INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS

RENAN SARAIVA DOS SANTOS

SEGMENTAÇÃO SEMÂNTICA PARA PERCEPÇÃO AMBIENTAL EM VEÍCULOS BAJA SAE: UM ESTUDO COMPARATIVO ENTRE FCN-8S E U-NET

Cajazeiras-PB
2025

RENAN SARAIVA DOS SANTOS

SEGMENTAÇÃO SEMÂNTICA PARA PERCEPÇÃO AMBIENTAL EM VEÍCULOS BAJA SAE: UM ESTUDO COMPARATIVO ENTRE FCN-8S E U-NET

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba-*Campus* Cajazeiras, como parte dos requisitos para a obtenção do Título de Bacharel em Engenharia de Controle e Automação, sob Orientação do Prof. Dr. Raphaell Maciel de Sousa**.**

Cajazeiras-PB
2025

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

S237s Santos, Renan Saraiva dos.

Segmentação semântica para percepção ambiental em veículos Baja SAE : um estudo comparativo entre FCN-8S e U-Net / Renan Saraiva dos Santos.– 2025.

39f. : il.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador(a): Prof. Dr. Raphaell Maciel de Sousa.

1. Controle automático. 2. Baja SAE. 3. Redes neurais convolucionais. 4. Desempenho veicular. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 681.5(043.2)

RENAN SARAIVA DOS SANTOS

SEGMENTAÇÃO SEMÂNTICA PARA PERCEPÇÃO AMBIENTAL EM VEÍCULOS BAJA SAE: UM ESTUDO COMPARATIVO ENTRE FCN-8S E U-NET

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, *Campus* Cajazeiras, como parte dos requisitos para a obtenção do Título de Bacharel em Engenharia de Controle e Automação.

Aprovado em 19 de dezembro de 2025.

BANCA EXAMINADORA

gov.br Documento assinado digitalmente
RAPHAELL MACIEL DE SOUSA
Data: 23/01/2026 20:29:38-0300
Verifique em https://validar.iti.gov.br

Raphaell Maciel de Sousa – IFPB-*Campus* Cajazeiras
Orientador

gov.br Documento assinado digitalmente
LEANDRO HONORATO DE SOUZA SILVA
Data: 23/01/2026 15:23:13-0300
Verifique em https://validar.iti.gov.br

Leandro Honorato de Souza Silva – IFPB-*Campus* Cajazeiras
Examinador 1

gov.br Documento assinado digitalmente
FABIO ARAUJO DE LIMA
Data: 23/01/2026 15:37:55-0300
Verifique em https://validar.iti.gov.br

Fábio Araújo de Lima – IFPB-*Campus* Cajazeiras
Examinador 2

Dedico este trabalho à minha família, que sempre acreditou em mim mesmo nos momentos mais desafiadores. Aos meus pais, por todo amor, paciência e apoio incondicional, que foram a base da minha jornada e o combustível para nunca desistir. E à minha namorada, pelo carinho, compreensão e por estar ao meu lado em cada etapa desta caminhada, compartilhando sonhos, desafios e conquistas. E a todos que, de alguma forma, fizeram parte dessa caminhada, meu sincero agradecimento por acreditarem que este sonho era possível.

AGRADECIMENTOS

Ao Instituto Federal da Paraíba (IFPB), *Campus* Cajazeiras, expresso minha mais profunda gratidão. Cada projeto, cada conversa com professores e colegas, e cada obstáculo superado contribuíram para que eu me tornasse alguém mais preparado, confiante e apaixonado pela área que escolhi seguir. O incentivo à pesquisa, à inovação e ao pensamento crítico foram essenciais para que eu pudesse construir este trabalho com propósito e convicção.

Ao meu professor orientador, Raphaell Maciel, deixo um agradecimento sincero e especial. Sua paciência, disponibilidade e dedicação constante foram fundamentais para o desenvolvimento do conhecimento técnico. A forma como conduziu cada orientação, sempre com clareza e incentivo, foi determinante para que eu conseguisse transformar ideias abstratas em resultados concretos.

Aos colaboradores e auxiliares que participaram das etapas iniciais deste trabalho, registro minha profunda gratidão. A ajuda de cada um foi essencial para que as fases de preparação e coleta de dados fossem realizadas com eficiência e leveza.

Aos amigos e colegas de curso Felliph Nascimento, Julio Matos, Miguel Ângelo e Henrique Sobral e demais, deixo meus agradecimentos. Foram vocês que tornaram esta caminhada mais leve, divertida e inesquecível. Cada conversa, cada noite em claro estudando para provas e atividades, cada momento de desabafo ou comemoração contribuiu para tornar essa jornada diferenciada. Sem vocês, o curso teria sido apenas uma sequência de disciplinas; com vocês, tornou-se uma experiência de mudanças, maturidade, companheirismo e memórias que levarei comigo por toda a vida.

À minha namorada, Hellen Kelly, por todo o amor e carinho. Oferecendo palavras de incentivo e compreensão quando mais precisei. Agradeço por compartilhar comigo não apenas os desafios, mas também as pequenas conquistas que tornaram tão significativa.

E, acima de tudo, à minha família, dedico esta conquista, Francisca Saraiva, José Nilton e Isael Saraiva. Vocês foram a base de tudo: o suporte nos momentos de dúvida, o incentivo constante quando as forças pareciam faltar, e o amor incondicional que me manteve firme até o fim. A cada desafio enfrentado, encontrei em vocês a motivação para seguir em frente. Esta vitória não é apenas minha, mas de todos que estiveram ao meu lado acreditando, mesmo quando os resultados ainda não eram visíveis. Sem o apoio, a paciência e a fé de vocês, nada disso seria possível.

RESUMO

A percepção ambiental em tempo real apresenta-se como um desafio inovador para otimizar a segurança e o desempenho dos veículos na competição Baja SAE. Este trabalho apresenta um estudo comparativo (*benchmarking*) entre as arquiteturas de redes neurais convolucionais *Fully Convolutional Network* e U-Net, com a finalidade de determinar o modelo que oferece o melhor balanço entre acurácia de segmentação e potencial de eficiência para futura implantação em sistemas embarcados de baixo custo. A metodologia foi sistematicamente estruturada seguindo o ciclo *PACE* (*Plan, Analyze, Construct, Execute*), iniciando com a criação de um conjunto de dados customizado, composto por 138 imagens de competições reais, as quais foram rotuladas manualmente com sete classes de interesse. A avaliação quantitativa, baseada nas métricas de *Intersection over Union* (*IoU*), Coeficiente de *Dice* e Acurácia Categórica, demonstrou a superioridade da arquitetura *FCN-8s*. O modelo *FCN-8s* alcançou um *IoU* de 0,7324 e um Coeficiente de *Dice* de 0,8319, superando a *U-Net*, que obteve 0,6838 e 0,8071, respectivamente. Conclui-se que, embora a FCN-8s apresente maior precisão de segmentação para este domínio, a seleção final para a implantação embarcada dependerá de uma análise subsequente do desempenho computacional (tempo de inferência e uso de memória) no *hardware* alvo, o que exigirá a conversão dos modelos para o formato *TensorFlow Lite*.

Palavras-chave: segmentação semântica; Baja SAE; redes neurais convolucionais; sistemas embarcados.

ABSTRACT

Low-cost environmental awareness represents an innovative challenge to improve the safety and performance of vehicles in the Baja SAE competition. This work presents a comparative study (benchmarking) between the Fully Convolutional Network (FCN-8s) and U-Net convolutional neural network architectures, aiming to determine the model that offers the best balance between segmentation accuracy and efficiency potential for future deployment on low-cost embedded systems. The methodology was systematically structured following the PACE (Plan, Analyze, Construct, Execute) cycle, beginning with the creation of a custom dataset composed of 138 images from actual competitions, which were manually labeled with seven classes of interest. The quantitative evaluation, based on Intersection over Union (IoU), Dice Coefficient, and Categorical Accuracy metrics, demonstrated the superiority of the FCN-8s architecture. The FCN-8s model achieved an IoU of 0.7324 and a Dice Coefficient of 0.8319, surpassing the U-Net, which obtained 0.6838 and 0.8071, respectively. It is concluded that, although FCN-8s exhibits greater segmentation precision for this domain, the final selection for embedded deployment will depend on a subsequent analysis of computational performance (inference time and memory usage) on the target hardware, which will require converting the models to the TensorFlow Lite format.

Keywords: semantic segmentation; Baja SAE; convolutional neural networks; embedded systems.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SAE – Society of Automotive Engineers International.

GPUs – Graphics Processing Units.

FCN – Fully Convolutional Network.

IoU – Intersection over Union.

Loss – Função de Perda.

CNNs – Convolutional Neural Networks.

PACE – Plan, Analyze, Construct, Execute.

MLOps – Machine Learning Operations.

SUMÁRIO

1 INTRODUÇÃO

A competição Baja SAE Brasil representa um dos maiores desafios de engenharia para acadêmicos de graduação, propondo o projeto, a construção e a validação de um protótipo de veículo *off-road* de alto desempenho. Embora a robustez mecânica seja o alicerce do projeto, o sucesso em provas de longa duração (enduro) está intrinsecamente ligado à capacidade do sistema de interagir de forma segura com um ambiente não estruturado e hostil. Nesse cenário, a percepção ambiental em tempo real emerge não apenas como um diferencial, mas como um fator crítico de sobrevivência do protótipo. Em condições de visibilidade degradada e fadiga do piloto, a capacidade automática de identificar obstáculos, delimitar o trajeto navegável e detectar outros competidores torna-se determinante para evitar colisões e atolamentos, ações fundamentais para a otimização da pilotagem e, sobretudo, para a garantia da segurança operacional (Fang; Cai, 2021).

Tradicionalmente, a interpretação do ambiente depende exclusivamente do piloto. Contudo, a evolução dos sistemas eletrônicos e da inteligência artificial abre precedente para o desenvolvimento de sistemas de assistência ao piloto, aumentando a consciência situacional. Dentre as tecnologias disponíveis, a visão computacional, por meio da análise de imagens digitais, oferece uma solução rica em informações e de custo relativamente baixo. Especificamente, a técnica de segmentação semântica, que consiste em classificar cada pixel de uma imagem em uma categoria pré-definida, destaca-se como uma abordagem poderosa para uma compreensão densa e detalhada da cena. Através dela, é possível gerar um mapa completo do ambiente, distinguindo com precisão áreas de “pista”, “gramado”, “lama”, “obstáculo”, “cone”, “pessoa” e carro”.

Apesar de seu potencial, a implementação de modelos de segmentação semântica de última geração, baseados em redes neurais profundas, impõe um desafio substancial. Tais modelos demandam elevado poder computacional, geralmente suprido por Unidades de Processamento Gráfico dedicadas, que são inviáveis em um protótipo Baja SAE devido a restrições severas de custo, consumo energético, peso e dissipação térmica. A aplicação desta tecnologia é classificada como inovadora justamente por seu ineditismo no cenário atual da competição: até o momento, não há registros na literatura técnica ou nos boxes da competição de equipes que tenham validado a implementação embarcada de um sistema de percepção densa dessa natureza em seus veículos *off-road*. A concretização dessa proposta depende, portanto, da superação de uma lacuna tecnológica: a adaptação de arquiteturas de redes neurais para

operar eficientemente em sistemas embarcados de baixo custo, como a *Raspberry Pi* (Silva, 2024).

Diante do exposto, este trabalho propõe um estudo comparativo (*benchmarking*) entre duas das mais influentes arquiteturas de segmentação semântica, a *Fully Convolutional Network (FCN)* e a *U-Net*, com o propósito de avaliar seu desempenho e determinar sua viabilidade para a aplicação em veículos Baja SAE. A escolha dessas arquiteturas justifica-se por apresentarem abordagens distintas e fundamentais para a segmentação densa, sendo amplamente reconhecidas na literatura por sua eficácia (Long; Shelhamer; Darrell, 2015).

A relevância desta pesquisa reside em sua dupla contribuição. Do ponto de vista acadêmico, realiza-se uma análise de desempenho de arquiteturas clássicas em um domínio de aplicação novo e desafiador, para o qual não existem conjuntos de dados públicos disponíveis. Do ponto de vista prático e tecnológico, este estudo representa o passo inicial para o desenvolvimento de um sistema de percepção ambiental de baixo custo que pode ser integrado aos veículos da competição, constituindo uma inovação com potencial para aumentar a competitividade e a segurança das equipes.

O presente trabalho está estruturado em cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica sobre os conceitos de segmentação semântica e as arquiteturas *FCN* e *U-Net*. O Capítulo 3 detalha a metodologia empregada na construção do dataset, na implementação e no treinamento dos modelos. O Capítulo 4 apresenta e discute os resultados quantitativos e qualitativos obtidos. Por fim, o Capítulo 5 expõe as conclusões do estudo, suas limitações e aponta direções para trabalhos futuros.

2 OBJETIVOS

Este capítulo delineia os propósitos norteadores desta pesquisa, definindo a meta central e os passos metodológicos necessários para alcançá-la.

2.1 OBJETIVO GERAL

Realizar uma análise comparativa de desempenho entre as arquiteturas de redes neurais convolucionais FCN e U-Net para a tarefa de segmentação semântica, mediante a construção de uma base de dados aplicada ao ambiente da competição Baja SAE, com vistas à futura implementação em sistemas embarcados.

2.2 OBJETIVOS ESPECÍFICOS

Para que o objetivo geral fosse alcançado, foram estabelecidos os seguintes objetivos específicos:

- construir um conjunto de dados customizado, composto por imagens representativas do ambiente da competição Baja SAE, e realizar a rotulagem manual para a tarefa de segmentação semântica;
- desenvolver e treinar os algoritmos de redes neurais baseando-se nas arquiteturas *FCN-8s* e *U-Net*, utilizando a base de dados desenvolvida e um ambiente computacional alinhado à plataforma-alvo;
- comparar quantitativamente e avaliar o desempenho dos modelos treinados por meio de métricas de avaliação padrão para segmentação, como *Intersection over Union* (*IoU*), Coeficiente de *Dice*, Acurácia Categórica e Função de Perda (*Loss*);
- analisar os resultados obtidos para determinar qual arquitetura apresenta o balanço mais promissor entre acurácia de segmentação e potencial de eficiência computacional para a aplicação embarcada.

3 REVISÃO DE LITERATURA

Este capítulo contém a exposição ordenada do assunto tratado, apresentando os conceitos e as obras com maior relevância para a pesquisa desenvolvida.

3.1 REDES NEURAIS CONVOLUCIONAIS (*CNN*)

As Redes Neurais Convolucionais (*Convolutional Neural Networks - CNNs*) constituem uma classe de modelos de aprendizado profundo que se tornaram o padrão-ouro para tarefas de análise de imagens (Lecun; Bengio; Hinton, 2015). Sua arquitetura é inspirada no córtex visual humano e se mostra extremamente eficaz na extração de hierarquias de características espaciais a partir de dados com topologia de grade.

3.1.1 Fully Convolutional Network (FCN)

A camada de convolução é o elemento fundamental de uma *CNN*, responsável por aprender a representar as características locais da entrada. A operação central é a convolução, que consiste em deslizar um pequeno filtro (ou *kernel*) sobre a entrada, calculando o produto escalar em cada posição. Esse mecanismo de compartilhamento de pesos (*weight sharing*) através do *kernel* é o que permite à rede detectar padrões (como arestas ou texturas) independentemente de sua posição na imagem (Lecun *et al.*, 1998).

Considere uma entrada I bidimensional (por exemplo, um mapa de características de uma camada anterior ou a própria imagem de entrada) e um filtro K. A operação de convolução $(I.K)$ em uma posição (x, y) é definida, na prática da literatura de *Deep Learning*, como uma correlação-cruzada (*cross-correlation*), conforme a Equação 1 (Goodfellow; Bengio; Courville, 2016).

$$(I.K)(x,y) = \sum_{i} \sum_{j} I(x-i, y-j)K(i,j) \quad (1)$$

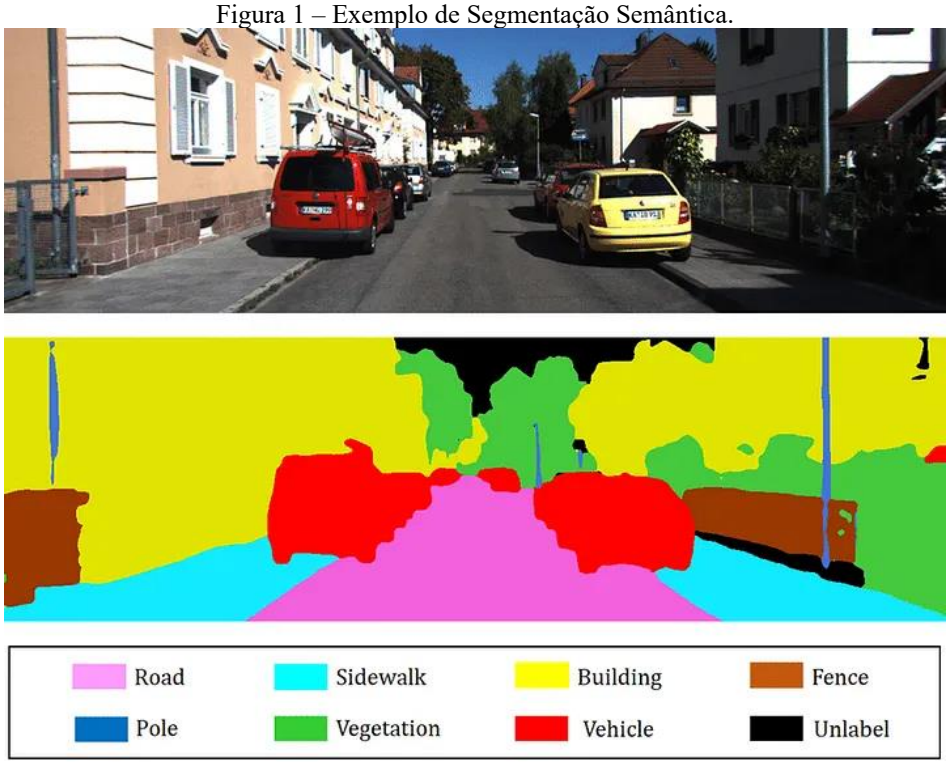
Onde i *e* j percorrem as dimensões do filtro K.

Se uma imagem de entrada possui múltiplas canais, o filtro também terá a mesma profundidade de canais, e a convolução é realizada sobre todos os canais, somando-se os resultados (Guimarães, 2025). A saída de cada filtro é um mapa de características *2D*. Se a camada utiliza *N* filtros, a saída será um volume *3D* com *N* mapas de características.

3.2 SEGMENTAÇÃO SEMÂNTICA

A segmentação semântica é uma tarefa de visão computacional que visa atribuir um rótulo de classe a cada pixel de uma imagem (Long; Shelhamer; Darrell, 2015), realizando uma classificação densa. Diferentemente da classificação de imagens (que atribui um único rótulo à imagem inteira), a segmentação particiona a imagem em regiões semanticamente coerentes (Garcia-Garcia *et al.*, 2018), conforme ilustrado na Figura 1.

Figura 1 – Exemplo de Segmentação Semântica.



Fonte: Jeong, Yoon, Park (2018).

3.3 ARQUITETURAS PARA SEGMENTAÇÃO SEMÂNTICA

A transição das Redes Neurais Convolucionais (CNNs) de tarefas de classificação de imagem para a segmentação semântica exigiu o desenvolvimento de arquiteturas especializadas. O desafio central reside em realizar uma predição densa, classificando cada pixel, e não apenas a imagem inteira. Para isso, a maioria das arquiteturas modernas adota um paradigma de codificador-decodificador (*encoder-decoder*). O codificador, tipicamente uma rede de classificação pré-treinada, é responsável por extrair características hierárquicas e reduzir a resolução espacial. O decodificador, por sua vez, tem a tarefa de realizar o upsampling desses mapas de características para reconstruir o mapa de segmentação na resolução original da entrada. Diversas arquiteturas foram propostas na literatura para otimizar essa tarefa. Entre

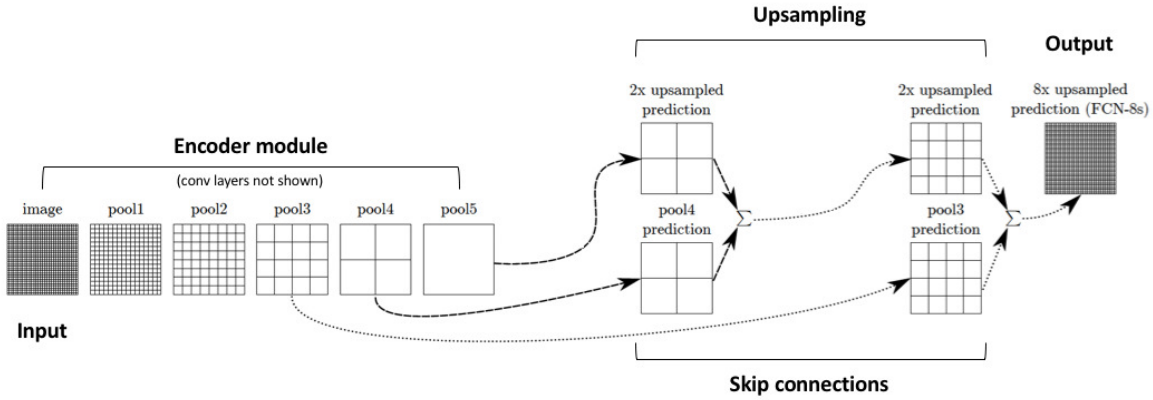
as mais influentes e que servem de base para muitos trabalhos subsequentes, destacam-se a *Fully Convolutional Network* (FCN) e a U-Net, que são os objetos de estudo deste trabalho.

3.3.1 Fully Convolutional Network (FCN)

A *FCN*, proposta por Long, Shelhamer e Darrell (2015), foi uma arquitetura seminal que adaptou com sucesso as *CNNs* de classificação para a tarefa de segmentação. A principal inovação foi a substituição das camadas totalmente conectadas (*fully connected*) por camadas convolucionais 1 x 1, permitindo que a rede processasse imagens de qualquer tamanho e gerasse um mapa de calor como saída. Para refinar os detalhes da segmentação, a *FCN* introduziu o conceito de *skip connections* (conexões de atalho), que combinam informações de diferentes escalas da rede (Long; Shelhamer; Darrell, 2015).

Devido às operações de *pooling* e *stride* nas camadas convolucionais iniciais (o *encoder*), a resolução espacial dos mapas de características é progressivamente reduzida. Para recuperar o mapa de segmentação para a resolução da imagem original, a *FCN* emprega camadas de convolução transposta (*transposed convolution*), também conhecidas como "deconvolução" ou *upsampling* (Zeiler; Fergus, 2014). Esta operação é o inverso da convolução e permite que a rede aprenda a expandir a resolução espacial, preenchendo os detalhes perdidos. A Figura 2 ilustra o conceito de *upsampling* em uma *FCN*.

Figura 2 – Ilustração do Conceito de *Upsampling* em uma *FCN*.



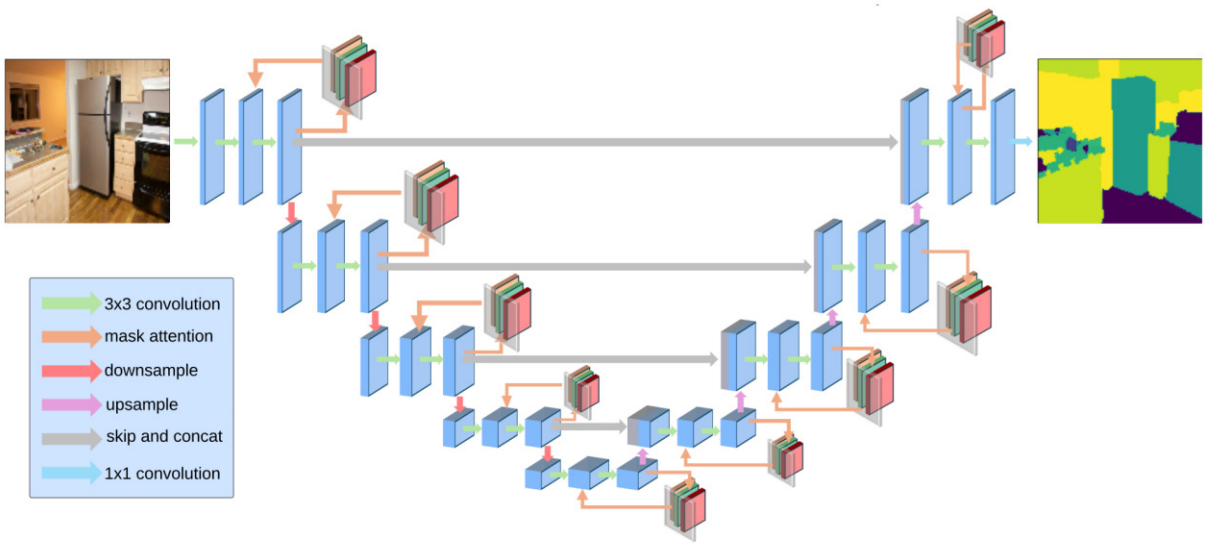
Fonte: Noori, Shaker, Azeez (2022).

3.3.2 Arquitetura U-Net

Desenvolvida por Ronneberger, Fischer e Brox (2015), a *U-Net* notabilizou-se por sua arquitetura simétrica em formato de "U", conforme esquematizado na Figura 3. Tal arquitetura

é composta por um caminho de contração (*encoder*) e um caminho de expansão (*decoder*). O seu diferencial reside nas proeminentes *skip connections*, que concatenam os mapas de características de alta resolução do *encoder* com os mapas correspondentes no *decoder*, resultando em segmentações com limites de objetos muito bem definidos.

Figura 3 – Arquitetura U-Net simplificada.



Fonte: Cheng *et al.* (2025).

3.4 MÉTRICAS DE AVALIAÇÃO

A avaliação quantitativa do desempenho de modelos de segmentação semântica é fundamental para comparar diferentes arquiteturas e compreender sua eficácia na tarefa de classificação de pixel a pixel. As métricas são calculadas comparando a máscara de segmentação predita pelo modelo com a máscara de referência, conhecida como *ground truth*.

3.4.1 Intersection over Union (IoU)

A métrica *IoU*, também denominada Coeficiente de Jaccard, é uma das mais amplamente utilizadas na avaliação de segmentação de imagens (Everingham *et al.,* 2010). Ela mede a similaridade entre dois conjuntos de amostras e é calculada como a razão entre a área de interseção e a área de união entre a máscara predita e a máscara de *ground truth*, conforme a Equação 2.

$$\frac{Area\ of\ Intersection}{Area\ of\ Union} = \frac{A \cup B}{A \cap B} \tag{2}$$

O valor do *IoU* varia de 0 a 1, onde 0 indica nenhuma sobreposição entre as máscaras e 1 representa uma sobreposição perfeita. Para tarefas de segmentação multiclasse, o *IoU* é frequentemente calculado para cada classe individualmente e, em seguida, uma média é obtida para fornecer uma medida geral do desempenho do modelo em todas as classes.

3.4.2 Coeficiente de DICE (DSC) ou F1-Score

O Coeficiente de *Dice* (*DSC*), também conhecido como *F1-Score* ou Índice de *Sørensen-Dice*, é outra métrica comum para avaliar a similaridade espacial entre dois objetos segmentados (Sorensen, 1948). Tal métrica, é definida como duas vezes a área de interseção entre a máscara predita e a máscara de *ground truth*, dividida pela soma das áreas das duas máscaras, como mostra a Equação 3.

$$Dice\ Coefficient = \frac{2.area\ od\ overlapped}{total\ area} \tag{3}$$

O *DSC* também varia de 0 a 1. Embora sejam correlacionadas, o *DSC* tende a ser mais sensível a pequenas discrepâncias em objetos menores e pode penalizar mais severamente predições incorretas em comparação com o *IoU*.

3.4.3 Acurácia Categórica (Categorical Accuracy)

A Acurácia Categórica, no contexto da segmentação semântica, representa a proporção de pixels que foram classificados corretamente pelo modelo em relação ao número total de pixels na imagem. A Equação 4 mostra como é calculada.

$$Acurácia = \frac{Número\ de\ Pixels\ Classificados\ Corretamente}{Número\ Total\ de\ Pixels} \tag{4}$$

Embora seja uma métrica intuitiva, a acurácia pode ser enganosa em casos de desequilíbrio de classes, onde classes majoritárias dominam o cálculo e podem mascarar um desempenho insatisfatório em classes minoritárias (Garcia-Garcia *et al.*, 2018). Por exemplo, em uma imagem com predominância de "gramado", um modelo que classifica a maioria dos pixels como "gramado" pode ter uma alta acurácia, mesmo que falhe ao detectar pequenos "obstáculos". Por isso, *IoU* e *Dice* são geralmente preferidos para uma avaliação mais robusta em segmentação.

3.4.4 Função de Perda (Loss Function)

A Função de Perda é o mecanismo matemático que viabiliza o aprendizado da rede neural. Ela opera como o objetivo de otimização, fornecendo um valor escalar diferenciável que quantifica o erro entre a predição e o *ground truth*. É através da minimização desta função que o algoritmo de *Backpropagation* calcula os gradientes necessários para atualizar os pesos da rede (Goodfellow; Bengio; Courville, 2016).

Para tarefas de segmentação semântica, onde cada pixel é classificado em uma das N classes, a função de perda mais comum é a Entropia Cruzada Categórica (*Categorical Cross-Entropy*). Para um único *pixel i* e N classes, a perda é calculada pela Equação 5.

$$L = -\sum_{C=1}^{N} \quad y_{i,c} log\ (P_{i,c}) \tag{5}$$

Onde:

$y_{i,c}$: é 1 se o pixel i pertence à classe c (no *ground truth*), e 0 caso contrário;

$P_{i,c}$: é a probabilidade predita pelo modelo para o pixel i pertencer à classe c.

A Entropia Cruzada Categórica penaliza fortemente as predições incorretas com alta confiança, guiando a rede a ajustar seus pesos para que as probabilidades preditas se aproximem das distribuições verdadeiras de cada pixel.

4 METODOLOGIA

A presente seção detalha a abordagem metodológica adotada para a realização deste trabalho, abrangendo a classificação da pesquisa, as estratégias de levantamento e análise de dados, e a descrição das atividades desenvolvidas. O fluxo de trabalho foi estruturado com base no ciclo *PACE* (*Plan*, *Analyze*, *Construct*, *Execute*), um *framework* que organiza o desenvolvimento de projetos de forma sistemática e iterativa, alinhado aos princípios de *MLOps* (*Machine Learning Operations*) para garantir a reprodutibilidade, o monitoramento e a qualidade da engenharia de *machine learning*.

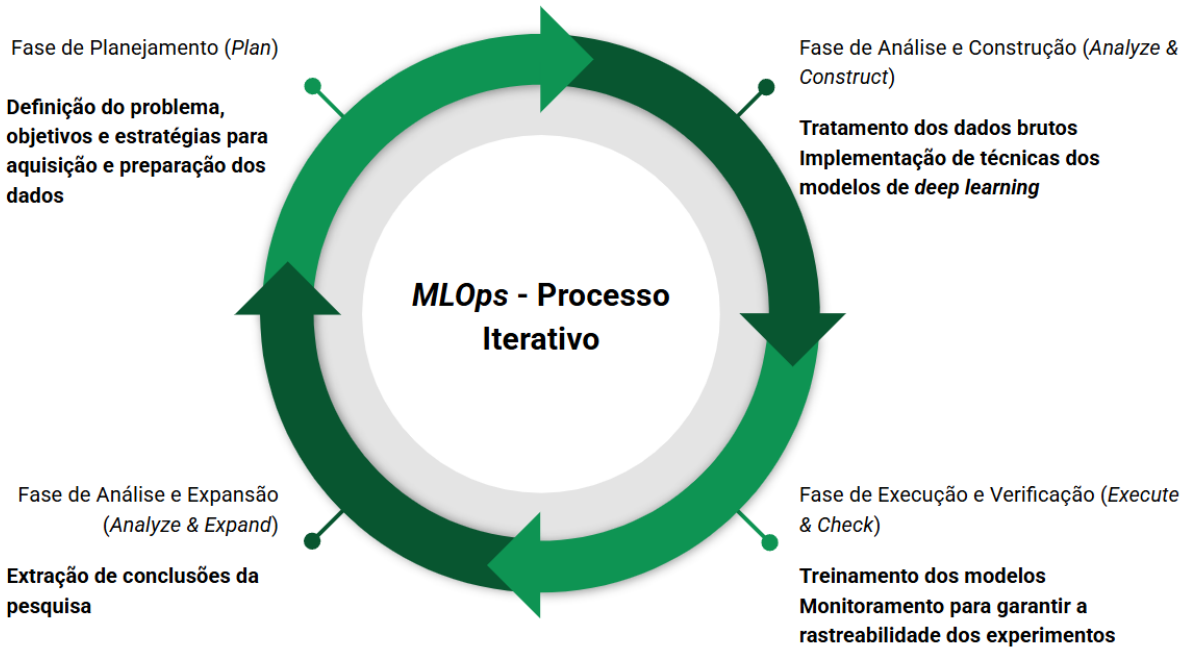
4.1 CLASSIFICAÇÃO DA PESQUISA

A pesquisa pode ser caracterizada quanto à abordagem, como quantitativa, pois envolve a coleta e análise de dados numéricos (métricas de desempenho dos modelos, tempos de inferência) com o objetivo de quantificar e comparar a performance das arquiteturas *FCN-8s* e *U-Net*. Quanto à natureza, classifica-se como pesquisa aplicada, uma vez que busca gerar conhecimento com um objetivo prático e direto: desenvolver um sistema de percepção ambiental para veículos Baja SAE que seja eficiente e de baixo custo. Quanto aos objetivos, a pesquisa possui natureza descritiva e explicativa, pois caracteriza e compara o desempenho dos modelos e busca identificar os fatores que determinam suas diferenças de performance. Quanto aos procedimentos, trata-se de uma pesquisa experimental, devido ao treinamento e validação controlada dos modelos, combinada com uma pesquisa bibliográfica para o embasamento teórico.

4.2 FLUXO DE TRABALHO METODOLÓGICO (*PACE*)

O desenvolvimento do projeto seguiu as quatro fases do ciclo *PACE*, conforme detalhado no fluxograma da Figura 4, que ilustra o itinerário da pesquisa, desde a concepção dos dados até a análise final dos modelos.

Figura 4 – Fluxograma Metodológico baseado no Ciclo *PACE*.



Fonte: Autoria própria.

4.3 FASE DE PLANEJAMENTO (*PLAN*)

Nesta fase inicial, foram definidos o escopo do problema, os objetivos, os recursos necessários e as estratégias para a aquisição e preparação dos dados.

4.3.1 Coleta e Estruturação do Conjunto de Dados

Dada a inexistência de conjuntos de dados públicos e anotados para o domínio específico da competição Baja SAE, a primeira etapa do planejamento consistiu na curadoria e criação de um conjunto de dados. Foi compilado um total de 138 imagens, cuja aquisição seguiu uma estratégia de fontes mistas para garantir variabilidade: aproximadamente 80% das amostras foram obtidas do repositório oficial da SAE Brasil (imagens registradas pela organização no Baja SAE Nacional 2025), 15% foram capturadas por autoria própria *in loco* durante testes de campo, e os 5% restantes consistem em registros da competição de Michigan 2024 (Enduro) obtidos via *web*.

Visando a padronização necessária para a arquitetura da Rede Neural Convolucional, todas as imagens passaram por um pré-processamento de redimensionamento espacial, resultando em tensores com dimensões fixas de (576, 640, 3) (altura, largura e canais RGB). O *dataset* abrange diferentes tipos de terreno e condições de iluminação, e os dados brutos foram

estruturados nos diretórios *images*/ e *annotated*/ para garantir a rastreabilidade e versionamento no *pipeline* de treinamento.

4.3.2 Definição das Classes de Segmentação

Com base nos requisitos de percepção ambiental para um veículo *off-road*, foram definidas sete classes de interesse: carro, pessoa, gramado, pista, obstáculo, cone e lama. Uma oitava classe, *background*, foi implicitamente definida para representar todas as demais áreas não rotuladas.

4.3.3 Escolha de Ferramentas e Arquiteturas

Selecionou-se o ecossistema Python como base de desenvolvimento, utilizando o *framework TensorFlow* com a *API Keras*. É importante ressaltar que as arquiteturas U-Net e FCN-8s não foram obtidas de bibliotecas de modelos pré-compilados; ambas foram implementadas integralmente através da construção manual das camadas.

A codificação dos algoritmos baseou-se rigorosamente nas descrições topológicas e diagramas apresentados na literatura original de cada arquitetura. Essa abordagem de implementação própria permitiu o controle total sobre os hiperparâmetros e a adaptação necessária das camadas de entrada e saída para as dimensões específicas dos dados deste projeto.

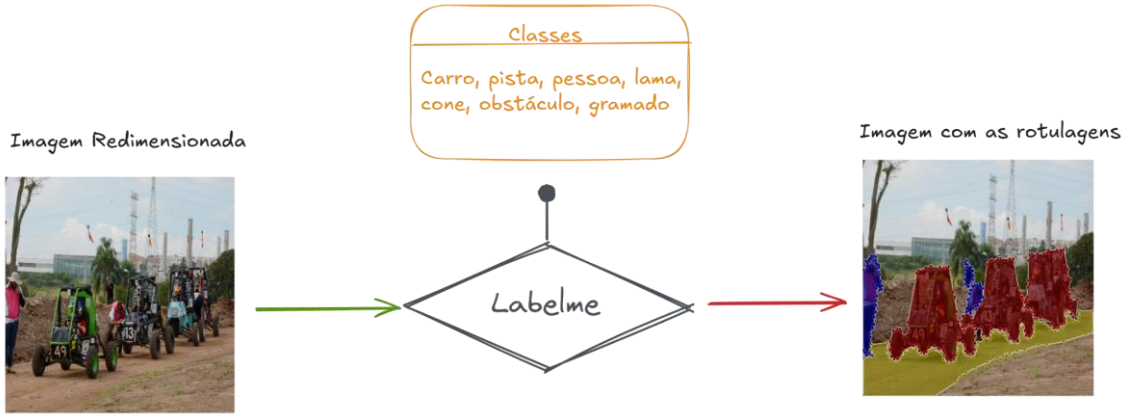
4.4 FASE DE ANÁLISE E CONSTRUÇÃO (*ANALYZE & CONSTRUCT*)

Esta fase compreendeu o tratamento dos dados brutos e a implementação técnica dos modelos de *deep learning*.

4.4.1 Anotação e Geração de Máscaras

As imagens coletadas foram rotuladas manualmente utilizando a ferramenta *LabelMe*. Para cada imagem, foi gerado um arquivo *JSON* contendo os polígonos que delimitam cada objeto de interesse e sua respectiva classe. Conforme implementado, a função *create_multi_masks* processa os arquivos *JSON* para gerar as máscaras de segmentação. A Figura 5 apresenta um esquema que ilustra essa etapa do processo.

Figura 5 – Processo de Rotulagem de Imagens com o *LabelMe* para Segmentação Semântica.

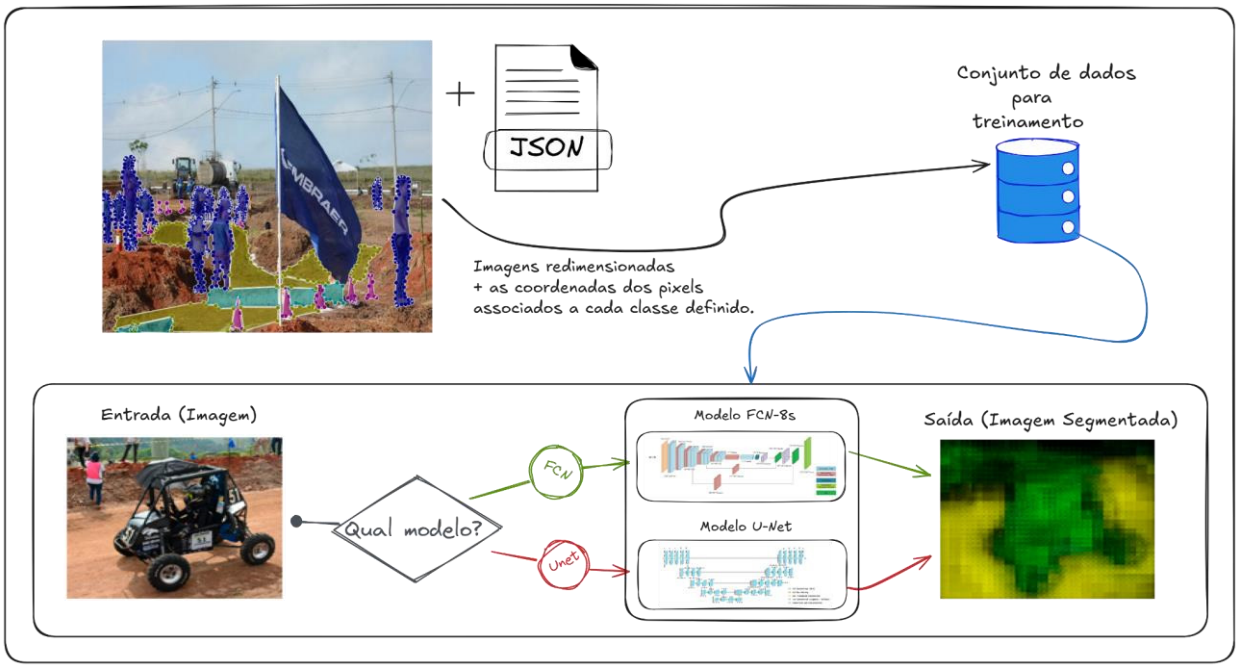


Fonte: Autoria própria.

Para cada imagem, foi criada uma matriz tridimensional de formato (Altura, Largura, *N_Classes*), onde cada um dos Nº canais corresponde a uma máscara binária para uma classe específica, servindo como o *ground truth* para o treinamento supervisionado.

A Figura 6 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 6 – Fluxo de Processamento para Segmentação Semântica de Imagens com Redes Neurais Convolucionais.

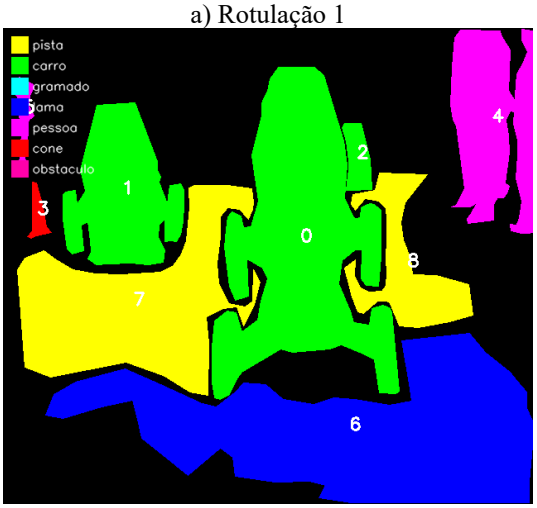


Fonte: Autoria própria.

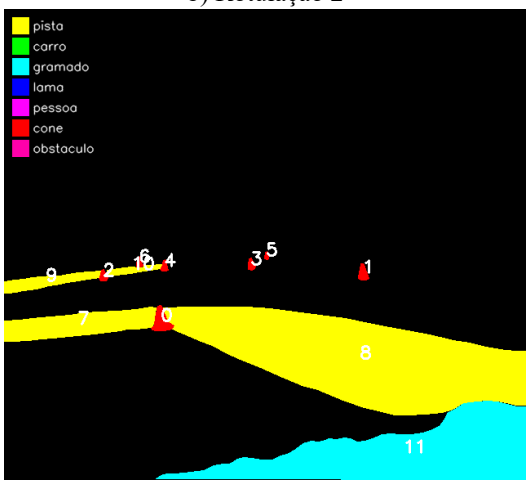
As Figuras 7 (a, b, c e d) exemplificam as máscaras de segmentação, que servem como rótulos (*ground truth*) para o treinamento supervisionado. Esses rótulos fornecem a informação fundamental, pixel-a-pixel, permitindo que a rede neural aprenda a associar os padrões visuais de entrada à sua correspondente classe semântica (como 'pista', 'carro', ‘pessoa’ etc.). O objetivo final é capacitar o modelo a generalizar esse aprendizado, realizando a segmentação precisa em imagens inéditas.

Figura 7 – Processamento de Imagens e Anotações, Associando Imagens às *Labels* Feitas no *Labelme*.

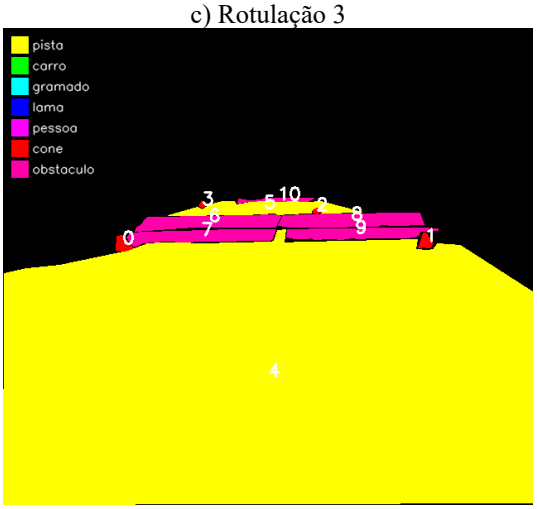
a) Rotulação 1



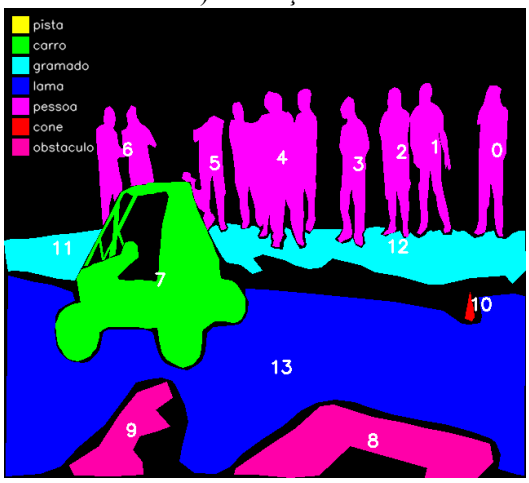
b) Rotulação 2



c) Rotulação 3



d) Rotulação 4



Fonte: Autoria própria.

4.4.2 Implementação das Arquiteturas

U-Net: foi construída com uma estrutura simétrica composta por um caminho de contração (*encoder*) com 5 blocos convolucionais e um caminho de expansão (*decoder*) com 4 blocos. Cada bloco convolucional (*conv_block*) consiste em duas camadas *Conv2D* (com *kernel* 3 x 3 e *padding* "*same*"), onde cada uma é imediatamente seguida por uma camada

BatchNormalization e uma ativação *relu*.

O caminho de contração (*encoder_block*) aplica um *conv_block* e, em seguida, uma camada *MaxPooling2D* (2 x 2) para subamostragem. O caminho de expansão (*decoder_block*) utiliza uma camada *Conv2DTranspose* (2 x 2, *strides* 2) para o *upsampling*, concatena sua saída com os mapas de características correspondentes do encoder (via *skip connections*) e, em seguida, aplica um *conv_block* para refinar os mapas de características.

A implementação dessa arquitetura foi parametrizada pela função *U-Net (pretrained=False, base=1)*. A Tabela 1 detalha as camadas, os filtros e as dimensões de saída, assumindo uma imagem de entrada de (576, 640, 3) e o parâmetro *base*=1.

Tabela 1 – Arquitetura *U-Net* Customizada (*pretrained=False*, *base*=1).

Etapa	**Bloco/Camada**	**Filtros**	**Dimensão da Saída (H, W, F)**	**Conexão (*Skip*)**
Entrada	*Input*	-	(576, 640, 3)	
Encoder	Bloco Enc 1 (*encoder_block*)	64		
	conv_block (2x [Conv, BN, ReLU])	*64*	(576, 640, 64)	Salva s1
	MaxPooling2D (2x2)	-	(288, 320, 64)	
	Bloco *Enc 2 (encoder_block)*	*128*		
	conv_block (2x [Conv, BN, ReLU])	*128*	(288, 320, 128)	Salva s2
	MaxPooling2D (2x2)	-	(144, 160, 128)	
	Bloco *Enc 3 (encoder_block)*	*256*		
	conv_block (2x [Conv, BN, ReLU])	*256*	(144, 160, 256)	Salva s3
	MaxPooling2D (2x2)	-	(72, 80, 256)	
	Bloco *Enc 4 (encoder_block)*	*512*		
	conv_block (2x [Conv, BN, ReLU])	*512*	(72, 80, 512)	Salva s4
	MaxPooling2D (2x2)	-	(36, 40, 512)	
Bottleneck	Bloco Central *(conv_block)*	*1024*		
	conv_block (2x [Conv, BN, ReLU])	*1024*	(36, 40, 1024)	
Decoder	Bloco *Dec 1 (decoder_block)*	*512*		
	Conv2DTranspose (2x2)	*512*	(72, 80, 512)	

Etapa	Bloco/Camada	Filtros	Dimensão da Saída (H, W, F)	Conexão (*Skip*)
	Concatenação	-	(72, 80, 1024)	Recebe s4
	conv_block (2x [Conv, BN, ReLU])	*512*	(72, 80, 512)	
	Bloco *Dec 2 (decoder_block)*	*256*		
	Conv2DTranspose (2x2)	*256*	(144, 160, 256)	
	Concatenação	-	(144, 160, 512)	Recebe s3
	conv_block (2x [Conv, BN, ReLU])	*256*	(144, 160, 256)	
	Bloco *Dec 3 (decoder_block)*	*128*		
	Conv2DTranspose (2x2)	*128*	(288, 320, 128)	
	Concatenação	-	(288, 320, 256)	Recebe s2
	conv_block (2x [Conv, BN, ReLU])	*128*	(288, 320, 128)	
	Bloco Dec 4 (decoder_block)	*64*		
	Conv2DTranspose (2x2)	*64*	(576, 640, 64)	
	Concatenação	-	(576, 640, 128)	Recebe s1
	conv_block (2x [Conv, BN, ReLU])	*64*	(576, 640, 64)	
Saída	*Conv2D (1x1)*	*n_classes*	(576, 640, n_classes)	

Fonte: Autoria própria.

Além da arquitetura, os hiperparâmetros de compilação e treinamento definidos no código são sumarizados na Tabela 2.

Tabela 2 – Hiperparâmetros de Configuração e Compilação do Modelo *U-Net*.

Parâmetro	Valor	Descrição
Ativação (Camadas Internas)	*relu*	Conforme conv_block
Normalização	*BatchNormalization*	Conforme conv_block
Inicializador de *Kernel*	*glorot_uniform (default)*	Padrão do Keras para Conv2D
Otimizador	*Adam*	Conforme código
Taxa de Aprendizado	*1e-4*	Conforme código
Caso Binário (*n_classes=1*)		
Função de Perda	*BinaryCrossentropy*	Para segmentação binária

Parâmetro	Valor	Descrição
Ativação de Saída	*sigmoid*	Para segmentação binária
Métrica Principal	*MeanIoU(num_classes=2)*	*IoU* para fundo/frente
Caso Multiclasse (n_classes > 1)		
Função de Perda	*CategoricalCrossentropy*	Para segmentação multiclasse
Ativação de Saída	*softmax*	Para segmentação multiclasse

Fonte: Autoria própria.

FCN-8s: a implementação seguiu a arquitetura clássica, utilizando um *backbone* inspirado na *VGG*. As camadas densas foram substituídas por convoluções 1 x 1, e a segmentação final é refinada pela fusão de predições de três escalas distintas da rede (*pool3*, *pool4* e a camada final), combinadas através de operações de *Add* após o devido *upsampling* com camadas *Conv2DTranspose*.

A arquitetura implementada utiliza a VGG16 *(include_top=False, weights="imagenet")* como *backbone*, conforme detalhado na Tabela 3. A escolha intencional pela VGG16, em detrimento de arquiteturas mais leves como a *MobileNet*, justifica-se pela necessidade de estabelecer um comparativo fiel à proposta original da FCN-8s (Long et al., 2015). O objetivo foi avaliar o desempenho de uma rede densa e com alta capacidade de extração de características (VGG16) em contraste com a topologia baseada em *encoder-decoder* simétrico da U-Net. Dessa forma, isolam-se as variáveis arquiteturais, utilizando a FCN-VGG16 como o padrão-ouro de acurácia (*upper bound* de capacidade), ainda que à custa de maior carga computacional. É assumida uma entrada de (576, 640, 3).

Tabela 3 – Arquitetura *FCN-8s (Backbone VGG16)*.

Etapa	Camada/Bloco *VGG16*	Dimensão Saída (H, W, F)	Extração (para *Decoder*)
Entrada	*Input*	(576, 640, 3)	
Backbone	*block1_pool*	(288, 320, 64)	
(*Encoder*)	*block2_pool*	(144, 160, 128)	
	block3_pool	(72, 80, 256)	Salva f3 (Stride 8)
	block4_pool	(36, 40, 512)	Salva f4 (Stride 16)
	block5_pool	(18, 20, 512)	Salva f5 (Stride 32)
Decoder	*Caminho 1 (de f5)*		
(Fusão)	*Conv2D (1x1) em f5*	(18, 20, *n_classes*)	

Etapa	Camada/Bloco *VGG16*	Dimensão Saída (H, W, F)	Extração (para *Decoder*)
	Conv2DTranspose (2x2)	(36, 40, *n_classes*)	(*Upsample* x2)
	Caminho 2 (de f4)		
	Conv2D (1x1) em f4	(36, 40, *n_classes*)	
	Fusão 1 (f5 + f4)		
	Add()	(36, 40, n_classes)	
	Conv2DTranspose (2x2)	(72, 80, *n_classes*)	(*Upsample* x2)
	Caminho 3 (de f3)		
	Conv2D (1x1) em f3	(72, 80, *n_classes*)	
	Fusão 2 (f3 + f4 + f5)		
	Add()	(72, 80, *n_classes*)	
	Upsampling Final		
	Conv2DTranspose (8x8)	(576, 640, *n_classes*)	*(Upsample x8)*
Saída	*Activation*	(576, 640, *n_classes*)	

Fonte: Autoria própria.

Os parâmetros de compilação definidos no código para o modelo *FCN-8s* estão sumarizados na Tabela 4.

Tabela 4 – Hiperparâmetros de Configuração e Compilação do Modelo *FCN-8s*.

Parâmetro	Valor	Descrição
Backbone	*VGG16*	Pré-treinado (*ImageNet*), congelado
Otimizador	*Adam*	Conforme código
Taxa de Aprendizado	1e-4	Conforme código
Caso Binário (*n_classes=1*)		
Função de Perda	*BinaryCrossentropy*	Para segmentação binária
Ativação de Saída	*sigmoid*	Para segmentação binária
Métrica Principal	*MeanIoU*(num_classes=2)	*IoU* para fundo/frente
Caso Multiclasse (*n_classes > 1*)		
Função de Perda	*CategoricalCrossentropy*	Para segmentação multiclasse
Ativação de Saída	*softmax*	Para segmentação multiclasse
Métricas	*CategoricalAccuracy, MeanIoU*	Acurácia e *IoU*

Fonte: Autoria própria.

4.4.3 Definição da Função de Perda e Otimizador

Para ambos os modelos, foi utilizada a função de perda *categorical_crossentropy*, adequada para problemas de classificação multiclasse pixel a pixel. O otimizador escolhido foi o Adam, com uma taxa de aprendizado (*learning rate*) de 1 e -4, conhecido por sua eficiência e robustez em problemas de visão computacional.

4.5 FASE DE EXECUÇÃO E VERIFICAÇÃO (*EXECUTE & CHECK*)

Nesta fase, os modelos foram treinados, e seu desempenho foi sistematicamente monitorado, aplicando práticas de *MLOps* para garantir a rastreabilidade dos experimentos.

4.5.1 Processo de Treinamento

A execução dos experimentos e o treinamento dos modelos foram realizados em uma estação de trabalho equipada com processador *AMD Ryzen 7* (arquitetura de 8 núcleos físicos e 16 *threads*), operando em conjunto com uma unidade de processamento gráfico (GPU) com frequência de clock de até 2000 MHz.

O pipeline de treinamento foi orquestrado pelo *script* desenvolvido, que implementa um ciclo de aprendizado ao longo de 300 épocas.

4.6 FASE DE ANÁLISE E EXPANSÃO (*ANALYZE & EXPAND*)

Na fase final, os melhores modelos foram avaliados de forma conclusiva, e os resultados foram analisados para extrair as conclusões da pesquisa.

4.6.1 Análise Quantitativa

Os *checkpoints* dos modelos com melhor desempenho foram carregados e avaliados em um conjunto de teste nunca visto. As métricas de *IoU*, Coeficiente de *Dice*, Acurácia Categórica e Perda Final foram calculadas para realizar a comparação objetiva entre a *U-Net* e a *FCN-8s*.

4.6.2 Análise Qualitativa e de Inferência

Conforme o *script inferencia.py*, foi simulado um cenário de implantação. Os modelos foram convertidos para o formato *TensorFlow Lite (.tflite)*, uma versão otimizada para dispositivos com recursos limitados. A inferência foi executada em um conjunto de 50 imagens

de teste, e o tempo de processamento de cada imagem foi cronometrado para estimar o desempenho em potencial no *hardware*-alvo. As máscaras de saída foram visualmente inspecionadas para identificar pontos fortes e fracos de cada arquitetura na segmentação de diferentes classes, com a opção de mesclar a máscara com a imagem original para melhor visualização (*Background* = *True*).

5 RESULTADOS E ANÁLISES

Na Tabela 5 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, Coeficiente de *Dice*, *Intersection over Union* e a Acurácia Categórica (*Categorical Accuracy*).

Tabela 5 – 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.

A avaliação comparativa dos modelos *U-Net* e *FCN-8s* apresenta variáveis importantes do desempenho deles. Conforme apresentado na Tabela 1, a arquitetura *FCN-8s* demonstrou um desempenho superior em diversas métricas 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 os resultados de acurácia favoreçam a *FCN-8s*, este trabalho reconhece a importância crítica da análise de desempenho computacional (como *FPS*, tamanho do modelo, consumo de *RAM* e, principalmente, a compatibilidade) para aplicações embarcadas. A arquitetura *FCN-8s*, por ser baseada na *VGG16* e possuir um *decoder* mais complexo, apresenta alta complexidade computacional. Isso a torna, em seu formato atual, incompatível com *hardwares* de capacidade restrita como a *Raspberry Pi 3*. Sua implementação prática exigiria uma conversão para o formato *TensorFlow Lite*, com otimizações de quantização.

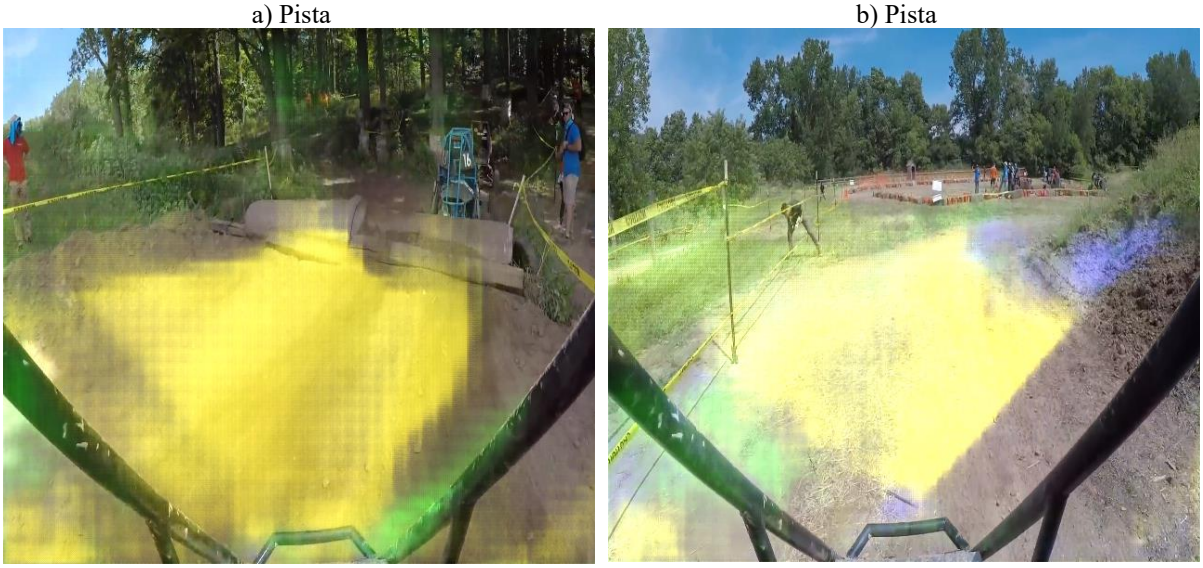
Embora as métricas de acurácia, como o coeficiente Dice e IoU, favoreçam ligeiramente a arquitetura FCN-8s, a viabilidade de implantação em sistemas embarcados exige uma análise rigorosa do custo computacional. A comparação estrutural entre os modelos revela uma discrepância significativa: a FCN-8s, fundamentada na densa arquitetura VGG16, possui aproximadamente [18.035.704] milhões de parâmetros, resultando em um modelo com tamanho estimado de [89+] MB. Em contraste, a U-Net implementada demonstra ser substancialmente mais leve, contabilizando [7.691.325] milhões de parâmetros (uma redução

de [~44,14]%) e ocupando apenas [22,4] MB em disco.

A avaliação qualitativa do modelo selecionado é apresentada na Figura 8, que compila as máscaras de segmentação geradas pela U-Net sobre o conjunto de validação. A disposição das amostras busca demonstrar a robustez da rede frente à complexidade progressiva dos cenários: as subfiguras (a) e (b) validam a detecção da geometria básica da Pista em condições de iluminação controlada; em (c) e (d), observa-se a capacidade do modelo em distinguir texturas de solo, segmentando corretamente áreas de Lama adjacentes ao traçado; os casos (e) e (f) evidenciam a identificação de obstáculos dinâmicos (Veículo) mesmo em terrenos acidentados; por fim, as amostras (g) e (h) ilustram o cenário de maior entropia, onde a rede realiza a segmentação multiclasse simultânea de Lama, Veículo e Pista, confirmando sua aptidão para interpretar a sobreposição de elementos típica do ambiente real do Baja SAE.

Figura 8 – Predições da *U-Net* Usando as Imagens do Conjunto de Validação.

a) Pista b) Pista



c) Lama e pista

d) Lama e pista

e) Lama e Veículo

f) Lama e Veículo

g) Lama, Veículo e Pista

h) Lama, Veículo e Pista

Fonte: Adaptado do Baja SAE 2025.

5.1 ANÁLISE CRÍTICA E PRÓXIMOS PASSOS

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 em *hardwares* como *Raspberry Pi.*

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 4* 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* durante a execução;
- tamanho final do modelo (em disco);
- compatibilidade com otimizações adicionais (como quantização de redes).

Essas medidas são essenciais para garantir a viabilidade da operação em tempo real nos veículos Baja SAE e permitir futuras expansões do sistema de visão computacional.

6 CONCLUSÃO

O presente trabalho teve como objetivo central desenvolver e avaliar arquiteturas de *deep learning* para segmentação semântica, visando sua futura implementação em um sistema de visão computacional embarcado para veículos da competição Baja SAE.

Para atingir este objetivo, foram implementadas, treinadas e comparadas duas das arquiteturas mais influentes na área: a *U-Net* e a *FCN-8s*. A análise quantitativa dos resultados, apresentada no capítulo anterior, permitiu validar a eficácia de ambas as redes na tarefa de segmentação das classes de interesse (pista, obstáculos, pessoas etc.).

Os resultados demonstraram que, em termos de métricas de acurácia, a arquitetura *FCN-8s* apresentou um desempenho superior (*IoU*: 0,7324; *Dice*: 0,8319) em comparação com a *U-Net* (*IoU*: 0,6838; *Dice*: 0,8071) no conjunto de dados utilizado. Este achado sugere que a estratégia da *FCN-8s*, de fundir mapas de características de diferentes escalas (8s, 16s, 32s), foi mais eficaz para capturar os detalhes semânticos da cena.

No entanto, este trabalho também identificou que a acurácia de segmentação é apenas uma das variáveis na equação para um sistema embarcado de tempo real. A principal hipótese da pesquisa — a viabilidade de execução em tempo real — ainda requer validação prática. Conforme discutido, a análise de desempenho computacional (tempo de inferência/*FPS*, uso de *RAM*, tamanho do modelo) é uma etapa crítica e mandatória que não foi coberta nesta fase do projeto.

Como desdobramento e principal encaminhamento para trabalhos futuros, estabelece-se a necessidade imediata de converter os modelos treinados (com foco prioritário na *FCN-8s*, devido à sua maior acurácia) para o formato *TensorFlow Lite*. Esta etapa é essencial para otimizar os modelos para dispositivos com restrições de processamento e memória. Subsequentemente, deverão ser conduzidos testes rigorosos de inferência no *hardware*-alvo para quantificar o desempenho real.

A projeção deste trabalho, uma vez superada a etapa de validação em *hardware*, é de grande repercussão para a equipe Baja SAE. A implementação bem-sucedida de um sistema de segmentação semântica em tempo real estabelece a fundação técnica para o desenvolvimento de sistemas de navegação autônoma, permitindo ao veículo identificar caminhos transitáveis e de obstáculos. Este projeto, portanto, não se encerra em si, mas serve como um pilar essencial para a próxima geração de sistemas de percepção e controle autônoma.

REFERÊNCIAS

CHENG, A.; YIN, C.; CHANG, Y.; PING, H.; LI, S., NAZARIAN, S.; BOGDAN, P. MaskAttn-UNet: a mask attention-driven framework for universal low-resolution image segmentation. **arXiv preprint arXiv:2503.10686**, 2025. Disponível em: https://arxiv.org/pdf/2503.10686. Acesso em 01 set. 2025.

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. Disponível em https://link.springer.com/article/10.1007/S11263-009-0275-4. Acesso em 23 out. 2025.

FANG, R.; CAI, C. Computer vision based obstacle detection and target tracking for autonomous vehicles. *In*: MATEC WEB OF CONFERENCES, 336. EDP Sciences, 2021. p. 07004. Disponível em https://www.matec-conferences.org/articles/matecconf/pdf/2021/05/matecconf_cscns20_07004.pdf. Acesso em 05 set. 2025.

GARCIA-GARCIA, A.; ORTS-ESCOLANO, S.; OPREA, S.; VILLENA-MARTINEZ, V.; GARCIA-RODRIGUEZ, J. A review on deep learning techniques applied to semantic segmentation. **ArXiv:1704.06857**, 2018. Disponível em: https://arxiv.org/abs/1704.06857. Acesso em 26 out. 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge: The MIT Press, 2016. Disponível em: https://synapse.koreamed.org/pdf/10.4258/hir.2016.22.4.351. Acesso em 09 jun. 2025.

GUIMARÃES, M. R. de M. **HARS1DE:** arquitetura de hardware para processamento de CNNs1D na borda. 2025. Dissertação de Mestrado (Programa de Pós-Graduação em Sistemas e Computação) – Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte. Disponível em https://repositorio.ufrn.br/server/api/core/bitstreams/2a8955fa-d958-4576-b945-5b5ccbea5770/content. Acesso em 23 ago. 2025.

JEONG, J.; YOON, T. S.; PARK, J. B. Towards a meaningful 3D map using a 3D lidar and a camera. **Sensors**, v. 18, n. 8, p. 2571, 2018. Disponível em: https://www.mdpi.com/1424-8220/18/8/2571. Acesso em 07 mai. 2025.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436-444, 2015. DOI: https://doi.org/10.1038/nature14539. Disponível em: https://www.nature.com/articles/nature14539. Acesso em 09 out. 2025.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, 1998. DOI: 10.1109/5.726791. Disponível em: https://ieeexplore.ieee.org/document/726791. Acesso em 16 out. 2025.

LONG, J.; SHELHAMER, E.; DARRELL, T**.** Fully convolutional networks for semantic segmentation. CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings**, IEEE, p. 3431-3440, 2015. Disponível em: https://openaccess.thecvf.com/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf. Acesso em 09 nov. 2025.

NOORI, A. Y.; SHAKER, S. H.; AZEEZ, R. A. Semantic segmentation of urban street scenes using deep learning. **Webology**, v. 19, n. 1, 2022. DOI: 10.14704/WEB/V19I1/WEB19156. Disponível em: https://www.researchgate.net/publication/358057201_Semantic_Segmentation_of_Urban_Street_Scenes_Using_Deep_Learning. Acesso em 4 nov. 2025.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: convolutional networks for biomedical image segmentation. *In*: INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION. Cham: Springer international publishing, p. 234-241, 2015. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28. Acesso em 15 nov. 2025.

SILVA, I. C. L. **APES-SOFT:** uma abordagem eficiente em classificação de objetos 3D em nuvens de pontos com redes neurais convolucionais. 2024. Dissertação (Mestrado em Engenharia Elétrica) – Centro de Tecnologia, Universidade Federal do Ceará, 2024. Disponível em: https://repositorio.ufc.br/handle/riufc/78667. Acesso em 21 jul. 2025.

SORENSEN, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. **Biologiske skrifter**, v. 5, p. 1-34, 1948. Disponível em: https://cir.nii.ac.jp/crid/1571135649789292416. Acesso em 23 out. 2025.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. *In*: EUROPEAN CONFERENCE ON COMPUTER VISION. Cham: Springer International Publishing, p. 818-833, 2014. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53. Acesso em 19 nov. 2025.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Cajazeiras - Código INEP: 25008978
Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

Dissertação TCC

Assunto:	Dissertação TCC
Assinado por:	Renan Saraiva
Tipo do Documento:	Dissertação
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Renan Saraiva dos Santos**, **DISCENTE (202112240024) DE BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO - CAMPUS CAJAZEIRAS**, em 26/01/2026 09:22:08.

Este documento foi armazenado no SUAP em 10/02/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1760921
Código de Autenticação: e244ff0427