias tiradas por inspetores durante a análise de estruturas e equipamentos. Acompanhando essas fotografias, são também catalogados os resultados da inspeção, na forma de categorias de corrosão regulamentadas pela ASTM (*American Society for Testing and Materials*), e metadados sobre o item inspecionado. Para os dados utilizados neste trabalho, essas informações são armazenadas em um banco de dados com milhares de entradas, das quais uma parcela é selecionada para a construção de um conjunto de dados. O conjunto construído tem 14,134 entradas, abrangendo diferentes tipos de objetos e estruturas e possuindo diversas configurações de captura de imagem: diferentes câmeras, ambientes, condições de iluminação e resoluções.

Esse conjunto é utilizado como fundação para o desenvolvimento da base de dados de segmentação de corrosão proposta nesse trabalho. Contudo, como a construção de bases de dados de segmentação de imagens é uma tarefa bastante custosa, elaborou-se um *pipeline* de anotação, integrando o *Segment Anything Model* [19] à *Computer Vision Annotation Tool* (CVAT)<sup>1</sup>, para facilitar anotações. Além disso, devido a limitações de recursos e de tempo, apenas imagens de flanges, consideradas de menor complexidade, são anotadas. Para agilizar o processo, também contrataram-se anotadores profissionais, por meio da plataforma online *fiverr*<sup>2</sup>, que utilizaram o *pipeline* deste trabalho como apoio na anotação dos dados. No momento da escrita deste trabalho, o conjunto de dados de segmentação contém 1,106 exemplos de corrosão em flanges anotados no nível do pixel.

#### B. Escolha dos Modelos

O SOTA para segmentação de imagens é dominado por arquiteturas baseadas em *transformers*. Como são custosas a termos computacionais, podem não ser apropriadas para uso em borda. A Tabela I, então, apresenta três colunas de dados. Na primeira, compara resultados de inferência dos modelos escolhidos na Seção II, usando o conjunto de dados ADE20k [20] como base, e o *mIoU* (*mean Intersection over Union*) [21] como métrica de avaliação. Por simplicidade, as análises foram baseadas apenas em resultados já publicados na literatura. Na segunda, introduz métrica simples para comparar os modelos

<sup>1</sup><http://www.cvat.ai/>

<sup>2</sup><http://www.fiverr.com/>

a termos de suas eficiências computacionais, dividindo seus resultados de *benchmark* pelo número de operações de ponto flutuante (*Floating-Point Operations* – FLOPs) necessários para inferência, em bilhões (*giga*). Na terceira, traz o número absoluto de parâmetros de cada modelo, em milhões.

TABELA I  
RESULTADOS COMPARADOS NO DATASET ADE20K (mIoU)

| Modelo                            | ADE20K mIoU | mIoU/GFLOPs | Parâmetros |
|-----------------------------------|-------------|-------------|------------|
| SegFormer-b0                      | 37.4        | 4.45        | 3.7        |
| DeepLabV3 (ResNet101 backbone)    | 44.1        | 0.17        | 6.4        |
| SegFormer-b2                      | 46.5        | 0.75        | 27.5       |
| SETR                              | 48.6        | 0.13        | 318.3      |
| SegFormer-b4                      | 50.3        | 0.52        | 61.1       |
| ConvNext-XL++                     | 54.0        | 0.02        | 391.0      |
| TransNeXt-small (Mask2Former)     | 54.1        | ?           | 69.0       |
| TransNeXt-base (Mask2Former)      | 54.7        | ?           | 109.0      |
| Mask2Former (SWIN-Large backbone) | 56.1        | 0.14        | 215.0      |
| OneFormer (DiNAT backbone)        | 58.3        | 0.16        | 223.0      |
| Mask DINO                         | 60.8        | 0.04        | 223.0      |
| ViT-Adapter-L                     | 61.5        | ?           | 571.0      |

Considerando mIoU/GFLOPs, os modelos SegFormer apresentam desempenho superior a todos os outros modelos com dados relatados na literatura, sendo inicialmente escolhidos. Restam para comparação apenas outras duas opções, posto que não possuem dados de *benchmark* relatados no ADE20k: ViT-Adapter e TransNeXt. O ViT-Adapter, por ser um modelo com grande número de parâmetros, é descartado. Os modelos TransNeXt, por outro lado, mostram melhores resultados e contagens de parâmetros semelhantes aos maiores SegFormers. No entanto, essa família não possui versão leve – o menor TransNeXt possui quase 50 milhões de parâmetros. Então, privilegiando a simplicidade, optou-se pela família SegFormer para avaliar e comparar tanto modelos leves (SegFormer-b0) quanto modelos de alta capacidade (SegFormer-b5).

### C. Treinamento de Modelos

Os modelos SegFormer são pré-treinados, por padrão, no conjunto de dados ImageNet-1k [22]. Adicionalmente, são carregados pesos pré-treinados no conjunto de dados Cityscapes [23] – com imagens de tamanho  $1024 \times 1024$  pixels. O *pipeline* de treinamento consiste em duas etapas principais: pré-processamento e treino. Na primeira etapa, as imagens de entrada e as máscaras de segmentação são transformadas, normalizadas e redimensionadas para  $128 \times 128$  para o SegFormer-b0 e para  $1024 \times 1024$  para o SegFormer-b5. Em seguida, são utilizadas técnicas de aumento de dados (*data augmentation*), como inversão horizontal, redução de resolução e variação de cores planckiana.

Para a segunda etapa, desenvolveu-se um fluxo de paralelização multi-nó e multi-GPU que utiliza o Slurm [24] para orquestrar múltiplos processos distribuídos por vários nós de computação e, para garantir a paralelização do modelo, utiliza-se o módulo *Distributed Data Parallel* (DDP) do PyTorch [25]. Para treinar os modelos maiores, foram utilizados 4 nós, com 256 núcleos de vCPU (dois processadores AMD EPYC 7742), 8 GPUs NVIDIA A100 de 80GB e 2 Terabytes de RAM cada. Para os menores, foi escolhido um único nó de mesma especificação.

Para os resultados apresentados, treinam-se os dois modelos com o mesmo cronograma de taxa de aprendizado e a mesma política de parada antecipada. Utiliza-se também o otimizador

Adam, com taxa de aprendizado fixa de 0.00002. O cronograma inicialmente define a taxa de aprendizado como 0.001 e, toda vez que a *loss* de validação do modelo não melhora por 5 épocas consecutivas, reduz-se a taxa pela metade. A política de parada antecipada define que, caso a *loss* de validação do modelo não melhore por 30 épocas consecutivas, o treinamento deve ser interrompido. Tanto o modelo final quanto o melhor modelo geral — considerando a *loss* de validação como métrica de avaliação de qualidade — têm seus *checkpoints* salvos.

A métrica escolhida para avaliar os modelos treinados, assim como na revisão bibliográfica, é o mIoU, comumente utilizado em avaliações de segmentação de imagens (e.g. [10], [15], [18]). O mIoU equilibra o desempenho entre as classes, atribuindo o mesmo peso a cada uma independentemente da quantidade de exemplos ou da frequência de pixels. Essa propriedade garante que a métrica não favoreça desproporcionalmente classes dominantes, o que é especialmente importante em conjuntos de dados com distribuições de classes desequilibradas. Como a base de dados utilizada neste trabalho conta com apenas dois *labels* – *corrosão* e *background* – e como, na maioria das imagens, as regiões de *background* são muito maiores do que as de *corrosão*, lidar com esse tipo de desequilíbrio torna-se crucial.

### D. Processamento do Conjunto de Dados

O conjunto de dados de segmentação de *corrosão offshore* deste trabalho é composto pelas imagens dos objetos e por um arquivo de anotações no formato *json*. No entanto, como o SegFormer espera pares imagem/rótulo como entrada, desenvolveu-se fluxo de pré-processamento para gerar, a partir do arquivo de anotações, os rótulos em *bitmap* necessários.

Para dividir o conjunto de dados em seções de treino e validação, utiliza-se um método baseado em *k-fold* estratificado [26], buscando garantir distribuições adequadas de níveis de *corrosão* tanto entre *folds* quanto entre os subconjuntos de treino e validação. Para isso, utilizam-se as categorias de *corrosão* da ASTM como parâmetro, evitando amostragem aleatória. Inicia-se separando as imagens por categoria ASTM. Então, aplica-se a geração de *folds*, criando subconjuntos de treino e validação para cada classe ASTM. Por fim, esses subconjuntos são mesclados, com resultados explicitados pela Tabela II.

TABELA II  
QUANTIDADE DE AMOSTRAS DISTRIBUÍDAS POR *fold*

|                  | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
|------------------|--------|--------|--------|--------|--------|
| <b>Treino</b>    | 881    | 882    | 884    | 887    | 890    |
| <b>Validação</b> | 225    | 224    | 222    | 219    | 215    |

## IV. RESULTADOS

A Figura 1 apresenta resultados comparativos de mIoU para os modelos, com o SegFormer-b0 à esquerda e o SegFormer-b5 à direita. Ambos os modelos apresentam comportamento

apropriado, reduzindo progressivamente as *losses* de treinamento e validação – esses gráficos foram omitidos por questões de espaço – enquanto melhoram o mIoU. Considerando-se as médias dos *folds*, o modelo menor atinge 0,675 de mIoU, com o maior chegando a 0,7277 – uma diferença expressiva de 5,27 pontos percentuais. Além disso, os modelos demonstram comportamento consistente em todos os cinco *folds*, destacando a robustez da metodologia e do conjunto de dados.

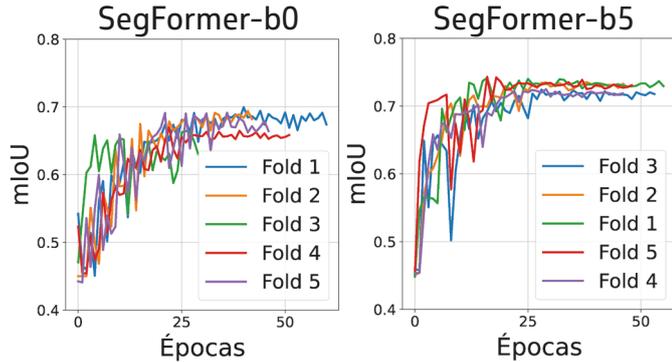


Fig. 1. Métricas de avaliação, usando mIoU. À esquerda: SegFormer-b0. À direita: Segformer-b5.

Para análises visuais, utilizam-se dois cenários. Primeiro, apresentam-se as inferências dos modelos em imagens pertencentes a seus respectivos conjuntos de validação, comparando as previsões com as máscaras anotadas. Segundo, geram-se visualizações para imagens que não pertencem a nosso conjunto de dados, buscando avaliar capacidades de generalização. Em ambos os casos, utilizam-se os modelos que alcançaram a menor perda de validação dentre os *folds*.

Para o primeiro cenário, geram-se visualizações compostas por três elementos: a fotografia original, a máscara de segmentação anotada e o resultado – na forma da máscara predita sobreposta sobre a imagem original. A Figura 2 ilustra um exemplo de resultado, destacando a diferença de comportamento entre modelos: enquanto o SegFormer-b5 se aproxima bastante da anotação, o SegFormer-b0 produz máscaras de contorno irregular, que não envolvem completamente o objeto corroído.

Para o segundo cenário, utilizam-se imagens não anotadas. Em todos os casos, o SegFormer-b0 está no lado esquerdo das figuras, enquanto o SegFormer-b5 está no lado direito. A Figura 3 apresenta um exemplo de flange. Nela, ambos os modelos são capazes de identificar o piso corroído no fundo da imagem, uma característica que não existe no conjunto de dados de treinamento. A Figura 4, por sua vez, mostra um dos tipos de item mais distintos em relação às flanges do conjunto de treinamento: imagens de guarda-corpos, que comumente contêm extensas regiões de fundo, frequentemente incluindo outros objetos analisáveis. Mesmo assim, ambos os modelos apresentam bom desempenho, demonstrando fortes capacidades de generalização.

Para avaliar o desempenho dos modelos sob diferentes restrições de recursos, dois cenários com configurações de hardware distintas são testados. No primeiro cenário, as exe-

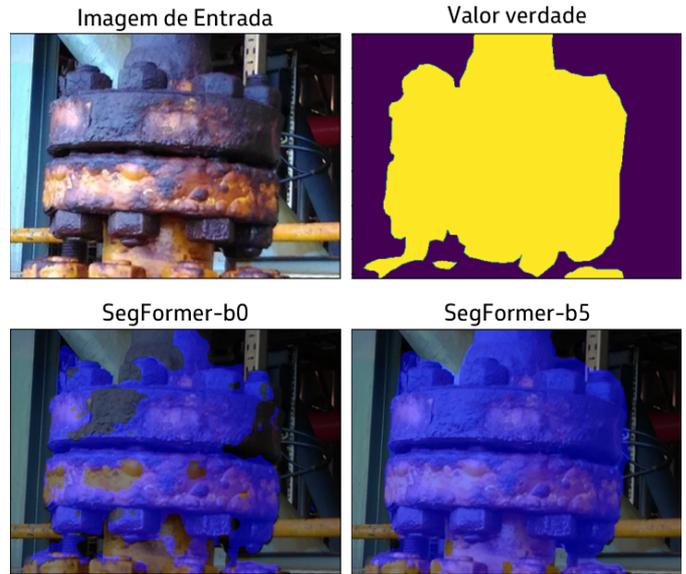


Fig. 2. Visualização de flange do conjunto de avaliação. Em sentido horário: imagem de entrada; máscara anotada; visualização da predição do SegFormer-b0; predição do SegFormer-b5.

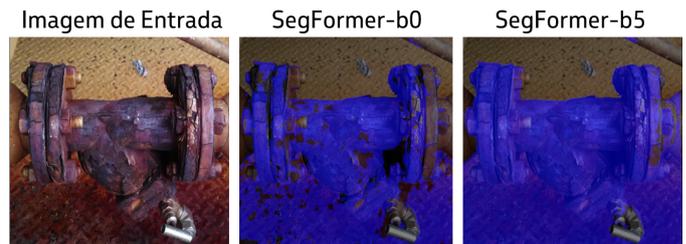


Fig. 3. Flange com corrosão no *background*. À esquerda: imagem do conjunto de teste. Ao centro: inferência do SegFormer-B0. À direita: SegFormer-b5.

cuções são realizadas sem limitações, com ambos os modelos utilizando uma GPU NVIDIA A100 80GB para inferência. No segundo, recursos são limitados, sendo usadas apenas as vCPUs disponíveis nos nós de processamento. Além disso, para cada cenário, dois tamanhos de entrada diferentes são testados: O primeiro tamanho,  $256 \times 256$ , representa um tamanho padrão de *patch* para tarefas de ML em visão computacional; o segundo tamanho,  $1024 \times 1024$ , corresponde à tarefa de pré-treinamento do SegFormer no conjunto Cityscapes e está mais próximo de resoluções nativas das câmeras usadas para adquirir imagens de corrosão em ambientes *offshore*.

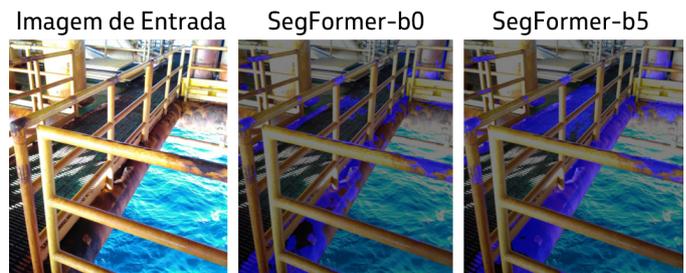


Fig. 4. Guarda corpo do conjunto de teste. À esquerda: imagem do conjunto de teste. Ao centro: inferência do SegFormer-B0. À direita: SegFormer-b5.

Resultados são calculados como a média de 100 execuções

independentes, e são exibidos na Tabela III. Custos computacionais para o modelo leve são significativamente menores: em comparação com o SegFormer-b5, tempos de processamento para o SegFormer-b0 diminuem por mais de 4 vezes, e o consumo de memória por mais de 5. Em particular, o uso de memória para o modelo maior torna-se proibitivo para dispositivos de borda em imagens de resolução mais alta, usando mais de 20Gb de RAM para a inferência. Contudo, em resoluções mais baixas, o modelo maior se mantém viável.

TABELA III  
TESTES DE DESEMPENHO PARA CENÁRIOS USANDO CPU E GPU

|             | SegFormer-b0 |           | SegFormer-b5 |          |           |          |
|-------------|--------------|-----------|--------------|----------|-----------|----------|
|             | CPU          | GPU       | CPU          | GPU      | CPU       | GPU      |
|             | 1024         | 256       | 1024         | 1024     | 256       | 1024     |
| CPU time    | 2.24 s       | 10.94 ms  | 14.81 ms     | 12.28 s  | 58.17 ms  | 66.63 ms |
| GPU time    | -            | 2.42 ms   | 14.32 ms     | -        | 12.73 ms  | 64.74 ms |
| Peak Memory | 4.38 Gb      | 111.41 Mb | 4.41 Gb      | 22.97 Gb | 621.00 Mb | 23.04 Gb |

## V. CONCLUSÕES E TRABALHOS FUTUROS

Nesse trabalho, iniciou-se a construção de uma base de dados de segmentação de imagens focada em avaliação de corrosão, usando como base dados reais provenientes de fluxos de manutenção da indústria de O&G. Esse conjunto foi utilizado para realizar experimentos com arquiteturas SOTA de segmentação de imagens. Nos experimentos, foram avaliados os desempenhos dos modelos sob diferentes condições de restrição de recursos. Os resultados obtidos são promissores, demonstrando que a base de dados e a metodologia propostas podem ser usadas para avaliar corrosão de forma eficaz em cenários *offshore*. Além disso, sugerem potencial para generalização, considerando corrosão em outros contextos. Como proposta de trabalhos futuros, pretende-se expandir a base de dados para que, quando completa, torne-se pública. Além disso, pretende-se desenvolver aplicação de borda consolidada, integrável a avaliações de corrosão na indústria de O&G.

## REFERÊNCIAS

- [1] T. Y. C. Loureiro, "Produção offshore no brasil: Construindo novos caminhos," <https://www.presalpetroleo.gov.br/wp-content/uploads/2024/06/Producao-offshore-no-Brasil-Construindo-novos-caminhos.pdf>, 2024, acessado em 15/01/2025.
- [2] S. Sookpong, S. Phimsiri, T. Tosawadi, P. Choppradit, V. Suttichaya, C. Utintu, and E. Thamwiwatthana, "Comparison of corrosion segmentation techniques on oil and gas offshore critical assets," in *20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Nakhon Phanon, Thailand, May 2023, pp. 1–5.
- [3] A. Hussein Khalaf, Y. Xiao, N. Xu, B. Wu, H. Li, B. Lin, Z. Nie, and J. Tang, "Emerging ai technologies for corrosion monitoring in oil and gas industry: A comprehensive review," *Engineering Failure Analysis*, vol. 155, p. 107735, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1350630723006891>
- [4] S. S. Narayana, "Synergizing edge computing and cloud integration for enhanced efficiency in the oil and gas industry," *International Journal of Computer Engineering & Technology*, vol. 15, pp. 15–17, Apr. 2024.
- [5] S. K. Fondevik, A. Stahl, A. A. Transeth, and O. O. Knudsen, "Image segmentation of corrosion damages in industrial inspections," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020, pp. 787–792.
- [6] Z. Li, P. Shao, M. Zhao, K. Yan, G. Liu, L. Wan, X. Xu, and K. Li, "Optimized deep learning for steel bridge bolt corrosion detection and classification," *Journal of Constructional Steel Research*, vol. 215, p. 108570, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0143974X24001202>
- [7] W. Nash, L. Zheng, and N. Birbilis, "Deep learning corrosion detection with confidence," *NPJ Materials Degradation*, vol. 6, Mar. 2022. [Online]. Available: <https://doi.org/10.1038/s41529-022-00232-6>
- [8] T. Zhu, S. Zhu, T. Zheng, H. Ding, W. Song, and C. Li, "Heu-net: hybrid attention residual block-based network with external skip connections for metal corrosion semantic segmentation," *The Visual Computer*, vol. 40, Feb. 2024. [Online]. Available: <https://doi.org/10.1007/s00371-023-02846-6>
- [9] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *CoRR*, vol. abs/1606.00915, 2016. [Online]. Available: <http://arxiv.org/abs/1606.00915>
- [10] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Álvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *CoRR*, vol. abs/2105.15203, 2021. [Online]. Available: <https://arxiv.org/abs/2105.15203>
- [11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: <https://arxiv.org/abs/2010.11929>
- [12] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *CoRR*, vol. abs/2012.15840, 2020. [Online]. Available: <https://arxiv.org/abs/2012.15840>
- [13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *CoRR*, vol. abs/2103.14030, 2021. [Online]. Available: <https://arxiv.org/abs/2103.14030>
- [14] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," *CoRR*, vol. abs/2112.01527, 2021. [Online]. Available: <https://arxiv.org/abs/2112.01527>
- [15] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask dino: Towards a unified transformer-based framework for object detection and segmentation," 2022. [Online]. Available: <https://arxiv.org/abs/2206.02777>
- [16] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," 2023. [Online]. Available: <https://arxiv.org/abs/2205.08534>
- [17] D. Shi, "Transnext: Robust foveal visual perception for vision transformers," 2024. [Online]. Available: <https://arxiv.org/abs/2311.17132>
- [18] J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," 2022. [Online]. Available: <https://arxiv.org/abs/2211.06220>
- [19] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: <https://arxiv.org/abs/2304.02643>
- [20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *CoRR*, vol. abs/1608.05442, 2016. [Online]. Available: <http://arxiv.org/abs/1608.05442>
- [21] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.
- [22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *CoRR*, vol. abs/1604.01685, 2016. [Online]. Available: <http://arxiv.org/abs/1604.01685>
- [24] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Job Scheduling Strategies for Parallel Processing*, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 44–60.
- [25] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala, "Pytorch distributed: Experiences on accelerating data parallel training," *CoRR*, vol. abs/2006.15704, 2020. [Online]. Available: <https://arxiv.org/abs/2006.15704>
- [26] S. Prusty, S. Patnaik, and S. K. Dash, "Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. Volume 4 - 2022, 2022. [Online]. Available: <https://www.frontiersin.org/journals/nanotechnology/articles/10.3389/fnano.2022.972421>