

Desenvolvimento de um Sistema de Segmentação Semântica Embarcado

SANTOS, R. S. (IFPB, Cajazeiras), SOUSA, R. M. (IFPB, Cajazeiras), SILVA, M. A. L. (IFPB, Cajazeiras) e PEREIRA, J. F. M. (IFPB, Cajazeiras)

E-mails: renan.saraiva@academico.ifpb.edu.br, raphaell.sousa@ifpb.edu.br, miguel.lacerda@academico.ifpb.edu.br e jose.muniz@academico.ifpb.edu.br.

Área de conhecimento (Tabela CNPq): 1.03.01.01-1 Computabilidade e Modelos de Computação

Palavras-chave: benchmarking; off-road; visão computacional embarcado

1. Introdução

A competição Baja SAE Brasil desafia estudantes de engenharia a desenvolverem veículos *off-road*, nos quais a percepção ambiental em tempo real é essencial para garantir segurança e desempenho em terrenos adversos. Nesse contexto, a segmentação semântica — técnica de visão computacional que classifica cada pixel de uma imagem de acordo com sua categoria — destaca-se como ferramenta fundamental para a identificação de elementos críticos da pista (MORAIS, 2020). No entanto, modelos de segmentação baseados em redes neurais profundas exigem elevado poder computacional, como o fornecido por *GPUs* dedicadas, o que dificulta sua aplicação em sistemas embarcados com restrições de custo e consumo energético, características típicas dos veículos da competição Baja SAE (SILVA, 2024).

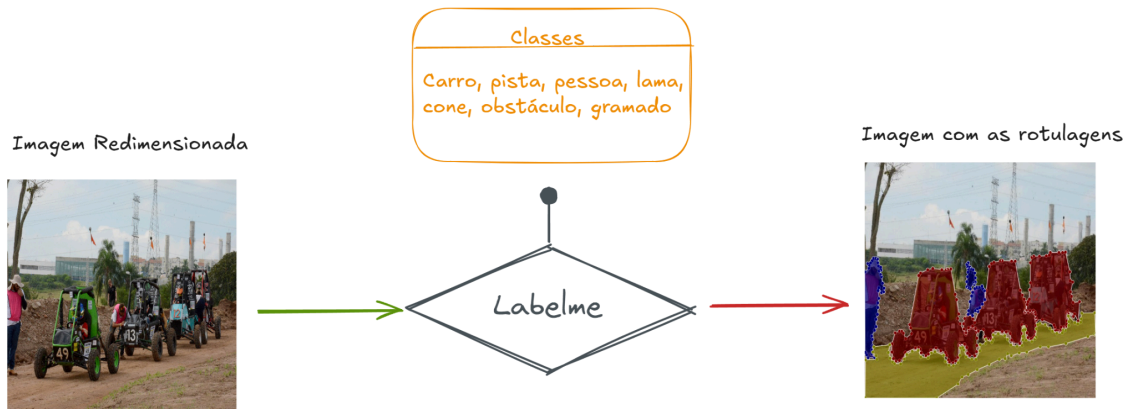
Diante desse desafio, esta pesquisa propõe um estudo comparativo (*benchmarking*) entre as arquiteturas de redes neurais *Fully Convolutional Network (FCN)* e *U-Net*, com o objetivo de identificar qual modelo oferece o melhor equilíbrio entre acurácia de segmentação e eficiência computacional. O objetivo geral é viabilizar a futura implementação de um sistema de segmentação embarcado em plataformas de baixo custo, como a *Raspberry Pi*. Como objetivos específicos, destacam-se: (i) a construção de um *dataset* rotulado com imagens reais da competição Baja SAE; (ii) o treinamento e validação dos modelos em ambiente *Linux (Ubuntu)*, mais próximo da arquitetura da *Raspberry Pi 3*; e (iii) a comparação dos modelos quanto às métricas de acurácia e potencial de embarque, considerando o desempenho computacional e a compatibilidade com bibliotecas otimizadas como *TensorFlow Lite*.

2. Materiais e métodos

A metodologia adotada seguiu o ciclo *PACE (Plan, Analyze, Construct, Execute)*. Na fase de Planejamento, definiu-se a *Raspberry Pi 3* como plataforma-alvo, as arquiteturas *U-Net* e *FCN-8s* como modelos para o estudo de *benchmarking*, e as classes de interesse para segmentação — carro, pessoa, gramado, pista, obstáculo, cone e lama — com base em cenários da competição Baja SAE.

Na fase de Análise, foi criado um conjunto de dados (*dataset*) composto por imagens obtidas durante o Baja SAE Nacional 2025 e a Prova de Enduro de *Michigan* 2024. Essas imagens foram submetidas a um processo de pré-processamento e posteriormente rotuladas manualmente, *pixel a pixel*, utilizando o *software LabelMe*. A Figura 1 apresenta um esquema que ilustra essa etapa do processo.

Figura 1 - Processo de Rotulagem de Imagens com o *Labelme* para Segmentação Semântica.



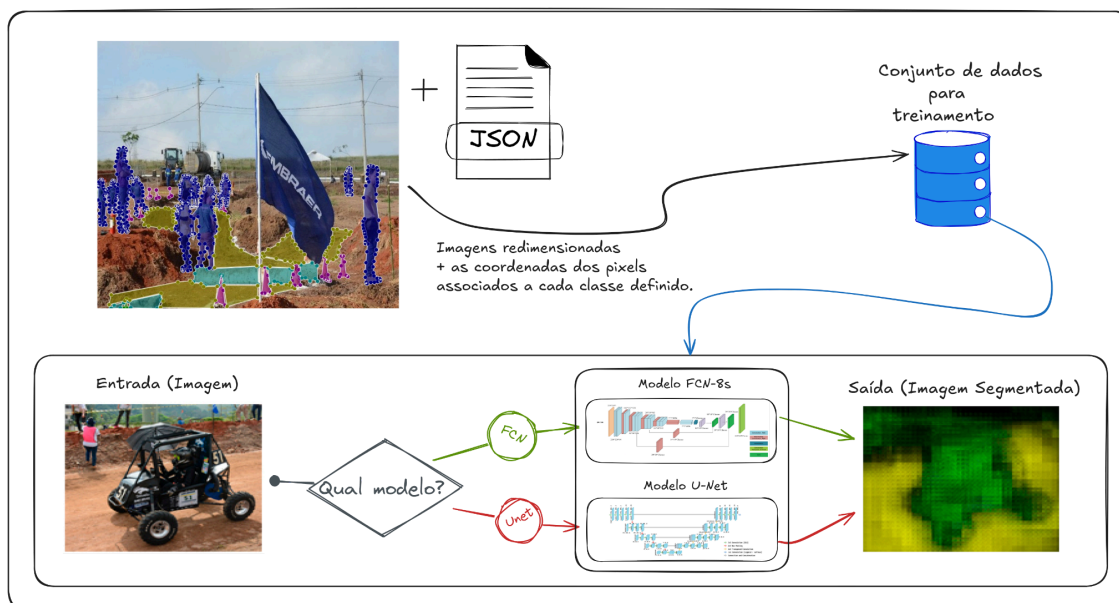
Fonte: Autoria própria (2025).

Na etapa de Construção, foram implementadas as arquiteturas *FCN-8s* — com *skip connections* provenientes das camadas *pool3*, *pool4* e da camada convolucional final — e *U-Net*, caracterizada por uma estrutura *encoder-decoder* simétrica, também com *skip connections* e camadas específicas. Ambas as redes foram finalizadas com uma camada de convolução 1×1 , seguida por função de ativação *softmax* para classificação por *pixel*.

O treinamento dos modelos foi realizado em *GPU*, utilizando as bibliotecas *TensorFlow* e o otimizador *Adam*. Durante esse processo, foram monitoradas métricas como o Coeficiente de *Dice*, *IoU* (*Intersection over Union*), *Loss* e Acurácia, além da exploração de técnicas de otimização, como poda de rede (*pruning*). Foram utilizadas 138 imagens no total, com 300 épocas de treinamento.

A Figura 2 apresenta o diagrama do projeto. O conjunto de dados utilizado para o treinamento é composto por imagens e suas respectivas coordenadas de rótulo, obtidas a partir da rotulação manual realizada na ferramenta *LabelMe*. Após o treinamento, uma imagem do conjunto de validação pode ser selecionada para inferência, etapa em que os modelos processam a entrada e geram como saída a respectiva imagem segmentada.

Figura 2 - Fluxo de Processamento para Segmentação Semântica de Imagens com Redes Neurais Convolucionais.



Fonte: Autoria própria (2025).

Finalmente, na Execução, os modelos foram comparados no ambiente *Ubuntu*, por meio de métricas de segmentação. Para isso, foi utilizada uma implementação customizada do *TensorBoard*, composta por duas classes principais: *TrainValTensorBoard* e *TensorBoardMask*. A classe *TrainValTensorBoard* foi responsável por registrar separadamente os valores das métricas de validação e treinamento, permitindo a análise visual simultânea desses dados no painel do *TensorBoard*. Já a classe *TensorBoardMask* possibilitou a geração e visualização das máscaras segmentadas previstas pelos modelos durante o processo de treinamento.

3. Resultados e discussão

Na Tabela 1 estão sintetizados os resultados dos parâmetros obtidos após os treinamentos e validações dos modelos. Para este *benchmarking*, pensou-se utilizar as variáveis: Função de Perda (*Loss*), Coeficiente *Dice*, *Intersection over Union* (*IoU*) e a Acurácia Categórica (*Categorical Accuracy*)

Tabela 1 – Métricas de Avaliação dos Modelos U-Net e FCN-8s.

Modelo	Loss	Dice	IoU	Categorical Accuracy
U-Net	0,3651	0,8071	0,6838	0,8653
FCN - 8s	0,3706	0,8319	0,7324	0,8939

Fonte: Autoria própria (2025).

A avaliação comparativa dos modelos *U-Net* e *FCN-8s* apresenta variáveis importantes do desempenho dos mesmos. Conforme apresentado na Tabela 1, a arquitetura *FCN-8s* demonstrou um desempenho superior em diversas métricas cruciais de segmentação, alcançando valores de Coeficiente de *Dice* (0,8319), *IoU* (0,7324) e Acurácia Categórica (0,8939) mais elevados em comparação com a *U-Net* (*Dice*: 0,8071, *IoU*: 0,6838, Acurácia Categórica: 0,8653). Por outro lado, a *U-Net* apresentou uma *loss* ligeiramente menor (0,3651 contra 0,3706 da *FCN-8s*), o que pode indicar um ajuste sutilmente diferente durante o treinamento.

Embora este trabalho tenha se concentrado na avaliação de métricas de acurácia, como *Dice*, *IoU* e acurácia categórica, reconhece-se a importância da análise de desempenho computacional (como FPS, tamanho do modelo e consumo de *RAM*) para aplicações embarcadas. No entanto, tais métricas ainda não foram medidas diretamente na *Raspberry Pi 3*, plataforma-alvo desta pesquisa. A execução dos treinamentos e validações foi realizada em ambiente *Linux (Ubuntu)*, cuja estrutura é semelhante à do sistema operacional utilizado na *Raspberry*, proporcionando maior proximidade com o ambiente futuro de implantação.

A análise computacional completa será realizada como etapa futura, incluindo testes reais de inferência no *hardware* embarcado, visando avaliar a viabilidade prática dos modelos em tempo real e sob restrições de processamento e memória.

5. Considerações finais

Embora as métricas de acurácia favoreçam a arquitetura *FCN-8s*, a decisão sobre qual modelo será embarcado deve considerar também aspectos de desempenho computacional, ainda não avaliados diretamente na *Raspberry Pi 3*. A realização dos treinamentos e validações em sistema operacional *Ubuntu* garantiu maior aderência à arquitetura da plataforma embarcada, possibilitando um ambiente de testes mais próximo da realidade futura.

Como próximos passos, planeja-se a conversão dos modelos para o formato *TensorFlow Lite*, mais leve e adequado para dispositivos embarcados com restrições de memória e processamento. Essa conversão é necessária, uma vez que a *Raspberry Pi 3* apresenta limitações de compatibilidade com bibliotecas utilizadas na fase de desenvolvimento. Além disso, será conduzida a análise prática do desempenho dos modelos no dispositivo embarcado, com foco em tempo de inferência (*FPS*), uso de *RAM*, tamanho do modelo, e compatibilidade com otimizações adicionais, como quantização de redes e aceleração por hardware dedicado. Essas medidas são essenciais para garantir a operação em tempo real nos veículos Baja SAE e permitir futuras expansões do sistema de visão computacional.

Agradecimentos

Os autores agradecem ao Instituto Federal da Paraíba – Campus Cajazeiras pelo incentivo e fomento às iniciativas de pesquisa acadêmica, que possibilitam o desenvolvimento de soluções inovadoras e aplicadas à realidade tecnológica atual. Agradecem também ao Laboratório de Sistemas Inteligentes (LABSIN), coordenado pelos professores Dr. Raphael Sousa e Dr. Leandro Honorato, pelo suporte técnico, infraestrutura e orientações fundamentais durante a realização deste trabalho.

Referências

MORAIS, G. A. P. D. Aprendizagem profunda por reforço combinada com controlador recursivo robusto para controle lateral baseado em visão computacional. 2020 (Doctoral dissertation, Universidade de São Paulo). Disponível em: <https://www.teses.usp.br/teses/disponiveis/18/18153/tde-25082020-092618/en.php>. Acesso em: 03/01/2025.

SILVA, I. C. L. APES-SOFT: uma abordagem eficiente em classificação de objetos 3d em nuvens de pontos com redes neurais convolucionais. 2024. Disponível em: <https://repositorio.ufc.br/handle/riufc/78667>. Acesso em: 09/01/2025.